A Congruence Rule Format for Name-Passing Process Calculi

Marcelo Fiore [1] and Sam Staton [2]

Computer Laboratory, University of Cambridge

Abstract

We introduce a GSOS-like rule format for name-passing process calculi. Specifications in this format correspond to theories in nominal logic. The intended models of such specifications arise by initiality from a general categorical model theory. For operational semantics given in this rule format, a natural behavioural equivalence — a form of open bisimilarity — is a congruence.

Introduction

A significant strand of research in semantics concerns defining and establishing properties of formats for operational rules. By moving away from a particular syntax and semantics one can simultaneously study a whole class of calculi, becoming instead concerned with the intrinsic nature of the kinds of system that the formats allow.

In this vein, the present work provides an analysis of the congruence properties of bisimilarity for name-passing systems, with the π-calculus [16] being the paradigmatic example. Specifically we ask: What is a name-passing process calculus, and when is its behavioural equivalence a congruence?

A variety of rule formats have been proposed for conventional process calculi [see e.g. 1]. The most relevant existing format to the present article is a positive version of the GSOS rule format of Bloom et al. [4]. The positive

[1] Research partially supported by an EPSRC Advanced Research Fellowship
[2] Research partially supported by EPSRC grant GR/T22049/01

GSOS format only permits operational semantics given by rules of the form

$$\frac{\bigcup_{i=1}^{l}\left\{X_i \xrightarrow{a_{ij}} Y_{ij} \mid 1 \le j \le m_i\right\}}{\mathsf{op}(X_1,\ldots,X_l) \xrightarrow{a} C[\vec{X},\vec{Y}]}$$

with all the variables distinct, $l \ge 0$ the arity of op, with $m_i \ge 0$, and with $C[\vec{X},\vec{Y}]$ a context with free variables including at most the X_i's and Y_{ij}'s. In this setting, Bloom et al. showed that various properties hold of the induced system, and, in particular, that bisimilarity is a congruence.

The present article presents a rule format that is relevant for name-passing process calculi. This format is interesting in itself, but may also be viewed as a stepping stone towards a more thorough and rigorous understanding of some important phenomena in modern semantics research.

Conventional rule formats are inappropriate in the context of name-passing

The GSOS format, like most existing formats, is inadequate for name-passing calculi, for a variety of reasons. Firstly, the syntax of name-passing calculi typically involves binding, and the semantics is specified using capture-avoiding substitution. To properly define capture-avoiding substitution on terms it is necessary to work with terms up-to α-equivalence. This means that the semantics, a transition system, must be defined over terms up-to α-equivalence.

A second complication is that rule-based definitions of name-passing calculi typically have side-conditions on the rules. For instance, Milner, Parrow and Walker [16] have proposed the following specification of scope opening.

$$\frac{P \xrightarrow{\bar{x}y} P'}{\nu y.\, P \xrightarrow{\bar{x}(w)} P'\{w/y\}} \quad \begin{array}{l} y \neq x \\ w \notin \mathrm{fn}(\nu y.\, P') \end{array}$$

Thus a crucial role is played by side-conditions that ensure the distinctness and freshness of names.

A third complication of name-passing systems is that the appropriate notion of bisimilarity is not the usual notion for labelled transition systems. Indeed, there is considerable debate as to what the most appropriate notion of bisimilarity is. But few would argue that the π-calculus processes

$$P_1 = \nu a.\, \bar{c}a.\, 0 \qquad \text{and} \qquad P_2 = \nu a.\, \bar{c}a.\, (\nu b.\, \bar{b}d.\, 0)$$

should be distinguished, even though

$$P_1 =_\alpha \nu d.\, \bar{c}d.\, 0 \xrightarrow{\bar{c}(d)} 0$$

— a transition that cannot be *exactly* matched by P_2, as d is free in P_2. The point here is that in any definition of bisimulation for the π-calculus, any bound data in the labels is required to be suitably fresh. Thus notions of freshness are important at this basic level.

Even with such adjustments to the usual notions of bisimulation, bisimilarity is in general not a congruence for the full π-calculus. For instance, in the π-calculus with a matching operator, neither

$$Q_1 = 0 \qquad \text{nor} \qquad Q_2 = [a = b].\, \bar{a}b.\, 0$$

can perform any action; the context $c(a).\,(-) \mid \bar{c}b.\, 0$ will distinguish the processes, though, because

$$c(a).\, Q_2 \mid \bar{c}b.\, 0 \xrightarrow{\tau} [b = b].\, \bar{b}b.\, 0 \mid 0 \xrightarrow{\bar{b}b} 0 \mid 0 \quad ,$$

a trace that cannot be matched by $c(a).\, Q_1 \mid \bar{c}b.\, 0$. In this article, we resolve this difficulty by adopting an approach related to the ideas of *open bisimilarity* introduced by Sangiorgi [19]. We say that a (ground) bisimulation is a **wide-open** bisimulation if it is closed under all substitutions, and wide-open bisimilarity is defined to be the greatest such bisimulation. Wide-open bisimilarity is a congruence for the π-calculus, and it is with this notion that we are concerned. It is a notion that has been studied by various authors, under different names [e.g. 17, 5, 9]. Furthermore, as will be seen, it arises naturally from our mathematical model.

Transition system specifications as nominal logic theories

Specifications of name-passing systems refer explicitly to the freshness of name parameters, and so it is helpful to see these specifications as theories in a logical framework that has facilities for such assertions. In this article, we adopt Pitts's work on nominal logic [18]. A nominal logic theory is a conventional first-order theory in which certain functions and relation symbols must be present, and these must satisfy particular axioms. For instance, there must be a sort of names, and a 'freshness' relation symbol permitting assertions of the form "the name a is fresh for the expression x".

Since a nominal logic theory is only a first-order theory, it is relatively easy to understand what a specification is. For instance, the conventional presentations of the π-calculus are essentially first-order, given a fixed interpretation for notions of freshness.

The fixed interpretation of certain parts of the language is important in nominal logic. As any other first-order theory, it is straightforward to interpret a nominal logic theory set-theoretically. But the notions of name, and of freshness, suggest a particular, canonical interpretation, and for these reasons it is more natural to study models of nominal logic within the universe of *nominal sets*.

Through the axioms of nominal logic, α-equivalence is built in to the equality of the logic. On the other hand, it is important that, at the level of the *syntax* of the logic, seemingly α-equivalent terms are distinguished. For example, in the usual rule for late input semantics,

$$\frac{—}{c(a).\,P \xrightarrow{c(a)} P} \tag{1}$$

one could naively presume that "a is binding in the metavariable P", and therefore that "$c(a).\,P =_\alpha c(c).\,P$". But the rule

$$\frac{—}{c(c).\,P \xrightarrow{c(c)} P} \tag{2}$$

is more restrictive than rule (1). Under a naive, first-order logic interpretation of rule (2), the term $c(a).\,[c = c].\,0$ is inactive; notice that

$$c(a).\,[c = c].\,0 \;\neq_\alpha\; c(c).\,[c = c].\,0 \quad .$$

Thus it is important to be careful, and indeed formal, about these issues.

For such reasons it is necessary to restrict the class of permitted rules. With the wrong input rule (2) above, the π-calculus context $c(a).\,(-)$ is able to distinguish between two bisimilar processes, 0 and $[c = c].\,0$ (even in the context of wide-open bisimilarity).

When working informally, authors often adopt (sometimes tacitly) a 'Barendregt variable convention' [e.g. 20, Conv. 1.10] to eschew such counter-intuitive and absurd behaviour. When rules are considered as formulas of nominal logic, these problems can be understood formally, and without reference to informal conventions. In this setting, through our rule format, we impose conditions about the appearance of variables in rules, and by doing so, we are able to make precise the way that these problems can be prevented.

We stress that there are other logical frameworks that deal with names, binding and freshness; consider, for example the $FO\lambda^{\Delta\nabla}$ framework of Miller and Tiu [15]. In fact, our transition system specifications are not nominal logic theories per se, but rather syntactic structures that give rise to theories, and it should also be possible to extract $FO\lambda^{\Delta\nabla}$ theories from our transition system specifications.

Ziegler et al. [24] have proposed a format for name-passing specifications in the

$FO\lambda^{\Delta\nabla}$ framework, and have a congruence result. By working with schematic metavariables, they are able to avoid some of the uglier aspects of name binding that we encounter here, but this is arguably at the cost of taking a higher level of abstraction, a step removed from the day-to-day intuitions of the working operational semanticist.

The best way to consider and develop rule formats for systems with binding and freshness remains an important matter for debate and research.

A congruence result from a categorical model theory

The main result of this article is that for any name-passing system defined by a specification in our format, wide-open bisimilarity is a congruence. One way to prove this result would be to adapt a proof of a conventional result, e.g. from [1, 4], to this setting. Here, we adopt a different, model-theoretic approach.

The fundamental work of Turi and Plotkin [22] exposed the specification of a system in the GSOS format as the specification of a coalgebra by initial algebra recursion. The conditions about the structure and appearance of variables in the GSOS format have been shown to amount to a naturality condition for recursion data. By taking this category-theoretic approach, GSOS-like specifications can be understood at an abstract level. The categorical notions are relevant not just for conventional bisimilarity over syntax without binding, but, by changing the base category and the (behaviour/syntax) endofunctors involved, for wide-open bisimilarity and syntax with binding and substitution [see e.g. 9].

In this article, a connection is made between this abstract theory and the concrete notions of transition system specification that we introduce. From this perspective, the difficult part of the congruence result is the proof of naturality for a certain family of functions. This is not a trivial exercise by any means, but at least the general structure and nature of the result can be guided by model-theoretic considerations.

Other related work

The rule format presented here, which first appeared in [8], is not the only rule format for name-passing systems. The closest result to ours is that of Ziegler et al. [24], mentioned above, although our result is obtained in quite a different way.

In earlier work, Bernstein [2] demonstrated that her congruence format admits a specification of the π-calculus. Bernstein's format does not explictly cater for variable binding and substitution — this has to be encoded. For this reason, the π-calculus equivalence that she considers is a peculiar variant of open bisimilarity, in which the terms 0 and $[a = a].\,0$ are considered distinct.

In other work, Weber and Bloom [23] have designed a framework for adding GSOS-like operators to the π-calculus. There is a built-in restriction operator, which distributes over certain operators. For the binding in this restriction operator, syntax is considered up-to α-equivalence, while other operators involving binding are considered in the style of higher-order abstract syntax. A congruence result is established for a notion of equivalence that appears to be open bisimilarity, although this is not made explicit.

More broadly, we recall that there have been substantial set-theoretic studies of the model theory for specifications involving variable-binding [e.g. 3, 11, 14], although none of these studies provide congruence results for name-passing calculi.

Open bisimilarity

A potential criticism of wide-open bisimilarity is that it treats bound input actions in the same way as bound output actions. While it is reasonable to close under all substitutions for bound input data, it is perhaps less reasonable to perform this closure for bound output data. To resolve this anomaly, Sangiorgi [19, Sec. 7] has proposed the notion of **open bisimilarity**. For instance, the processes

$$P_1 = \nu a.\, \bar{c}a.\, [a = c].\, \tau.\, 0 \qquad \text{and} \qquad P_2 = \nu a.\, \bar{c}a.\, 0 \tag{3}$$

are open bisimilar, although not wide-open bisimilar.

The rule format that we present here does not guarantee the congruence of genuine open bisimilarity. For instance, the following rule is legal in our format.

$$\frac{P \xrightarrow{\bar{c}(a)} Q \quad P' \xrightarrow{\bar{c}d} Q'}{P \,|\, P' \xrightarrow{\tau} \{d/a\}Q \,|\, Q'} \tag{4}$$

This rule is a (perhaps perverse) modification of the usual rule for communication, that allows output actions to synchronise with bound output actions, rather than with input actions. In the presence of this rule, for the processes introduced in (3) we have a trace

$$P_1 \,|\, (\bar{c}c.\, 0) \xrightarrow{\tau} ([c = c].\, \tau.\, 0) \,|\, 0 \xrightarrow{\tau} 0 \,|\, 0$$

that cannot be matched by $(P_2 \mid (\bar{c}c.\,0))$. Thus, with this rule, the context $((-) \mid (\bar{c}c.\,0))$ is able to distinguish the open bisimilar processes P_1 and P_2.

(In the format of Ziegler et al. [24], the rule (4) is forbidden by the type system.)

In the future we intend to redevelop the present work in the context of genuine open bisimilarity, as a first step towards more sophisticated calculi. We remark that Ghani, Yemane, and Victor [13] have already proposed a categorical model theory.

Remark. In [13], non-determinism is introduced by a free semilattice monad on a category of states with substitutions, directly. This renders the model inappropriate for the π-calculus because, intuitively, the next-state function for the π-calculus does not entirely respect the substitution structure of their model. It seems that this problem could be remedied by adopting the structured coalgebra approach that we take here, or alternatively by introducing non-determinism via the powerobject functor of their presheaf topos. Details have to be worked out.

Synopsis

This article comprises four sections. In the first, we introduce various important nominal logic theories, and introduce a notion of transition system specification as a particular kind of nominal logic theory. In the second section, we study the model theory of nominal logic, by considering models of theories in the category of nominal sets, and related categories. The 'intended model' of a transition system specification is seen to be the initial model of a theory in the category of nominal sets.

We devote the third section to introducing and explaining our conditions on transition system specifications. In the fourth section, we develop a correspondence with the categorical model theory, and thus establish the congruence result.

Acknowledgements

We enjoyed discussing this work with many people, and are grateful for various suggestions, including those from anonymous referees. Part of this article is based on S. Staton's Ph.D. research [21], and he would like to thank his examiners, A. M. Pitts and G. D. Plotkin.

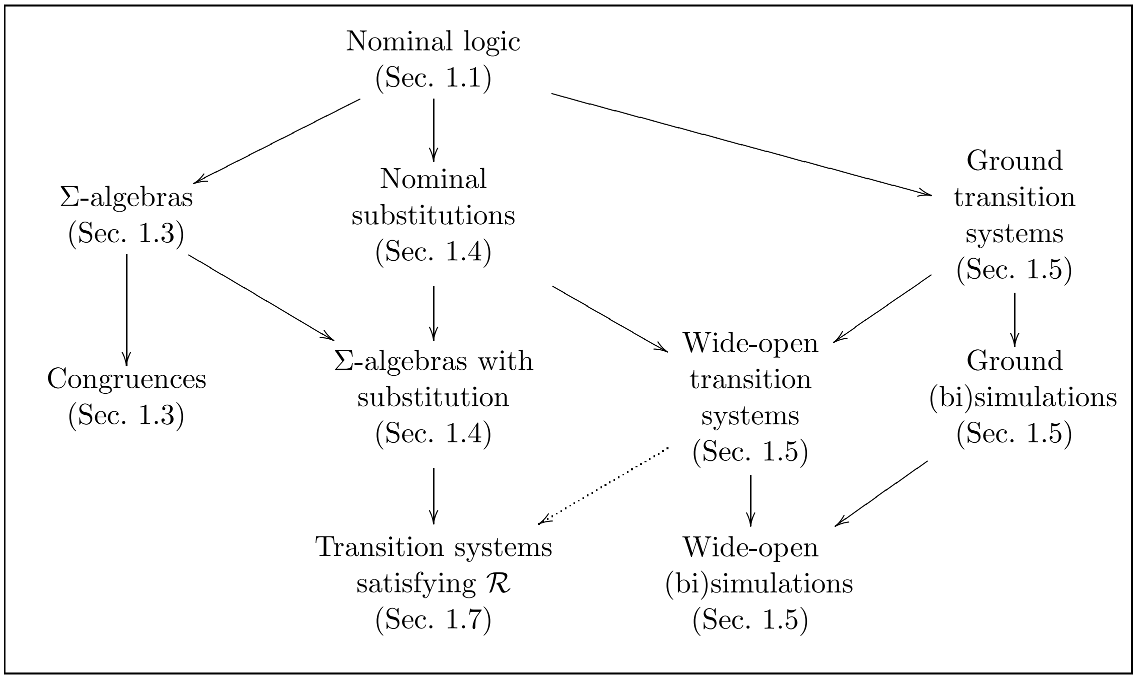
Nominal logic (Sec. 1.1)
Σ-algebras (Sec. 1.3)
Nominal substitutions (Sec. 1.4)
Ground transition systems (Sec. 1.5)
Congruences (Sec. 1.3)
Σ-algebras with substitution (Sec. 1.4)
Wide-open transition systems (Sec. 1.5)
Ground (bi)simulations (Sec. 1.5)
Transition systems satisfying $\mathcal{R}$ (Sec. 1.7)
Wide-open (bi)simulations (Sec. 1.5)

Fig. 1. The nominal logic theories introduced in Section 1. (A solid arrow $A \to B$ indicates that the signature and axioms of A are included in the signature and axioms of B. The arrow from wide-open transition systems to transition systems satisfying $\mathcal{R}$ is dotted because the signature is included although the axioms are not.)

1 Specifications of name-passing calculi

We begin this section by recalling Pitts's nominal logic (Sec. 1.1). The concepts that we introduce in the remainder of this section fall into three distinct classes.

- There are nominal logic theories of nominal substitution (Sec. 1.4), and of transition system and bisimulation (Sec. 1.5).
- There are two formal, syntactic notions: signatures for abstract syntax with variable binding (Sec. 1.2) and transition system specifications (Sec. 1.6).
- There are nominal logic theories that are extracted from these notions (Sec. 1.3, Sec. 1.7 resp.).

In Figure 1 we summarise the nominal logic theories that are introduced in this section, and the relationships between them.

1.1 Nominal logic

We recall some aspects of Pitts's nominal logic [18].

For our purposes, a **nominal logic signature** is a signature for multi-sorted first-order logic that satisfies the following additional requirements.

- There is a distinguished sort N of names.
- For every sort X there is a sort $[\mathsf{N}]\mathsf{X}$. (Informally: $[\mathsf{N}]\mathsf{X}$ is the sort of (name,term) pairs up-to α-equivalence.)
- For every sort X there is an function symbol $\mathsf{bind} : (\mathsf{N}, \mathsf{X}) \to [\mathsf{N}]\mathsf{X}$, written infix: $\mathsf{bind}(\mathsf{a}, t)$ is written $\langle \mathsf{a} \rangle t$. (Informally: $\langle \mathsf{a} \rangle t$ is t with a bound in it.)
- For every sort X, there is an function symbol $\mathsf{swap} : (\mathsf{N}, \mathsf{N}, \mathsf{X}) \to \mathsf{X}$, written infix: $\mathsf{swap}(\mathsf{a}, \mathsf{a}', t)$ is written $(\mathsf{a}\,\mathsf{a}') \cdot t$. (Informally: $(\mathsf{a}\,\mathsf{a}') \cdot t$ is the term t with a and a' swapped.)
- For every sort X, there is a distinguished relation $\#$ with arity (N, X). (Informally: $\mathsf{a}\#t$ means that a is fresh for t.)

The function symbols and relation symbols of a nominal logic signature will be called **non-logical** if they are surplus to these requirements.

A **nominal logic theory** is a theory of first-order logic (with equality) whose signature is a nominal logic signature, and whose axioms include those described in Figure 2. The axioms that are not required by nominal logic will be called **non-logical axioms**.

In Figure 2, axiom (F4), we have written $\mathsf{a}\#\vec{\mathtt{x}}$ as an abbreviation for the finite conjunction of formulas $\mathsf{a}\#\mathtt{x}_i$, with $\mathtt{x}_i$ ranging over $\vec{\mathtt{x}}$. We use this convention throughout the article.

The reader is reminded that α-equivalence plays no role in the syntax of terms for a nominal logic signature: despite the suggestive notation for the bind symbol, we are working with conventional first-order logic. On the other hand, α-equivalence is present in the equality of a nominal logic theory, for terms of sort $[\mathsf{N}]\mathsf{X}$, as a result of axioms (A1) and (A2).

1.2 Signatures for abstract syntax with variable binding

The syntax of the π-calculus is built from various operators. For instance, the input phrase $c(a).t$, which will be written $\mathsf{inp}(c, \langle a \rangle t)$, has one name parameter c and one term parameter t with the name a bound; the output phrase $\bar{c}d.t$ will be written $\mathsf{out}(c, d, t)$, and has two name parameters c, d, and one term parameter t with no names bound. We will also use the restriction phrase $\nu a.t$, written $\mathsf{res}(\langle a \rangle t)$, with one term parameter with a name bound in it; the parallel phrase $t \,|\, t'$, written $\mathsf{par}(t, t')$, which has two term parameters, neither with any names bound; and the inactive process 0, written nil, which is a constant, and has no parameters.

Properties of swapping (for every sort X):

$$\forall \mathsf{a} : \mathsf{N}.\ \forall \mathtt{x} : \mathsf{X}.\ (\mathsf{a}\ \mathsf{a}) \cdot \mathtt{x} = \mathtt{x} \tag{S1}$$

$$\forall \mathsf{a}, \mathsf{a}' : \mathsf{N}.\ \forall \mathtt{x} : \mathsf{X}.\ (\mathsf{a}\ \mathsf{a}') \cdot (\mathsf{a}\ \mathsf{a}') \cdot \mathtt{x} = \mathtt{x} \tag{S2}$$

Name permutation:

$$\forall \mathsf{a}, \mathsf{a}' : \mathsf{N}.\ (\mathsf{a}\ \mathsf{a}') \cdot \mathsf{a} = \mathsf{a}' \tag{S3}$$

Equivariance (for every $n \in \mathbb{N}$, every function symbol $f : (\mathsf{X}_1, \ldots, \mathsf{X}_n) \to \mathsf{X}$ and every relation symbol R of arity $(\mathsf{X}_1, \ldots, \mathsf{X}_n)$):

$$\forall \mathsf{a}, \mathsf{a}' : \mathsf{N}.\ \forall \mathtt{x}_1 : \mathsf{X}_1.\ \ldots\ \forall \mathtt{x}_n : \mathsf{X}_n.\quad (\mathsf{a}\ \mathsf{a}') \cdot f(\mathtt{x}_1, \ldots, \mathtt{x}_n) = f((\mathsf{a}\ \mathsf{a}') \cdot \mathtt{x}_1, \ldots, (\mathsf{a}\ \mathsf{a}') \cdot \mathtt{x}_n) \tag{E3}$$

$$\forall \mathsf{a}, \mathsf{a}' : \mathsf{N}.\ \forall \mathtt{x}_1 : \mathsf{X}_1.\ \ldots\ \forall \mathtt{x}_n : \mathsf{X}_n.\quad R(\mathtt{x}_1, \ldots, \mathtt{x}_n) \implies R((\mathsf{a}\ \mathsf{a}') \cdot \mathtt{x}_1, \ldots, (\mathsf{a}\ \mathsf{a}') \cdot \mathtt{x}_n) \tag{E4}$$

Pitts [18] explicitly includes axioms for equivariance of swap (E1), of # (E2), and of bind (E5).

