A Simpler Proof Theory for Nominal Logic

James Cheney

University of Edinburgh
(jcheney@inf.ed.ac.uk)

Abstract. Nominal logic is a variant of first-order logic equipped with a "fresh-name quantifier" N and other features useful for reasoning about languages with bound names. Its original presentation was as a Hilbert axiomatic theory, but several attempts have been made to provide more convenient Gentzen-style sequent or natural deduction calculi for nominal logic. Unfortunately, the rules for N in these calculi involve complicated side-conditions, so using and proving properties of these calculi is difficult. This paper presents an improved sequent calculus $NL^{\Rightarrow}$ for nominal logic. Basic results such as cut-elimination and conservativity with respect to nominal logic are proved. Also, $NL^{\Rightarrow}$ is used to solve an open problem, namely relating nominal logic's N-quantifier and the self-dual ∇-quantifier of Miller and Tiu's $FO\lambda^{\nabla}$.

1 Introduction

Gabbay and Pitts [8] have introduced a new way of reasoning about names and binding, in which α-equivalence and capture-avoiding substitution can be defined in terms of the basic concepts of *swapping* and *freshness*. This approach provides a cleaner treatment of α-equivalence than the classical first-order approach in which α-equivalence and capture-avoiding substitution are defined by mutual recursion. On the other hand, unlike higher-order techniques for dealing with names and binding, the semantics of this model of name-binding is relatively straightforward, so well-understood mathematical tools like structural induction can be used to reason about syntax with bound names.

These ideas have been incorporated into a logic called *nominal logic* [12]. Nominal logic is typed, first-order equational logic augmented with:

- *name-types* $\nu, \nu', \ldots$ inhabited by countably many *names* $\mathsf{a}, \mathsf{b}, \ldots$;
- a *swapping operation* $(- -) \cdot - : \nu \to \nu \to \tau \to \tau$ for each name-type ν and type τ, which acts on values by exchanging occurrences of names;
- a *freshness* relation $- \# - : \nu \to \tau \to o$[1] for each name-type ν and type τ, that holds between a name and a value independent of the name;
- an *abstraction type constructor* $\langle - \rangle -$ and *abstraction function symbol* $\langle - \rangle - : \nu \to \tau \to \langle \nu \rangle \tau$ which constructs values equal up to consistent renaming, axiomatized as follows:

$$\forall a, b, x, y. \langle a \rangle x = \langle b \rangle y \iff (a = b \wedge x = y) \vee (a \# y \wedge x = (a\ b) \cdot y)\ ;$$

- a *some/any fresh-name quantifier* N that is self-dual ($\neg \mathsf{N}\mathsf{a}.\varphi \iff \mathsf{N}\mathsf{a}.\neg\varphi$);
- and *freshness* and *equivariance* principles which state that *fresh names can always be chosen* and *truth is preserved by name-swapping*, respectively.

[1] o is the type of propositions

1.1 The Problem

This paper is concerned with developing simple rules for reasoning with the И-quantifier. Pitts' original formalization of nominal logic was a Hilbert-style collection of first-order axioms (which we call NL). There were no new inference rules for И. Instead, И was defined using the axiom scheme $\forall \overline{x}.(И a.\varphi \iff \exists a.a \# \overline{x} \wedge \varphi)$, where $FV(\varphi) \subseteq \{a, \overline{x}\}$. While admirable from a reductionist point of view, Hilbert systems have well-known deficiencies for modeling actual reasoning. Instead, Gentzen-style *natural deduction* and *sequent* systems provide a more intuitive approach to formal reasoning in which logical connectives are explained as *proof-search* operations. Gentzen systems are especially useful for computational applications, such as automated deduction and logic programming. A sequent calculus formalization would also be convenient for relating nominal logic with other logics by proof-theoretic translations.

Gentzen-style rules for И have been considered in previous work. Pitts [12] proposed sequent and natural deduction rules for И based on the observation that

$$\forall a.(a \# \overline{x} \supset \varphi(a, \overline{x})) \supset И a.\varphi(a, \overline{x}) \supset \exists a.(a \# \overline{x} \wedge \varphi(a, \overline{x})) \ .$$

These rules (see Figure 1(NL)) are symmetric, emphasizing И's self-duality. However, they are not closed under substitution, which greatly complicates the the proof of cut-elimination or proof-normalization properties.

Gabbay [6] introduced Fresh Logic (FL), an intuitionistic natural deduction calculus for nominal logic, and studied semantic issues including soundness and completeness as well as proving proof-normalization. Gabbay and Cheney [7] presented a similar sequent calculus called FL_{Seq}. In FL, Gabbay introduced a technical device called *slices* for obtaining rules that are closed under substitution. Technically, a slice $\varphi[a\#\overline{u}]$ of a formula φ is a decomposition of the formula as $\varphi(a, \overline{x})[\overline{u}/\overline{x}]$ for fresh variables $\overline{x}$, such that a does not appear in any of the $\overline{u}$. Slices were also used in the FL_{Seq} rules (see Figure 1(FL_{Seq})). The slice-based rules shown in Figure 1(FL_{Seq}) *are* closed under substitution, so proving cut-elimination for these rules is relatively straightforward once several technical lemmas involving slices have been proved. Noting that the FL_{Seq} rules are structurally similar to $\forall L$ and $\exists R$, respectively, Gabbay and Cheney observed that alternate rules in which $И L$ was similar to $\exists L$ and $И R$ similar to $\forall R$ were possible (see Figure 1(FL'_{Seq})). These rules seem simpler and more deterministic; however, they still involve slices.

Gabbay and Cheney presented a proof-theoretic semantics for nominal logic programming based on FL_{Seq}. However, this analysis suggested an interpretation of И-quantified formulas that was radically different from the approach used in the αProlog nominal logic programming language [2]. The proof-search interpretation of $И a.\varphi$ suggested by FL_{Seq} is "search for a slice $\varphi[a\#\overline{u}]$ of φ and substitution t for a such that $t \# \overline{u}$ and solve $\varphi(t, \overline{u})$", while in αProlog, the interpretation of $И \mathsf{a}.\varphi$ is "generate a fresh name a' and solve $\varphi(\mathsf{a}')$". The approach motivated by the FL_{Seq} proof-theoretic semantics seems much more complicated than experience with αProlog suggests.

Gabbay and Cheney also gave a translation from $FO\lambda^{\nabla}$, a logic introduced by Miller and Tiu that also includes a self-dual quantifier, ∇ [9] into FL_{Seq}. This translation was sound (mapped derivable sequents to derivable sequents), but incomplete

$$\frac{\Gamma, a \# \overline{x} \Rightarrow \varphi, \Delta \quad (\dagger)}{\Gamma \Rightarrow \mathsf{И}a.\varphi, \Delta}\ \mathsf{И}R \qquad \frac{\Gamma, a \# \overline{x}, \varphi \Rightarrow \Delta \quad (\dagger)}{\Gamma, \mathsf{И}a.\varphi \Rightarrow \Delta}\ \mathsf{И}L \qquad (NL)$$

$$\frac{\Gamma \vdash u \# \overline{t} \quad \Gamma \vdash \varphi[u/\mathsf{a}] \quad (*)}{\Gamma \vdash \mathsf{И}\mathsf{a}.\varphi}\ \mathsf{И}I \qquad \frac{\Gamma \vdash \mathsf{И}\mathsf{a}.\varphi \qquad \Gamma \vdash u \# \overline{t} \quad \Gamma, \varphi[u/\mathsf{a}] \vdash \psi\ (*)}{\Gamma \Rightarrow \psi}\ \mathsf{И}E \qquad (FL)$$

$$\frac{\Gamma, u \# \overline{t} \Rightarrow \varphi[u/a] \quad (*)}{\Gamma, u \# \overline{t} \Rightarrow \mathsf{И}a.\varphi}\ \mathsf{И}R \qquad \frac{\Gamma, u \# \overline{t}, \varphi[u/a] \Rightarrow \psi \quad (*)}{\Gamma, u \# \overline{t}, \mathsf{И}a.\varphi \Rightarrow \psi}\ \mathsf{И}L \qquad (FL_{Seq})$$

$$\frac{\Gamma, a \# \overline{t} \Rightarrow \varphi \quad (*), (**)}{\Gamma \Rightarrow \mathsf{И}a.\varphi}\ \mathsf{И}R \qquad \frac{\Gamma, a \# \overline{t}, \varphi \Rightarrow \psi \quad (*), (**)}{\Gamma, \mathsf{И}a.\varphi \Rightarrow \psi}\ \mathsf{И}L \qquad (FL'_{Seq})$$

$$\frac{\Sigma \# \mathsf{a} : \Gamma \Rightarrow \varphi \quad (\mathsf{a} \notin \Sigma)}{\Sigma : \Gamma \Rightarrow \mathsf{И}\mathsf{a}.\varphi}\ \mathsf{И}R \qquad \frac{\Sigma \# \mathsf{a} : \Gamma, \varphi \Rightarrow \psi \quad (\mathsf{a} \notin \Sigma)}{\Sigma : \Gamma, \mathsf{И}\mathsf{a}.\varphi \Rightarrow \psi}\ \mathsf{И}L \qquad (NL^{\Rightarrow})$$

$$(\dagger)\ \overline{x} = FV(\Gamma, \mathsf{И}a.\varphi, \Delta) \quad (*)\ \varphi = \varphi[a\#\overline{t}] \quad (**)\ a \notin FV(\Gamma, \psi)$$

Fig. 1. Evolution of rules for И

(mapped some non-derivable sequents to derivable ones). Gabbay and Cheney conjectured that their translation would be complete relative to $FO\lambda^{\nabla}$ extended with weakening and exchange for ∇.

