A Cellular Howe Theorem

Peio Borthelle
Univ. Grenoble Alpes, Univ. Savoie Mont Blanc, CNRS, LAMA
73000, Chambéry, France

Tom Hirschowitz
Univ. Grenoble Alpes, Univ. Savoie Mont Blanc, CNRS, LAMA
73000, Chambéry, France

Ambroise Lafont
Gallinette, Inria Nantes, IMT Atlantique
Nantes, France

Abstract

We introduce a categorical framework for operational semantics, in which we define substitution-closed bisimilarity, an abstract analogue of the open extension of Abramsky's applicative bisimilarity. We furthermore prove a congruence theorem for substitution-closed bisimilarity, following Howe's method. We finally demonstrate that the framework covers the call-by-name and call-by-value variants of λ-calculus in big-step style. As an intermediate result, we generalise the standard framework of Fiore et al. for syntax with variable binding to the skew-monoidal case.

CCS Concepts: • **Theory of computation → Semantics and reasoning**; **Categorical semantics**; **Operational semantics**.

Keywords: operational semantics, category theory, bisimilarity, congruence, Howe's method

ACM Reference Format:
Peio Borthelle, Tom Hirschowitz, and Ambroise Lafont. 2020. A Cellular Howe Theorem. In *Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS '20), July 8–11, 2020, Saarbrücken, Germany*. ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3373718.3394738

1 Introduction

1.1 Motivation

In research on programming language design and implementation, ideas are often presented on *one*, simple example. E.g., abstract interpretation, separation logic, or gradual typing were all presented on a single language, and later adapted to other settings. Usually, the presented ideas are thought of as widely applicable and their scope is clear to the experts, but no attempt is made at delimiting it precisely and formally. As a consequence, these ideas cannot be freely reused, even in slightly different contexts, they always have to be adapted, and reproved.

We think the reason for this is that appropriate mathematical concepts are missing: there is no widely accepted notion of programming language, so that we cannot state properties like "for all programming languages of such shape, the following idea works". Such a general notion should account for both

(i) the interaction between syntax and dynamics, as involved in, e.g., structural operational semantics [28], or in statements or proofs of results like type soundness, congruence of program equivalence, or compiler correctness, and
(ii) denotational semantics, in the sense of including not only operational, syntactic models but also others, typically ones in which program equivalence is coarser.

1.2 Context

In recent work [19], Hirschowitz proposed a new abstract approach to operational semantics, and demonstrated its expressive power by proving abstract versions of classic results in process algebra, including for the first time an abstract soundness result for *bisimulation up to context* in the presence of variable binding. Bisimulation up to context is an efficient technique [29] for proving program equivalences, which had previously been proved correct at a similar level of generality [6], but only without binding.

Briefly, in the new setting, a language equipped with a collection of operational semantics rules is viewed as a monad $\mathscr{T}$ on a *transition category* $\mathscr{C}$, typically a category of labelled graphs. The idea is that, on vertices, $\mathscr{T}$ defines the syntax of the considered language: for any $X \in \mathscr{C}$, the vertices of $\mathscr{T}(X)$ are terms with free variables in vertices of X. Similarly, transitions in $\mathscr{T}(X)$ are derivation trees following the given rules, with axioms in transitions of X. So in particular $\mathscr{T}(0)$ is precisely the syntactic transition system.

Now, properties like congruence of bisimilarity or soundness of bisimulation up to context may be expressed in this setting, and their proofs in [19] rely on two crucial properties of $\mathscr{T}$.

- First, it is *familial* [7, 10], and even *cellular* in a sense close to Garner and Hirschowitz [16]. Familiality yields abstract analogues of syntactic notions like contexts and partial derivation trees, and cellularity enforces well-formedness conditions on the collection of premises of each transition rule, very roughly the fact

LICS '20, July 8–11, 2020, Saarbrücken, Germany

ACM ISBN 978-1-4503-7104-9/20/07...$15.00
https://doi.org/10.1145/3373718.3394738

that the premises of any transition $f(x_1, \ldots, x_n) \to x'$ consist of transitions from $x_1, \ldots, x_n$.
- Furthermore, in order to prove, e.g., congruence of bisimilarity for a free algebra $\mathscr{T}(X)$, the second property that we need is that the monad multiplication $\mu_X : \mathscr{T}^2(X) \to \mathscr{T}(X)$ is a functional bisimulation. Let us call this *compositionality* of $\mathscr{T}$ at X.