Freshness (for every sort X, and every finite sequence of sorts $\vec{\mathsf{X}}$):

$$\forall \mathsf{a}, \mathsf{a}' : \mathsf{N}.\ \forall \mathtt{x} : \mathsf{X}.\ \mathsf{a}\#\mathtt{x} \wedge \mathsf{a}'\#\mathtt{x} \implies (\mathsf{a}\ \mathsf{a}') \cdot \mathtt{x} = \mathtt{x} \tag{F1}$$

$$\forall \mathsf{a}, \mathsf{a}' : \mathsf{N}.\ \mathsf{a}\#\mathsf{a}' \iff \mathsf{a} \neq \mathsf{a}' \tag{F2}$$

$$\forall \vec{\mathtt{x}} : \vec{\mathsf{X}}.\ \exists \mathsf{a} : \mathsf{N}.\ \mathsf{a}\#\vec{\mathtt{x}} \tag{F4}$$

Properties of abstraction (for every sort X):

$$\forall \mathsf{a}, \mathsf{a}' : \mathsf{N}.\ \forall \mathtt{x}, \mathtt{x}' : \mathsf{X}.\quad \langle \mathsf{a} \rangle \mathtt{x} = \langle \mathsf{a}' \rangle \mathtt{x}' \iff (\mathsf{a} = \mathsf{a}' \wedge \mathtt{x} = \mathtt{x}') \vee (\mathsf{a}'\#\mathtt{x} \wedge \mathtt{x}' = (\mathsf{a}\ \mathsf{a}') \cdot \mathtt{x}) \tag{A1}$$

$$\forall \mathtt{y} : [\mathsf{N}]\mathsf{X}.\ \exists \mathsf{a} : \mathsf{N}.\ \exists \mathtt{x} : \mathsf{X}.\ \mathtt{y} = \langle \mathsf{a} \rangle \mathtt{x} \tag{A2}$$

Fig. 2. Axioms of nominal logic [18, App. A].

Thus we are led to the following, essential standard, notion of signature.

Definition 1.1. An **algebraic binding signature** Σ consists of a finite set of operators together with, for each operator op, a **name-arity** $\mathrm{ar}_{\mathsf{N}}(\mathsf{op}) \in \mathbb{N}$ and a **term-arity** $\mathrm{ar}_{\mathsf{X}}(\mathsf{op}) \in \mathbb{N}$. To each $j \in [1, \mathrm{ar}_{\mathsf{X}}(\mathsf{op})]$ is associated a **binding depth** $\mathrm{bdep}_{\mathsf{op}}(j) \in \mathbb{N}$.

(Here, and throughout this article, for natural numbers $a, b \in \mathbb{N}$, we write $[a, b]$ for the interval $\{n \in \mathbb{N} \mid a \leq n \leq b\}$.)

For the fragment of the π-calculus recalled above we have an algebraic binding signature Σ_π with operators $\{\mathsf{inp}, \mathsf{out}, \mathsf{res}, \mathsf{par}, \mathsf{nil}\}$. Arities are assigned as follows.

op	$\mathrm{ar}_{\mathsf{N}}(\mathsf{op})$	$\mathrm{ar}_{\mathsf{X}}(\mathsf{op})$	$\mathrm{bdep}_{\mathsf{op}}$
inp	1	1	$\mathrm{bdep}_{\mathsf{inp}}(1) = 1$
out	2	1	$\mathrm{bdep}_{\mathsf{out}}(1) = 0$
res	0	1	$\mathrm{bdep}_{\mathsf{res}}(1) = 1$
par	0	2	$\mathrm{bdep}_{\mathsf{par}}(j) = 0 \quad (j = 1, 2)$
nil	0	0	

The **terms** of an algebraic binding signature Σ are inductively defined as follows. We fix a set $\mathtt{N}$ of name variables and a set $\mathtt{X}$ of term variables. (We use a typewriter font for variables, throughout this article.) Then:

- every term variable $\mathtt{x} \in \mathtt{X}$ is a term, and
- for any operator op, any name variables $\mathsf{c}_1, \ldots, \mathsf{c}_{\mathrm{ar}_{\mathsf{N}}(\mathsf{op})}$, $\mathsf{a}^1_1, \ldots, \mathsf{a}^1_{\mathrm{bdep}_{\mathsf{op}}(1)}, \ldots$, $\mathsf{a}^{\mathrm{ar}_{\mathsf{X}}(\mathsf{op})}_1, \ldots, \mathsf{a}^{\mathrm{ar}_{\mathsf{X}}(\mathsf{op})}_{\mathrm{bdep}_{\mathsf{op}}(\mathrm{ar}_{\mathsf{X}}(\mathsf{op}))}$, and any terms $t_1, \ldots, t_{\mathrm{ar}_{\mathsf{X}}(\mathsf{op})}$, there is a term

  $$\mathsf{op}\left(\mathsf{c}_1, \ldots, \mathsf{c}_{\mathrm{ar}_{\mathsf{N}}(\mathsf{op})}, \langle \mathsf{a}^1_1, \ldots, \mathsf{a}^1_{\mathrm{bdep}_{\mathsf{op}}(1)}\rangle t_1, \ldots \langle \mathsf{a}^{\mathrm{ar}_{\mathsf{X}}(\mathsf{op})}_1, \ldots, \mathsf{a}^{\mathrm{ar}_{\mathsf{X}}(\mathsf{op})}_{\mathrm{bdep}_{\mathsf{op}}(\mathrm{ar}_{\mathsf{X}}(\mathsf{op}))}\rangle t_{\mathrm{ar}_{\mathsf{X}}(\mathsf{op})}\right)$$

  which we will often abbreviate as

  $$\mathsf{op}\left((\mathsf{c}_i)_{i\in[1,\mathrm{ar}_{\mathsf{N}}(\mathsf{op})]}, \left(\langle \mathsf{a}^j_k\rangle_{k\in[1,\mathrm{bdep}_{\mathsf{op}}(j)]} t_j\right)_{j\in[1,\mathrm{ar}_{\mathsf{X}}(\mathsf{op})]}\right) \quad .$$

For instance, the terms for the π-calculus signature Σ_π are the π-calculus raw terms.

Notice that there is no α-equivalence at this level of terms for algebraic binding signatures, just as there is no α-equivalence among terms for nominal logic signatures (Sec. 1.1). For instance, if a and a' are different name variables, then the terms of the π-calculus signature $\mathsf{inp}(\mathsf{c}, \langle \mathsf{a}\rangle \mathsf{nil})$ and $\mathsf{inp}(\mathsf{c}, \langle \mathsf{a}'\rangle \mathsf{nil})$ are distinct. Some parameters are called 'binders', and the notation $\langle \ldots \rangle$ is used to indicate this, but this is only notation: α-equivalence will be introduced later, by adding axioms, and by working with particular notions of model. To be clear, we will refer to terms that are not subject to α-equivalence as **raw terms**.

For any signature Σ, we write $T_\Sigma(\mathtt{N}, \mathtt{X})$ for the set of terms of the signature with free name variables in $\mathtt{N}$ and free term variables in $\mathtt{X}$. We write $\Sigma(\mathtt{N}, \mathtt{X})$ for the set of terms that are basic expressions, i.e. that involve exactly one operator.

Every algebraic binding signature gives rise to a nominal logic signature with one non-logical sort X, and a non-logical function symbol for each operator of the algebraic signature. The terms (in $T_\Sigma(\mathtt{N},\mathtt{X})$) for an algebraic binding signature Σ are exactly the terms for the corresponding nominal logic signature that only involve variables (from $\mathtt{N}$, $\mathtt{X}$) of name and term sorts, and that do not involve the swap symbol. For example, the nominal logic signature arising from the π-calculus signature Σ_π has five function symbols:

$$\mathsf{inp} : (\mathsf{N}, [\mathsf{N}]\mathsf{X}) \to \mathsf{X}, \qquad \mathsf{out} : (\mathsf{N},\mathsf{N},\mathsf{X}) \to \mathsf{X}, \qquad \mathsf{res} : [\mathsf{N}]\mathsf{X} \to \mathsf{X},$$
$$\mathsf{par} : (\mathsf{X},\mathsf{X}) \to \mathsf{X} \qquad \text{and} \qquad \mathsf{nil} : () \to \mathsf{X} \quad .$$

The nominal logic axioms ensure, for instance, that there is a theorem

$$\forall \mathsf{a}, \mathsf{a}', \mathsf{c} : \mathsf{N}.\, \mathsf{inp}(\mathsf{c}, \langle \mathsf{a} \rangle \mathsf{nil}) = \mathsf{inp}(\mathsf{c}, \langle \mathsf{a}' \rangle \mathsf{nil})$$

in the theory associated to the signature Σ_π.

Fixing an algebraic binding signature Σ, the nominal logic theory of **congruence** contains the structure associated to Σ together with a relation symbol R of arity (X,X).

The theory of congruence has the non-logical axioms that say that R is an equivalence relation (reflexivity, symmetry, and transitivity). There is also a non-logical axiom for each operator of the signature, stating that the relation R must respect that operator. For instance, the theory of congruence for the π-calculus includes the following axiom for the input operator.

$$\forall \mathsf{a}, \mathsf{c} : \mathsf{N}.\ \forall \mathsf{x}, \mathsf{y} : \mathsf{X}.\ \mathsf{x} \,\mathsf{R}\, \mathsf{y} \implies \mathsf{inp}(\mathsf{c}, \langle \mathsf{a} \rangle \mathsf{x})\ \mathsf{R}\ \mathsf{inp}(\mathsf{c}, \langle \mathsf{a} \rangle \mathsf{y})$$

1.4 Theory of nominal substitution

Nominal logic provides facilities for describing name-permutation, and this is essential for the axiomatisation of α-equivalence. Name-passing calculi, however, involve non-injective substitutions of names. Consider the crucial role of substitution in the rule for communication for the π-calculus (taken from Milner et al. [16]):

$$\frac{P \xrightarrow{\bar{x}y} P' \qquad Q \xrightarrow{x(z)} Q'}{P \,|\, Q \xrightarrow{\tau} P' \,|\, Q'\{{}^{y}\!/_{z}\}} \quad .$$

We introduce the nominal logic signature of **nominal substitution**: it has one non-logical sort, X, and one function symbol $\mathsf{sub} : (\mathsf{N}, [\mathsf{N}]\mathsf{X}) \to \mathsf{X}$, written infix: $\mathsf{sub}(\mathsf{a}', \langle \mathsf{a}\rangle t)$ is written $[\mathsf{a}'/\mathsf{a}]t$. (Informally: $[\mathsf{a}'/\mathsf{a}]t$ is a' substituted for a in t. We are perhaps breaking with convention by writing the substitution on the left, rather than on the right.) Thus we have an algebraic binding signature with an operator sub and arities $\mathrm{ar}_{\mathsf{N}}(\mathsf{sub}) = 1$, $\mathrm{ar}_{\mathsf{X}}(\mathsf{sub}) = 1$, $\mathrm{bdep}_{\mathsf{sub}}(1) = 1$.

The theory of nominal substitution has four non-logical axioms:

(1) Identity:
$\forall \mathsf{a} : \mathsf{N}.\, \forall \mathtt{x} : \mathsf{X}.\, [\mathsf{a}/\mathsf{a}]\mathtt{x} = \mathtt{x}.$
(2) Weakening:
$\forall \mathsf{a}, \mathsf{b} : \mathsf{N}.\, \forall \mathtt{x} : \mathsf{X}.\, \mathsf{a}\#\mathtt{x} \implies [\mathsf{b}/\mathsf{a}]\mathtt{x} = \mathtt{x}$
(3) Contraction:
$\forall \mathsf{a}, \mathsf{b}, \mathsf{c} : \mathsf{N}.\, \forall \mathtt{x} : \mathsf{X}.\, [\mathsf{c}/\mathsf{b}][\mathsf{b}/\mathsf{a}]\mathtt{x} = [\mathsf{c}/\mathsf{b}][\mathsf{c}/\mathsf{a}]\mathtt{x}.$
(4) Permutation:
$\forall \mathsf{a}, \mathsf{b}, \mathsf{c}, \mathsf{d} : \mathsf{N}.\, \forall \mathtt{x} : \mathsf{X}.\, \mathsf{c} \neq \mathsf{b} \neq \mathsf{a} \neq \mathsf{d} \implies [\mathsf{d}/\mathsf{b}][\mathsf{c}/\mathsf{a}]\mathtt{x} = [\mathsf{c}/\mathsf{a}][\mathsf{d}/\mathsf{b}]\mathtt{x}.$

These axioms for substitution are essentially standard [see e.g. 10, Defn. 3.1]. Notice that we have the theorem

$$\forall \mathtt{x} : \mathsf{X}.\ \forall \mathsf{a}, \mathsf{b} : \mathsf{N}.\ \mathsf{b}\#\mathtt{x} \implies [\mathsf{b}/\mathsf{a}]\mathtt{x} = (\mathsf{b}\ \mathsf{a}) \cdot \mathtt{x} \quad .$$

For any algebraic binding signature Σ, we define the nominal logic theory of **Σ-algebras with substitution** by combining the signature for Σ with the theory of nominal substitution, and adding an axiom for every operator in Σ, ensuring that substitution respects that operator, avoiding the capture of bound names. For instance, the theory of Σ_π-algebras with substitution includes the following axiom to ensure that substitution respects the restriction structure in the π-calculus.

$$\forall \mathsf{a},\ \mathsf{b}, \mathsf{c} : \mathsf{N}.\ \forall \mathtt{x} : \mathsf{X}.\ \mathsf{a} \neq \mathsf{b} \neq \mathsf{c} \implies [\mathsf{c}/\mathsf{a}]\mathsf{res}(\langle \mathsf{b}\rangle \mathtt{x}) = \mathsf{res}(\langle \mathsf{b}\rangle([\mathsf{c}/\mathsf{a}]\mathtt{x}))$$

1.5 Theory of transition systems and bisimulation

We now introduce nominal logic theories of behaviour for name-passing systems.

The nominal logic theory of **ground transition systems** has one non-logical sort X, and four relation symbols:

(1) The input transition relation, $\xrightarrow{-?(-)}$, with arity $(\mathsf{X}, \mathsf{N}, \mathsf{N}, \mathsf{X})$;
(2) The output transition relation, $\xrightarrow{-!-}$, with arity $(\mathsf{X}, \mathsf{N}, \mathsf{N}, \mathsf{X})$;

(3) The bound output transition relation, $\xrightarrow{-!(-)}$, with arity $(\mathsf{X}, \mathsf{N}, \mathsf{N}, \mathsf{X})$;
(4) The silent transition relation, $\xrightarrow{\tau}$, with arity (X, X).

The relation symbols will be written infix as indicated.

Notice that, although the input and bound output data is to be thought of as binding, we do not use the binding sort $[\mathsf{N}]\mathsf{X}$. Instead, we equip the theory with two non-logical axioms, ensuring that the data which is 'binding' is in fact fresh, as follows.

(1) $\forall \mathsf{x}, \mathsf{y} : \mathsf{X}.\ \forall \mathsf{a}, \mathsf{c} : \mathsf{N}.\ \mathsf{x} \xrightarrow{\mathsf{c?(a)}} \mathsf{y} \implies \mathsf{a}\#\mathsf{c} \wedge \mathsf{a}\#\mathsf{x}.$
(2) $\forall \mathsf{x}, \mathsf{y} : \mathsf{X}.\ \forall \mathsf{a}, \mathsf{c} : \mathsf{N}.\ \mathsf{x} \xrightarrow{\mathsf{c!(a)}} \mathsf{y} \implies \mathsf{a}\#\mathsf{c} \wedge \mathsf{a}\#\mathsf{x}.$

Remark. We will not consider models of nominal logic theories until Section 2. When we do, however, it will become clear (Prop. 4.1) that models of the theory of ground transition systems correspond exactly with models of the modified theory where input and output relations are of arity $(\mathsf{X}, \mathsf{N}, [\mathsf{N}]\mathsf{X})$ and the axioms are omitted — i.e., where data really is binding. It is a matter of taste as to which presentation is preferable; we adopt this more explicit approach because it seems closer in spirit to the original work of Milner et al. [16] and to much of the work that has followed from this.

The nominal logic theory of **ground simulation** is formed by extending the theory of ground transition systems with a relation R of arity (X, X), and four additional axioms.

(1) $\forall \mathsf{x}, \mathsf{x}', \mathsf{y} : \mathsf{X}.\ \forall \mathsf{a}, \mathsf{c} : \mathsf{N}.\ \mathsf{a}\#\mathsf{y} \wedge \mathsf{xRy} \wedge \mathsf{x} \xrightarrow{\mathsf{c?(a)}} \mathsf{x}' \implies \exists \mathsf{y}' : \mathsf{X}.\ \mathsf{x}'\mathsf{R}\mathsf{y}' \wedge \mathsf{y} \xrightarrow{\mathsf{c?(a)}} \mathsf{y}'$
(2) $\forall \mathsf{x}, \mathsf{x}', \mathsf{y} : \mathsf{X}.\ \forall \mathsf{a}, \mathsf{c} : \mathsf{N}.\ \mathsf{xRy} \wedge \mathsf{x} \xrightarrow{\mathsf{c!a}} \mathsf{x}' \implies \exists \mathsf{y}' : \mathsf{X}.\ \mathsf{x}'\mathsf{R}\mathsf{y}' \wedge \mathsf{y} \xrightarrow{\mathsf{c!a}} \mathsf{y}'$
(3) $\forall \mathsf{x}, \mathsf{x}', \mathsf{y} : \mathsf{X}.\ \forall \mathsf{a}, \mathsf{c} : \mathsf{N}.\ \mathsf{a}\#\mathsf{y} \wedge \mathsf{xRy} \wedge \mathsf{x} \xrightarrow{\mathsf{c!(a)}} \mathsf{x}' \implies \exists \mathsf{y}' : \mathsf{X}.\ \mathsf{x}'\mathsf{R}\mathsf{y}' \wedge \mathsf{y} \xrightarrow{\mathsf{c!(a)}} \mathsf{y}'$
(4) $\forall \mathsf{x}, \mathsf{x}', \mathsf{y} : \mathsf{X}.\ \mathsf{xRy} \wedge \mathsf{x} \xrightarrow{\tau} \mathsf{x}' \implies \exists \mathsf{y}' : \mathsf{X}.\ \mathsf{x}'\mathsf{R}\mathsf{y}' \wedge \mathsf{y} \xrightarrow{\tau} \mathsf{y}'$

The nominal logic theory of **ground bisimulation** adds to the theory of ground simulation the same four axioms again, but with the roles of x, x' and y, y' interchanged.

The nominal logic theory of **wide-open transition systems** has one non-logical sort, and all the symbols and axioms of the theory of ground transition systems and of the theory of nominal substitutions. We include *no* additional axioms about the interplay between the transition relations and the substitution function.

The nominal logic theory of **wide-open bisimulation** combines the theory of ground bisimulation with the theory of nominal substitution. We further include the additional axiom:

$$\forall \mathsf{a}, \mathsf{b} : \mathsf{N}.\ \forall \mathsf{x}, \mathsf{y} : \mathsf{X}.\ \mathsf{xRy} \implies ([\mathsf{b}/\mathsf{a}]\mathsf{x})\mathsf{R}([\mathsf{b}/\mathsf{a}]\mathsf{y}) \quad .$$

1.6 Transition system specifications

We now introduce our basic notion of *transition system specification*. For now, we take a purely formal approach, treating a specificaton as syntactic data. In Section 1.7 we explain how a specification gives rise to a nominal logic theory.

Consider a set $\mathtt{N}$ of name variables. A **label term** is an element of the set $Lab(\mathtt{N})$, given as the following disjoint union:

$$Lab(\mathtt{N}) = \mathtt{N} \times \mathtt{N}\ +\ \mathtt{N} \times \mathtt{N}\ +\ \mathtt{N} \times \mathtt{N}\ +\ 1 \quad .$$

Label terms in the four summands of $Lab(\mathtt{N})$ are respectively to be thought of as input labels (written $\mathsf{c?(a)}$), output labels (written $\mathsf{c!d}$), bound output labels (written $\mathsf{c!(a)}$), and silent labels (written τ).

We define functions $\mathrm{fn}, \mathrm{bn} : Lab(\mathtt{N}) \to \mathcal{P}(\mathtt{N})$ that assign to each label the set of its bound and free variables. Precisely:

$l \in Lab(\mathtt{N})$	$\mathrm{fn}(l)$	$\mathrm{bn}(l)$
$\mathsf{c?(a)}$	$\{\mathsf{c}\}$	$\{\mathsf{a}\}$
$\mathsf{c!d}$	$\{\mathsf{c}, \mathsf{d}\}$	$\emptyset$
$\mathsf{c!(a)}$	$\{\mathsf{c}\}$	$\{\mathsf{a}\}$
τ	$\emptyset$	$\emptyset$

Definition 1.2. Let Σ be an algebraic binding signature. A **formal rule structure** over Σ is given by sets $\mathtt{N}$ and $\mathtt{X}$ of name and term variables together with a finite set of premises over Σ, $\mathtt{N}$ and $\mathtt{X}$, and a conclusion over Σ, $\mathtt{N}$ and $\mathtt{X}$.

A premise over Σ, $\mathtt{N}$ and $\mathtt{X}$ is a triple in $\mathtt{X} \times Lab(\mathtt{N}) \times \mathtt{X}$.

A conclusion over Σ, $\mathtt{N}$ and $\mathtt{X}$ is a triple in $\Sigma(\mathtt{N}, \mathtt{X}) \times Lab(\mathtt{N}) \times T_{\Sigma+\mathsf{sub}}(\mathtt{N}, \mathtt{X})$. Here, $(\Sigma + \mathsf{sub})$ is the algebraic binding signature obtained by adding to Σ the function symbol sub for nominal substitution.

We refer to the three components of a premise or a conclusion as the *source*, the *label* and the *target*.

A **transition system specification** over Σ is a set of formal rule structures over Σ.

Remark. We will not give meaning to transition system specifications until the next subsection, but the reader familiar with the GSOS format of Bloom et al. [4] will already notice a schematic difference from that format. Whereas Bloom et al. consider both positive and negative premises, we only allow one kind of premise here ('positive') . It appears that the developments of this article could be straightforwardly revised to accommodate the two kinds of premise; we work in the less sophisticated setting primarily for simplicity. The goals of Bloom et al. are, after all, different from ours.

As a first example, part of the transition system specification $\mathcal{R}_\pi$ for the π-calculus is given in Figure 3. Here, it is illustrative to add two operators, match and mismch, to the signature Σ_π, with arities

$$\mathrm{ar}_{\mathsf{N}}(\mathsf{match}) = \mathrm{ar}_{\mathsf{N}}(\mathsf{mismch}) = 2; \qquad \mathrm{ar}_{\mathsf{X}}(\mathsf{match}) = \mathrm{ar}_{\mathsf{X}}(\mathsf{mismch}) = 1;$$
$$\mathrm{bdep}_{\mathsf{match}}(1) = \mathrm{bdep}_{\mathsf{mismch}}(1) = 0.$$

Formal rule structures are written in the usual "premises over conclusion" style, and with the triples comprising the premises, and the conclusion, written with the transition arrow.

This suggestive notation should provide the reader with an inkling of how a transition system specification can be understood as a set of rules in nominal logic. We will make this precise in the following subsection. Here, though, we note that side-conditions play an important role in defining name-passing calculi, and yet they are entirely absent from our notion of formal rule structure. For instance, the rule for bound output in parallel, Figure 3(e), is missing the side condition "$\mathtt{a} \notin \mathrm{fn}(\mathtt{x}')$", that would ensure that no free names of $\mathtt{x}'$ are captured; in the rule for mismatch, Figure 3(f), one might anticipate a side condition $\mathsf{c} \neq \mathsf{d}$, for the mismatch operator to be of any use. In fact, all the side-conditions that appear in the π-calculus are of these two forms, involving freshness and distinctness of names.

It would be clumsy to establish general results about rules with arbitrary side-conditions and so, in this article, we avoid this by adopting a convention whereby every rule structure is equipped with two side-conditions:

- distinct name variables are interpreted by distinct names;
- bound names in the conclusion label must be fresh for the conclusion source.