In this paper we present a simplified sequent calculus for nominal logic, called $NL^{\Rightarrow}$, in which slices are not needed in the rules for И (or anywhere else), and which seems more compatible with the proof-search reading of И in αProlog. Following Urban, Pitts, and Gabbay [14, 6], we employ a new syntactic class of *name-symbols* $\mathsf{a}, \mathsf{b}, \ldots$. Like variables, such name-symbols may be bound (by И), but unlike variables, two distinct name-symbols are always regarded as denoting distinct name values. In place of slices, we introduce variable contexts that encode information about freshness. Specifically, contexts $\Sigma\#\mathsf{a}{:}\nu$ may be formed by adjoining a *fresh name-symbol* a which is also assumed to be semantically fresh for any value mentioned in Σ. Our rules for И (Figure 1($NL^{\Rightarrow}$)) are in the spirit of the original rules and are very simple.

Besides the sequent calculus itself, we present two applications. First, we verify that $NL^{\Rightarrow}$ and Pitts' axiomatization NL are equivalent. Second, we present and prove the soundness and completeness of a new translation from $FO\lambda^{\nabla}$ to nominal logic, solving a problem left unsolved by Gabbay and Cheney. We have also found that the original translation is complete relative to $FO\lambda^{\nabla}$ extended with ∇-weakening and contraction.

The structure of this paper is as follows: Section 2 presents the sequent calculus $NL^{\Rightarrow}$ along with proofs of structural properties and conservativity of $NL^{\Rightarrow}$ relative to NL. In Section 3, we present sound and complete translations from $FO\lambda^{\nabla}$ (with and without ∇-weakening and exchange) to $NL^{\Rightarrow}$. Section 4 discusses additional related and future work, and Section 5 concludes.

2 Sequent Calculus

The sequent calculus in this section is a generalization of the one presented in Chapter 4 of the author's dissertation [5]. Full proofs can be found there and in a companion technical report [3].

2.1 Syntax and Well-Formedness

The types τ, terms t, and formulas φ of $NL^{\Rightarrow}$ are generated by the following grammar:

$$\tau ::= o \mid \delta \mid \nu \mid \tau \to \tau' \mid \langle\nu\rangle\tau \qquad t, u ::= c \mid \mathsf{a} \mid \lambda x{:}\tau.t \mid t\ u \mid x$$
$$\varphi, \psi ::= \top \mid \bot \mid t \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \varphi \supset \psi \mid \forall x{:}\tau.\varphi \mid \exists x{:}\tau.\varphi \mid \mathit{И}\mathsf{a}{:}\nu.\varphi$$

The base types are datatypes δ, name-types ν, and the type o of propositions; additional types are formed using the function and abstraction type constructors. Variables x, y are drawn from a countably infinite set V; also, name-symbols a, b are drawn from a disjoint countably infinite set A. The letters a, b are typically used for terms of some name-sort ν. Note that λ-terms are included in this language and are handled in a traditional fashion. In particular, terms are considered equal up to $\alpha\beta\eta$-equivalence. Similarly, $\forall, \exists$, and $\mathit{И}$-quantified formulas are identified up to α-equivalence. We assume given a signature that maps constant symbols c to types τ, and containing at least the following declarations:

$$\mathit{eq}_\tau : \tau{\to}\tau{\to}o \quad \mathit{fresh}_{\nu\tau} : \nu{\to}\tau{\to}o \quad \mathit{swap}_{\nu\tau} : \nu{\to}\nu{\to}\tau{\to}\tau \quad \mathit{abs}_{\nu\tau} : \nu{\to}\tau{\to}\langle\nu\rangle\tau$$

for all name-types ν and types τ. The notations $t \approx u$, $t \# u$, $(t\ u) \cdot v$, and $\langle t\rangle u$ are syntactic sugar for $\mathit{eq}\ t\ u$, $\mathit{fresh}\ t\ u$, $\mathit{swap}\ t\ u\ v$, and $\mathit{abs}\ t\ u$, respectively.

The *contexts* used in $NL^{\Rightarrow}$ are generated by the grammar:

$$\Sigma ::= \cdot \mid \Sigma, x{:}\tau \mid \Sigma\#\mathsf{a}{:}\nu$$

We often abbreviate $\cdot, x{:}\tau$ and $\cdot\#\mathsf{a}{:}\nu$ to $x{:}\tau$ and $\mathsf{a}{:}\nu$ respectively, and may omit type declarations when no ambiguity ensues. We write ω for a term that may be either a name-symbol a or a variable x. The functions $FV(-)$, $FN(-)$, $FVN(-)$ calculate the sets of free variables, name-symbols, or both variables and name-symbols of a term or formula. Note that abstraction $\langle-\rangle-$ is just a function symbol and does not bind its first argument (which may be any term of type ν), and so $FN(\langle a\rangle t) = FN(a) \cup FN(t)$, whereas $\mathit{И}\mathsf{a}.\varphi$ does bind a, so $FN(\mathit{И}\mathsf{a}.\varphi) = FN(\varphi) - \{\mathsf{a}\}$. We write $\omega{:}\tau \in \Sigma$ if the binding $\omega{:}\tau$ is present in Σ. We write $\Sigma; \Sigma'$ for the result of concatenating two contexts such that $FVN(\Sigma) \cap FVN(\Sigma') = \varnothing$.

Remark 1. The inclusion of λ-terms and identification of terms and formulas with bound names up to α-equivalence may be objectionable because it appears that we are circularly attempting to define binding in terms of binding. This is not the case. A key contribution of Gabbay and Pitts' approach is that it shows how one can formally justify a traditional, informal approach to binding syntax by constructing syntax trees modulo α-equivalence as simple mathematical objects in a particularly clever way [8][5, Ch. 3–4]. We assume that this or some other standard technique for dealing with binding in nominal logic's terms and formulas is acting behind the scenes.

We write $\Sigma \vdash t : \tau$ or $\Sigma \vdash \varphi : o$ to indicate that t is a well-formed term of type τ or φ is a well-formed formula. From the point of view of typechecking, the freshness information given by the context is irrelevant. There are only two nonstandard rules for typechecking:

$$\frac{\omega{:}\tau \in \Sigma}{\Sigma \vdash \omega : \tau} \qquad \frac{\Sigma\#\mathsf{a}{:}\nu \vdash \varphi : o}{\Sigma \vdash \mathit{И}\mathsf{a}{:}\nu.\varphi : o}$$

Terms viewed as formulas must, as usual, be of type o. Quantification using $\forall$ and $\exists$ is only allowed over types not mentioning o; $\mathit{И}$-quantification is only allowed over name-types.

Let $Tm_\Sigma = \{t \mid \Sigma \vdash t : \tau\}$ be the set of well-formed terms in context Σ. We associate a set of freshness formulas $|\Sigma|$ to each context Σ as follows:

$$|\cdot| = \varnothing \qquad |\Sigma, x : \tau| = |\Sigma| \qquad |\Sigma\#\mathsf{a} : \nu| = |\Sigma| \cup \{\mathsf{a} \# t \mid t \in Tm_\Sigma\}$$

For example, $\mathsf{a} \# x, \mathsf{b} \# \mathsf{a}$, and $\mathsf{b} \# f\ x\ y \in |x{:}\tau\#\mathsf{a}{:}\nu, y{:}\tau'\#\mathsf{b}{:}\nu'|$ (provided $f : \tau \to \tau' \to \sigma$ is a function symbol). We say that Σ is stronger than Σ' ($\Sigma' \leq \Sigma$) if $Tm_{\Sigma'} \subseteq Tm_\Sigma$ and $|\Sigma'| \subseteq |\Sigma|$. For example, $\mathsf{a}, x \leq x\#\mathsf{a}, y$.

Lemma 1 (Term Weakening). *If $\Sigma \vdash t : \tau$ and $\Sigma \leq \Sigma'$ then $\Sigma' \vdash t : \tau$.*

Lemma 2 (Term Substitution). *If $\Sigma \vdash t : \tau$ and $\Sigma, x : \tau; \Sigma' \vdash u : \tau'$ then $\Sigma; \Sigma' \vdash u[t/x] : \tau'$.*

2.2 The Rules

Judgments are of the form $\Sigma : \Gamma \Rightarrow \Delta$, where Σ is a context and Γ, Δ are multisets of formulas. We define classical and intuitionistic versions of $NL^{\Rightarrow}$. *Classical* $NL^{\Rightarrow}$ is based on the classical sequent calculus $\mathbf{G3c}$ [11] (see Figure 2), whereas *Intuitionistic* $NL^{\Rightarrow}$ ($INL^{\Rightarrow}$) is based on the intuitionistic calculus $\mathbf{G3im}$ (in which $\supset$, $\forall R$, and $\exists L$-rules are restricted to a single-conclusion form). Both versions include two additional *logical rules*, $\mathit{И}L$ and $\mathit{И}R$, shown in Figure 1($NL^{\Rightarrow}$). In addition, $NL^{\Rightarrow}$ includes several *nonlogical rules* (Figure 4) defining the properties of swapping, equality, freshness and abstraction. Figure 5 lists some admissible rules.

Many of the nonlogical rules correspond to first-order universal axioms of nominal logic (Figure 3), which may be incorporated into sequent systems in a uniform fashion using the Ax rule without affecting cut-elimination [11]. The remaining nonlogical rules are as follows. Rule A_2 expresses an invertibility property for abstractions: two abstractions are equal only if they are structurally equal or equal by virtue of A_1. A_3 says that all values of abstraction type are formed using the abstraction function symbol. The F rule expresses the freshness principle: that a name fresh for a given context may always be chosen. Finally, the $\Sigma\#$ rule allows freshness information to be extracted from the context Σ. It states that in context Σ, any constraint in $|\Sigma|$ is valid.

2.3 Structural Properties

We now list some routinely-verified properties of $NL^{\Rightarrow}$ derivations. We write $\vdash_n J$ to indicate that judgment J has a derivation of height at most n.

$$\frac{}{\Sigma : \Gamma, p\,\overline{t} \Rightarrow p\,\overline{t}, \Delta}\ hyp$$

$$\frac{}{\Sigma : \Gamma \Rightarrow \top, \Delta}\ \top R \qquad \frac{}{\Sigma : \Gamma, \bot \Rightarrow \Delta}\ \bot L$$

$$\frac{\Sigma : \Gamma \Rightarrow \varphi, \Delta \quad \Sigma : \Gamma \Rightarrow \psi, \Delta}{\Sigma : \Gamma \Rightarrow \varphi \wedge \psi, \Delta}\ \wedge R \qquad \frac{\Sigma : \Gamma, \varphi_1, \varphi_2 \Rightarrow \Delta}{\Sigma : \Gamma, \varphi_1 \wedge \varphi_2 \Rightarrow \Delta}\ \wedge L$$

$$\frac{\Sigma : \Gamma \Rightarrow \varphi_1, \varphi_2, \Delta}{\Sigma : \Gamma \Rightarrow \varphi_1 \vee \varphi_2, \Delta}\ \vee R \qquad \frac{\Sigma : \Gamma, \varphi \Rightarrow \Delta \quad \Gamma, \psi \Rightarrow \Delta}{\Sigma : \Gamma, \varphi \vee \psi \Rightarrow \Delta}\ \vee L$$

$$\frac{\Sigma : \Gamma, \varphi \Rightarrow \psi, \Delta}{\Sigma : \Gamma \Rightarrow \varphi \supset \psi, \Delta}\ \supset R \qquad \frac{\Sigma : \Gamma \Rightarrow \varphi, \Delta \quad \Sigma : \Gamma, \psi \Rightarrow \Delta}{\Sigma : \Gamma, \varphi \supset \psi \Rightarrow \Delta}\ \supset L$$

$$\frac{\Sigma, x : \Gamma \Rightarrow \varphi, \Delta \quad (x \notin \Sigma)}{\Sigma : \Gamma \Rightarrow \forall x.\varphi, \Delta}\ \forall R \qquad \frac{\Sigma \vdash t : \sigma \quad \Sigma : \Gamma, \forall x{:}\tau.\varphi, \varphi[t/x] \Rightarrow \Delta}{\Sigma : \Gamma, \forall x{:}\tau.\varphi \Rightarrow \Delta}\ \forall L$$

$$\frac{\Sigma \vdash t : \sigma \quad \Sigma : \Gamma \Rightarrow \exists x{:}\tau.\varphi, \varphi[t/x], \Delta}{\Sigma : \Gamma \Rightarrow \exists x{:}\tau.\varphi, \Delta}\ \exists R \qquad \frac{\Sigma, x : \Gamma, \varphi \Rightarrow \Delta \quad (x \notin \Sigma)}{\Sigma : \Gamma, \exists x.\varphi \Rightarrow \Delta}\ \exists L$$

$$\frac{\Sigma : \Gamma, t \approx t \Rightarrow \Delta}{\Sigma : \Gamma \Rightarrow \Delta}\ \approx R \qquad \frac{\Sigma : \Gamma, t \approx u, P(t), P(u) \Rightarrow \Delta}{\Sigma : \Gamma, t \approx u, P(t) \Rightarrow \Delta}\ \approx S$$

Fig. 2. Classical typed first-order equational logic (**G3c**)

Lemma 3 (Weakening). *If* $\vdash_n \Sigma : \Gamma \Rightarrow \Delta$ *is derivable then so is* $\vdash_n \Sigma : \Gamma, \varphi \Rightarrow \Delta$.

Lemma 4 (Context Weakening). *If* $\vdash_n \Sigma : \Gamma \Rightarrow \Delta$ *and* $\Sigma \leq \Sigma'$ *then* $\vdash_n \Sigma' : \Gamma \Rightarrow \Delta$

Lemma 5 (Substitution). *If* $\vdash_n \Sigma \vdash t : \tau$ *and* $\Sigma, x{:}\tau; \Sigma' : \Gamma \Rightarrow \Delta$ *then* $\vdash_n \Sigma; \Sigma' : \Gamma[t/x] \Rightarrow \Delta[t/x]$.

The remaining structural transformations do not preserve the height of derivations. However, they do preserve the logical height of the derivation, which is defined as follows.

Definition 1. *The* logical height *of a derivation is the maximum number of logical rules in any branch of the derivation. We write* $\vdash^l_n J$ *to indicate that* J *has a derivation of logical height* $\leq n$.

Lemma 6 (Admissibility of EVL**,** EVR**).** *If* $\vdash^l_n \Sigma : \Gamma, (a\ b) \cdot \varphi \Rightarrow \Delta$*, then so is* $\vdash^l_n \Sigma : \Gamma, \varphi \Rightarrow \Delta$*. Similarly, if* $\vdash^l_n \Sigma : \Gamma \Rightarrow (a\ b) \cdot \varphi, \Delta$ *is derivable, then so is* $\vdash^l_n \Sigma : \Gamma \Rightarrow \varphi, \Delta$.

Lemma 7 (Admissibility of hyp^***).** *The judgment* $\Sigma : \Gamma, \varphi \Rightarrow \varphi, \Delta$ *is derivable for any* φ.

Proof (Sketch). Induction on the construction of φ. The only new case is for $\varphi = \mathsf{И}\mathsf{a}.\psi(\mathsf{a}, \overline{x})$. By induction we know that $\Sigma\#\mathsf{a}\#\mathsf{b} : \Gamma, \psi(\mathsf{b}, \overline{x}) \Rightarrow \psi(\mathsf{b}, \overline{x})$. Using equivariance we have $\Sigma\#\mathsf{a}\#\mathsf{b} : \Gamma, (\mathsf{a}\ \mathsf{b}) \cdot \psi(\mathsf{a}, \overline{x}) \Rightarrow \psi(\mathsf{b}, \overline{x})$. Since $\overline{x} \subset FV(\Sigma)$, using $\Sigma\#$ we know that $\mathsf{a} \# \overline{x}, \mathsf{b} \# \overline{x}$, hence $(\mathsf{a}\ \mathsf{b}) \cdot \overline{x} \approx \overline{x}$, so using equational reasoning we have $(\mathsf{a}\ \mathsf{b}) \cdot \psi(\mathsf{a}, \overline{x}) \approx \psi(\mathsf{b}, \overline{x})$. Then using $\mathsf{И}L$ and $\mathsf{И}R$ we can conclude $\Sigma : \Gamma, \mathsf{И}\mathsf{a}.\psi \Rightarrow \mathsf{И}\mathsf{a}.\psi, \Delta$. □

$$
\begin{array}{llll}
(S_1) & (a\ a)\cdot x \approx x & (E_4) & (a\ b)\cdot \lambda x.e[x] \approx \lambda x.(a\ b)\cdot e[(a\ b)\cdot x] \\
(S_2) & (a\ b)\cdot(a\ b)\cdot x \approx x & (F_1) & a \# x \wedge b \# x \supset (a\ b)\cdot x \approx x \\
(S_3) & (a\ b)\cdot a \approx b & (F_2) & a \# b \quad (a:\nu, b:\nu', \nu \not\equiv \nu') \\
(E_1) & (a\ b)\cdot c \approx c & (F_3) & a \# a \supset \bot \\
(E_2) & (a\ b)\cdot(t\ u) \approx ((a\ b)\cdot t)\ ((a\ b)\cdot u) & (F_4) & a \# b \vee a \approx b \\
(E_3) & p(\overline{x}) \supset p((a\ b)\cdot \overline{x}) & (A_1) & a \# y \wedge x \approx (a\ b)\cdot y \supset \langle a\rangle x \approx \langle b\rangle y
\end{array}
$$