(For now, these are only to be understood informally.) A drawback of this approach is that some duplication in the rules may be necessary. For instance, to attain the usual output behaviour it is necessary to split the usual rule in two, as shown in Figure 3(a–b). One can envisage a notion of formal rule structure with *explicit* side conditions from which a finite family of formal rule structures in the form of Definition 1.2 can be derived, but we will not dwell

(a) **Output** of the channel name

$\mathtt{X} = \{\mathsf{x}\}$, $\mathtt{N} = \{\mathsf{c}\}$

$$\frac{—}{\mathsf{out}(\mathsf{c},\mathsf{c},\mathsf{x}) \xrightarrow{\mathsf{c!c}} \mathsf{x}}$$

(b) **Output** of a name distinct from the channel name

$\mathtt{X} = \{\mathsf{x}\}$, $\mathtt{N} = \{\mathsf{c},\mathsf{d}\}$

$$\frac{—}{\mathsf{out}(\mathsf{c},\mathsf{d},\mathsf{x}) \xrightarrow{\mathsf{c!d}} \mathsf{x}}$$

(c) **Input**

$\mathtt{X} = \{\mathsf{x}\}$, $\mathtt{N} = \{\mathsf{a},\mathsf{c}\}$

$$\frac{—}{\mathsf{inp}(\mathsf{c},\langle\mathsf{a}\rangle\mathsf{x}) \xrightarrow{\mathsf{c?(a)}} \mathsf{x}}$$

(d) **Match** for input transitions where the channel is the match name

$\mathtt{X} = \{\mathsf{x},\mathsf{y}\}$, $\mathtt{N} = \{\mathsf{a},\mathsf{c}\}$

$$\frac{\mathsf{x} \xrightarrow{\mathsf{c?(a)}} \mathsf{y}}{\mathsf{match}(\mathsf{c},\mathsf{c},\mathsf{x}) \xrightarrow{\mathsf{c?(a)}} \mathsf{y}}$$

(e) **Parallel** for bound output on the right

$\mathtt{X} = \{\mathsf{x},\mathsf{x}',\mathsf{y},\mathsf{y}'\}$, $\mathtt{N} = \{\mathsf{a},\mathsf{c}\}$

$$\frac{\mathsf{x}' \xrightarrow{\mathsf{c!(a)}} \mathsf{y}'}{\mathsf{par}(\mathsf{x},\mathsf{x}') \xrightarrow{\mathsf{c!(a)}} \mathsf{par}(\mathsf{x},\mathsf{y}')}$$

(f) **Mismatch** for input transitions over the second mismatch name

$\mathtt{X} = \{\mathsf{x},\mathsf{y}\}$, $\mathtt{N} = \{\mathsf{c},\mathsf{d},\mathsf{e}\}$

$$\frac{\mathsf{x} \xrightarrow{\mathsf{d?(e)}} \mathsf{y}}{\mathsf{mismch}(\mathsf{c},\mathsf{d},\mathsf{x}) \xrightarrow{\mathsf{d?(e)}} \mathsf{y}}$$

(g) **Communication** with output on the left, and distinct channel and data

$\mathtt{X} = \{\mathsf{x},\mathsf{x}',\mathsf{y},\mathsf{y}'\}$, $\mathtt{N} = \{\mathsf{a},\mathsf{c},\mathsf{d}\}$

$$\frac{\mathsf{x} \xrightarrow{\mathsf{c!d}} \mathsf{y} \qquad \mathsf{x}' \xrightarrow{\mathsf{c?(a)}} \mathsf{y}'}{\mathsf{par}(\mathsf{x},\mathsf{x}') \xrightarrow{\tau} \mathsf{par}(\mathsf{y},[\mathsf{d}/\mathsf{a}]\mathsf{y}')}$$

(h) **Scope closure** for output on the left

$\mathtt{X} = \{\mathsf{x},\mathsf{x}',\mathsf{y},\mathsf{y}'\}$, $\mathtt{N} = \{\mathsf{a},\mathsf{c}\}$

$$\frac{\mathsf{x} \xrightarrow{\mathsf{c!(a)}} \mathsf{y} \qquad \mathsf{x}' \xrightarrow{\mathsf{c?(a)}} \mathsf{y}'}{\mathsf{par}(\mathsf{x},\mathsf{x}') \xrightarrow{\tau} \mathsf{res}\,(\langle\mathsf{a}\rangle\mathsf{par}(\mathsf{y},\mathsf{y}'))}$$

(i) **Restriction** for bound output

$\mathtt{X} = \{\mathsf{x},\mathsf{y}\}$, $\mathtt{N} = \{\mathsf{a},\mathsf{b},\mathsf{c}\}$

$$\frac{\mathsf{x} \xrightarrow{\mathsf{c!(b)}} \mathsf{y}}{\mathsf{res}(\langle\mathsf{a}\rangle\mathsf{x}) \xrightarrow{\mathsf{c!(b)}} \mathsf{res}(\langle\mathsf{a}\rangle\mathsf{y})}$$

(j) **Scope opening**

$\mathtt{X} = \{\mathsf{x},\mathsf{y}\}$, $\mathtt{N} = \{\mathsf{a},\mathsf{c}\}$

$$\frac{\mathsf{x} \xrightarrow{\mathsf{c!a}} \mathsf{y}}{\mathsf{res}(\langle\mathsf{a}\rangle\mathsf{x}) \xrightarrow{\mathsf{c!(a)}} \mathsf{y}}$$

Fig. 3. Examples of rule structures for the π-calculus. Note that the side conditions are implicit.

on that here.

1.7 Theories from transition system specifications

Consider an algebraic binding signature Σ, and let $\mathcal{R}$ be a transition system specification over Σ. The nominal logic theory of **transition systems satisfying $\mathcal{R}$** is given as follows.

The nominal logic signature is based on the signature arising from Σ; this has one non-logical sort X and a function symbol for each operator in Σ. We add to this signature the signature for nominal substitution, i.e. the function symbol $\mathsf{sub} : (\mathsf{N}, [\mathsf{N}]\mathsf{X}) \to \mathsf{X}$ (see Sec. 1.4), and the four relation symbols of the theory of ground transition systems (see Sec. 1.5).

The theory has two kinds of non-logical axioms. The axioms of the first kind are the axioms for Σ-algebra with substitution, taken from Section 1.3, ensuring that substitution respects the operators of Σ in an appropriate way. The second kind of axiom comes from treating the formal rule structures as logical rules, taking into account the implicit side conditions. For instance, the rule for bound output in parallel, Figure 3(e), gives rise to the following axiom.

$$\begin{aligned}&\forall \mathsf{x}, \mathsf{x}', \mathsf{y} : \mathsf{X}.\ \forall \mathsf{a}, \mathsf{c} : \mathsf{N}.\\ &\qquad \mathsf{a}\#\mathsf{c} \quad\wedge\quad \mathsf{a}\#\mathsf{par}(\mathsf{x},\mathsf{x}') \quad\wedge\quad \mathsf{x}' \xrightarrow{\mathsf{c!(a)}} \mathsf{y}' \implies \mathsf{par}(\mathsf{x},\mathsf{x}') \xrightarrow{\mathsf{c!(a)}} \mathsf{par}(\mathsf{x},\mathsf{y}')\end{aligned}$$

More generally, we consider an arbitrary formal rule structure $\mathtt{R}$ in $\mathcal{R}$. The hypothesis of the corresponding axiom is derived from the premises of $\mathtt{R}$ together with the implicit side conditions; the conclusion is derived from the conclusion of $\mathtt{R}$, as follows. Suppose that $\mathtt{R}$ has name and term variables $\mathtt{N}$ and $\mathtt{X}$, and premises *Prem* and conclusion ($\underline{\mathtt{src}}, \underline{\mathtt{l}}, \underline{\mathtt{tar}}$); then the corresponding axiom is as follows.

$$\begin{aligned}&\forall \mathtt{N} : \mathsf{N}.\, \forall \mathtt{X} : \mathsf{X}.\\ &\qquad \bigwedge_{\substack{\mathtt{a},\mathtt{b}\in\mathtt{N}\\ \mathtt{a}\neq\mathtt{b}}} \mathtt{a}\#\mathtt{b} \ \wedge \bigwedge_{\mathtt{a}\in \mathrm{bn}(\underline{\mathtt{l}})} \mathtt{a}\#(\underline{\mathtt{src}}, \mathrm{fn}(\underline{\mathtt{l}})) \ \wedge \bigwedge_{(\mathtt{x},\mathtt{l},\mathtt{y})\in \mathit{Prem}} \mathtt{x} \xrightarrow{\mathtt{l}} \mathtt{y}\\ &\qquad\qquad \implies \underline{\mathtt{src}} \xrightarrow{\underline{\mathtt{l}}} \underline{\mathtt{tar}} \quad .\end{aligned}$$

Here, we have used some notation that is standard in infinitary logic. Suppose that $\mathtt{N} = \{\mathsf{a}_1, \ldots \mathsf{a}_n\}$; then when we write "$\forall \mathtt{N} : \mathsf{N}.\ \Psi$" we intend

$$\forall \mathsf{a}_1 : \mathsf{N}. \ldots \forall \mathsf{a}_n : \mathsf{N}.\ \Psi \quad .$$

The reader might expect that the theory associated to $\mathcal{R}$ will include the two axioms of the theory of ground transition systems (Sec. 1.5). Indeed, these axioms are consistent with our theory, because of the side conditions on the rules. However, we do *not* include these axioms in the theory, primarily because they are not Horn clauses over the transition relations, and as such they would make the reasons for the existence of an initial model more complicated (see Section 2.5).

2 Models

This section is concerned with models of the theories introduced in the previous section, and most particularly with models of transition system specifications. We introduce nominal sets in Sec. 2.1, and explain in Sec. 2.2 how nominal logic is to be modelled in nominal sets. It is useful to consider models for algebraic binding signatures (i.e., algebras) in categories other than nominal sets, and we begin the study of this in Sec. 2.3. The category of models of nominal substitutions is an important universe in which algebras of binding signatures can be studied (Sec. 2.4). With a proper understanding of models of syntax, we are ready to consider models of rules (Sec. 2.5) and thus the question of congruence of bisimilarity can be phrased (Sec. 2.6).

2.1 Nominal sets

We briefly review and set notation for nominal sets, as introduced by Gabbay and Pitts [e.g. 12]. Later, in Section 2.2, we will explain that nominal sets are the appropriate place in which to interpret nominal logic theories, and we here recall how some of the core features of nominal logic — names, freshness, binding — can be understood in the setting of nominal sets.

For the remainder of this article we fix an infinite set $\mathcal{N}$ of names. Recall that the symmetric group $\mathsf{Sym}(\mathcal{N})$ on $\mathcal{N}$ comprises all permutations of $\mathcal{N}$, under the operation of composition. For names $a, b \in \mathcal{N}$, as usual, we write $(a\,b)$ for the permutation on $\mathcal{N}$ that swaps a and b and fixes all other names.

A left action of $\mathsf{Sym}(\mathcal{N})$ is a set X equipped with a function

$$\cdot_X : \mathsf{Sym}(\mathcal{N}) \times X \to X$$

(written infix) which is such that for any element $x \in X$ we have $\mathsf{id}_{\mathcal{N}} \cdot_X x = x$ and, for any $\sigma, \tau \in \mathsf{Sym}(\mathcal{N})$, that $(\tau\sigma) \cdot_X x = \tau \cdot_X (\sigma \cdot_X x)$.

A finite set of names $N \subseteq_{\mathrm{f}} \mathcal{N}$ is said to **support** an element x of a $\mathsf{Sym}(\mathcal{N})$-action $(X, \cdot_X)$ if every permutation $\sigma \in \mathsf{Sym}(\mathcal{N})$ that fixes every element of N

also fixes x. A **nominal set** is a $\mathsf{Sym}(\mathcal{N})$-action in which every element has finite support. It follows that every element x of a nominal set has a *least* support, which we denote $\mathsf{supp}(x)$.

We let **Nom** be the category of nominal sets and equivariant functions, i.e. functions between the underlying sets that are compatible with the actions.

The set $\mathcal{N}$ of names has nominal-set structure; the permutation action is given by evaluation, i.e. $\sigma \cdot_{\mathcal{N}} a = \sigma(a)$. The category **Nom** has colimits and finite limits, and the functor $\mathbf{Nom} \to \mathbf{Set}$, that forgets the permutation action structure, preserves them. So, in particular, sums and products are inherited from **Set**.

An equivariant relation is a relation between nominal sets that is compatible with the permutation actions — i.e., it is a relation in the category of nominal sets. For an example, we consider a nominal set X, and define a relation $\# \subseteq \mathcal{N} \times X$, by letting $a\#x$ iff $a \notin \mathsf{supp}(x)$. This relation is equivariant because if $a\#x$ then also $(\sigma(a))\#(\sigma \cdot_X x)$, for every permutation σ.

For any nominal set X we have the nominal set $[\mathcal{N}]X$: intuitively, it is the set of (name,element) pairs up-to α-equivalence. The carrier set is the quotient

$$[\mathcal{N}]X = (\mathcal{N} \times X)/{\sim_{[\mathcal{N}]X}}$$

where $(a, x) \sim_{[\mathcal{N}]X} (a', x')$ if for any $b \in \mathcal{N}$ such that $b\#x$ and $b\#x'$ we have $(b\,a) \cdot_X x = (b\,a') \cdot_X x'$. We write $\langle a \rangle x$ for the equivalence class $(a, x)_{[\mathcal{N}]X}$. The $\mathsf{Sym}(\mathcal{N})$-action of $[\mathcal{N}]X$ is inherited from that of the product.

2.2 Models of nominal logic theories

The category **Nom** is a topos, and so any first-order theory, indeed any nominal logic theory, can be interpreted in **Nom**. It is reasonable, though, to restrict attention to those models in which the features that are required by nominal logic are interpreted in a particular way. This can be seen as an extension of the usual convention that equality in first-order logic should be interpreted as equality in the model.

A **structure** $\mathcal{M}$ for a nominal logic signature is given as follows: to each sort X is associated a nominal set $[\![\mathsf{X}]\!]_{\mathcal{M}}$; to each function symbol $\mathsf{op} : (\mathsf{X}_1, \ldots \mathsf{X}_n) \to \mathsf{X}$ is associated an equivariant function $[\![\mathsf{op}]\!]_{\mathcal{M}} : [\![\mathsf{X}_1]\!]_{\mathcal{M}} \times \cdots \times [\![\mathsf{X}_n]\!]_{\mathcal{M}} \to [\![\mathsf{X}]\!]_{\mathcal{M}}$; to each relation symbol R of arity $(\mathsf{X}_1, \ldots \mathsf{X}_n)$ is associated an equivariant relation $[\![\mathsf{R}]\!]_{\mathcal{M}} \subseteq [\![\mathsf{X}_1]\!]_{\mathcal{M}} \times \cdots \times [\![\mathsf{X}_n]\!]_{\mathcal{M}}$. This is subject to the following requirements.

- The sort N of names must be interpreted as the nominal set $\mathcal{N}$ of names.

- If a sort X is interpreted as the nominal set X, then the sort $[\mathsf{N}]\mathsf{X}$ must be interpreted as the nominal set $[\mathcal{N}]X$, and the bind symbol must be interpreted as the evident quotient map $\mathcal{N} \times X \twoheadrightarrow [\mathcal{N}]X$.
- If a sort X is interpreted as the nominal set X then the corresponding swap symbol must be interpreted in terms of the permutation action of X.
- For each sort X the corresponding relation symbol $\#$ must be interpreted as the $\#$ relation in **Nom**.

A **morphism** between two structures, $f : \mathcal{M} \to \mathcal{M}'$, is specified by an equivariant function $f_{\mathsf{X}} : [\![\mathsf{X}]\!]_{\mathcal{M}} \to [\![\mathsf{X}]\!]_{\mathcal{M}'}$ for every sort X, provided the functions $\{f_{\mathsf{X}}\}$ respect the interpretations of the function and relation symbols appropriately. In fact, it is sufficient to specify equivariant functions $f_{\mathsf{X}} : [\![\mathsf{X}]\!]_{\mathcal{M}} \to [\![\mathsf{X}]\!]_{\mathcal{M}'}$ for the non-logical sorts only, since the $[\mathcal{N}](-)$ operator extends to an endofunctor on **Nom**.

Terms and formulas in the signature are interpreted according to the usual set-theoretic interpretation of first-order logic. A **model** $\mathcal{M}$ for a nominal logic theory is defined to be a structure for the signature in which all the axioms of the theory hold. It is sufficient to verify the non-logical axioms, because the nominal logic axioms have been designed to be sound for structures in **Nom** [18, Theorem 1].

In this way we arrive at notions of model for the theories introduced in Section 1. It will often be convenient to abbreviate "a model of the theory of nominal substitutions" as "a nominal substitution"; "a model of the theory of transition systems satisfying $\mathcal{R}$" as "a transition system satisfying $\mathcal{R}$"; and so on.

2.3 Models for algebraic binding signatures

If one is concerned only with algebraic binding signatures (Defn. 1.1), then the full structure of **Nom** is not needed. A **binding model category** $\mathcal{C}$ is a category with finite products and coproducts, and a distinguished object 'of names' $\mathcal{N}_{\mathcal{C}} \in \mathcal{C}$ and a 'binding' endofunctor $[\mathcal{N}]_{\mathcal{C}} : \mathcal{C} \to \mathcal{C}$.

Consider an algebraic binding signature Σ, and let $\mathcal{C}$ be a binding model category. A **Σ-algebra** in $\mathcal{C}$ is given by an object $X \in \mathcal{C}$ together with, for each op in Σ, a morphism

$$\mathsf{op}_X : \mathcal{N}_{\mathcal{C}}^{\mathrm{ar}_{\mathsf{N}}(\mathsf{op})} \times [\mathcal{N}]_{\mathcal{C}}^{\mathrm{bdep}_{\mathsf{op}}(1)} X \times \cdots \times [\mathcal{N}]_{\mathcal{C}}^{\mathrm{bdep}_{\mathsf{op}}(\mathrm{ar}_{\mathsf{X}}(\mathsf{op}))} X \ \to\ X \quad .$$

(Here, we write $\mathcal{N}_{\mathcal{C}}^{n}$ for the n-fold product of $\mathcal{N}_{\mathcal{C}}$, and write $[\mathcal{N}]_{\mathcal{C}}^{n}$ for the n-fold application of the endofunctor $[\mathcal{N}]_{\mathcal{C}}$.)

A **homomorphism** between Σ-algebras is given by a morphism in $\mathcal{C}$ between the underlying objects, that respects the interpretations of the operators. Σ-algebras in $\mathcal{C}$, and homomorphisms between them, form a category.

To give a Σ-algebra in $\mathcal{C}$ is to give an algebra for the endofunctor

$$\Sigma_{\mathcal{C}}(-) \;=\; \coprod_{\mathsf{op}\in\Sigma} \left(\mathcal{N}_{\mathcal{C}}^{\mathrm{ar}_{\mathsf{N}}(\mathsf{op})} \times [\mathcal{N}]_{\mathcal{C}}^{\mathrm{bdep}_{\mathsf{op}}(1)}(-) \times \cdots \times [\mathcal{N}]_{\mathcal{C}}^{\mathrm{bdep}_{\mathsf{op}}(\mathrm{ar}_{\mathsf{X}}(\mathsf{op}))}(-)\right)$$

on $\mathcal{C}$. Indeed, the category of Σ-algebras in $\mathcal{C}$ is isomorphic to the category of algebras for the endofunctor $\Sigma_{\mathcal{C}}$.

The initial Σ-algebra over some $X \in \mathcal{C}$, when it exists, will be denoted $T_{\Sigma,\mathcal{C}}(X)$. In all the examples that we will consider, the object $T_{\Sigma,\mathcal{C}}(X)$ exists for all $X \in \mathcal{C}$, and so $T_{\Sigma,\mathcal{C}}$ extends to a monad on $\mathcal{C}$ in the usual fashion.

A first example of a binding model category is the category **Set** of sets: there, the distinguished set of names is a chosen set $\mathtt{N}$ of name variables, and the binding endofunctor is given by the product functor $\mathtt{N} \times (-)$. Henceforth, we will write $\mathbf{Set}_{\mathtt{N}}$ to indicate the binding model category **Set** with the chosen name variables $\mathtt{N}$. For a set $\mathtt{X}$ of term variables, the set $\Sigma_{\mathbf{Set}_{\mathtt{N}}}(\mathtt{X})$ is the set of basic expressions of raw syntax, as considered in Section 1. The set $T_{\Sigma,\mathbf{Set}_{\mathtt{N}}}(\mathtt{X})$ contains all terms of raw syntax, previously written as $T_{\Sigma}(\mathtt{N},\mathtt{X})$.

A second example is the category **Nom** of nominal sets, using the nominal set of names and the binding operator there. Algebras for an algebraic binding signature are exactly the structures for the corresponding nominal logic signature. The set $T_{\Sigma,\mathbf{Nom}}(\emptyset)$ contains all Σ-terms up-to α-equivalence.

Consider a binding model category $\mathcal{C}$ that moreover has pullbacks, and let X be a Σ-algebra. An equivalence relation $X \leftarrow R \rightarrow X$ is a **Σ-congruence** if there is a Σ-algebra structure over R making the following diagram commute in $\mathcal{C}$.

$$\begin{tikzcd}
\Sigma_{\mathcal{C}}X \arrow[d] & \Sigma_{\mathcal{C}}R \arrow[l] \arrow[r] \arrow[d] & \Sigma_{\mathcal{C}}X \arrow[d] \\
X & R \arrow[l] \arrow[r] & X
\end{tikzcd}$$

It is straightforward to verify that a congruence between two Σ-algebras in **Nom** is the same thing as a model of the corresponding nominal logic theory of congruence (Sec. 1.3).

2.4 Syntax with substitutions

A third example of a binding model category is the category **NSub** of nominal substitutions and model morphisms between them (see Sec. 1.4). The category **NSub** inherits the structure of a binding model category from **Nom**, as we now explain. The nominal substitution structure for products and coproducts is given componentwise, so that the forgetful functor $\mathbf{NSub} \to \mathbf{Nom}$ preserves them. The object $\mathcal{N}$ is the carrier of exactly one nominal substitution, given by

$$[c/a]a = c; \qquad\qquad [c/b]a = a, \quad \text{where } a \neq b.$$

The endofunctor $[\mathcal{N}](-)$ on **Nom** lifts along the forgetful functor $\mathbf{NSub} \to \mathbf{Nom}$ as follows. For any nominal substitution $X \in \mathbf{NSub}$, we define a substitution action of $[\mathcal{N}]X$ by

$$[c/b]\langle a\rangle x = \langle a\rangle[c/b]x \quad \text{where } b \neq a \neq c.$$

The nominal substitution $T_{\Sigma,\mathbf{NSub}}(\emptyset)$ contains all Σ-terms up-to α-equivalence: its nominal carrier set is $T_{\Sigma,\mathbf{Nom}}(\emptyset)$. Moreover, it is equipped with a substitution action, which provides the usual notion of capture-avoiding name-for-name substitution for terms. Indeed, $T_{\Sigma,\mathbf{NSub}}(\emptyset)$ is the initial model of the nominal logic theory of Σ-algebras with substitutions (Sec. 1.4).

2.5 The intended model of a transition system specification

Consider an algebraic binding signature Σ (Defn. 1.1), and a transition system specification $\mathcal{R}$ for it (Defn. 1.2). The corresponding nominal logic theory may have many models, but the model of interest in operational semantics is that which is initial in the category of (nominal) models. In this model, the only transitions are those that are provable from the rules in $\mathcal{R}$.

The initial model is constructed as follows. The interpretation of the non-logical sort, X, is the set $T_{\Sigma,\mathbf{Nom}}(\emptyset)$ of Σ-terms up-to-α-equivalence. The interpretation of the operators of Σ is given accordingly. The substitution structure is that of $T_{\Sigma,\mathbf{NSub}}(\emptyset)$.