Fig. 3. Equational and freshness axioms

$$
\frac{\Sigma : \Gamma, \overline{P}, Q_1 \Rightarrow \Delta \quad \cdots \quad \Sigma : \Gamma, \overline{P}, Q_n \Rightarrow \Delta}{\Sigma : \Gamma, \overline{P} \Rightarrow \Delta}\ Ax \qquad \bigwedge \overline{P} \supset \bigvee \overline{Q} \text{ an axiom instance}
$$

$$
\frac{\Sigma : \Gamma, \langle a\rangle t \approx \langle b\rangle u, a \approx b, t \approx u \Rightarrow \Delta \quad \Sigma : \Gamma, \langle a\rangle t \approx \langle b\rangle u, a \# u, t = (a\ b)\cdot u \Rightarrow \Delta}{\Sigma : \Gamma, \langle a\rangle t \approx \langle b\rangle u \Rightarrow \Delta}\ A_2
$$

$$
\frac{\Sigma \vdash t : \langle \nu\rangle \sigma \quad \Sigma, a{:}\nu, x{:}\sigma : \Gamma, t \approx \langle a\rangle x \Rightarrow \Delta \quad (a, x \notin \Sigma)}{\Sigma : \Gamma \Rightarrow \Delta}\ A_3
$$

$$
\frac{\Sigma \# \mathsf{a} : \Gamma \Rightarrow \Delta \quad (\mathsf{a} \notin \Sigma)}{\Sigma : \Gamma \Rightarrow \Delta}\ F \qquad \frac{\Sigma : \Gamma, t \# u \Rightarrow \Delta \quad (t \# u \in |\Sigma|)}{\Sigma : \Gamma \Rightarrow \Delta}\ \Sigma\#
$$

Fig. 4. Nonlogical rules

$$
\frac{\Sigma : \Gamma \Rightarrow \Delta}{\Sigma : \Gamma, \varphi \Rightarrow \Delta}\ W \qquad \frac{}{\Sigma : \Gamma, \varphi \Rightarrow \varphi, \Delta}\ hyp^* \qquad \frac{\Sigma : \Gamma \Rightarrow \varphi, \Delta \quad \Sigma : \Gamma', \varphi \Rightarrow \Delta'}{\Sigma : \Gamma, \Gamma' \Rightarrow \Delta, \Delta'}\ cut
$$

$$
\frac{\Sigma : \Gamma, \varphi, \varphi \Rightarrow \Delta}{\Sigma : \Gamma, \varphi \Rightarrow \Delta}\ C \qquad \frac{\Sigma : \Gamma, (a\ b)\cdot\varphi \Rightarrow \Delta}{\Sigma : \Gamma, \varphi \Rightarrow \Delta}\ EVL \qquad \frac{\Sigma : \Gamma \Rightarrow (a\ b)\cdot\varphi, \Delta}{\Sigma : \Gamma \Rightarrow \Delta, \varphi}\ EVR
$$

Fig. 5. Some admissible rules of $NL^{\Rightarrow}$

Lemma 8 (Inversion). *The $\supset L$, $\exists L$, $\wedge L$, and $\vee L$ rules are invertible, in the sense of lemma 2.3.5 and 4.2.8 of Negri and von Plato [11]. In addition, $И L$ is invertible: if $\vdash^l_n \Sigma : \Gamma, И\mathsf{a}.\varphi \Rightarrow \Delta$ is derivable then so is $\vdash^l_n \Sigma\#\mathsf{a} : \Gamma, \varphi \Rightarrow \Delta$ for fresh a.*

Lemma 9 (Contraction). *If $\vdash^l_n \Sigma : \Gamma, \varphi, \varphi \Rightarrow \Delta$ is derivable then so is $\vdash^l_n \Sigma : \Gamma, \varphi \Rightarrow \Delta$.*

2.4 Cut-Elimination

Lemma 10 (Admissibility of Cut). *If $\Sigma : \Gamma \Rightarrow \Delta, \varphi$ and $\Sigma : \Gamma', \varphi \Rightarrow \Delta'$ have cut-free derivations then so does $\Sigma : \Gamma, \Gamma' \Rightarrow \Delta, \Delta'$.*

Proof (Sketch). We show the most interesting case, that for principal cuts on $И$-quantified formulas. In this case, the derivations are of the form

$$
\frac{\begin{array}{c}\Pi \\ \Sigma\#\mathsf{a} : \Gamma \Rightarrow \varphi, \Delta\end{array}}{\Sigma : \Gamma \Rightarrow И\mathsf{a}.\varphi, \Delta}\ И R \qquad \frac{\begin{array}{c}\Pi' \\ \Sigma\#\mathsf{a} : \Gamma', \varphi \Rightarrow \Delta'\end{array}}{\Sigma : \Gamma', И\mathsf{a}.\varphi \Rightarrow \Delta'}\ И L
$$

where without loss of generality we assume that the same fresh name $\mathsf{a} \notin \Sigma$ was used in both sub-derivations. Since φ is smaller than $И\mathsf{a}.\varphi$, we can obtain a derivation Π'' of $\Sigma\#\mathsf{a} : \Gamma, \Gamma' \Rightarrow \Delta, \Delta'$ from Π and Π' by the induction hypothesis. Then

$$\dfrac{\begin{array}{c}\Pi'' \\ \Sigma\#\mathsf{a} : \Gamma, \Gamma' \Rightarrow \Delta, \Delta'\end{array}}{\Sigma : \Gamma, \Gamma' \Rightarrow \Delta, \Delta'}\ F$$

follows using rule F. □

Theorem 1 (Cut-elimination). *If $\Sigma : \Gamma \Rightarrow \Delta$ has any derivation then it has a cut-free derivation.*

Corollary 1 (Consistency). *There is no derivation of $\Sigma : \cdot \Rightarrow \bot$.*

Corollary 2 (Orthognality). *Suppose $\Sigma : \Gamma \Rightarrow \Delta$ and Γ, Δ have no subterms of the form $\langle a \rangle t$ (respectively, $\lambda x.t$). Then there is a derivation of $\Sigma : \Gamma \Rightarrow \Delta$ that does not use any nonlogical rules involving abstraction (respectively, λ).*

2.5 Conservativity

In this section, we show that $NL^{\Rightarrow}$ is conservative relative to Pitts' original axiomatization NL [12]. That is, every theorem of NL is provable in $NL^{\Rightarrow}$, and no new theorems become provable. For convenience, we assume that the same underlying first-order sequent calculus is used for NL and $NL^{\Rightarrow}$.

Write $\vdash_{NL} \Sigma : \Gamma \Rightarrow \Delta$ if there is a first-order equational sequent proof of $\Sigma : \Gamma, \Gamma' \Rightarrow \Delta$. for some set of NL axioms Γ'. Write $\vdash_{NL^{\Rightarrow}} \Sigma : \Gamma \Rightarrow \Delta$ if $\Sigma : \Gamma \Rightarrow \Delta$ is derivable in $NL^{\Rightarrow}$ without using any rules involving λ. Write $\vdash_{IX}$ for the intuitionistic version of provability in system X, that is, provability using only single-conclusion sequents.

We translate NL formulas φ to $NL^{\Rightarrow}$ formulas φ^* by replacing all subformulas of the form $И a.\varphi(a)$ with $И\mathsf{a}.\varphi^*(\mathsf{a})$, for fresh name-symbols a. This translation is uniquely defined up to α-equivalence. For example, $(И a.И b.p(a,b))^* = И\mathsf{a}.И\mathsf{b}.p(\mathsf{a},\mathsf{b})$.

To prove the reverse direction of conservativity, it is necessary to show that $NL^{\Rightarrow}$ sequents involving fresh name-symbols and contexts $\Sigma\#\mathsf{a}$ are equivalent to sequents involving only variables.

Lemma 11 (Name-Elimination). *Suppose Σ mentions only variables and $\vdash^l_n \Sigma\#\mathsf{a} : \Gamma[\mathsf{a}] \Rightarrow \Delta[\mathsf{a}]$. Then $\vdash^l_n \Sigma, a : \Gamma[a], a \# \Sigma \Rightarrow \Delta[a]$, where $a \# \Sigma$ is an abbreviation for $\{a \# x \mid x \in \Sigma\}$.*

Theorem 2 (Conservativity). *$\vdash_{(I)NL} \Sigma : \Gamma \Rightarrow \Delta$ if and only if $\vdash_{(I)NL^{\Rightarrow}} \Sigma : \Gamma^* \Rightarrow \Delta^*$*

Remark 2 (Semantics). Conservativity justifies $NL^{\Rightarrow}$'s description as a sequent calculus for nominal logic. Although this paper focuses exclusively on proof theory at the expense of more traditional model theoretic semantics, conservativity guarantees that $NL^{\Rightarrow}$ inherits Pitts' nominal set semantics for nominal logic (as well as suffering from the same completeness problem). Space constraints preclude further discussion; however, these issues are considered in detail in Cheney's dissertation and a paper in preparation.

3 A Sound and Complete Translation of $FO\lambda^{\nabla}$

Miller and Tiu introduced a sequent calculus called $FO\lambda^{\nabla}$, which abbreviates "First-order Logic with λ-terms and the ∇-quantifier" [9]. Like И, the ∇ quantifier is self-dual. However, И and ∇ have distinctly different properties. Nominal logic and $FO\lambda^{\nabla}$ have similar aims (reasoning about languages in which binding and fresh name-generation play an important role), so it is of interest to determine the relationship between $FO\lambda^{\nabla}$ and $INL^{\Rightarrow}$. Also, $FO\lambda^{\nabla}$ has only been studied using proof theory, but nominal logic has a well-understood semantics [12], so relating the two systems may also elucidate the semantics of $FO\lambda^{\nabla}$.

In $FO\lambda^{\nabla}$, formulas are generalized to *formulas-in-context* $\sigma \triangleright \varphi$, where σ is a list of *local parameters* (variables introduced by ∇) and φ is a formula built out of first-order connectives and quantifiers or $\nabla x.\psi$. We abbreviate "formula-in-context" to "c-formula". Local parameter contexts are subject to α-renaming, so that $a \triangleright p(a)$ and $b \triangleright p(b)$ are considered equal c-formulas. However, c-formulas are not considered equivalent up to reordering or extension of the contexts. Thus, $a, b \triangleright p(a)$, $a \triangleright p(a)$, and $b, a \triangleright p(a)$ are all considered different c-formulas.

The sequent calculus rules dealing with ∇ are as follows:

$$\frac{\Sigma : \Gamma \Rightarrow (\sigma, x) \triangleright \varphi}{\Sigma : \Gamma \Rightarrow \sigma \triangleright \nabla x.\varphi}\ \nabla R \qquad \frac{\Sigma : \Gamma, (\sigma, x) \triangleright \varphi \Rightarrow \mathcal{A}}{\Sigma : \Gamma, \sigma \triangleright \nabla x.\varphi \Rightarrow \mathcal{A}}\ \nabla L$$

where in either case x must not already appear in σ or Σ. However, x *may* appear in some other local context.

Most of the other sequent rules of $FO\lambda^{\nabla}$ are standard, except for the presence of local contexts. For example,

$$\frac{\Sigma : \Gamma, \sigma \triangleright \varphi, \sigma \triangleright \psi \Rightarrow \mathcal{A}}{\Sigma : \Gamma, \sigma \triangleright \varphi \wedge \psi \Rightarrow \mathcal{A}}\ \wedge L \qquad \frac{\Sigma : \Gamma \Rightarrow \sigma \triangleright \varphi \quad \Sigma : \Gamma \Rightarrow \sigma \triangleright \psi}{\Sigma : \Gamma \Rightarrow \sigma \triangleright \varphi \wedge \psi}\ \wedge R$$

are the rules dealing with $\wedge$. The only exceptions are the $\forall$ and $\exists$ rules. In $\forall R$ and $\exists L$, the bound variable is "lifted" to show its dependence on local parameters. Dually, in $\forall L$ and $\exists R$, the term substituted for the bound variable may depend on local parameters. Here are the $\forall$-rules; the rules for $\exists$ are similar.

$$\frac{\Sigma, h{:}\overline{\tau_\sigma} \to \tau : \Gamma \Rightarrow \sigma \triangleright A[h\ \overline{\sigma}/x]}{\Sigma : \Gamma \Rightarrow \sigma \triangleright \forall_\tau x.A}\ \forall R \qquad \frac{\Sigma, \sigma \vdash t : \tau \quad \Sigma : \Gamma, \sigma \triangleright A[t/x] \Rightarrow \mathcal{C}}{\Sigma : \Gamma, \sigma \triangleright \forall_\tau x.A \Rightarrow \mathcal{C}}\ \forall L$$

Although ∇ and И have some properties in common and seem to have similar motivations, the relation between them is not obvious. For example, $INL^{\Rightarrow}$ includes name-types, and И may only quantify over them; $FO\lambda^{\nabla}$ has no name-types, and ∇ may quantify over any simple type. In addition, И admits weakening ($\varphi \iff$ И$\mathsf{a}.\varphi$ where $\mathsf{a} \notin FN(\varphi)$) and exchange (И$\mathsf{a}$.И$\mathsf{b}.\varphi \iff$ Иb.И$\mathsf{a}.\varphi$), and satisfies $\forall x.\varphi(x) \supset$ И$\mathsf{a}.\varphi(\mathsf{a}) \supset \exists x.\varphi(x)$. None of these inferences are derivable with ∇ substituted for И. On the other hand, ∇ commutes with all propositional connectives, $\forall$, and $\exists$, while И only commutes with propositional connectives.

Gabbay and Cheney studied the problem of embedding $FO\lambda^{\nabla}$ into nominal logic. They presented a translation (which we call T_{GC}) from $FO\lambda^{\nabla}$ to FL_{Seq} satisfying

a soundness property: if J is derivable in $FO\lambda^{\nabla}$ then its translation $[\![J]\!]$ is derivable in FL_{Seq}. However, their translation did not satisfy the corresponding completeness property: some non-derivable judgments of $FO\lambda^{\nabla}$ were translated to derivable FL_{Seq} judgments. In particular, the translation failed to reconcile the different behavior of $\mathsf{И}$ and ∇ with respect to weakening and exchange principles.

In the rest of this section, we present a modified translation and prove its soundness and completeness. We also sketch a proof that the original translation is complete with respect to $FO\lambda^{\nabla}$ with ∇-weakening and exchange. Full proofs will be given in a companion technical report [4].

Our translation T departs from T_{GC} in two ways. First, T_{GC} translated c-formulas such as $x \triangleright \varphi \wedge \psi$ by first using $\mathsf{И}$-quantifiers for the local context, then translating $\varphi \wedge \psi$, and finally substituting $n(\mathsf{a})$ for x, resulting in $\mathsf{И}\mathsf{a}.[\![\varphi]\!][n(\mathsf{a})/x] \wedge [\![\psi]\!][n(\mathsf{a})/x]$. In this approach, the head symbol of a translated c-formula was hidden beneath a sequence of $\mathsf{И}$-quantifiers, which made T_{GC} difficult to analyze. Instead, our translation delays $\mathsf{И}$-quantification as long as possible and preserves the head symbol for most formulas: for example, the prior example translates to $[\![x \triangleright \varphi]\!] \wedge [\![x \triangleright \psi]\!]$. Any $\mathsf{И}$-quantification is delayed as long as possible, that is, until the base case for atomic formulas.

The second change is the translation of atomic formulas. As noted earlier, the validity of c-formulas is sensitive to both the *order* and *number* of local parameters in context. To deal with this, we relativize atomic formulas to their local contexts. This is accomplished by adding an argument to each atomic formula symbol for a list of names representing the local context. Let ν^* be a type with constructors $nil : \nu^*$ and $cons : \nu \to \nu^* \to \nu^*$, that is, a type of lists of names. We use a conventional comma-separated list notation for lists: $[\mathsf{a}, \mathsf{b}, \mathsf{c}] = cons(\mathsf{a}, cons(\mathsf{b}, cons(\mathsf{c}, nil)))$. The translation of an atomic c-formula $\sigma \triangleright p\bar{t}$ is $\mathsf{И}\bar{\mathsf{a}}.p^*\ [\bar{\mathsf{a}}]\ \bar{t}[\overline{n_\tau(\mathsf{a})}/\sigma]$, where if $p : \overline{\tau} \to o$ then $p^* : \nu^* \to \overline{\tau} \to o$.

Otherwise, T is similar to T_{GC}. Ordinary $\forall$ and $\exists$-quantified values are lifted to *equivariant* functions applied to lists of names. For example, $\sigma \triangleright \forall x{:}\tau'.p(x)$ was translated to $\mathsf{И}\bar{\mathsf{a}}.\forall h{:}\tau_1 \to \cdots \tau_n \to \tau'.ev(h) \supset p(h\ \overline{n_\tau(\mathsf{a})})$, where each a_i is the name representing x_i, and $ev(x) = \forall a : \nu.a \# x$.