With these parts of the model fixed, the permitted interpretations of the four transition relations form a complete lattice, because the only relevant axioms here — those arising from the rule structures in $\mathcal{R}$, and the four instances of the equivariance axiom (Figure 2, E4) — are all Horn clauses. The initial model uses the least permitted interpretation, in which the only transitions are those that can be justified by a rule in $\mathcal{R}$, or by axiom (E4).

For a first example, notice that the initial model of the transition system specification $\mathcal{R}_\pi$ for the π-calculus has four equivariant relations over the π-calculus terms up-to α-equivalence, which provide the usual transition relations for the

π-calculus. It is perhaps conventional to derive these transition relations without the equivariance axiom (E4). Even without this axiom, though, we have, by rule induction, that the least relation is equivariant.

Recall that the axioms of the theory of ground transition systems (Sec. 1.5) are consistent with the theory associated to $\mathcal{R}$. In fact, it is straightforward to prove the following result by rule induction.

Proposition 2.1. *The initial transition system satisfying a transition system specification is a wide-open transition system.* $\square$

As an aside, we remark that the notions of ground and wide-open transition systems are perhaps too generous a model of name-passing. For example, we could add to the π-calculus an operator $\mathsf{broadcast}$ with one name parameter and no term parameters. The semantics of this operator is given by the following two formal rule structures.

$$\frac{—}{\mathsf{broadcast}(\mathtt{d}) \xrightarrow{\mathtt{c!d}} \mathsf{broadcast}(\mathtt{d})} \qquad \frac{—}{\mathsf{broadcast}(\mathtt{d}) \xrightarrow{\mathtt{d!d}} \mathsf{broadcast}(\mathtt{d})}$$

Informally, $\mathsf{broadcast}(d)$ will persistently output d on any channel. This may be against the reader's intuitions about the π-calculus, because the term $\mathsf{broadcast}(d)$ is able to output on a channel which it does not 'know'. Such concerns can be eliminated by axiomatising a notion of reasonable model [see e.g. 5, 7]. One can straightforwardly introduce restrictions on the appearance of name variables in transition system specifications, so as to prevent such phenomena.

2.6 Bisimilarity

Consider a ground transition system (Sec. 1.5). A **ground bisimulation** on this system is a model of the theory of ground bisimulation in which this system models the transition system component. The class of such bisimulations on a system forms a complete lattice, as usual, and the greatest ground bisimulation we call **ground bisimilarity**.

For the initial model of the π-calculus specification, ground bisimulation is almost the notion proposed by Sangiorgi [19, Defn. 3.9]. The one difference is that equivariance is not enforced in the definition there. Observe, though, that for every bisimulation in the sense of [19], there is a least equivariant relation containing it, and this relation is also a bisimulation. Thus ground bisimilarity in the initial model is exactly the notion proposed in [19].

In the same way, a notion of **wide-open bisimilarity** is defined for wide-open transition systems, and, due to Proposition 2.1, for the initial model of a transition system specification. We are thus able to investigate the question: *For which transition system specifications is wide-open bisimilarity a congruence in the initial model?*

3 Rule format

We introduce conditions on rule structures, designed to guarantee that wide-open bisimilarity is a congruence in the initial model. In Section 3.2 we justify each of the conditions.

Throughout this section we fix an algebraic binding signature, Σ (Defn. 1.2) and a formal rule structure $\mathtt{R}$ over it (Defn. 1.2). We suppose that $\mathtt{R}$ has variables from $\mathtt{N}$ and $\mathtt{X}$, with premise set *Prem* and conclusion with source

$$\underline{\mathtt{src}} \;=\; \underline{\mathsf{op}}\left((\underline{\mathsf{c}}_i)_{i\in[1,\mathrm{ar}_{\mathsf{N}}(\underline{\mathsf{op}})]},\, \left(\langle \underline{\mathsf{a}}^j_k\rangle_{k\in[1,\mathrm{bdep}_{\underline{\mathsf{op}}}(j)]}\underline{\mathtt{x}}_j\right)_{j\in[1,\mathrm{ar}_{\mathsf{X}}(\underline{\mathsf{op}})]}\right) \;,$$

label $\underline{\mathtt{l}}$, and target $\underline{\mathtt{tar}}$. We distinguish entities appearing in the conclusion by underlining them.

3.1 Conditions on rule structures

In Figure 4 we present conditions that we expect to hold for rule structures. Conditions (1) and (2) are the conditions of the GSOS format [4] considered in this context. Conditions (3–8) relate to the freshness of the names that appear in binding position. To specify these conditions precisely it is necessary to formalise the notions of bound and free names that are implicit in rule structures.

Associating names to variables. From here on we assume that Conditions (1) and (2) hold of $\mathtt{R}$. We then assign to each term variable $\mathtt{x} \in \mathtt{X}$ the set $\mathtt{BN}(\mathtt{x}) \subseteq \mathtt{N}$ of name variables that are binding in $\mathtt{x}$. For instance, in the input rule, of Figure 3(c), $\mathtt{BN}(\mathtt{x}) = \{\mathsf{a}\}$, and in the parallel rule, of Figure 3(e), $\mathtt{BN}(\mathtt{x}) = \mathtt{BN}(\mathtt{x}') = \emptyset$, while $\mathtt{BN}(\mathtt{y}') = \{\mathsf{a}\}$.

To define $\mathtt{BN}$ in general we use the fact that, since Conditions (1) and (2) are satisfied, we have a bijection

$$\mathtt{X} \;\cong\; [1,\mathrm{ar}_{\mathsf{X}}(\underline{\mathsf{op}})] + \coprod_{j\in[1,\mathrm{ar}_{\mathsf{X}}(\underline{\mathsf{op}})]} \left\{(\mathtt{x},\mathtt{l},\mathtt{y}) \in \mathit{Prem} \;\middle|\; \mathtt{x} = \underline{\mathtt{x}}_j\right\} \qquad (5)$$

whose inverse maps $j \in [1,\mathrm{ar}_{\mathsf{X}}(\underline{\mathsf{op}})]$ to $\underline{\mathtt{x}}_j$, and $\iota_j(\mathtt{x},\mathtt{l},\mathtt{y})$ to $\mathtt{y}$. Now:

GSOS-like conditions:

(1) Every term variable appears exactly once in the conclusion source and the premise targets.

(2) The source of every premise appears in the conclusion source.

Conditions relating to name binding:

(3) For each term variable in the conclusion source, the binding names are distinct.

$$\forall j \in [1, \mathrm{ar}_{\mathsf{X}}(\underline{\mathtt{op}})],\ k, k' \in [1, \mathrm{bdep}_{\underline{\mathtt{op}}}(j)].\ \underline{\mathtt{a}}^j_k = \underline{\mathtt{a}}^j_{k'} \implies k = k'$$

(4) No free names bind in the conclusion source.

$$\forall j \in [1, \mathrm{ar}_{\mathsf{X}}(\underline{\mathtt{op}})],\ k \in [1, \mathrm{bdep}_{\underline{\mathtt{op}}}(j)].\ \underline{\mathtt{a}}^j_k \notin \mathtt{FN}(\underline{\mathtt{src}}, \mathit{Prem})$$

(5) For each premise, binding names in the label are fresh for the source.

$$\forall (\mathtt{x}, \mathtt{l}, \mathtt{y}) \in \mathit{Prem}.\ \mathrm{bn}(\mathtt{l}) \cap \mathtt{FN}(\mathtt{x}) = \emptyset$$

(6) Free names of the conclusion label are not bound in the source or premises.

$$\mathrm{fn}(\underline{\mathtt{l}}) \cap \mathtt{BN}(\underline{\mathtt{src}}, \mathit{Prem}) = \emptyset$$

(7) No names become unbound in the induced transition.

$$\mathtt{FN}(\underline{\mathtt{tar}}) \cap \mathtt{BN}(\underline{\mathtt{src}}, \mathit{Prem}) = \emptyset$$

(8) Variable binding in the conclusion target is appropriate.

$$\mathcal{WF}(\underline{\mathtt{tar}})$$

Fig. 4. Conditions on rule structures.

- For $j \in [1, \mathrm{ar}_{\mathsf{X}}(\underline{\mathtt{op}})]$ we let $\mathtt{BN}(\underline{\mathtt{x}}_j) = \left\{ \underline{\mathtt{a}}^j_k \,\middle|\, k \in [1, \mathrm{bdep}_{\underline{\mathtt{op}}}(j)] \right\}$.
- For $(\mathtt{x}, \mathtt{l}, \mathtt{y}) \in \mathit{Prem}$ we let $\mathtt{BN}(\mathtt{y}) = \mathtt{BN}(\mathtt{x}) \cup \mathrm{bn}(\mathtt{l})$.

Finally, we write $\mathtt{BN}(\underline{\mathtt{src}}, \mathit{Prem}) \subseteq \mathtt{N}$ for the set

$$\mathtt{BN}(\underline{\mathtt{src}}, \mathit{Prem}) = \bigcup_{\mathtt{x} \in \mathtt{X}} \mathtt{BN}(\mathtt{x})$$

of all name variables that appear in binding position in the conclusion source or the premise labels.

We now associate to each variable $\mathtt{x} \in \mathtt{X}$ a set $\mathtt{FN}(\mathtt{x}) \subseteq \mathtt{N}$, which approximates (from the point of view of the rule) the names that appear free when the variable $\mathtt{x}$ is instantiated. To do this we first define the set $\mathtt{FN}(\underline{\mathtt{src}}, \mathit{Prem}) \subseteq \mathtt{N}$

that approximates the names that will be free in the conclusion source when it is instantiated.

$$\mathtt{FN}(\underline{\mathtt{src}}, \mathit{Prem}) \;=\; \left\{\underline{\mathsf{c}}_i \;\middle|\; i \in [1, \mathrm{ar}_{\mathsf{N}}(\underline{\mathsf{op}})]\right\} \;\cup \bigcup_{(\mathtt{x},\mathtt{l},\mathtt{y})\in \mathit{Prem}} \mathrm{fn}(\mathtt{l}) \setminus \mathtt{BN}(\mathtt{x})$$

Finally, for any $\mathtt{x} \in \mathtt{X}$, we let

$$\mathtt{FN}(\mathtt{x}) = \mathtt{FN}(\underline{\mathtt{src}}, \mathit{Prem}) \cup \mathtt{BN}(\mathtt{x}) \quad .$$

The function $\mathtt{FN}$ extends to compound terms with substitutions. For $\mathtt{t} \in T_{(\Sigma+\mathsf{sub}),\mathbf{Set}_{\mathtt{N}}}\mathtt{X}$ with

$$\mathtt{t} = \mathsf{op}\left((\mathsf{c}_i)_{i\in[1,\mathrm{ar}_{\mathsf{N}}(\mathsf{op})]}, \left(\langle \mathsf{a}^j_k\rangle_{k\in[1,\mathrm{bdep}_{\mathsf{op}}(j)]}\mathtt{t}_j\right)_{j\in[1,\mathrm{ar}_{\mathsf{X}}(\mathsf{op})]}\right)$$

we define $\mathtt{FN}(\mathtt{t}) \subseteq \mathtt{N}$ by

$$\begin{aligned}\mathtt{FN}(\mathtt{t}) \;=\;& \{\mathsf{c}_i \mid i \in [1, \mathrm{ar}_{\mathsf{N}}(\mathsf{op})]\} \;\cup \\ & \bigcup\left\{\mathtt{FN}(\mathtt{t}_j) \;\setminus \left\{\mathsf{a}^j_k \;\middle|\; k \in [1, \mathrm{bdep}_{\mathsf{op}}(j)]\right\} \;\middle|\; j \in [1, \mathrm{ar}_{\mathsf{X}}(\mathsf{op})]\right\}\end{aligned}$$

As an example, consider the scope closure rule structure, Figure 3(h), where we have $\mathtt{FN}(\underline{\mathtt{src}}, \mathit{Prem}) = \{\mathsf{c}\}$, while $\mathtt{BN}(\mathtt{y}) = \mathtt{BN}(\mathtt{y}') = \{\mathsf{a}\}$, and so we have $\mathtt{FN}(\mathtt{y}) = \mathtt{FN}(\mathtt{y}') = \{\mathsf{a}, \mathsf{c}\}$. However, $\mathtt{FN}(\underline{\mathtt{tar}}) = \{\mathsf{c}\}$.

For the communication rule structure, Figure 3(g), we have

$$\mathtt{FN}(\underline{\mathtt{src}}, \mathit{Prem}) = \mathtt{FN}(\mathtt{y}) = \{\mathsf{c}, \mathsf{d}\}$$

while $\mathtt{BN}(\mathtt{y}') = \{\mathsf{a}\}$ and so $\mathtt{FN}(\mathtt{y}') = \{\mathsf{a}, \mathsf{c}, \mathsf{d}\}$. However, $\mathtt{FN}(\underline{\mathtt{tar}}) = \{\mathsf{c}, \mathsf{d}\}$. (Recall that $[\mathsf{d}/\mathsf{a}]\mathsf{y}'$ is an abbreviation for $\mathsf{sub}(\mathsf{d}, \langle\mathsf{a}\rangle\mathsf{y}')$.)

Well-formed conclusion targets. Condition (8) asserts that the predicate $\mathcal{WF}$ holds of the conclusion target. Informally, this predicate requires that a binding variable is not used to bind in one term variable in the conclusion source and in a different term variable in the conclusion target. For instance, consider a $\mathsf{strange}$ operator taking two term parameters, the first one with a binder, and the following rule structure.

$$\frac{—}{\mathsf{strange}(\langle\mathsf{a}\rangle\mathtt{x}, \mathtt{x}') \xrightarrow{\tau} \mathsf{res}(\langle\mathsf{a}\rangle\mathsf{par}(\mathtt{x}, \mathtt{x}'))}$$

Here the scope of the binder a in the conclusion target encompasses both $\mathtt{x}$ and $\mathtt{x}'$, but was previously only binding in $\mathtt{x}$; thus the conclusion target is not well-formed.

Formally, the predicate $\mathcal{WF}$ is defined by induction on the structure of the set $T_{(\Sigma+\mathsf{sub}),\mathbf{Set}_{\mathtt{N}}}(\mathtt{X})$, as follows.

- For $\mathtt{x} \in \mathtt{X}$, we always let $\mathcal{WF}(\mathtt{x})$.
- For $\mathtt{t} = \mathsf{op}\left((\mathsf{c}_i)_{i\in[1,\mathrm{ar}_{\mathsf{N}}(\mathsf{op})]}, \left(\langle \mathsf{a}^j_k\rangle_{k\in[1,\mathrm{bdep}_{\mathsf{op}}(j)]}\mathtt{t}_j\right)_{j\in[1,\mathrm{ar}_{\mathsf{X}}(\mathsf{op})]}\right)$, we let $\mathcal{WF}(\mathtt{t})$ if: for all $j \in [1, \mathrm{ar}_{\mathsf{X}}(\mathsf{op})]$ we have $\mathcal{WF}(\mathtt{t}_j)$ and, furthermore, for all $k \in [1, \mathrm{bdep}_{\mathsf{op}}(j)]$, if $\mathsf{a}^j_k \in \mathtt{BN}(\underline{\mathtt{src}}, \mathit{Prem})$ then for all $\mathtt{x}$ appearing in $\mathtt{t}_j$ we have $\mathsf{a}^j_k \in \mathtt{FN}(\mathtt{x})$.

3.2 Necessity of conditions

If one of Conditions (1–8) is violated then wide-open bisimilarity need not be a congruence for the induced transition system. The reasons suggested by Bloom et al. [4, App. A] justify Conditions (1) and (2).

To justify Conditions (3–8), we give some examples of extensions of the π-calculus.

For all these examples, it is helpful to consider two names $a, b \in \mathcal{N}$, and define two π-calculus terms

$$t_1 = \mathsf{nil}\ , \qquad t_2 = \mathsf{match}(a, b, \mathsf{nil}) \quad .$$

Crucially, t_1 and t_2 are bisimilar, since neither process can perform any actions, but t_2 has two free names, while t_1 has none.

Now, to justify **Condition (3)**, we consider an operator $\mathsf{if\text{-}fresh}$, which takes no name parameters but one term parameter with two binders, with semantics given by the formal rule structure

$$\frac{—}{\mathsf{if\text{-}fresh}(\langle \mathsf{a}\rangle\langle \mathsf{a}\rangle \mathtt{x}) \xrightarrow{\tau} \mathtt{x}}$$

which violates Condition (3). If the $\mathsf{if\text{-}fresh}$ construct was allowed, the semantics would be that if $b\#\langle a\rangle t$ then $\mathsf{if\text{-}fresh}(\langle b\rangle\langle a\rangle t)$ performs a τ transition to t, because in that case $\langle b\rangle\langle a\rangle t = \langle a\rangle\langle a\rangle t$. Thus the context $\mathsf{if\text{-}fresh}(\langle b\rangle\langle a\rangle(-))$ would distinguish the bisimilar terms t_1 and t_2. **Condition (4)** is justified in a similar manner, for instance by considering an operator $\mathsf{if\text{-}fresh\text{-}2}$ with one name parameter and one term parameter with a binder, and with behaviour specified by the following rule structure.

$$\frac{—}{\mathsf{if\text{-}fresh\text{-}2}(\mathsf{c}, \langle \mathsf{c}\rangle \mathtt{x}) \xrightarrow{\tau} \mathtt{x}}$$

For **Condition (5)**, we consider an operator $\mathsf{tau\text{-}if\text{-}inp}$ with one name parameter and one term parameter, and no binders. The behaviour of this operator is specified by the following formal rule structure, which violates Condition (5).

$$\frac{\mathsf{x} \xrightarrow{\mathsf{c?(a)}} \mathsf{y}}{\mathsf{tau\text{-}if\text{-}inp}(\mathsf{a}, \mathsf{x}) \xrightarrow{\tau} \mathsf{y}}$$

Informally, if t can perform an input action specifically with data a, to become t', then $\mathsf{tau\text{-}if\text{-}inp}(a, t)$ will perform a silent action to t'.

The context $\mathsf{tau\text{-}if\text{-}inp}(a, \mathsf{inp}(c, \langle b \rangle(-)))$ will distinguish the bisimilar terms t_1 and t_2. For $\mathsf{inp}(c, \langle b \rangle(t_1))$ can perform an input of a (on c), while $\mathsf{inp}(c, \langle b \rangle(t_2))$ cannot.

To justify **Condition (6)**, we consider an operator $\mathsf{in\text{-}to\text{-}out}$ which has one term parameter, and no binders. The behaviour of this operator is specified by the following formal rule structure, which violates Condition (6).

$$\frac{\mathsf{x} \xrightarrow{\mathsf{c?(a)}} \mathsf{y}}{\mathsf{in\text{-}to\text{-}out}(\mathsf{x}) \xrightarrow{\mathsf{c!a}} \mathsf{y}}$$

So, whenever t can perform an input action, then $\mathsf{in\text{-}to\text{-}out}(t)$ can perform an output action with the same channel and data. The context $\mathsf{in\text{-}to\text{-}out}(\mathsf{inp}(c, \langle a \rangle(-)))$ will distinguish the bisimilar terms t_1 and t_2. For we have the transition

$$\begin{aligned} \mathsf{in\text{-}to\text{-}out}(\mathsf{inp}(c, \langle a \rangle t_1)) &= \mathsf{in\text{-}to\text{-}out}(\mathsf{inp}(c, \langle a \rangle \mathsf{nil})) \\ &=_\alpha \mathsf{in\text{-}to\text{-}out}(\mathsf{inp}(c, \langle b \rangle \mathsf{nil})) \\ &\xrightarrow{c?(b)} \mathsf{in\text{-}to\text{-}out}(\mathsf{inp}(c, \langle b \rangle \mathsf{nil})) \end{aligned}$$

which cannot be matched by $\mathsf{in\text{-}to\text{-}out}(\mathsf{inp}(c, \langle a \rangle t_2))$.

Condition (7) prohibits rule structures such as the following.

$$\frac{\text{—}}{\mathsf{res}(\langle \mathsf{a} \rangle \mathsf{x}) \xrightarrow{\tau} \mathsf{x}}$$

Informally: restrictions can be silently forgotten. If this rule structure was permitted, we would have a sequence of transitions

$$\mathsf{res}(\langle a \rangle(\mathsf{out}(c, a, t_1))) =_\alpha \mathsf{res}(\langle b \rangle(\mathsf{out}(c, b, t_1))) \xrightarrow{\tau} \mathsf{out}(c, b, t_1) \xrightarrow{c!b} \mathsf{nil}$$

which cannot be matched by $\mathsf{res}(\langle a \rangle(\mathsf{out}(c, a, t_2)))$. Thus the context $\mathsf{res}(\langle a \rangle(\mathsf{out}(c, a, (-))))$ is able to distinguish the bisimilar terms t_1 and t_2.

Finally, we consider **Condition (8)**. Recall the $\mathsf{strange}$ operator, with semantics given by the following axiom, that violates this condition.

$$\frac{\text{—}}{\mathsf{strange}(\langle \mathsf{a} \rangle \mathsf{x}, \mathsf{x'}) \xrightarrow{\tau} \mathsf{res}(\langle \mathsf{a} \rangle \mathsf{par}(\mathsf{x}, \mathsf{x'}))}$$

The context $\mathsf{strange}(\langle a\rangle(-),\mathsf{out}(c,b,\mathsf{nil}))$ can distinguish between the bisimilar processes t_1 and t_2. For we have the sequence of transitions

$$\begin{aligned}
\mathsf{strange}(\langle a\rangle t_1,\mathsf{out}(c,b,\mathsf{nil})) &= \mathsf{strange}(\langle a\rangle\mathsf{nil},\mathsf{out}(c,b,\mathsf{nil}))\\
&=_\alpha \mathsf{strange}(\langle b\rangle\mathsf{nil},\mathsf{out}(c,b,\mathsf{nil}))\\
&\xrightarrow{\tau} \mathsf{res}(\langle b\rangle\mathsf{par}(\mathsf{nil},\mathsf{out}(c,b,\mathsf{nil})))\\
&\xrightarrow{c!(b)} \mathsf{par}(\mathsf{nil},\mathsf{nil})
\end{aligned}$$

which cannot be matched by $\mathsf{strange}(\langle a\rangle t_2,\mathsf{out}(c,b,\mathsf{nil}))$.

4 Congruence of bisimilarity

In this section we introduce a categorical model theory for our rule format, and, by relating this with the model theory of our logic (Sec. 2), we arrive at our congruence result.

From the categorical perspective, the formal transition system specifications provide the definition of behavioural coalgebras by initial algebra recursion. With this in mind, we redevelop our notions of behavioural model — ground/wide-open transition systems and bisimulations — in the coalgebraic setting (Sec. 4.1), and subsequently give a categorical model theory for the semantics of name-passing systems (Sec. 4.2). We explain how every transition system specification gives rise to an abstract rule (Sec. 4.3), and we conclude this section (in Sec. 4.4) by explaining that the semantics induced from the categorical model theory corresponds exactly with the intended model of a specification given in Sec. 2.5. Thus the congruence result from the categorical model theory is relevant in the logical setting.

4.1 Transition systems as coalgebras

We define an endofunctor L_g on **Nom** by

$$L_\mathrm{g}(-) = \mathcal{N}\times[\mathcal{N}](-) \ + \ \mathcal{N}\times\mathcal{N}\times(-) \ + \ \mathcal{N}\times[\mathcal{N}](-) \ + \ (-) \quad .$$

For any nominal set X, an element of $L_\mathrm{g}X$ defines either an input (bin) behaviour (i.e. a channel name and a resumption state with one name bound); an output (out) behaviour, with the output data paired rather than bound; or a bound output (bout), or silent (tau) behaviour.