The new translation is shown in full in Figure 6. The function $[\![\cdot]\!]$ translates judgments, contexts, and c-formulas of $FO\lambda^{\nabla}$ to judgments, formula multisets, and formulas of $INL^{\Rightarrow}$ respectively. Note that the context Σ is translated to a set of hypotheses $ev(x)$, one for each $x \in \Sigma$. Here are two examples of the new translation. The formula $\nabla x.p \iff p$ is translated to $\mathsf{И}\mathsf{a}.p^*\ [\mathsf{a}] \iff p^*\ []$. Likewise, we translate $\nabla x, y.p\ x\ y \iff \nabla y, x.p\ x\ y$ to $\mathsf{И}\mathsf{a}, \mathsf{b}.p^*\ [\mathsf{a}, \mathsf{b}]\ (n(\mathsf{a}))(n(\mathsf{b})) \iff \mathsf{И}\mathsf{b}, \mathsf{a}.p^*\ [\mathsf{b}, \mathsf{a}]\ (n(\mathsf{a}))\ (n(\mathsf{b}))$. Neither of these translated formulas is derivable in nominal logic.

Lemma 12. *If $\Sigma \vdash_{FO\lambda^{\nabla}} t : \tau$ then $\Sigma \vdash_{INL^{\Rightarrow}} t : \tau$; in addition, $\Sigma : [\![\Sigma]\!] \Rightarrow ev(t)$. Also, if $\Sigma : \Gamma \Rightarrow \mathcal{A}$ is well-formed then so is $[\![\Sigma : \Gamma \Rightarrow \mathcal{A}]\!]$.*

Proposition 1 (Soundness). *If $\Sigma : \Gamma \Rightarrow \mathcal{A}$ is derivable in $FO\lambda^{\nabla}$ then $[\![\Sigma : \Gamma \Rightarrow \mathcal{A}]\!]$ is derivable in $INL^{\Rightarrow}$.*

Proof. Similar to, but simpler than, the proof for T_{GC}. □

$$
\begin{array}{rl}
[\![\sigma \triangleright \top]\!] = \top & [\![\sigma \triangleright \varphi \vee \psi]\!] = [\![\sigma \triangleright \varphi]\!] \vee [\![\sigma \triangleright \psi]\!] \\
[\![\sigma \triangleright \bot]\!] = \bot & [\![\sigma \triangleright \varphi \supset \psi]\!] = [\![\sigma \triangleright \varphi]\!] \supset [\![\sigma \triangleright \psi]\!] \\
[\![\sigma \triangleright p\ \overline{t}]\!] = \mathit{И}\overline{\mathsf{a}}.p^*\ [\overline{\mathsf{a}}]\ (\overline{t[\overline{n_\tau(\mathsf{a})}/\sigma]}) & [\![\sigma \triangleright \forall x{:}\tau.\varphi]\!] = \forall h{:}\overline{\tau_\sigma}{\to}\tau.ev(h) \supset [\![\sigma \triangleright \varphi[h\sigma/x]]\!] \\
[\![\sigma \triangleright \varphi \wedge \psi]\!] = [\![\sigma \triangleright \varphi]\!] \wedge [\![\sigma \triangleright \psi]\!] & [\![\sigma \triangleright \exists x{:}\tau.\varphi]\!] = \exists h{:}\overline{\tau_\sigma}{\to}\tau.ev(h) \wedge [\![\sigma \triangleright \varphi[h\sigma/x]]\!] \\
[\![\cdot]\!] = \cdot & [\![\sigma \triangleright \nabla x{:}\tau.\varphi]\!] = [\![\sigma, x{:}\tau \triangleright \varphi]\!] \\
 & [\![\Sigma, x{:}\tau]\!] = [\![\Sigma]\!], ev(x) \quad (ev(x) = \forall a{:}\nu.a \mathbin{\#} x)
\end{array}
$$

$$
[\![\Sigma : \Gamma \Rightarrow \mathcal{A}]\!] = \Sigma : [\![\Sigma]\!], [\![\Gamma]\!] \Rightarrow [\![\mathcal{A}]\!]
$$

Fig. 6. Translation T from $FO\lambda^{\nabla}$ to $INL^{\Rightarrow}$

Theorem 3 (Completeness). *If $[\![\Sigma : \Gamma \Rightarrow \mathcal{A}]\!]$ is derivable in $INL^{\Rightarrow}$ then $\Sigma : \Gamma \Rightarrow \mathcal{A}$ is derivable in $FO\lambda^{\nabla}$.*

Proof (Sketch). We break the proof into the following steps:

1. Identify two normal forms for $INL^{\Rightarrow}$ proofs, and show that proofs of translated sequents can be normalized.
2. Show that proofs of the first normal form are proofs of initial sequents.
3. Show that proofs of the second normal form correspond to applications of $FO\lambda^{\nabla}$ rules.

In the analysis to follow, it simplifies matters to eliminate as many nonlogical rules as possible from derivations. By the orthogonality property, we need not consider the rules for abstraction in translated derivations, since abstractions are not used in the translation. In addition, the nonlogical rules F_3 and F_4 can also be eliminated, as we shall now show.

Lemma 13. *Suppose Σ has no name-variables. If $\Sigma \vdash a : \nu$, then for some $\mathsf{a} \in \Sigma$, $\Sigma : \cdot \Rightarrow a \approx \mathsf{a}$.*

Proposition 2. *If $[\![\Sigma : \Gamma \Rightarrow \mathcal{A}]\!]$ is derivable then it has a derivation that does not use F_3 or F_4.*

Proof. To show that F_3 cannot be used in a derivation of a translated sequent, note that $[\![\Gamma]\!]$ and $[\![\mathcal{A}]\!]$ do not mention equality or freshness, and the formulas $[\![\Sigma]\!] = \forall a.a \mathbin{\#} x_1, \ldots, \forall a.a \mathbin{\#} x_n$ cannot be instantiated to $x_i \mathbin{\#} x_i$ since the variables x_i are not of name-type. We can therefore show that no sequent occurring in the derivation of a translated sequent can contain $a \mathbin{\#} a$ using methods similar to those used for consistency and orthogonality.

Consider a subderivation ending with F_4, of the form

$$
\frac{\Sigma : \Gamma, a \mathbin{\#} b \Rightarrow \varphi \quad \Sigma : \Gamma, a \approx b \Rightarrow \varphi}{\Sigma : \Gamma \Rightarrow \varphi}
$$

Name-variables are never introduced in translated derivations, so by Lemma 13, we have $\Sigma \Rightarrow a \approx \mathsf{a}$, $\Sigma : \cdot \Rightarrow b \approx \mathsf{b}$ for some $\mathsf{a}, \mathsf{b} \in \Sigma$. If $\mathsf{a} = \mathsf{b}$ then clearly $\Sigma : \cdot \Rightarrow a \approx b$, so we can use the second subderivation and cut to derive $\Sigma : \Gamma \Rightarrow \varphi$. On the other hand, if $\mathsf{a} \neq \mathsf{b}$ then clearly $\Sigma : \cdot \Rightarrow \mathsf{a} \mathbin{\#} \mathsf{b}$ and also $\Sigma : \cdot \Rightarrow a \mathbin{\#} b$. Using cut and the subderivation $\Sigma : \Gamma, a \mathbin{\#} b$ we can derive $\Sigma : \Gamma \Rightarrow \varphi$. □

Definition 2. *A derivation is in* first normal form *if it uses only the rules $\mathsf{И}L$, $\mathsf{И}R$, hyp, and nonlogical rules.*

A derivation beginning with a left- or right-rule is in second normal form *provided that if the toplevel rule is $\forall L$, $\forall R$, $\exists L$, or $\exists R$, then the next rule used is $\supset L$, $\supset R$, $\wedge L$, or $\wedge R$, respectively.*

Before proving that translated derivations always have normal forms, we need some additional technical machinery. We write $\hat{\varphi}(t)$ for the formula $ev(t) \supset \varphi(t)$; translations of universal c-formulas are always of the form $\forall x.\hat{\varphi}(x)$. We write $\hat{\Gamma}(\bar{t})$ for a set of formulas $\hat{\varphi_1}(t_n), \ldots, \hat{\varphi_n}(t_n)$ such that $\forall x.\hat{\varphi_i}(x) \in [\![\Gamma]\!]$ for each i.