To introduce non-determinism we consider the covariant powerset endofunctor $\mathcal{P}$ on **Nom**. For any nominal set X, the (not-necessarily-equivariant) subsets S of X are equipped with the pointwise permutation action:

$$\sigma \cdot_{\mathcal{P}X} S = \{\sigma \cdot_X x \mid x \in S\} \quad .$$

With this action, there may be subsets of X without finite support. We define a nominal set

$$\mathcal{P}X = \{S \subseteq X \mid S \text{ has finite support}\}$$

with the above group action. (In fact, $\mathcal{P}X$ is the powerobject of X when the category of nominal sets is considered as a topos.)

An equivariant function $f : X \to Y$ gives rise to an equivariant function $\mathcal{P}f : \mathcal{P}X \to \mathcal{P}Y$, by direct image. Thus we have an endofunctor $\mathcal{P}$ on **Nom**.

We write B_{g} for the composite endofunctor $\mathcal{P}L_{\mathrm{g}}$ on **Nom**. A **B_{g}-coalgebra** is a pair (X, h) of a nominal set X and an equivariant function $h : X \to B_{\mathrm{g}}X$. The set X should be thought of as a set of states, and the function h assigns to each state a set of possible behaviours.

If (X, h) is a B_{g}-coalgebra, then we say that an equivariant relation $R \subseteq X \times X$ is a **B_{g}-bisimulation** if there is a B_{g}-coalgebra with carrier R making the following diagram commute in **Nom**.

$$\begin{tikzcd}
R \arrow[rr, hook] \arrow[d] & & X \times X \arrow[d, "h\times h"] \\
B_{\mathrm{g}}R \arrow[r] & B_{\mathrm{g}}(X \times X) \arrow[r, "\langle B_{\mathrm{g}}\pi_1, B_{\mathrm{g}}\pi_2\rangle"'] & B_{\mathrm{g}}X \times B_{\mathrm{g}}X
\end{tikzcd} \tag{6}$$

These coalgebraic notions are related with models of nominal logic theories (Sec. 1.5) as follows.

Proposition 4.1.

(1) To give a B_{g}-coalgebra is to give a ground transition system.
(2) To give a B_{g}-bisimulation on a B_{g}-coalgebra is to give a model of the theory of ground bisimulation. □

A proof of these statements is provided in Appendix A. For item (1), it is convenient to establish the correspondence between ground transition systems and equivariant relations of the form

$$\longrightarrow \subseteq X \times L_{\mathrm{g}}X$$

which straightforwardly correspond to B_{g}-coalgebras.

To model wide-open transition systems, and wide-open bisimulation, we must combine the coalgebraic notions of transition system and of bisimulation with the theory of nominal substitution. To this end, let $U : \mathbf{NSub} \to \mathbf{Nom}$ be the functor that forgets the substitution structure (Sec. 2.4). A **U-structured B_g-coalgebra** is given by a pair (X, h) of a nominal substitution X and a B_g-coalgebra structure $h : UX \to B_\mathrm{g}(UX)$. (This terminology should not be confused with the different usage considered by Corradini et al. [e.g. 6].)

If (X, h) is a U-structured B_g-coalgebra, then we say that a relation $R \subseteq X \times X$ in **NSub**, i.e. a substitution-closed equivariant relation, is a **U-structured B_g-bisimulation** if $UR \subseteq UX \times UX$ is a B_g-bisimulation on the underlying B_g-coalgebra. (Here, we have used the fact that the forgetful functor $U : \mathbf{NSub} \to \mathbf{Nom}$ preserves products, as mentioned in Sec. 2.4.)

Structured coalgebras and bisimulation are compared with the notions of wide-open transition system and wide-open bisimulation in the following corollary of Proposition 4.1.

Proposition 4.2.

(1) *To give a U-structured B_g-coalgebra is to give a wide-open transition system.*
(2) *To give a U-structured B_g-bisimulation on a U-structured B_g-coalgebra is to give a model of the theory of wide-open bisimulation.* □

4.2 Mathematical Operational Semantics for name-passing calculi

We begin this section with a standard parameterised-recursion theorem, and explain why the recursion data can be thought of as an abstract kind of GSOS rule. A congruence result is obtained by considering a naturality condition.

In this subsection, we fix an algebraic binding signature, Σ (see Defn. 1.2).

Proposition 4.3. *For every equivariant function*

$$\varrho : \Sigma_{\mathbf{Nom}}\Big((T_{\Sigma,\mathbf{Nom}}\emptyset) \times B_\mathrm{g}(T_{\Sigma,\mathbf{Nom}}\emptyset)\Big) \to B_\mathrm{g}(T_{\Sigma,\mathbf{Nom}}\emptyset)$$

there is a unique B_g-coalgebra $\varrho^\sharp : T_{\Sigma,\mathbf{Nom}}\emptyset \to B_\mathrm{g}(T_{\Sigma,\mathbf{Nom}}\emptyset)$ making the following diagram commute.

$$\begin{tikzcd}
\Sigma_{\mathbf{Nom}}(T_{\Sigma,\mathbf{Nom}}\emptyset) \arrow[r, "\cong"] \arrow[d, "\Sigma_{\mathbf{Nom}}\langle \mathsf{id},\varrho^\sharp\rangle"'] & T_{\Sigma,\mathbf{Nom}}\emptyset \arrow[d, "\varrho^\sharp"] \\
\Sigma_{\mathbf{Nom}}\Big((T_{\Sigma,\mathbf{Nom}}\emptyset) \times B_\mathrm{g}(T_{\Sigma,\mathbf{Nom}}\emptyset)\Big) \arrow[r, "\varrho"'] & B_\mathrm{g}(T_{\Sigma,\mathbf{Nom}}\emptyset)
\end{tikzcd} \tag{7}$$

□

(This result is a consequence of a general parameterised recursion theorem about initial algebras in categories with products.)

The recursion data ϱ used in this proposition can be seen as a GSOS-like rule, as follows. The hom-set

$$\mathbf{Nom}\Big(T_{\Sigma,\mathbf{Nom}}\emptyset,\ B_{\mathrm{g}}(T_{\Sigma,\mathbf{Nom}}\emptyset)\Big)$$

of B_{g}-coalgebras with carrier $T_{\Sigma,\mathbf{Nom}}\emptyset$ inherits a partial order structure from B_{g}: we let $h \leq k : T_{\Sigma,\mathbf{Nom}}\emptyset \to B_{\mathrm{g}}(T_{\Sigma,\mathbf{Nom}}\emptyset)$ if $h(t) \subseteq k(t)$ for every term $t \in T_{\Sigma,\mathbf{Nom}}\emptyset$. We define an operator Φ_ϱ on this partial order, taking a coalgebra $h : T_{\Sigma,\mathbf{Nom}}\emptyset \to B_{\mathrm{g}}(T_{\Sigma,\mathbf{Nom}}\emptyset)$ to the following composite, $\Phi_\varrho(h)$:

$$\begin{tikzcd}
T_{\Sigma,\mathbf{Nom}}\emptyset \arrow[d, "\cong"] \\
\Sigma_{\mathbf{Nom}}(T_{\Sigma,\mathbf{Nom}}\emptyset) \arrow[d, "\Sigma_{\mathbf{Nom}}\langle \mathsf{id},h\rangle"] \\
\Sigma_{\mathbf{Nom}}(T_{\Sigma,\mathbf{Nom}}\emptyset \times B_{\mathrm{g}}(T_{\Sigma,\mathbf{Nom}}\emptyset)) \arrow[d, "\varrho"] \\
B_{\mathrm{g}}(T_{\Sigma,\mathbf{Nom}}\emptyset)\ .
\end{tikzcd}$$

Intuitively, the operator Φ_ϱ first extracts the outermost operator of a term. It then derives the behaviours, using the coalgebra h, for the subterms below the outermost operator. At this point, we have a term in which each top-level parameter is associated with a behaviour. We then apply the function ϱ to arrive at a behaviour for the term itself. We can think of the resulting coalgebra as describing the behaviour of terms via ϱ, based on the behaviour h. A coalgebra h will be said to **satisfy** ϱ if $\Phi_\varrho(h) \leq h$. Proposition 4.3 says that Φ_ϱ has a unique fixed point, $\varrho^\sharp$, that satisfies ϱ. When ϱ is thought of as a transition system specification, this fixed point can be thought of as the intended model of ϱ.

The partial order is, in fact, a complete lattice, with the join of a family of coalgebras given by pointwise union. If Φ_ϱ is a monotone operator, then $\varrho^\sharp$ is the least pre-fixed point of Φ_ϱ, i.e. it is the least coalgebra that satisfies ϱ.

From another perspective, it is difficult to think of the recursion data ϱ as a rule because it applies only to the specific case of terms in $T_{\Sigma,\mathbf{Nom}}\emptyset$, whereas a rule should apply to models in some generality. Thus an **abstract rule** is a natural transformation of the form

$$\rho\ :\ \Sigma_{\mathbf{Nom}}(U(-) \times B_{\mathrm{g}}(U(-))) \to B_{\mathrm{g}}(U(T_{\Sigma,\mathbf{NSub}}-))\ \ :\ \mathbf{NSub} \to \mathbf{Nom}\quad .$$

An abstract rule gives rise to recursion data, as the following composite:

$$\begin{tikzcd}
\Sigma_{\mathbf{Nom}}((T_{\Sigma,\mathbf{Nom}}\emptyset) \times B_{\mathrm{g}}(T_{\Sigma,\mathbf{Nom}}\emptyset)) \arrow[d, "="] \\
\Sigma_{\mathbf{Nom}}(U(T_{\Sigma,\mathbf{NSub}}\emptyset) \times B_{\mathrm{g}}(U(T_{\Sigma,\mathbf{NSub}}\emptyset))) \arrow[d, "\rho(T_{\Sigma,\mathbf{NSub}}\emptyset)"] \\
B_{\mathrm{g}}(U(T_{\Sigma,\mathbf{NSub}}T_{\Sigma,\mathbf{NSub}}\emptyset)) \arrow[d, "B_{\mathrm{g}}U\mu_{\emptyset}"] \\
B_{\mathrm{g}}(U(T_{\Sigma,\mathbf{NSub}}\emptyset)) \arrow[d, "="] \\
B_{\mathrm{g}}(T_{\Sigma,\mathbf{Nom}}\emptyset)
\end{tikzcd}$$

using the multiplication μ of the monad $T_{\Sigma,\mathbf{NSub}}$ that flattens a term made of closed terms into a closed term.

Theorem 4.4. *If a U-structured B_{g}-coalgebra with carrier $T_{\Sigma,\mathbf{NSub}}(\emptyset)$ is defined by an abstract rule, then the greatest U-structured B_{g}-bisimulation is a Σ-congruence.* □

This theorem does not follow immediately from the developments of Turi and Plotkin [22] because we have used a novel notion of structured coalgebra. The result can be reduced to their model, however, because the forgetful functor $\mathbf{NSub} \to \mathbf{Nom}$ has a right adjoint (see [9, Sec. 3], [8, Sec. 1]). Interestingly, though, the right adjoint is *not* necessary for the result; we will explicate the matter more thoroughly in future work, for it is beyond the scope of the present article [though see 8, Sec 1].

4.3 From formal rule structures to abstract rules

Throughout this subsection we fix an algebraic binding signature, Σ (see Defn. 1.2) and a formal rule structure $\mathtt{R}$ over it (Defn. 1.2) that satisfies all the conditions of Figure 4. We suppose that $\mathtt{R}$ has variables from $\mathtt{N}$ and $\mathtt{X}$, with premise set *Prem* and conclusion with source

$$\underline{\mathtt{src}} \;=\; \underline{\mathtt{op}}\left((\underline{\mathtt{c}}_i)_{i\in[1,\mathrm{ar}_{\mathtt{N}}(\underline{\mathtt{op}})]},\left(\langle \underline{\mathtt{a}}_k^j\rangle_{k\in[1,\mathrm{bdep}_{\underline{\mathtt{op}}}(j)]}\underline{\mathtt{x}}_j\right)_{j\in[1,\mathrm{ar}_{\mathtt{X}}(\underline{\mathtt{op}})]}\right) \;,$$

label $\underline{\mathtt{l}}$, and target $\underline{\mathtt{tar}}$.

We will explain how $\mathtt{R}$ gives rise to an abstract rule

$$[\![\mathtt{R}]\!] \;:\; \Sigma_{\mathbf{Nom}}(U(-) \times B_{\mathrm{g}}(U(-))) \to B_{\mathrm{g}}(U(T_{\Sigma,\mathbf{NSub}}-)) \;:\; \mathbf{NSub} \to \mathbf{Nom} \quad .$$

For a nominal substitution X, a **valuation** $\mathcal{V}$ of the rule structure $\mathtt{R}$ is a pair of functions $(\mathcal{V}_{\mathsf{N}} : \mathtt{N} \to \mathcal{N}, \mathcal{V}_{\mathsf{X}} : \mathtt{X} \to X)$ between sets.

To each valuation $\mathcal{V}$ we will assign an **archetypal parameter**

$$\mathcal{V}(\underline{\mathtt{src}}, \mathit{Prem}) \ \in\ \Sigma_{\mathbf{Nom}}((UX) \times B_{\mathrm{g}}(UX)) \quad .$$

This is to be thought of as a simultaneous instantiation of both the conclusion source and of the premises.

First, for each $j \in [1, \mathrm{ar}_{\mathsf{X}}(\underline{\mathsf{op}})]$, we instantiate the premises with source $\underline{\mathtt{x}}_j$, by defining $\mathcal{V}(\mathit{Prem}[j]) \in B_{\mathrm{g}}(UX)$.

$$\begin{aligned}\mathcal{V}(\mathit{Prem}[j]) \ = \ & \Big\{\mathsf{bin}(\mathcal{V}_{\mathsf{N}}(\mathtt{c}), \langle\mathcal{V}_{\mathsf{N}}(\mathtt{a})\rangle\mathcal{V}_{\mathsf{X}}(\mathtt{y})) \ \Big|\ (\underline{\mathtt{x}}_j, \mathtt{c}?(\mathtt{a}), \mathtt{y}) \in \mathit{Prem}\Big\} \\ & \cup \Big\{\mathsf{out}(\mathcal{V}_{\mathsf{N}}(\mathtt{c}), \mathcal{V}_{\mathsf{N}}(\mathtt{d}), \mathcal{V}_{\mathsf{X}}(\mathtt{y})) \ \Big|\ (\underline{\mathtt{x}}_j, \mathtt{c}!\mathtt{d}, \mathtt{y}) \in \mathit{Prem}\Big\} \\ & \cup \Big\{\mathsf{bout}(\mathcal{V}_{\mathsf{N}}(\mathtt{c}), \langle\mathcal{V}_{\mathsf{N}}(\mathtt{a})\rangle\mathcal{V}_{\mathsf{X}}(\mathtt{y})) \ \Big|\ (\underline{\mathtt{x}}_j, \mathtt{c}!(\mathtt{a}), \mathtt{y}) \in \mathit{Prem}\Big\} \\ & \cup \Big\{\mathsf{tau}(\mathcal{V}_{\mathsf{X}}(\mathtt{y})) \ \Big|\ (\underline{\mathtt{x}}_j, \tau, \mathtt{y}) \in \mathit{Prem}\Big\}\end{aligned}$$

Now the archetypal parameter $\mathcal{V}(\underline{\mathtt{src}}, \mathit{Prem})$ is given by

$$\underline{\mathsf{op}}\left(\begin{array}{l}(\mathcal{V}_{\mathsf{N}}(\underline{\mathsf{c}}_i))_{i\in[1,\mathrm{ar}_{\mathsf{N}}\underline{\mathsf{op}}]}\,, \\ \Big(\langle\mathcal{V}_{\mathsf{N}}(\underline{\mathsf{a}}^j_k)\rangle_{k\in[1,\mathrm{bdep}_{\underline{\mathsf{op}}}(j)]}\,\big(\mathcal{V}_{\mathsf{X}}(\underline{\mathtt{x}}_j), \mathcal{V}(\mathit{Prem}[j])\big)\Big)_{j\in[1,\mathrm{ar}_{\mathsf{X}}(\underline{\mathsf{op}})]}\end{array}\right) .$$

Archetypal result. To each valuation $\mathcal{V}$ we assign an **archetypal result** $\mathcal{V}(\underline{\mathtt{l}}, \underline{\mathtt{tar}}) \in L_{\mathrm{g}}(U(T_{\Sigma,\mathbf{NSub}}X))$. This is to be thought of as a simultaneous instantiation of both the conclusion label and of the conclusion target.

First, we consider how to instantiate the conclusion target. The conclusion target is a term in the theory of Σ-algebras with substitutions, while $T_{\Sigma,\mathbf{NSub}}(X)$ is a model of this theory. Let η_X be the unit of the monad $T_{\Sigma,\mathbf{NSub}}$, exhibiting every element of X as a term in $T_{\Sigma,\mathbf{NSub}}(X)$. Then the pair

$$(\mathcal{V}_{\mathsf{N}} : \mathtt{N} \to \mathcal{N},\ \mathtt{X} \xrightarrow{\mathcal{V}_{\mathsf{X}}} X \xrightarrow{\eta_X} T_{\Sigma,\mathbf{NSub}}(X))$$

provides a valuation of the variables of $\underline{\mathtt{tar}}$ into the model $T_{\Sigma,\mathbf{NSub}}(X)$. Through this valuation, the term $\underline{\mathtt{tar}}$ has an interpretation as an element of the set $T_{\Sigma,\mathbf{NSub}}(X)$. The nominal set underlying $T_{\Sigma,\mathbf{NSub}}(X)$ is $T_{\Sigma,\mathbf{Nom}}(UX)$, and we let $\mathcal{V}(\underline{\mathtt{tar}})$, in $T_{\Sigma,\mathbf{Nom}}(UX)$, be this element.

The archetypal result $\mathcal{V}(\underline{\mathtt{l}}, \underline{\mathtt{tar}})$ is dependent on the kind of conclusion label, as follows:

$$\begin{aligned}
&\text{for } \underline{\mathtt{l}} = \mathsf{c}?(\mathsf{a}),\ \ \mathcal{V}(\underline{\mathtt{l}}, \underline{\mathtt{tar}}) = \mathsf{bin}(\mathcal{V}_{\mathsf{N}}(\mathsf{c}), \langle \mathcal{V}_{\mathsf{N}}(\mathsf{a})\rangle \mathcal{V}(\underline{\mathtt{tar}}));\\
&\text{for } \underline{\mathtt{l}} = \mathsf{c}!\mathsf{d},\ \ \mathcal{V}(\underline{\mathtt{l}}, \underline{\mathtt{tar}}) = \mathsf{out}(\mathcal{V}_{\mathsf{N}}(\mathsf{c}), \mathcal{V}_{\mathsf{N}}(\mathsf{d}), \mathcal{V}(\underline{\mathtt{tar}}));\\
&\text{for } \underline{\mathtt{l}} = \mathsf{c}!(\mathsf{a}),\ \ \mathcal{V}(\underline{\mathtt{l}}, \underline{\mathtt{tar}}) = \mathsf{bout}(\mathcal{V}_{\mathsf{N}}(\mathsf{c}), \langle \mathcal{V}_{\mathsf{N}}(\mathsf{a})\rangle \mathcal{V}(\underline{\mathtt{tar}}));\\
&\text{for } \underline{\mathtt{l}} = \tau,\ \ \mathcal{V}(\underline{\mathtt{l}}, \underline{\mathtt{tar}}) = \mathsf{tau}(\mathcal{V}(\underline{\mathtt{tar}})).
\end{aligned}$$

Abstract rules. The archetypal parameter of a valuation represents the smallest parameter that should be considered with that valuation. The same valuation, however, is also adequate for overspecified parameters; i.e. those that more than fulfill the premises. Formally, thus, we say that a valuation $\mathcal{V}$ is **adequate** for a parameter $s \in \Sigma_{\mathbf{Nom}}((UX) \times B_{\mathrm{g}}(UX))$ if $\mathcal{V}_{\mathsf{N}}$ is injective, and $\mathcal{V}_{\mathsf{N}}(\mathrm{bn}(\underline{\mathtt{l}}))\#s$, and if there are $\beta_j \subseteq L_{\mathrm{g}}(UX)$, for $j \in [1, \mathrm{ar}_{\mathsf{X}}(\underline{\mathsf{op}})]$, such that

$$s \;=\; \underline{\mathsf{op}}\left(\begin{array}{l}(\mathcal{V}_{\mathsf{N}}(\underline{\mathsf{c}}_i))_{i\in[1,\mathrm{ar}_{\mathsf{N}}(\underline{\mathsf{op}})]}\,,\\ \left(\langle \mathcal{V}_{\mathsf{N}}(\underline{\mathsf{a}}^j_k)\rangle_{k\in[1,\mathrm{bdep}_{\underline{\mathsf{op}}}(j)]}\,(\mathcal{V}_{\mathsf{X}}(\underline{\mathtt{x}}_j), \beta_j)\right)_{j\in[1,\mathrm{ar}_{\mathsf{X}}(\underline{\mathsf{op}})]}\end{array}\right)$$

and such that $\mathcal{V}(\mathit{Prem}[j]) \subseteq \beta_j$ for all $j \in [1, \mathrm{ar}_{\mathsf{X}}(\underline{\mathsf{op}})]$.

For every parameter $s \in \Sigma_{\mathbf{Nom}}((UX) \times B_{\mathrm{g}}(UX))$ we can thus derive a set $[\![\mathtt{R}]\!]_X(s) \subseteq L_{\mathrm{g}}(U(T_{\Sigma,\mathbf{NSub}}X))$ of possible results:

$$[\![\mathtt{R}]\!]_X(s) \;=\; \{\mathcal{V}(\underline{\mathtt{l}}, \underline{\mathtt{tar}}) \mid \mathcal{V} \text{ is an adequate valuation for } s\}$$

Proposition 4.5. *The mapping $s \mapsto [\![\mathtt{R}]\!]_X(s)$ yields an equivariant function*

$$[\![\mathtt{R}]\!]_X : \Sigma_{\mathbf{Nom}}((UX) \times B_{\mathrm{g}}(UX)) \to B_{\mathrm{g}}(U(T_{\Sigma,\mathbf{NSub}}X)) \quad .$$

Proof. We will first show that, for any $s \in \Sigma_{\mathbf{Nom}}((UX) \times B_{\mathrm{g}}(UX))$, and any permutation $\sigma \in \mathsf{Sym}(\mathcal{N})$,

$$[\![\mathtt{R}]\!]_X(\sigma \cdot_{\Sigma_{\mathbf{Nom}}((UX)\times B_{\mathrm{g}}(UX))} s) \;=\; \left\{\sigma \cdot_{L_{\mathrm{g}}(U(T_{\Sigma,\mathbf{NSub}}X))} b \,\middle|\, b \in [\![\mathtt{R}]\!]_X(s)\right\} \quad . \qquad (8)$$

In fact, because every permutation σ has an inverse, it is sufficient to prove that

$$\{\sigma \cdot b \mid b \in [\![\mathtt{R}]\!]_X(s)\} \;\subseteq\; [\![\mathtt{R}]\!]_X(\sigma \cdot s) \quad .$$

Suppose that $b \in [\![\mathtt{R}]\!]_X(s)$. So there is an adequate valuation $\mathcal{V}$ for s for which $b = \mathcal{V}(\underline{\mathtt{l}}, \underline{\mathtt{tar}})$.