Lemma 14. *If Σ is a $FO\lambda^{\nabla}$ context, $\Sigma\#\bar{\mathsf{a}} \vdash t : \tau$ and $\Sigma\#\bar{\mathsf{a}} : [\![\Sigma]\!] \Rightarrow ev(t)$ then $\Sigma \vdash t : \tau$.*

Lemma 15. *If Σ is a $FO\lambda^{\nabla}$ context, $\Sigma\#\bar{\mathsf{a}} : [\![\Sigma]\!], [\![\Gamma]\!], \hat{\Gamma}(\bar{t}) \Rightarrow ev(t)$ then either $\Sigma : [\![\Gamma]\!] \Rightarrow \varphi$ has a normal derivation for any formula φ, or $\Sigma\#\bar{\mathsf{a}} : [\![\Sigma]\!] \Rightarrow ev(t)$.*

Lemma 16. *If $\Sigma : \Gamma \Rightarrow \varphi$ has a derivation using only* nonbranching *nonlogical rules, then it has either a first normal form derivation or one that starts with F or a logical rule.*

Proposition 3. *If $[\![\Sigma : \Gamma \Rightarrow \mathcal{A}]\!]$ is derivable, then it has a normal derivation.*

Proof (Sketch). First, by Corollary 2 and Proposition 2, $[\![\Sigma : \Gamma \Rightarrow \mathcal{A}]\!]$ must have a derivation that does not use the rules A_1, A_2, A_3, F_3 or F_4.

Because of subtleties involved in the interaction between the F and $\forall L$ rule, we need a stronger induction hypothesis. We prove that if $\Sigma\#\bar{\mathsf{a}} : [\![\Sigma]\!], [\![\Gamma]\!], \hat{\Gamma}(\bar{t}) \Rightarrow [\![\mathcal{A}]\!]$ has a derivation, then $\Sigma : [\![\Sigma]\!], [\![\Gamma]\!] \Rightarrow [\![\mathcal{A}]\!]$ has a normal derivation.

Using Lemma 16, the sequent either has a first normal form derivation (in which case we are done) or begins with F or a logical rule. If it starts with a propositional rule applied to an element of $[\![\Gamma]\!]$, then we are done. The induction steps for F and $\forall L$ are immediate. For $\exists L, \forall R$, we can use the invertibility of $\wedge L$ and $\supset R$ respectively and then use $\forall L$. This leaves the cases for $\exists R$ and for $\supset L$ applied to an element of $\hat{\Gamma}$. For $\supset L$ we must have subderivations of $\Sigma\#\bar{\mathsf{a}} \vdash t : \tau$ and $\Sigma\#\bar{\mathsf{a}} : [\![\Gamma]\!], \hat{\Gamma}(\bar{t}), \varphi(t) \Rightarrow [\![\mathcal{A}]\!]$. Using the lemmas we can show that the witnessing term t does not mention any names, and so we can construct a derivation starting with $\forall L$ and $\supset L$. In the similar case of $\exists R$, we also need the invertibility of $\wedge R$. □

We next show that if the derivation is in first normal form, then the $FO\lambda^{\nabla}$ sequent is derivable. We need two auxiliary facts.

Lemma 17. *Suppose $\overline{x}\#\bar{\mathsf{a}} \vdash t : \tau$ and $\pi \cdot [\bar{\mathsf{a}}] = [\bar{\mathsf{b}}]$. Then $\overline{x}\#\bar{\mathsf{a}}\#\bar{\mathsf{b}} : \cdot \Rightarrow \pi \cdot t \approx t[\mathsf{b}_1/\mathsf{a}_1, \ldots, \mathsf{b}_n/\mathsf{a}_n]$*

Lemma 18. *Suppose that Σ has no name-variables and Γ consists of freshness and equality formulas only. If $\Sigma : \Gamma, p\,\bar{t} \Rightarrow p\,\overline{u}$ then for some permutation π of names in Σ, we have $\Sigma : \Gamma \Rightarrow \pi \cdot \bar{t} \approx \overline{u}$.*

Proof. The proof is by induction on the structure of the derivation. Only the hypothesis and nonbranching nonlogical rules can be involved, of these cases, only F poses a challenge. In the case for F, the π obtained by induction may mention the fresh name a introduced by F; however, a cannot appear in t or u, so $\mathsf{b} = \pi^{-1}(\mathsf{a})$ must not appear in t, and so $\pi' = \pi \circ (\mathsf{a}\ \mathsf{b})$ also works since $\pi' \cdot t = \pi \cdot (\mathsf{a}\ \mathsf{b}) \cdot t = \pi \cdot t = u$. □

Proposition 4. *Let $[\![\Sigma : \Gamma \Rightarrow \mathcal{A}]\!]$ have a first-normal form derivation. Then $\Sigma : \Gamma \Rightarrow \mathcal{A}$ is derivable.*

Proof. If $[\![\Sigma : \Gamma \Rightarrow \mathcal{A}]\!]$ has a first normal form derivation, then $\mathcal{A}$ and some element $\mathcal{B}$ of Γ must be of the form $\sigma \triangleright \nabla \overline{x}.p\ \overline{t}$. Without loss of generality, we consider the case where no ∇-quantifiers appear. After stripping off the initial sequence of $\mathit{И}L$ and $\mathit{И}R$ rules, there must be a subderivation of

$$\Sigma \# \overline{\mathsf{a}} \# \overline{\mathsf{b}} : [\![\Sigma]\!], [\![\Gamma]\!], p^*\ [\overline{\mathsf{a}}]\ \theta(\overline{t}) \Rightarrow p^*\ [\overline{\mathsf{b}}]\ \theta'(\overline{u})$$

for some names $\overline{\mathsf{a}}, \overline{\mathsf{b}}$, where $\theta = [\overline{n(\mathsf{a})}/\sigma]$ and $\theta' = [\overline{n(\mathsf{b})}/\sigma']$. Note that θ and θ' are one-to-one and so invertible on on their ranges, and that $\Sigma \# \mathsf{a} \vdash \theta(\overline{t}) : \overline{\tau}$ (that is, none of the b appear in $\theta(\overline{t})$).

By Lemma 18, there must be a ground permutation π such that $\Sigma : \cdot \Rightarrow \pi \cdot ([\overline{\mathsf{a}}]\ \theta(\overline{t})) \approx [\overline{\mathsf{b}}]\ \theta'(\overline{u})$. Clearly, $\pi \cdot [\overline{\mathsf{a}}] = [\overline{\mathsf{b}}]$, so by Lemma 17 we have $\overline{u}[\overline{n(\mathsf{b})}/\sigma'] = \theta'(\overline{u}) \approx \pi \cdot \theta(\overline{t}) \approx \theta(\overline{t})[\mathsf{b}_1/\mathsf{a}_1, \ldots, \mathsf{b}_n/\mathsf{a}_n] = \overline{t}[\overline{n(\mathsf{b})}/\sigma]$. Since $[\overline{n(\mathsf{b})}/\sigma']$ is invertible, we have $\overline{u} \approx \overline{t}[\overline{n(\mathsf{b})}/\sigma][\sigma'/\overline{n(\mathsf{b})}] = \overline{t}[\sigma'/\sigma]$, which implies $\sigma \triangleright p\ \overline{t} \equiv_\alpha \sigma' \triangleright p\ \overline{u}$. □

Proof (Completeness Theorem). In $FO\lambda^\nabla$, ∇ commutes with all propositional connectives, $\forall$, and $\exists$. Therefore, every judgment is equivalent to one in which ∇-quantifiers only occur around atomic formulas, that is, in subformulas of the form $\nabla \overline{x}.p\ \overline{t}$. So it suffices to consider only judgments of this form.

The proof is by induction on the complexity of the judgment $\Sigma : \Gamma \Rightarrow \mathcal{A}$. If the normalized derivation is of the first form, then by Proposition 4, the sequent is derivable. If the normalized derivation is of the second form, there are many subcases, one for each possible starting left- or right-rule. The cases for propositional rules are straightforward. The remaining cases are those for $\forall$ and $\exists$. We will show that translated sequents derived using $\forall L/R$, $\exists L/R$ in $INL^{\Rightarrow}$ can be derived using $\forall L/R$ and $\exists L/R$ in $FO\lambda^\nabla$.

If the final step of the derivation is $\forall R$, then the derivation must be of the form

$$\dfrac{\dfrac{\Sigma, h : [\![\Sigma]\!], ev(h), [\![\Gamma]\!] \Rightarrow [\![\sigma \triangleright \varphi[h\sigma/x]]\!]}{\Sigma, h : [\![\Sigma]\!], [\![\Gamma]\!] \Rightarrow ev(h) \supset [\![\sigma \triangleright \varphi[h\sigma/x]]\!]}\ \supset R}{\Sigma : [\![\Sigma]\!], [\![\Gamma]\!] \Rightarrow \forall h.ev(h) \supset [\![\sigma \triangleright \varphi[h\sigma/x]]\!]}\ \forall R$$

Note that $[\![\Sigma]\!], ev(h) = [\![\Sigma, h]\!]$, so the topmost sequent is of the form $[\![\Sigma, h : \Gamma \Rightarrow \sigma \triangleright \varphi[h\sigma/x]]\!]$. By induction, $\Sigma, h : \Gamma \Rightarrow \sigma \triangleright \varphi[h\sigma/x]$ is derivable, and using $\forall R$, we conclude $\Sigma : \Gamma \Rightarrow \sigma \triangleright \forall x.\varphi$. The $\exists L$ case is similar.