We define a valuation $(\sigma \cdot \mathcal{V})$ into X by

$$\begin{aligned}
(\sigma \cdot \mathcal{V})_{\mathsf{N}}(\mathsf{a}) &= \sigma(\mathcal{V}_{\mathsf{N}}(\mathsf{a}))\\
(\sigma \cdot \mathcal{V})_{\mathsf{X}}(\mathtt{x}) &= \sigma \cdot_X (\mathcal{V}_{\mathsf{X}}(\mathtt{x})) \quad .
\end{aligned}$$

This valuation is adequate for $\sigma \cdot s$, and moreover, it is routine to verify that $(\sigma \cdot \mathcal{V})(\underline{\mathtt{l}}, \underline{\mathtt{tar}}) = \sigma \cdot (\mathcal{V}(\underline{\mathtt{l}}, \underline{\mathtt{tar}}))$ — for we have permuted everything in sight. Thus $\sigma \cdot b$ is in $[\![\mathtt{R}]\!]_X(\sigma \cdot s)$, and so

$$\{\sigma \cdot b \mid b \in [\![\mathtt{R}]\!]_X(s)\} \subseteq [\![\mathtt{R}]\!]_X(\sigma \cdot s)$$

as required. Hence (8) is established.

Since every $s \in \Sigma_{\mathbf{Nom}}((UX) \times B_{\mathrm{g}}(UX))$ has finite support, we know, then, that the set $[\![\mathtt{R}]\!]_X(s) \subseteq L_{\mathrm{g}}(U(T_{\Sigma,\mathbf{NSub}}X))$ is finitely supported: it follows from (8) that a permutation that fixes s also fixes $[\![\mathtt{R}]\!]_X(s)$. So $[\![\mathtt{R}]\!]_X(s)$ is in the nominal set $B_{\mathrm{g}}(U(T_{\Sigma,\mathbf{NSub}}X))$. Furthermore, we have already shown (8) that the mapping $s \mapsto [\![\mathtt{R}]\!]_X(s)$ is equivariant. $\square$

The following lemma is crucial.

Lemma 4.6. *Consider a finite set $N \subseteq_{\mathrm{f}} \mathcal{N}$ of names, and a nominal substitution X. For every adequate valuation for $s \in \Sigma_{\mathbf{Nom}}((UX) \times B_{\mathrm{g}}(UX))$, there is another adequate valuation for s with the same archetypal result, but that maps the binding variables of the rule, $\mathtt{BN}(\underline{\mathtt{src}}, \mathit{Prem}) \cup \mathrm{bn}(\underline{\mathtt{l}})$, outside of N.* $\square$

A proof of this lemma is provided in Appendix B. It uses all the conditions of Figure 4.

Theorem 4.7. *The family of functions $\{[\![\mathtt{R}]\!]_X\}_{X \in \mathbf{NSub}}$ is natural.*

Proof. Consider a homomorphism $f : X \to Y$ of nominal substitutions, and a parameter $s \in \Sigma_{\mathbf{Nom}}((UX) \times B_{\mathrm{g}}(UX))$. We must show that

$$B_{\mathrm{g}}(U(T_{\Sigma,\mathbf{NSub}}f))([\![\mathtt{R}]\!]_X(s)) \;=\; [\![\mathtt{R}]\!]_Y(\Sigma_{\mathbf{Nom}}((Uf) \times B_{\mathrm{g}}(Uf))(s))$$

That is, we must show that

$$\begin{aligned}\{L_{\mathrm{g}}(U(T_{\Sigma,\mathbf{NSub}}f))(\mathcal{V}(\underline{\mathtt{l}}, \underline{\mathtt{tar}})) \mid \mathcal{V} \text{ is an adequate valuation for } s\} \\ = \left\{ \mathcal{V}(\underline{\mathtt{l}}, \underline{\mathtt{tar}}) \;\middle|\; \begin{array}{l} \mathcal{V} \text{ is an adequate valuation} \\ \quad \text{for } \Sigma_{\mathbf{Nom}}((Uf) \times B_{\mathrm{g}}(Uf))(s) \end{array} \right\} \quad . \qquad (9)\end{aligned}$$

To see that the left-hand side of (9) is a subset of the right-hand side, we consider a valuation $\mathcal{V}$ that is adequate for s, and we construct the new valuation $(f \circ \mathcal{V}) = (\mathcal{V}_{\mathsf{N}}, f \circ \mathcal{V}_{\mathsf{X}})$. It is routine to verify that this valuation $(f \circ \mathcal{V})$ is adequate for $\Sigma_{\mathbf{Nom}}((Uf) \times B_{\mathrm{g}}(Uf))(s)$, and moreover that its archetypal result is $L_{\mathrm{g}}(U(T_{\Sigma,\mathbf{NSub}}f))(\mathcal{V}(\underline{\mathtt{l}}, \underline{\mathtt{tar}}))$.

Showing that the right-hand side of (9) is a subset of the left-hand side is more difficult. We consider a valuation $\mathcal{V}$ that is adequate for $\Sigma_{\mathbf{Nom}}((Uf) \times B_{\mathrm{g}}(Uf))(s)$, and we exhibit a valuation $\mathcal{V}'$ that is adequate for s and for which

$$\mathcal{V}(\underline{\mathtt{l}}, \underline{\mathtt{tar}}) = B_{\mathrm{g}}(U(T_{\Sigma,\mathbf{NSub}} f))(\mathcal{V}'(\underline{\mathtt{l}}, \underline{\mathtt{tar}})) \quad .$$

To begin, we recap the properties of $\mathcal{V}$ as an adequate valuation for

$$\Sigma_{\mathbf{Nom}}((Uf) \times B_{\mathrm{g}}(Uf))(s) \quad .$$

We have that $\mathcal{V}_{\mathsf{N}}$ is injective, and that

$$\Sigma_{\mathbf{Nom}}((Uf) \times B_{\mathrm{g}}(Uf))(s) \;=\; \underline{\mathsf{op}} \begin{pmatrix} (\mathcal{V}_{\mathsf{N}}(\underline{\mathsf{c}}_i))_{i \in [1, \mathrm{ar}_{\mathsf{N}}(\underline{\mathsf{op}})]}\, , \\ \Big(\langle \mathcal{V}_{\mathsf{N}}(\underline{\mathsf{a}}^j_k) \rangle_{k \in [1, \mathrm{bdep}_{\underline{\mathsf{op}}}(j)]} \, (\mathcal{V}_{\mathsf{X}}(\underline{\mathtt{x}}_j), \beta_j) \Big)_{j \in [1, \mathrm{ar}_{\mathsf{X}}(\underline{\mathsf{op}})]} \end{pmatrix} \tag{10}$$

where $\mathcal{V}(\mathit{Prem}[j]) \subseteq \beta_j$ for all $j \in [1, \mathrm{ar}_{\mathsf{X}}(\underline{\mathsf{op}})]$.

We now investigate the structure of s. From the functorial action of $\Sigma_{\mathbf{Nom}}$, we deduce that the operator-type of s must be $\underline{\mathsf{op}}$. So we have we have $c_i \in \mathcal{N}$ (for all $i \in [1, \mathrm{ar}_{\mathsf{N}}(\underline{\mathsf{op}})]$), and $a^j_k \in \mathcal{N}$ (for all $j \in [1, \mathrm{ar}_{\mathsf{X}}(\underline{\mathsf{op}})]$, $k \in [1, \mathrm{bdep}_j(\underline{\mathsf{op}})]$), and $x_j \in X$ and $\beta'_j \in B_{\mathrm{g}}(UX)$ (for all $j \in [1, \mathrm{ar}_{\mathsf{X}}(\underline{\mathsf{op}})]$), such that

$$s \;=\; \underline{\mathsf{op}} \begin{pmatrix} (c_i)_{i \in [1, \mathrm{ar}_{\mathsf{N}}(\underline{\mathsf{op}})]}\, , \\ \Big(\langle a^j_k \rangle_{k \in [1, \mathrm{bdep}_{\underline{\mathsf{op}}}(j)]} \, (x_j, \beta'_j) \Big)_{j \in [1, \mathrm{ar}_{\mathsf{X}}(\underline{\mathsf{op}})]} \end{pmatrix} . \tag{11}$$

By Lemma 4.6, we can assume, without loss of generality, that

$$\mathcal{V}_{\mathsf{N}}\Big(\mathtt{BN}(\underline{\mathtt{src}}, \mathit{Prem}) \cup \mathrm{bn}(\underline{\mathtt{l}})\Big) \cap \mathsf{supp}(s) \;=\; \emptyset \quad .$$

Thus, by definition of $[\mathcal{N}](-)$, and using Condition (3) and injectivity of $\mathcal{V}_{\mathsf{N}}$, we can also assume that $a^j_k = \mathcal{V}_{\mathsf{N}}(\underline{\mathsf{a}}^j_k)$ for every $j \in [1, \mathrm{ar}_{\mathsf{X}}(\underline{\mathsf{op}})]$, $k \in [1, \mathrm{bdep}_{\underline{\mathsf{op}}}(j)]$. Analysing (10) and (11), from the action of $\Sigma_{\mathbf{Nom}}((Uf) \times B_{\mathrm{g}}(Uf))$ we conclude that $c_i = \mathcal{V}_{\mathsf{N}}(\underline{\mathsf{c}}_i)$ for all $i \in [1, \mathrm{ar}_{\mathsf{N}}(\underline{\mathsf{op}})]$, and that

$$\text{(a)}\;\; f(x_j) = \mathcal{V}_{\mathsf{X}}(\underline{\mathtt{x}}_j) \qquad \text{and} \qquad \text{(b)}\;\; B_{\mathrm{g}}(Uf)(\beta'_j) = \beta_j \tag{12}$$

for all $j \in [1, \mathrm{ar}_{\mathsf{X}}(\underline{\mathsf{op}})]$.

For each $j \in [1, \mathrm{ar}_{\mathsf{X}}(\underline{\mathsf{op}})]$ and $l \in \mathit{Lab}(\mathtt{N})$, we define the sets $\mathtt{X}_{j,l} \subseteq \mathtt{X}$, $X_{j,l} \subseteq X$

as follows.

$$\begin{aligned}
\mathtt{X}_{j,l} &= \left\{ \mathtt{x} \in \mathtt{X} \;\middle|\; (\underline{\mathtt{x}}_j, l, \mathtt{x}) \in \mathit{Prem} \right\} \\
X_{j,\mathsf{c}?(\mathsf{a})} &= \left\{ x \in X \;\middle|\; \mathsf{bin}(\mathcal{V}_{\mathsf{N}}(\mathsf{c}), \langle \mathcal{V}_{\mathsf{N}}(\mathsf{a}) \rangle x) \in \beta'_j \right\} \\
X_{j,\mathsf{c}!\mathsf{d}} &= \left\{ x \in X \;\middle|\; \mathsf{out}(\mathcal{V}_{\mathsf{N}}(\mathsf{c}), \mathcal{V}_{\mathsf{N}}(\mathsf{d}), x) \in \beta'_j \right\} \\
X_{j,\mathsf{c}!(\mathsf{a})} &= \left\{ x \in X \;\middle|\; \mathsf{bout}(\mathcal{V}_{\mathsf{N}}(\mathsf{c}), \langle \mathcal{V}_{\mathsf{N}}(\mathsf{a}) \rangle x) \in \beta'_j \right\} \\
X_{j,\tau} &= \left\{ x \in X \;\middle|\; \mathsf{tau}(x) \in \beta'_j \right\}
\end{aligned}$$

We claim that, for each $j \in [1, \mathrm{ar}_{\mathsf{X}}(\underline{\mathsf{op}})]$ and $l \in \mathit{Lab}(\mathtt{N})$, we have

$$\mathcal{V}_{\mathsf{X}}(\mathtt{X}_{j,l}) \subseteq f(X_{j,l}) \quad . \tag{13}$$

Consider, for instance, some $(\underline{\mathtt{x}}_j, \mathsf{c}?(\mathsf{a}), \mathtt{x}) \in \mathit{Prem}$. So $\mathtt{x}$ is in $\mathtt{X}_{j,\mathsf{c}?(\mathsf{a})}$, and we must show that $\mathcal{V}_{\mathsf{X}}(\mathtt{x}) \in f(X_{j,l})$, i.e. that

$$\exists x \in X_{j,l}.\ \mathcal{V}_{\mathsf{X}}(\mathtt{x}) = f(x) \quad . \tag{14}$$

Because $\mathcal{V}$ is adequate, we have $\mathsf{bin}(\mathcal{V}_{\mathsf{N}}(\mathsf{c}), \langle \mathcal{V}_{\mathsf{N}}(\mathsf{a}) \rangle \mathcal{V}_{\mathsf{X}}(\mathtt{x})) \in \beta_j$. It follows from (12)(b) that there must exist $b' \in \beta'_j$ such that $\mathsf{bin}(\mathcal{V}_{\mathsf{N}}(\mathsf{c}), \langle \mathcal{V}_{\mathsf{N}}(\mathsf{a}) \rangle \mathcal{V}_{\mathsf{X}}(\mathtt{x})) = L_{\mathrm{g}}(Uf)(b')$. Indeed, we must have $x' \in X$ and $a \in \mathcal{N}$ such that $b' = \mathsf{bin}(\mathcal{V}_{\mathsf{N}}(\mathsf{c}), \langle a \rangle x')$ and $\langle a \rangle f(x') = \langle \mathcal{V}_{\mathsf{N}}(\mathsf{a}) \rangle \mathcal{V}_{\mathsf{X}}(\mathtt{x})$.

Consider a name $b \in \mathcal{N}$ that is fresh for β'_j and for x', a and $\mathcal{V}_{\mathsf{N}}(\mathsf{a})$. Since f is equivariant, we have $\langle a \rangle f(x') = \langle b \rangle f((a\ b) \cdot x')$. We let

$$x \;=\; (\mathcal{V}_{\mathsf{N}}(\mathsf{a})\ b)(a\ b) \cdot x'$$

and we know that $f(x) = \mathcal{V}_{\mathsf{X}}(\mathtt{x})$. To conclude (14), it remains for us to show that x is in $X_{j,\mathsf{c}?(\mathsf{a})}$.

We have made sure, using Lemma 4.6, that $\mathcal{V}_{\mathsf{N}}(\mathsf{a}) \# s$, and, because of Conditions (3) and (5), we know that $\mathcal{V}_{\mathsf{N}}(\mathsf{a}) \# \beta'_j$. So $(\mathcal{V}_{\mathsf{N}}(\mathsf{a})\ b) \cdot \beta'_j = \beta'_j$, and thus, by Condition (5) again, $\mathsf{bin}(\mathcal{V}_{\mathsf{N}}(\mathsf{c}), \langle \mathcal{V}_{\mathsf{N}}(\mathsf{a}) \rangle (x)) \in \beta'_j$ — so x is in $X_{j,l}$.

Other modes of action are treated in a similar manner, and thus (13) is established.

For every $j \in [1, \mathrm{ar}_{\mathsf{X}}(\underline{\mathsf{op}})]$, and every $l \in \mathit{Lab}(\mathtt{N})$, we have the following situation of sets and functions between them:

$$\begin{tikzcd}
\mathtt{X}_{j,l} \arrow[ddd, hook] \arrow[dr, dotted] \arrow[drr] & & \\
& X_{j,l} \arrow[r, two heads] \arrow[d, hook] & f(X_{j,l}) \arrow[dd, hook] \\
& X \arrow[dr, "f"] & \\
\mathtt{X} \arrow[rr, "\mathcal{V}_{\mathsf{X}}"'] & & Y
\end{tikzcd}$$

where the dotted arrow $\mathtt{X}_{j,l} \to X_{j,l}$ is the compsite

$$\mathtt{X}_{j,l} \xrightarrow{\mathcal{V}_{\mathsf{X}}} f(X_{j,l}) \xrightarrow{m_{j,l}} X_{j,l}$$

for a chosen section $m_{j,l} : f(X_{j,l}) \rightarrowtail X_{j,l}$ of the image $X_{j,l} \twoheadrightarrow f(X_{j,l})$.

We are now in a position to define a valuation $\mathcal{V}'$ into X. We let $\mathcal{V}'_{\mathsf{N}} = \mathcal{V}_{\mathsf{N}}$, and define $\mathcal{V}'_{\mathsf{X}}$ using the bijection (5):

- For $j \in [1, \mathrm{ar}_{\mathsf{X}}(\underline{\mathsf{op}})]$, we set $\mathcal{V}'_{\mathsf{X}}(\underline{\mathtt{x}}_j) = x_j$.
- For $(\underline{\mathtt{x}}_j, l, \mathtt{x}) \in \mathit{Prem}$, we set $\mathcal{V}'_{\mathsf{X}}(\mathtt{x}) = m_{j,l}(\mathcal{V}_{\mathsf{X}}(\mathtt{x}))$.

By construction,

$$s \;=\; \underline{\mathsf{op}} \left(\begin{array}{l} \big(\mathcal{V}'_{\mathsf{N}}(\underline{\mathsf{c}}_i)\big)_{i\in[1,\mathrm{ar}_{\mathsf{N}}(\underline{\mathsf{op}})]}\,, \\ \Big(\langle \mathcal{V}'_{\mathsf{N}}(\underline{\mathsf{a}}^j_k)\rangle_{k\in[1,\mathrm{bdep}_{\underline{\mathsf{op}}}(j)]}\,\big(\mathcal{V}'_{\mathsf{X}}(\underline{\mathtt{x}}_j), \beta'_j\big)\Big)_{j\in[1,\mathrm{ar}_{\mathsf{X}}(\underline{\mathsf{op}})]} \end{array} \right) \tag{15}$$

and $\mathcal{V}'(\mathit{Prem}[j]) \subseteq \beta'_j$ for all $j \in [1, \mathrm{ar}_{\mathsf{X}}(\underline{\mathsf{op}})]$. We have already ensured that $\mathcal{V}'_{\mathsf{N}}(\mathrm{bn}(\underline{\mathtt{l}}))\#s$, and so we can conclude that $\mathcal{V}'$ is adequate for s.

Moreover, by definition, we have that $(f \circ \mathcal{V}') = \mathcal{V}$, and so

$$\mathcal{V}(\underline{\mathtt{l}}, \underline{\mathtt{tar}}) = B_{\mathrm{g}}(U(T_{\Sigma,\mathbf{NSub}}f))(\mathcal{V}'(\underline{\mathtt{l}}, \underline{\mathtt{tar}})) \quad .$$

Thus (9) is proved, concluding the proof of Theorem 4.7. □

4.4 Congruence result

The class of abstract rules pointwise inherits a complete join semilattice structure from B_{g}. In particular, we have the following scenario. Let $\mathcal{R}$ be a transition system specification (Defn. 1.2) over some algebraic binding signature Σ (Defn. 1.1). If all the formal rule structures in $\mathcal{R}$ satisfy Conditions (1–8) of Figure 4, then $\mathcal{R}$ induces an abstract rule

$$[\![\mathcal{R}]\!] : \Sigma_{\mathbf{Nom}}((U(-)) \times B_{\mathrm{g}}(U(-))) \to B_{\mathrm{g}}(U(T_{\Sigma,\mathbf{NSub}}-)) \ \ : \mathbf{NSub} \to \mathbf{Nom}$$

by

$$[\![\mathcal{R}]\!]_X(s) = \bigcup_{\mathtt{R}\in\mathcal{R}} \Big([\![\mathtt{R}]\!]_X(s)\Big) \quad .$$

By Proposition 4.2, the induced U-structured B_{g}-coalgebra,

$$[\![\mathcal{R}]\!]^\sharp : T_{\Sigma,\mathbf{Nom}}\emptyset \to B_{\mathrm{g}}(T_{\Sigma,\mathbf{Nom}}\emptyset)$$

corresponds to a wide-open transition system.

Theorem 4.8. *For any transition system specification $\mathcal{R}$ in the format of Figure 4, the wide-open transition system corresponding to $[\![\mathcal{R}]\!]^\sharp$ is the initial transition system satisfying $\mathcal{R}$.*

Proof notes. The abstract rule induced by a transition system specification $\mathcal{R}$ gives rise to an operator $\Phi_{[\![\mathcal{R}]\!]}$ on the complete lattice

$$\mathbf{Nom}\Big(T_{\Sigma,\mathbf{Nom}}\emptyset,\ B_{\mathrm{g}}(T_{\Sigma,\mathbf{Nom}}\emptyset)\Big)$$

as described in Section 4.2 above.

Via Proposition 4.2, this lattice is isomorphic to the complete lattice of wide-open transition systems with fixed carrier, $T_{\Sigma,\mathbf{Nom}}\emptyset$, and with the corresponding fixed substitution structure, ordered by inclusion.

We claim that, on the lattice of wide-open transition systems, the pre-fixed points of this operator $\Phi_{[\![\mathcal{R}]\!]}$ are precisely those transition systems that satisfy $\mathcal{R}$. This is verified by tracing through the constructions in Section 4.3. Note that, for any rule $\mathtt{R}$ in $\mathcal{R}$, a valuation in the sense of Section 4.3 is nothing but a valuation of the universally quantified variables of the axiom corresponding to $\mathtt{R}$.

Next, we claim that the operator $\Phi_{[\![\mathcal{R}]\!]}$ is monotone. From the logical point of view, this is the case because we only consider positive rules.

Thus, by Tarski's fixed-point theorem, the operator $\Phi_{[\![\mathcal{R}]\!]}$ has a least pre-fixed point, which is a fixed point. By Proposition 4.3, this fixed-point is the coalgebra

$$[\![\mathcal{R}]\!]^\sharp : T_{\Sigma,\mathbf{Nom}}\emptyset \to B_{\mathrm{g}}(T_{\Sigma,\mathbf{Nom}}\emptyset) \quad .$$

From the model theoretic perspective, this least pre-fixed point is the least transition system satisfying $\mathcal{R}$ with carrier $T_{\Sigma,\mathbf{NSub}}\emptyset$, that is also a wide-open transition system. By Proposition 2.1, this fixed point is the initial transition system satisfying $\mathcal{R}$. □

By combining the results of this section, we arrive at our main result.

Corollary 4.9. *For the initial transition system satisfying a transition system specification in the format of Figure 4, wide-open bisimilarity is a congruence.* □

Appendices

A Proof of Proposition 4.1

Proposition 4.1.

(1) To give a B_g-coalgebra is to give a ground transition system.
(2) To give a B_g-bisimulation between two B_g-coalgebras is to give a model of the theory of ground bisimulation.

Proof. For item (1), we first define an intermediate notion. We say that a **pro-ground transition system** is a nominal set X together with an equivariant relation

$$\longrightarrow \subseteq X \times L_\mathrm{g}X \quad . \tag{A.1}$$

Pro-ground transition systems are in bijective correspondence with ground transition systems; we convert between the two notions using the following correspondence.

$$\begin{aligned}
x \xrightarrow{c?(a)} y &\iff x \longrightarrow \mathsf{bin}(c, \langle a\rangle y) \quad \text{and } a\#(c,y)\\
x \xrightarrow{c!d} y &\iff x \longrightarrow \mathsf{out}(c, d, y)\\
x \xrightarrow{c!(a)} y &\iff x \longrightarrow \mathsf{bout}(c, \langle a\rangle y) \ \text{ and } a\#(c,y)\\
x \xrightarrow{\tau} y &\iff x \longrightarrow \mathsf{tau}(y)
\end{aligned}$$

By definition of abstraction, it is reasonable to only consider fresh variables for binders. Conversely, since the pro-ground transition systems that we consider are equivariant, all fresh input/output names are treated uniformly. That is, if $x \xrightarrow{c?(a)} y$, and $a'\#(x,c)$, then we also have $x \xrightarrow{c?(a')} (a\,a')\cdot y$, and similarly for bound output transitions.

To conclude item (1), we remark that pro-ground transition systems are in bijective correspondence with B_g-coalgebras: this follows from the universal property of the powerobject construction $\mathcal{P}$ as a relation classifier. We convert between pro-ground transition systems and coalgebras $h : X \to B_\mathrm{g}X$ using the following correspondence.

$$x \longrightarrow b \quad \Longleftrightarrow \quad b \in h(x) \quad .$$

We now turn to item (2), and establish a correspondence between the two notions of bisimulation. The two step approach that we adopted for item (1) is also useful here. We define a notion of pro-ground (bi)simulation for pro-ground transition systems. Consider a pro-ground transition system $(X, \longrightarrow)$. A **pro-ground simulation** on this system is an equivariant relation $R \subseteq X \times X$ such that

$$\begin{array}{l} \text{if } xRx' \text{ and } x \longrightarrow b \\ \text{then there is } b' \in L_{\mathrm{g}}X \text{ such that } x' \longrightarrow b' \text{ and } b\,(L_{\mathrm{g}}R)\,b'. \end{array}$$

Here the object $L_{\mathrm{g}}R$ is considered as a relation over $L_{\mathrm{g}}X$; we have

$$\begin{aligned} L_{\mathrm{g}}R \;\cong\; & \{(\mathsf{bin}(c, \langle a\rangle x),\, \mathsf{bin}(c, \langle a\rangle x')) \mid a, c \in \mathcal{N},\; xRx'\} \\ \cup\; & \{(\mathsf{out}(c, d, x),\, \mathsf{out}(c, d, x')) \mid c, d \in \mathcal{N},\; xRx'\} \\ \cup\; & \{(\mathsf{bout}(c, \langle a\rangle x),\, \mathsf{bout}(c, \langle a\rangle x')) \mid a, c \in \mathcal{N},\; xRx'\} \\ \cup\; & \{(\mathsf{tau}(x),\, \mathsf{tau}(x')) \mid xRx'\} \quad . \end{aligned}$$

A notion of **pro-ground bisimulation** is defined accordingly.

Now, a pro-ground simulation is the same thing as a ground simulation on the corresponding ground transition system.

Suppose that R is a pro-ground simulation, and we will show that it is a ground simulation. Suppose that xRx', and consider the case where $x \xrightarrow{c?(a)} y$ in the corresponding ground transition system, with $a\#x'$. Note that $a \neq c$, since we have a ground transition system. Converting, we have $x \longrightarrow \mathsf{bin}(c, \langle a\rangle y)$ in the pro-ground transition system. Since R is a pro-ground simulation, we have $a' \in \mathcal{N}$ and $y' \in X$ such that $x' \longrightarrow \mathsf{bin}(c, \langle a'\rangle y')$ and $\langle a\rangle y\,([\mathcal{N}]R)\,\langle a'\rangle y'$. By definition of abstraction, we can assume that $a'\#y$, so that $(a\ a') \cdot y\,R\,y'$, and we can also assume that $a' \neq c$. Moreover, by assumption, $a\#x'$, and, as above, we have $a \neq c$. Thus, by the equivariance of the pro-ground relation $\longrightarrow \subseteq X \times L_{\mathrm{g}}X$, we have $x' \longrightarrow \mathsf{bin}\,(c, \langle a\rangle((a\,a') \cdot y'))$. Converting back, we have $x' \xrightarrow{c?(a)} (a\,a')y'$, and, by equivariance of R, we have $y\,R\,(a\,a') \cdot y'$. Following a similar argument for the other modes of communication, we conclude that R is indeed a ground simulation.

Conversely, suppose that R is a ground simulation. We will show that it is a pro-ground simulation. Suppose that xRx', and consider the case where $x \longrightarrow b$ in the corresponding pro-ground transition system, with b of input type. So we have $a, c \in \mathcal{N}$ and $y \in X$ such that $b = \mathsf{bin}(c, \langle a\rangle y)$. By definition of abstraction, we can assume that $a\#(x, x', c)$. Thus we have a ground transition $x \xrightarrow{c?(a)} y$, and, since R is a ground simulation, we have $y' \in Y$ such that yRy' and $x' \xrightarrow{c?(a)} y'$. Converting back, we deduce a pro-ground transition $x' \longrightarrow \mathsf{bin}(c, \langle a \rangle y')$. Following a similar argument for the other modes of communication, we conclude that R is indeed a pro-ground simulation.

By a symmetric argument, one shows that a ground bisimulation on a ground transition system is the same thing as a pro-ground bisimulation on the corresponding pro-ground transition system. To conclude item (2), we show that pro-ground bisimulations on pro-ground transition systems correspond to coalgebraic B_{g}-bisimulations on the corresponding B_{g}-coalgebras.

To this end, we consider a coalgebra $h : X \to B_{\mathrm{g}}X$, and the corresponding pro-ground transition system, $\longrightarrow \subseteq X \times L_{\mathrm{g}}X$.

Suppose that $R \subseteq X \times X$ is a B_{g}-bisimulation, so that there is a coalgebra structure over R making Diagram (6) commute,

$$\begin{tikzcd}
R \arrow[rr] \arrow[d] & & X \times X \arrow[d, "h\times h"] \\
B_{\mathrm{g}}R \arrow[r] & B_{\mathrm{g}}(X \times X) \arrow[r] & B_{\mathrm{g}}X \times B_{\mathrm{g}}X\,.
\end{tikzcd}$$

We will show that R is also a pro-ground bisimulation. Indeed, suppose that xRx', and that $x \longrightarrow b$. The coalgebra structure for R defines a subset $\beta_{x,x'} \subseteq L_{\mathrm{g}}R$ for which $B_{\mathrm{g}}\pi_1(\beta_{x,x'}) = h(x)$ and $B_{\mathrm{g}}\pi_2(\beta_{x,x'}) = h(x')$. We know that $b \in h(x)$, so we must have $b' \in \beta_{x,x'}$ such that $b = \pi_1(b')$. By definition, $\pi_2(b') \in h(x')$, so, converting, we have $x' \longrightarrow \pi_2(b')$, and certainly $b\,(L_{\mathrm{g}}R)\,\pi_2(b')$. Thus R is a pro-ground simulation; that it is a pro-ground bisimulation is established by a symmetric argument.

On the other hand, suppose that $R \subseteq X \times X$ is a pro-ground bisimulation, and we will show that it is also a B_{g}-bisimulation by exhibiting a coalgebra structure for R making Diagram (6) commute. One such appropriate structure maps a pair $(x, x') \in R$ to the set $\{b \in L_{\mathrm{g}}R \mid L_{\mathrm{g}}\pi_1(b) \in h(x), L_{\mathrm{g}}\pi_2(b) \in h(x')\}$ in $B_{\mathrm{g}}R$. We claim that

$$(R \to B_{\mathrm{g}}R \to (B_{\mathrm{g}}X \times B_{\mathrm{g}}X)) \;=\; (R \to (X \times X) \xrightarrow{h\times h} (B_{\mathrm{g}}X \times B_{\mathrm{g}}X)) \quad .$$

Under the componentwise, parameterwise order, the left-hand-side is trivially included in the right-hand-side. The inclusion of the right-hand-side in the left-hand-side follows from the definition of pro-ground bisimulation.

Thus we can conclude item (2): ground bisimulation is the same thing as coalgebraic B_{g}-bisimulation. $\square$

B Proof of Lemma 4.6

Lemma 4.6 was stated in Section 4.3 in the context of an algebraic binding signature Σ (see Defn. 1.2) and a formal rule structure over it (Defn. 1.2). It is assumed that the rule structure has variables from $\mathtt{N}$ and $\mathtt{X}$, and has premise set *Prem* and conclusion with source

$$\underline{\mathtt{src}} \;=\; \underline{\mathsf{op}}\left((\underline{\mathtt{c}}_i)_{i\in[1,\mathrm{ar}_{\mathsf{N}}(\underline{\mathsf{op}})]},\; \left(\langle \underline{\mathtt{a}}^j_k \rangle_{k\in[1,\mathrm{bdep}_{\underline{\mathsf{op}}}(j)]} \underline{\mathtt{x}}_j\right)_{j\in[1,\mathrm{ar}_{\mathsf{X}}(\underline{\mathsf{op}})]} \right) \;,$$

label $\underline{\mathtt{l}}$, and target $\underline{\mathtt{tar}}$. It is also assumed that the rule structure satisfies all the conditions of Figure 4.

Lemma 4.6. *Consider a finite set $N \subseteq_{\mathrm{f}} \mathcal{N}$ of names, and a nominal substitution X. For every adequate valuation for $s \in \Sigma_{\mathbf{Nom}}((UX)\times B_{\mathrm{g}}(UX))$, there is another adequate valuation for s with the same archetypal result, but that maps the binding variables of the rule,* $\mathtt{BN}(\underline{\mathtt{src}}, \mathit{Prem}) \cup \mathrm{bn}(\underline{\mathtt{l}})$*, outside of N.*

Proof. Pick an injection

$$\xi :\; (\mathtt{BN}(\underline{\mathtt{src}}, \mathit{Prem}) \cup \mathrm{bn}(\underline{\mathtt{l}})) \;\rightarrowtail\; \left(\mathcal{N} \setminus \left(N \cup \mathsf{im}(\mathcal{V}_{\mathsf{N}}) \cup \mathsf{supp}(s) \cup \bigcup_{\mathtt{x}\in\mathtt{X}} \mathsf{supp}(\mathcal{V}_{\mathsf{X}}(\mathtt{x}))\right)\right) \quad .$$

This is possible because the domain is finite, while the codomain is infinite.

We define a new valuation $\mathcal{V}'$ into X with

$$\begin{aligned}
\mathcal{V}'_{\mathsf{N}}(\mathtt{a}) &= \begin{cases} \xi(\mathtt{a}) & \text{if } \mathtt{a} \in (\mathtt{BN}(\underline{\mathtt{src}}, \mathit{Prem}) \cup \mathrm{bn}(\underline{\mathtt{l}})) \\ \mathcal{V}_{\mathsf{N}}(\mathtt{a}) & \text{otherwise} \end{cases} \\
\mathcal{V}'_{\mathsf{X}}(\mathtt{x}) &= (\mathcal{V}_{\mathsf{N}}(\mathtt{a})\ \xi(\mathtt{a}))_{\mathtt{a}\in(\mathtt{BN}(\mathtt{x})\cup \mathrm{bn}(\underline{\mathtt{l}}))} \cdot \mathcal{V}_{\mathsf{X}}(\mathtt{x})
\end{aligned}$$

Note that the order of swaps in the definition of $\mathcal{V}'_{\mathsf{X}}$ does not matter because both $\mathcal{V}$ and ξ are injective, and their images are disjoint on the set $(\mathtt{BN}(\underline{\mathtt{src}}, \mathit{Prem}) \cup \mathrm{bn}(\underline{\mathtt{l}}))$.

We will now show that $\mathcal{V}'$ is adequate for s (Sec. B.1) and that $\mathcal{V}$ and $\mathcal{V}'$ have the same archetypal results (Sec. B.2).

We will explain why $\mathcal{V}'$ is an adequate instantiation into s. By assumption, $\mathcal{V}$ is adequate for s. So $\mathcal{V}$ is injective, $\mathcal{V}_{\mathsf{N}}(\mathrm{bn}(\underline{\mathtt{l}}))\#s$, and

$$s \;=\; \underline{\mathsf{op}}\left(\begin{array}{l}\big(\mathcal{V}_{\mathsf{N}}(\underline{\mathsf{c}}_i)\big)_{i\in[1,\mathrm{ar}_{\mathsf{N}}(\underline{\mathsf{op}})]}\,,\\ \Big(\langle\mathcal{V}_{\mathsf{N}}(\underline{\mathsf{a}}^j_k)\rangle_{k\in[1,\mathrm{bdep}_{\underline{\mathsf{op}}}(j)]}\,\big(\mathcal{V}_{\mathsf{X}}(\underline{\mathtt{x}}_j),\beta_j\big)\Big)_{j\in[1,\mathrm{ar}_{\mathsf{X}}(\underline{\mathsf{op}})]}\end{array}\right)$$

where $\mathcal{V}(\mathit{Prem}[j])\subseteq\beta_j$ for all $j\in[1,\mathrm{ar}_{\mathsf{X}}(\underline{\mathsf{op}})]$.

We know that $\mathcal{V}'_{\mathsf{N}}$ is injective, because $\mathcal{V}$ and ξ are, and because $\mathcal{V}_{\mathsf{N}}(\mathtt{BN}(\underline{\mathtt{src}},\mathit{Prem})\cup\mathrm{bn}(\underline{\mathtt{l}}))$ is disjoint from $\mathsf{im}(\xi)$. Moreover, $\mathcal{V}'_{\mathsf{N}}(\mathrm{bn}(\underline{\mathtt{l}}))\#s$ by definition of ξ.

We let

$$\beta'_j \;=\; (\mathcal{V}_{\mathsf{N}}(\mathsf{a})\ \mathcal{V}'_{\mathsf{N}}(\mathsf{a}))_{\mathsf{a}\in(\mathtt{BN}(\underline{\mathtt{x}}_j)\cup\mathrm{bn}(\underline{\mathtt{l}}))}\cdot\beta_j$$

for each $j\in[1,\mathrm{ar}_{\mathsf{X}}(\underline{\mathsf{op}})]$, and we will show that

$$s \;=\; \underline{\mathsf{op}}\left(\begin{array}{l}\big(\mathcal{V}'_{\mathsf{N}}(\underline{\mathsf{c}}_i)\big)_{i\in[1,\mathrm{ar}_{\mathsf{N}}(\underline{\mathsf{op}})]}\,,\\ \Big(\langle\mathcal{V}'_{\mathsf{N}}(\underline{\mathsf{a}}^j_k)\rangle_{k\in[1,\mathrm{bdep}_{\underline{\mathsf{op}}}(j)]}\,\big(\mathcal{V}'_{\mathsf{X}}(\underline{\mathtt{x}}_j),\beta'_j\big)\Big)_{j\in[1,\mathrm{ar}_{\mathsf{X}}(\underline{\mathsf{op}})]}\end{array}\right)$$

and $\mathcal{V}'(\mathit{Prem}[j])\subseteq\beta'_j$ for each $j\in[1,\mathrm{ar}_{\mathsf{X}}(\underline{\mathsf{op}})]$.

By Conditions (4) and (5), $\underline{\mathsf{c}}_i\notin\mathtt{BN}(\underline{\mathtt{src}},\mathit{Prem})$, for all $i\in[1,\mathrm{ar}_{\mathsf{N}}(\underline{\mathsf{op}})]$. Moreover, since $\mathcal{V}$ is adequate for s, we know that $\mathcal{V}_{\mathsf{N}}(\underline{\mathsf{c}}_i)\in\mathsf{supp}(s)$, and that $\mathcal{V}_{\mathsf{N}}(\mathrm{bn}(\underline{\mathtt{l}}))\notin\mathsf{supp}(s)$, so we know that $\underline{\mathsf{c}}_i\notin\mathrm{bn}(\underline{\mathtt{l}})$. Thus we have

$$\underline{\mathsf{c}}_i\notin\Big(\mathtt{BN}(\underline{\mathtt{src}},\mathit{Prem})\cup\mathrm{bn}(\underline{\mathtt{l}})\Big)$$

so that $\mathcal{V}_{\mathsf{N}}(\underline{\mathsf{c}}_i)=\mathcal{V}'_{\mathsf{N}}(\underline{\mathsf{c}}_i)$.

We will now show that for each $j\in[1,\mathrm{ar}_{\mathsf{X}}(\underline{\mathsf{op}})]$, we have

$$\langle\mathcal{V}_{\mathsf{N}}(\underline{\mathsf{a}}^j_k)\rangle_{k\in[1,\mathrm{bdep}_{\underline{\mathsf{op}}}(j)]}\Big(\mathcal{V}_{\mathsf{X}}(\underline{\mathtt{x}}_j),\beta_j\Big) \;=\; \langle\mathcal{V}'_{\mathsf{N}}(\underline{\mathsf{a}}^j_k)\rangle_{k\in[1,\mathrm{bdep}_{\underline{\mathsf{op}}}(j)]}\Big(\mathcal{V}'_{\mathsf{X}}(\underline{\mathtt{x}}_j),\beta'_j\Big)\quad. \tag{B.1}$$

We first make the following observation. Consider a nominal set Y, and $x\in Y$, and a natural number n. Consider distinct names $a_1,b_1,\ldots,a_n,b_n\in\mathcal{N}$ such that for all $k\in[1,n]$ we have $b_k\#x$. Then, by definition of abstraction, we have

$$\langle a_k\rangle_{k\in[1,n]}x \;=\; \langle b_k\rangle_{k\in[1,n]}\Big((a_k\ b_k)_{k\in[1,n]}\cdot x\Big)\quad.$$

Equation (B.1) above is a particular case of this result, because of the restricted codomain of ξ, and because of Condition (3).

To conclude that $\mathcal{V}'$ is adequate for s, we explain why $\mathcal{V}'(\mathit{Prem}[j]) \subseteq \beta'_j$, for each $j \in [1, \mathrm{ar}_{\mathsf{X}}(\underline{\mathtt{op}})]$. Suppose, for instance, that there is a premise $\underline{\mathtt{x}}_j \xrightarrow{\mathtt{c?(a)}} \mathtt{y}$ in *Prem*. We will show that $\mathsf{bin}(\mathcal{V}'_{\mathsf{N}}(\mathtt{c}), \langle\mathcal{V}'_{\mathsf{N}}(\mathtt{a})\rangle\mathcal{V}'_{\mathsf{X}}(\mathtt{y})) \in \beta'_j$.

Since $\mathcal{V}$ is adequate, we know that $\mathsf{bin}(\mathcal{V}_{\mathsf{N}}(\mathtt{c}), \langle\mathcal{V}_{\mathsf{N}}(\mathtt{a})\rangle\mathcal{V}_{\mathsf{X}}(\mathtt{y})) \in \beta_j$. By definition,

$$(\mathcal{V}_{\mathsf{N}}(\mathtt{b})\ \mathcal{V}'_{\mathsf{N}}(\mathtt{b}))_{\mathtt{b}\in(\mathtt{BN}(\underline{\mathtt{x}}_j)\cup\mathrm{bn}(\underline{\mathtt{l}}))} \cdot \mathsf{bin}(\mathcal{V}_{\mathsf{N}}(\mathtt{c}), \langle\mathcal{V}_{\mathsf{N}}(\mathtt{a})\rangle\mathcal{V}_{\mathsf{X}}(\mathtt{y})) \in \beta'_j$$

so it suffices for us to prove that

$$\begin{aligned}(\mathcal{V}_{\mathsf{N}}(\mathtt{b})\ \mathcal{V}'_{\mathsf{N}}(\mathtt{b}))_{\mathtt{b}\in(\mathtt{BN}(\underline{\mathtt{x}}_j)\cup\mathrm{bn}(\underline{\mathtt{l}}))} &\cdot \mathsf{bin}(\mathcal{V}_{\mathsf{N}}(\mathtt{c}), \langle\mathcal{V}_{\mathsf{N}}(\mathtt{a})\rangle\mathcal{V}_{\mathsf{X}}(\mathtt{y})) \\ &= \mathsf{bin}(\mathcal{V}'_{\mathsf{N}}(\mathtt{c}), \langle\mathcal{V}'_{\mathsf{N}}(\mathtt{a})\rangle\mathcal{V}'_{\mathsf{X}}(\mathtt{y})) \quad . \end{aligned} \tag{B.2}$$

Independently of whether or not $\mathtt{c} \in \left(\mathtt{BN}(\underline{\mathtt{x}}_j) \cup \mathrm{bn}(\underline{\mathtt{l}})\right)$,

$$(\mathcal{V}_{\mathsf{N}}(\mathtt{b})\ \mathcal{V}'_{\mathsf{N}}(\mathtt{b}))_{\mathtt{b}\in(\mathtt{BN}(\underline{\mathtt{x}}_j)\cup\mathrm{bn}(\underline{\mathtt{l}}))} \cdot (\mathcal{V}_{\mathsf{N}}(\mathtt{c})) \ = \ \mathcal{V}'_{\mathsf{N}}(\mathtt{c}) \quad . \tag{B.3}$$

Indeed, for the case where $\mathtt{c} \in (\mathtt{BN}(\underline{\mathtt{x}}_j) \cup \mathrm{bn}(\underline{\mathtt{l}}))$, then this follows from the definition of $\mathcal{V}'_{\mathsf{N}}$. If, on the other hand, $\mathtt{c} \notin (\mathtt{BN}(\underline{\mathtt{x}}_j) \cup \mathrm{bn}(\underline{\mathtt{l}}))$, then $\mathtt{c} \in \mathtt{FN}(\underline{\mathtt{src}}, \mathit{Prem})$ by definition, and so, by Condition (4), $\mathtt{c}$ is not bound anywhere, hence $\mathcal{V}_{\mathsf{N}}(\mathtt{c}) = \mathcal{V}'_{\mathsf{N}}(\mathtt{c})$.

We will now show that

$$(\mathcal{V}_{\mathsf{N}}(\mathtt{b})\ \mathcal{V}'_{\mathsf{N}}(\mathtt{b}))_{\mathtt{b}\in(\mathtt{BN}(\underline{\mathtt{x}}_j)\cup\mathrm{bn}(\underline{\mathtt{l}}))} \cdot (\langle\mathcal{V}_{\mathsf{N}}(\mathtt{a})\rangle\mathcal{V}_{\mathsf{X}}(\mathtt{y})) = \langle\mathcal{V}'_{\mathsf{N}}(\mathtt{a})\rangle\mathcal{V}'_{\mathsf{X}}(\mathtt{y}) \quad . \tag{B.4}$$

If $\mathtt{a} \in \mathrm{bn}(\underline{\mathtt{l}})$, then (B.4) follows by definition of $\mathcal{V}'_{\mathsf{X}}(\mathtt{y})$. Otherwise, if $\mathtt{a} \notin \mathrm{bn}(\underline{\mathtt{l}})$, then we proceed as follows. The codomain of ξ ensures that $\mathcal{V}'_{\mathsf{N}}(\mathtt{a})\#\mathcal{V}_{\mathsf{X}}(\mathtt{y})$, so

$$\langle\mathcal{V}_{\mathsf{N}}(\mathtt{a})\rangle\mathcal{V}_{\mathsf{X}}(\mathtt{y}) \ = \ \langle\mathcal{V}'_{\mathsf{N}}(\mathtt{a})\rangle((\mathcal{V}_{\mathsf{N}}(\mathtt{a})\ \mathcal{V}'_{\mathsf{N}}(\mathtt{a})) \cdot \mathcal{V}_{\mathsf{X}}(\mathtt{y})) \quad .$$

By Condition (5), $\mathtt{a} \notin \mathtt{FN}(\underline{\mathtt{x}}_j)$, and so we have (B.4):

$$\begin{aligned}&(\mathcal{V}_{\mathsf{N}}(\mathtt{b})\ \mathcal{V}'_{\mathsf{N}}(\mathtt{b}))_{\mathtt{b}\in(\mathtt{BN}(\underline{\mathtt{x}}_j)\cup\mathrm{bn}(\underline{\mathtt{l}}))} \cdot (\langle\mathcal{V}_{\mathsf{N}}(\mathtt{a})\rangle\mathcal{V}_{\mathsf{X}}(\mathtt{y})) \\ &\qquad = (\mathcal{V}_{\mathsf{N}}(\mathtt{b})\ \mathcal{V}'_{\mathsf{N}}(\mathtt{b}))_{\mathtt{b}\in(\mathtt{BN}(\underline{\mathtt{x}}_j)\cup\mathrm{bn}(\underline{\mathtt{l}}))} \cdot \langle\mathcal{V}'_{\mathsf{N}}(\mathtt{a})\rangle((\mathcal{V}_{\mathsf{N}}(\mathtt{a})\ \mathcal{V}'_{\mathsf{N}}(\mathtt{a})) \cdot \mathcal{V}_{\mathsf{X}}(\mathtt{y})) \\ &\qquad = \langle\mathcal{V}'_{\mathsf{N}}(\mathtt{a})\rangle\left((\mathcal{V}_{\mathsf{N}}(\mathtt{b})\ \mathcal{V}'_{\mathsf{N}}(\mathtt{b}))_{\mathtt{b}\in(\mathtt{BN}(\underline{\mathtt{x}}_j)\cup\mathrm{bn}(\underline{\mathtt{l}}))} \cdot (\mathcal{V}_{\mathsf{N}}(\mathtt{a})\ \mathcal{V}'_{\mathsf{N}}(\mathtt{a})) \cdot \mathcal{V}_{\mathsf{X}}(\mathtt{y})\right) \\ &\qquad = \langle\mathcal{V}'_{\mathsf{N}}(\mathtt{a})\rangle((\mathcal{V}_{\mathsf{N}}(\mathtt{b})\ \mathcal{V}'_{\mathsf{N}}(\mathtt{b}))_{\mathtt{b}\in(\mathtt{BN}(\mathtt{y})\cup\mathrm{bn}(\underline{\mathtt{l}}))} \cdot \mathcal{V}_{\mathsf{X}}(\mathtt{y})) \\ &\qquad = \langle\mathcal{V}'_{\mathsf{N}}(\mathtt{a})\rangle\mathcal{V}'_{\mathsf{X}}(\mathtt{y}) \quad .\end{aligned}$$

Putting (B.3) and (B.4) together, we deduce (B.2), and thus $\mathcal{V}'$ is adequate for s.