If the final inference is $\forall L$, then the derivation must be of the form

$$\dfrac{\Sigma \vdash t : \overline{\tau_\sigma} \to \tau \qquad \dfrac{\Sigma : [\![\Sigma]\!], [\![\Gamma]\!] \Rightarrow ev(t) \quad \Sigma : [\![\Sigma]\!], [\![\Gamma]\!], [\![\sigma \triangleright \varphi[h\sigma/x]]\!][t/h] \Rightarrow [\![\mathcal{A}]\!]}{\Sigma : [\![\Sigma]\!], [\![\Gamma]\!], ev(t) \supset [\![\sigma \triangleright \varphi[h\sigma/x]]\!] \Rightarrow [\![\mathcal{A}]\!]}\ \supset L}{\Sigma : [\![\Sigma]\!], [\![\Gamma]\!], \forall h.ev(h) \supset [\![\sigma \triangleright \varphi[h\sigma/x]]\!] \Rightarrow [\![\mathcal{A}]\!]}\ \forall L$$

Since Σ does not mention name-constants, we have $\Sigma \vdash t : \overline{\tau_\sigma} \to \tau$ and also $\Sigma, \sigma \vdash t\ \sigma\ :\ \tau$ in $FO\lambda^\nabla$. Note that $[\![\sigma \triangleright \varphi[h\sigma/x]]\!][t/h] = [\![\sigma \triangleright \varphi[t\ \sigma/x]]\!]$ so we also have $\Sigma : [\![\Sigma]\!], [\![\Gamma]\!], [\![\sigma \triangleright \varphi[t\ \sigma/x]]\!] \Rightarrow [\![\mathcal{A}]\!]$, which is the same as $[\![\Sigma : \Gamma, \sigma \triangleright \varphi[t\ \sigma/x] \Rightarrow \mathcal{A}]\!]$. By induction, $\Sigma : \Gamma, \sigma \triangleright \varphi[t\ \sigma/x] \Rightarrow \mathcal{A}$ is derivable, and since $\Sigma, \sigma \vdash t\ \sigma : \tau$, we can use $\forall L$ to conclude that $\Sigma : \Gamma, \sigma \triangleright \forall x.\varphi \Rightarrow \mathcal{A}$. The $\exists R$ case is similar. □

Remark 3. If we modify the translation step for atomic formulas by defining $[\![\sigma \triangleright p\ \overline{t}]\!] = И\overline{\mathsf{a}}.p\ \overline{t}[\overline{n(\mathsf{a})}/\sigma]$ then we obtain a translation T_{WX} that is essentially the same as T_{GC}, and is complete with respect to $FO\lambda^\nabla$ with ∇-weakening and exchange principles.

We write $\theta : \sigma \hookrightarrow \sigma'$ to indicate that θ is a partial injective renaming mapping σ to σ'. We say that c-formulas are WX-equivalent ($\sigma \triangleright A \equiv_{WX} \sigma' \triangleright B$) if there is a $\theta : \sigma \hookrightarrow \sigma'$ such that $\theta(A) = B$. For example, $x, y \triangleright p(x, y) \equiv_{WX} y, x, z \triangleright p(x, y)$. Note that $\equiv_{WX}$ subsumes α-equivalence. Let $FO\lambda^\nabla_{WX}$ be $FO\lambda^\nabla$ except that atomic c-formulas are considered equal modulo $\equiv_{WX}$.

It is not difficult to show that the formulas $\nabla x.\varphi \iff \varphi$ (where $x \notin FV(\varphi)$) and $\nabla x.\nabla y.\varphi \iff \nabla y.\nabla x.\varphi$ are derivable in $FO\lambda^\nabla_{WX}$ for any formula φ. In addition, using the same techniques as above, we can show that the translation is sound and complete relative to $FO\lambda^\nabla_{WX}$. The proof is the same as that for completeness relative to $FO\lambda^\nabla$, except that we need to show that Proposition 4 holds for atomic c-formulas equal modulo $\equiv_{WX}$ instead of α-equivalence.

4 Related and Future Work

Besides previous formalizations of nominal logic by Pitts, Gabbay, and Cheney (surveyed in Section 1.1), several other logics and type systems have considered rules for И-quantified formulas or types. Caires and Cardelli [1] investigated a logic incorporating proof rules for И-quantified formulas based on maintaining a set of side-conditions involving freshness constraints. However, the freshness constraints are not formulas of their logic. These rules are similar in spirit to (and partly inspired) the slice-based rules of FL and FL_{Seq}. Another related system is the type system of Nanevski [10], which includes rules similar to those of FL for И-quantified types. A third closely related system is Schöpp and Stark's dependent type theory for names and binding [13], in which a bunched context is used to store freshness information. Our freshness contexts and rules for И are simpler special cases of the contexts and rules in their theory.

There are several directions for future work. $NL^\Rightarrow$ may be useful for developing an improved proof-theoretic semantics for nominal logic programming. Natural deduction calculi or type theories for nominal logic based on our approach could be used as the basis of proof checkers and interactive theorem provers for nominal logic. The existence of translations from $FO\lambda^\nabla$ to $NL^\Rightarrow$ suggest that $FO\lambda^\nabla$ can be interpreted using the semantics of nominal logic. Moreover, a semantic approach may lead to a simpler proof of the completeness of the translations.

5 Conclusions

This paper makes two contributions. First, we present a new sequent calculus for nominal logic which avoids the *slices* used in the rules for И in FL and FL_{Seq}. Instead,

our calculus deals with И using *freshness contexts* that encode freshness information as well as typing information. Although this is partly a matter of taste, we believe that our approach is easier to use and analyze and provides a more transparent reading of И as a proof search operation than any previous system. In particular, the proofs of cut-elimination and conservativity relative to Pitts' axiomatization seem simpler and require fewer technical lemmas than previous attempts.

The second contribution of this paper is an improved translation from $FO\lambda^{\nabla}$ to intuitionistic nominal logic ($INL^{\Rightarrow}$), which explains the behavior of the ∇-quantifier in terms of И. We show that $FO\lambda^{\nabla}$ can be soundly and completely interpreted in $INL^{\Rightarrow}$, so any argument carried out in $FO\lambda^{\nabla}$ can also safely be carried out in $INL^{\Rightarrow}$. In addition, we argued that the translation originally proposed by Gabbay and Cheney is complete relative to $FO\lambda^{\nabla}$ with weakening and exchange for ∇.

Acknowledgments Discussions with Ian Stark and Uli Schöpp and the anonymous reviewers' comments were of great value in improving this paper.

References

1. Luís Caires and Luca Cardelli. A spatial logic for concurrency–II. *Theoretical Computer Science*, 322(3):517–565, September 2004.
2. J. Cheney and C. Urban. Alpha-Prolog: A logic programming language with names, binding and alpha-equivalence. In *Proc. 20th Int. Conf. on Logic Programming (ICLP 2004)*, number 3132 in LNCS, pages 269–283, 2004.
3. James Cheney. A simpler proof theory for nominal logic. Technical Report EDI-INF-RR-0237, LFCS, University of Edinburgh, November 2004.
4. James Cheney. A sound and complete translation of generic judgments into nominal logic. Technical report, LFCS, University of Edinburgh, 2005. In preparation.
5. James R. Cheney. *Nominal Logic Programming*. PhD thesis, Cornell University, Ithaca, NY, August 2004.
6. M. J. Gabbay. Fresh logic: A logic of FM, 2003. Submitted.
7. M. J. Gabbay and J. Cheney. A proof theory for nominal logic. In *Proceedings of the 19th Annual IEEE Symposium on Logic in Computer Science (LICS 2004)*, pages 139–148, Turku, Finland, 2004.
8. M. J. Gabbay and A. M. Pitts. A new approach to abstract syntax with variable binding. *Formal Aspects of Computing*, 13:341–363, 2002.
9. Dale Miller and Alwen Tiu. A proof theory for generic judgments: extended abstract. In *Proc. 18th Symp. on Logic in Computer Science (LICS 2003)*, pages 118–127. IEEE Press, 2003.
10. Aleksandar Nanevski. Meta-programming with names and necessity. In *Proc. 8th ACM SIGPLAN Int. Conf. on Functional Programming*, pages 206–217. ACM Press, 2002.
11. Sara Negri and Jan von Plato. *Structural Proof Theory*. Cambridge University Press, 2001.
12. A. M. Pitts. Nominal logic, a first order theory of names and binding. *Information and Computation*, 183:165–193, 2003.
13. Ulrich Schöpp and Ian Stark. A dependent type theory with names and binding. In *Proceedings of the 2004 Computer Science Logic Conference*, number 3210 in Lecture notes in Computer Science, pages 235–249, Karpacz, Poland, 2004.
14. C. Urban, A. M. Pitts, and M. J. Gabbay. Nominal unification. *Theoretical Computer Science*, 323(1–3):473–497, 2004.