We now explain why $\mathcal{V}'(\underline{\mathtt{l}}, \underline{\mathtt{tar}}) = \mathcal{V}(\underline{\mathtt{l}}, \underline{\mathtt{tar}})$. First, we prove the following results for every $\mathtt{t} \in T_{\Sigma+\mathsf{sub}}(\mathtt{N}, \mathtt{X})$.

(B.5) The fresh bound names are disjoint from the original valuation for $\mathtt{t}$, i.e. $\mathcal{V}'_{\mathsf{N}}(\mathtt{BN}(\underline{\mathtt{src}}, \mathit{Prem}) \cup \mathrm{bn}(\underline{\mathtt{l}})) \cap \mathsf{supp}(\mathcal{V}(\mathtt{t})) = \emptyset$.

(B.6) Suppose that $\mathcal{WF}(\mathtt{t})$. Consider a set $\mathtt{C} \subseteq \mathtt{N}$, such that

$$(\mathtt{BN}(\underline{\mathtt{src}}, \mathit{Prem}) \cap \mathtt{FN}(\mathtt{t})) \cup \mathrm{bn}(\underline{\mathtt{l}}) \subseteq \mathtt{C}$$

and suppose that for all $\mathtt{x} \in \mathtt{X}$ appearing in $\mathtt{t}$ we have

$$(\mathcal{V}_{\mathsf{N}}(\mathsf{a})\ \mathcal{V}'_{\mathsf{N}}(\mathsf{a}))_{\mathsf{a}\in(\mathtt{BN}(\mathtt{x})\cup\mathtt{C})} \cdot \mathcal{V}_{\mathsf{X}}(\mathtt{x}) = \mathcal{V}'_{\mathsf{X}}(\mathtt{x}) \quad .$$

Then $(\mathcal{V}_{\mathsf{N}}(\mathsf{a})\ \mathcal{V}'_{\mathsf{N}}(\mathsf{a}))_{\mathsf{a}\in\mathtt{C}} \cdot \mathcal{V}(\mathtt{t}) = \mathcal{V}'(\mathtt{t})$.

The base cases, where $\mathtt{t}$ is a term variable, are trivial. For the inductive step, we consider an operator op in $(\Sigma + \mathsf{sub})$, together with: name variables $\mathsf{c}_i \in \mathtt{N}$, for $i \in [1, \mathrm{ar}_{\mathsf{N}}(\mathsf{op})]$; name variables $\mathsf{a}^j_k \in \mathtt{N}$, for $j \in [1, \mathrm{ar}_{\mathsf{X}}(\mathsf{op})]$, $k \in [1, \mathrm{bdep}_{\mathsf{op}}(j)]$; and terms $\mathtt{t}_j \in T_{\Sigma+\mathsf{sub}}(\mathtt{N}, \mathtt{X})$, for $j \in [1, \mathrm{ar}_{\mathsf{X}}(\mathsf{op})]$. We let

$$\mathtt{t} \;=\; \mathsf{op}\left((\mathsf{c}_i)_{i\in[1,\mathrm{ar}_{\mathsf{N}}(\underline{\mathsf{op}})]}\,,\, \left(\langle \mathsf{a}^j_k\rangle_{k\in[1,\mathrm{bdep}_{\underline{\mathsf{op}}}(j)]}\mathtt{t}_j\right)_{j\in[1,\mathrm{ar}_{\mathsf{X}}(\underline{\mathsf{op}})]}\right)$$

and we will explain why properties (B.5) and (B.6) hold of $\mathtt{t}$. For property (B.5), notice that, because of the codomain of ξ, we have

$$\mathcal{V}'_{\mathsf{N}}(\mathtt{BN}(\underline{\mathtt{src}}, \mathit{Prem}) \cup \mathrm{bn}(\underline{\mathtt{l}})) \cap \{\mathcal{V}_{\mathsf{N}}(\mathsf{c}_i) \mid i \in [1, \mathrm{ar}_{\mathsf{N}}(\mathsf{op})]\} \;=\; \emptyset \quad .$$

The induction hypotheses ensure that

$$\mathcal{V}'_{\mathsf{N}}(\mathtt{BN}(\underline{\mathtt{src}}, \mathit{Prem}) \cup \mathrm{bn}(\underline{\mathtt{l}})) \cap \mathsf{supp}(\mathcal{V}(\mathtt{t}_j)) = \emptyset$$

for each $j \in [1, \mathrm{ar}_{\mathsf{X}}(\mathsf{op})]$, and property (B.5) follows.

For the inductive step of property (B.6), we further assume that $\mathcal{WF}(\mathtt{t})$, and consider $\mathtt{C} \subseteq \mathtt{N}$ which is such that

$$(\mathtt{BN}(\underline{\mathtt{src}}, \mathit{Prem}) \cap \mathtt{FN}(\mathtt{t})) \cup \mathrm{bn}(\underline{\mathtt{l}}) \;\subseteq\; \mathtt{C}$$

and

$$(\mathcal{V}_{\mathsf{N}}(\mathsf{a})\ \mathcal{V}'_{\mathsf{N}}(\mathsf{a}))_{\mathsf{a}\in(\mathtt{BN}(\mathtt{x})\cup\mathtt{C})} \cdot \mathcal{V}_{\mathsf{X}}(\mathtt{x}) \;=\; \mathcal{V}'_{\mathsf{X}}(\mathtt{x})$$

for all $\mathtt{x}$ appearing in $\mathtt{t}$. We must show that $(\mathcal{V}_{\mathsf{N}}(\mathsf{a})\ \mathcal{V}'_{\mathsf{N}}(\mathsf{a}))_{\mathsf{a}\in\mathtt{C}} \cdot \mathcal{V}(\mathtt{t}) = \mathcal{V}'(\mathtt{t})$, i.e. that

- for all $i \in [1, \mathrm{ar}_{\mathsf{N}}(\mathsf{op})]$, $(\mathcal{V}_{\mathsf{N}}(\mathsf{a})\ \mathcal{V}'_{\mathsf{N}}(\mathsf{a}))_{\mathsf{a}\in\mathtt{C}}(\mathcal{V}_{\mathsf{N}}(\mathsf{c}_i)) = \mathcal{V}'_{\mathsf{N}}(\mathsf{c}_i)$; and

- for each $j \in [1, \mathrm{ar}_{\mathsf{X}}(\mathtt{op})]$,

$$(\mathcal{V}_{\mathsf{N}}(\mathtt{a})\ \mathcal{V}'_{\mathsf{N}}(\mathtt{a}))_{\mathtt{a}\in\mathtt{C}}\cdot(\langle\mathcal{V}_{\mathsf{N}}(\mathtt{a}^j_k)\rangle_{k\in[1,\mathrm{bdep}_{\mathtt{op}}(j)]}\mathcal{V}(\mathtt{t}_j)) \;=\; \langle\mathcal{V}'_{\mathsf{N}}(\mathtt{a}^j_k)\rangle_{k\in[1,\mathrm{bdep}_{\mathtt{op}}(j)]}\mathcal{V}'(\mathtt{t}_j)) \quad .$$

For any $i \in [1, \mathrm{ar}_{\mathsf{N}}(\mathtt{op})]$ we must show that $(\mathcal{V}_{\mathsf{N}}(\mathtt{a})\ \mathcal{V}'_{\mathsf{N}}(\mathtt{a}))_{\mathtt{a}\in\mathtt{C}}(\mathcal{V}_{\mathsf{N}}(\mathtt{c}_i)) = \mathcal{V}'_{\mathsf{N}}(\mathtt{c}_i)$. If $\mathtt{c}_i \in \mathtt{C}$ then this is trivial. Otherwise, if $\mathtt{c}_i \notin \mathtt{C}$, then $\mathtt{c}_i \in \mathtt{FN}(\underline{\mathtt{tar}})$ by definition, and yet $(\mathtt{BN}(\underline{\mathtt{src}}, \mathit{Prem}) \cap \mathtt{FN}(\underline{\mathtt{tar}})) \subseteq \mathtt{C}$, so we know that $\mathtt{c}_i \notin \mathtt{BN}(\underline{\mathtt{src}}, \mathit{Prem})$. Since $\mathrm{bn}(\underline{\mathtt{l}}) \subseteq \mathtt{C}$, we also know that $\mathtt{c}_i \notin \mathrm{bn}(\underline{\mathtt{l}})$. Thus, by definition of $\mathcal{V}'_{\mathsf{N}}$, we have $\mathcal{V}_{\mathsf{N}}(\mathtt{c}_i) = \mathcal{V}'_{\mathsf{N}}(\mathtt{c}_i)$, and certainly $(\mathcal{V}_{\mathsf{N}}(\mathtt{a})\ \mathcal{V}'_{\mathsf{N}}(\mathtt{a}))_{\mathtt{a}\in\mathtt{C}}(\mathcal{V}_{\mathsf{N}}(\mathtt{c}_i)) = \mathcal{V}'_{\mathsf{N}}(\mathtt{c}_i)$.

For $j \in [1, \mathrm{ar}_{\mathsf{X}}(\mathtt{op})]$, we must show that

$$(\mathcal{V}_{\mathsf{N}}(\mathtt{a})\ \mathcal{V}'_{\mathsf{N}}(\mathtt{a}))_{\mathtt{a}\in\mathtt{C}}\cdot(\langle\mathcal{V}_{\mathsf{N}}(\mathtt{a}^j_k)\rangle_{k\in[1,\mathrm{bdep}_{\mathtt{op}}(j)]}\mathcal{V}(\mathtt{t}_j)) \;=\; \langle\mathcal{V}'_{\mathsf{N}}(\mathtt{a}^j_k)\rangle_{k\in[1,\mathrm{bdep}_{\mathtt{op}}(j)]}\mathcal{V}'(\mathtt{t}_j) \quad .$$

As a first step in this direction, we let

$$\mathtt{C}_j \;=\; \mathtt{C} \cup \Big(\mathtt{BN}(\underline{\mathtt{src}}, \mathit{Prem}) \cap \Big\{\mathtt{a}^j_k \;\Big|\; k \in [1, \mathrm{bdep}_{\mathtt{op}}(j)]\Big\}\Big)$$

and we assert that

$$\begin{aligned}(\mathcal{V}_{\mathsf{N}}(\mathtt{a})\ \mathcal{V}'_{\mathsf{N}}(\mathtt{a}))_{\mathtt{a}\in\mathtt{C}} &\cdot (\langle\mathcal{V}_{\mathsf{N}}(\mathtt{a}^j_k)\rangle_{k\in[1,\mathrm{bdep}_{\mathtt{op}}(j)]}\mathcal{V}(\mathtt{t}_j)) \\ &= \langle\mathcal{V}'_{\mathsf{N}}(\mathtt{a}^j_k)\rangle_{k\in[1,\mathrm{bdep}_{\mathtt{op}}(j)]}((\mathcal{V}_{\mathsf{N}}(\mathtt{a})\ \mathcal{V}'_{\mathsf{N}}(\mathtt{a}))_{\mathtt{a}\in\mathtt{C}_j} \cdot \mathcal{V}(\mathtt{t}_j)) \quad .\end{aligned}$$

This step is established using property (B.5) of $\mathtt{t}_j$, and the definition of abstraction.

Finally, we conclude this inductive step by explaining that

$$(\mathcal{V}_{\mathsf{N}}(\mathtt{a})\ \mathcal{V}'_{\mathsf{N}}(\mathtt{a}))_{\mathtt{a}\in\mathtt{C}_j} \cdot \mathcal{V}(\mathtt{t}_j)) \;=\; \mathcal{V}'(\mathtt{t}_j) \quad .$$

We explain this using the induction hypothesis. We have assumed that $\mathcal{WF}(\mathtt{t})$, and it follows that $\mathcal{WF}(\mathtt{t}_j)$. It is clear that

$$(\mathtt{BN}(\underline{\mathtt{src}}, \mathit{Prem}) \cap \mathtt{FN}(\mathtt{t}_j)) \cup \mathrm{bn}(\underline{\mathtt{l}}) \subseteq \mathtt{C}_j \quad .$$

So it remains for us to consider $\mathtt{x} \in \mathtt{X}$ appearing in $\mathtt{t}_j$, and to show that

$$(\mathcal{V}_{\mathsf{N}}(\mathtt{a})\ \mathcal{V}'_{\mathsf{N}}(\mathtt{a}))_{\mathtt{a}\in(\mathtt{BN}(\mathtt{x})\cup\mathtt{C}_j)} \cdot \mathcal{V}_{\mathsf{X}}(\mathtt{x}) \;=\; \mathcal{V}'_{\mathsf{X}}(\mathtt{x}) \quad .$$

By assumption, we already have

$$(\mathcal{V}_{\mathsf{N}}(\mathtt{a})\ \mathcal{V}'_{\mathsf{N}}(\mathtt{a}))_{\mathtt{a}\in(\mathtt{BN}(\mathtt{x})\cup\mathtt{C})} \cdot \mathcal{V}_{\mathsf{X}}(\mathtt{x}) \;=\; \mathcal{V}'_{\mathsf{X}}(\mathtt{x})$$

and we are left to prove that, for all $k \in [1, \mathrm{bdep}_{\mathtt{op}}(j)]$, either $\mathtt{a}^j_k \in (\mathtt{BN}(\mathtt{x}) \cup \mathtt{C})$ or $(\mathcal{V}_{\mathsf{N}}(\mathtt{a}^j_k)\ \mathcal{V}'_{\mathsf{N}}(\mathtt{a}^j_k)) \cdot \mathcal{V}'_{\mathsf{X}}(\mathtt{x}) = \mathcal{V}'_{\mathsf{X}}(\mathtt{x})$. If $\mathcal{V}_{\mathsf{N}}(\mathtt{a}^j_k) = \mathcal{V}'_{\mathsf{N}}(\mathtt{a}^j_k)$ then we are done, so we consider the case where $\mathcal{V}_{\mathsf{N}}(\mathtt{a}^j_k) \neq \mathcal{V}'_{\mathsf{N}}(\mathtt{a}^j_k)$. Then, by definition of $\mathcal{V}'_{\mathsf{N}}$, we must have $\mathtt{a}^j_k \in \mathrm{bn}(\underline{\mathtt{l}})$ or $\mathtt{a}^j_k \in \mathtt{BN}(\underline{\mathtt{src}}, \mathit{Prem})$. In the former case, we know that bn($\underline{\mathtt{l}}$) $\subseteq$ C, so we are done. If $\mathsf{a}_k^j \in \mathtt{BN}(\underline{\mathtt{src}}, \mathit{Prem})$, then, since $\mathcal{WF}(\mathtt{t})$, we have $\mathsf{a}_k^j \in \mathtt{FN}(\mathtt{x})$. By definition, then, $\mathsf{a}_k^j \in \mathtt{BN}(\mathtt{x})$, as required.

Thus properties (B.5) and (B.6) are established.

We are now ready to prove that $\mathcal{V}(\underline{\mathtt{l}}, \underline{\mathtt{tar}}) = \mathcal{V}'(\underline{\mathtt{l}}, \underline{\mathtt{tar}})$. We first prove that

$$(\mathcal{V}_{\mathsf{N}}(\mathtt{a})\ \mathcal{V}'_{\mathsf{N}}(\mathtt{a}))_{\mathtt{a}\in \mathrm{bn}(\underline{\mathtt{l}})} \cdot \mathcal{V}(\underline{\mathtt{tar}}) \ = \ \mathcal{V}'(\underline{\mathtt{tar}}) \quad . \tag{B.7}$$

We do this by using property (B.6). By Condition (8), we have $\mathcal{WF}(\underline{\mathtt{tar}})$. Condition (7) says that $(\mathtt{BN}(\underline{\mathtt{src}}, \mathit{Prem}) \cap \mathtt{FN}(\underline{\mathtt{tar}})) = \emptyset$. Moreover, for every $\mathtt{x} \in \mathtt{X}$, we have

$$(\mathcal{V}_{\mathsf{N}}(\mathtt{a})\ \mathcal{V}'_{\mathsf{N}}(\mathtt{a}))_{\mathtt{a}\in (\mathtt{BN}(\mathtt{x})\cup \mathrm{bn}(\underline{\mathtt{l}}))} \cdot \mathcal{V}_{\mathsf{X}}(\mathtt{x}) \ = \ \mathcal{V}'_{\mathsf{X}}(\mathtt{x})$$

by definition. Thus $\mathtt{C} = \mathrm{bn}(\underline{\mathtt{l}})$ is a reasonable choice. Applying property (B.6), we conclude (B.7).

It is now straightforward to show that $\mathcal{V}(\underline{\mathtt{l}}, \underline{\mathtt{tar}}) = \mathcal{V}'(\underline{\mathtt{l}}, \underline{\mathtt{tar}})$. For instance, if $\underline{\mathtt{l}} = \mathtt{c}?(\mathtt{a})$, we have

$$\begin{aligned}
\mathcal{V}(\underline{\mathtt{l}}, \underline{\mathtt{tar}}) &= \mathsf{bin}\,(\mathcal{V}_{\mathsf{N}}(\mathtt{c}), \langle \mathcal{V}_{\mathsf{N}}(\mathtt{a})\rangle \mathcal{V}(\mathtt{t})) \\
&= \mathsf{bin}\,(\mathcal{V}'_{\mathsf{N}}(\mathtt{c}), \langle \mathcal{V}_{\mathsf{N}}(\mathtt{a})\rangle \mathcal{V}(\mathtt{t})) \\
&= \mathsf{bin}\,(\mathcal{V}'_{\mathsf{N}}(\mathtt{c}), \langle \mathcal{V}'_{\mathsf{N}}(\mathtt{a})\rangle ((\mathcal{V}_{\mathsf{N}}(\mathtt{a})\ \mathcal{V}'_{\mathsf{N}}(\mathtt{a})) \cdot \mathcal{V}(\mathtt{t}))) \\
&= \mathsf{bin}\,(\mathcal{V}'_{\mathsf{N}}(\mathtt{c}), \langle \mathcal{V}'_{\mathsf{N}}(\mathtt{a})\rangle \mathcal{V}'(\mathtt{t})) \quad .
\end{aligned}$$

Here, the second line is due to Condition (6); the third line uses property (B.5) for $\underline{\mathtt{tar}}$; and the fourth line follows from (B.7).

Thus Lemma 4.6 is proved. □

References

[1] L. Aceto, W. Fokkink, and C. Verhoef. Structural operational semantics. In *Handbook of Process Algebra*, chapter 1.3, pages 197–292. Elsevier, 1999.

[2] K. L. Bernstein. A congruence theorem for structural operational semantics of higher-order languages. In *Proc. LICS'98*, pages 153–164, 1998.

[3] B. Bloom and F. Vaandrager. SOS rule formats for parameterized and state-bearing processes. Draft, 1994.

[4] B. Bloom, S. Istrail, and A. R. Meyer. Bisimulation can't be traced. *J. ACM*, 42(1):232–268, 1995.

[5] G. L. Cattani and P. Sewell. Models for name-passing processes: interleaving and causal. *Inform. and Comput.*, 190(2):136–178, 2004.

[6] A. Corradini, R. Heckel, and U. Montanari. Compositional SOS and beyond: a coalgebraic view of open systems. *Theoretical Comput. Sci.*, 280(1–2):163–192, 2002.

[7] M. P. Fiore and S. Staton. Comparing operational models of name-passing process calculi. *Inform. and Comput.*, 204(4):524–560, 2006.

[8] M. P. Fiore and S. Staton. A congruence rule format for name-passing process calculi from mathematical structural operational semantics. In *Proc. LICS'06*, pages 49–58, 2006.

[9] M. P. Fiore and D. Turi. Semantics of name and value passing. In *Proc. LICS'01*, pages 93–104, 2001.

[10] M. P. Fiore, G. D. Plotkin, and D. Turi. Abstract syntax and variable binding. In *Proc. LICS'99*, pages 193–202, 1999.

[11] W. J. Fokkink and C. Verhoef. A conservative look at operational semantics with variable binding. *Inform. and Comput.*, 146(1):24–54, 1998.

[12] M. J. Gabbay and A. M. Pitts. A new approach to abstract syntax involving binders. *Formal Aspect. Comput.*, 13(3–5):341–363, 2002.

[13] N. Ghani, K. Yemane, and B. Victor. Relationally staged computations in calculi of mobile processes. In *Proc. CMCS'04*, volume 106 of *Electron. Notes Theor. Comput. Sci.*, pages 105–120, 2004.

[14] C. A. Middelburg. Variable binding operators in transition system specifications. *J. Log. Algebr. Program.*, 47:15–45, 2001.

[15] D. Miller and A. Tiu. A proof theory for generic judgments. *ACM Trans. Comput. Logic*, 6(4):749–783, 2005.

[16] R. Milner, J. Parrow, and D. Walker. A calculus of mobile processes (II). *Inform. and Comput.*, 100(1):41–77, 1992.

[17] J. Parrow and B. Victor. The Update calculus. In *Proc. AMAST'97*, volume 1349 of *LNCS*, 1997.

[18] A. M. Pitts. Nominal logic, a first order theory of names and binding. *Inform. and Comput.*, 186(2):165–193, 2003.

[19] D. Sangiorgi. A theory of bisimulation for the pi-calculus. *Acta Inform.*, 33(1):69–97, 1996.

[20] D. Sangiorgi and D. Walker. *The π-calculus: a theory of mobile processes.* CUP, 2001.

[21] S. Staton. Name-passing process calculi: operational models and structural operational semantics. Technical Report UCAM-CL-TR-688, University of Cambridge Computer Laboratory, 2007.

[22] D. Turi and G. Plotkin. Towards a mathematical operational semantics. In *Proc. LICS'97*, pages 280–291, 1997.

[23] S. Weber and B. Bloom. Metatheory of the π-calculus. Technical Report TR96-1564, Cornell University, 1996.

[24] A. Ziegler, D. Miller, and C. Palamidessi. A congruence format for name-passing calculi. In *Proc. SOS'05*, volume 156 of *Electron. Notes Theor. Comput. Sci.*, pages 169–189, 2006.