1.3 Overview

In this paper, we extend the approach to higher-order languages, taking as a running example the call-by-name, big step λ-calculus, equipped with the so-called open extension of Abramsky's *applicative* bisimilarity [2], which we here call *substitution-closed bisimilarity.*

Higher-order languages challenge the approach of [19] notably because transition rules rely on proper substitution, as opposed to mere renaming.

Indeed, if we follow one Fiore et al.'s categorical framework [11–13, 15, 17] for syntax with substitution in the presence of variable binding, the monad $\mathscr{T}$ we obtain is not familial. Informally, in order to model substitution, $\mathscr{T}$ features a form of explicit substitution [1], which turns out to be too flexible for familiality to work. Let us explain this in a bit more detail. In the case of pure λ-calculus, the set of terms is viewed as indexed over (finite) sets of free variables, and natively equipped with renaming operations. I.e., for any map $f : m \to n$ between finite ordinals, we get a map $X(f) : X(m) \to X(n)$ from terms with m free variables to terms with n free variables. Substitution is modelled as a map $X \otimes X \to X$, where elements of $(X \otimes X)(m)$ are 'explicit' substitutions $e(\!|e_1, \ldots, e_n|\!)$, with $e \in X(n)$ and $e_1, \ldots, e_n \in X(m)$. The problem is that such explicit substitutions are standardly considered equivalent modulo simple relations, e.g.,

$$(X(swap)(e))(\!|e_1, e_2|\!) = e(\!|e_2, e_1|\!),$$

where $swap : 2 \to 2$ swaps the two given elements.

It is precisely because of this quotienting that the obtained monad is not familial. In order to restore familiality and thus be able to follow the approach of [19], we switch to a more rigid notion of explicit substitution, which in fact forces us to change both the ambient category in examples and the general structure. The standard category for untyped calculi is the functor category $[\mathbf{Set}_f, \mathbf{Set}]$ of covariant presheaves on a skeleton of finite sets, with its standard monoidal structure. We now need to switch to mere $\mathbb{N}$-indexed families of sets, on which the relevant tensor product only yields *skew-monoidal* structure [4, 32]. Skew-monoidal structure is a weakening of plain monoidal structure, in which structural associativity and unitality isomorphisms may not be invertible.

As a final twist on this, syntax is standardly specified by a so-called *pointed-strong* endofunctor on the considered monoidal category, typically $[\mathbf{Set}_f, \mathbf{Set}]$, but the analogous endofunctor on $[\mathbb{N}, \mathbf{Set}]$ is not pointed strong. We thus need to resort to a weaker notion which we call *structurally strong*.

In summary, and this is our **first contribution**, we generalise the standard framework of pointed strong endofunctors on a monoidal category, to what we call *structurally strong* endofunctors Σ_0 on a skew-monoidal category. We characterise the syntax as the initial Σ_0*-monoid*, i.e., Σ_0-algebra with the syntax as the initial Σ_0*-monoid*, i.e., Σ_0-algebra with compatible monoid structure (Theorem 2.15), a characterisation we mechanically verify in Coq (see supplementary material), relying on the UniMath library [34]. Finally, we prove that the obtained monad $\mathscr{T}_0$ is familial, along with the fact that the natural transformation $\Sigma_0 \to \mathscr{T}_0$ preserves familiality, in a suitable sense (Theorem 2.19).

We thus obtain a framework for syntax with substitution in the presence of variable binding which lends itself to the familial/cellular approach of [19]. The next step is to model the dynamics. We first introduce *transition* Σ_0*-monoids*, which are intuitively transition systems whose vertices are equipped with Σ_0-monoid structure. Then, adapting ideas from Fiore [15] and Ahrens et al. [3, 18] to the cellular approach, we consider transition rules specified by a *syntactically free* endofunctor Σ_1 on transition Σ_0-monoids, models being given by a special kind of algebras, called *vertical*. Under finitarity hypotheses, as our **second contribution**, we characterise the syntactic transition system as the initial vertical algebra (Theorem 4.20).

Finally, we define *substitution-closed* bisimilarity, which in examples instantiates to the open extension of applicative bisimilarity. Our goal is thus to prove that substitution-closed bisimilarity is a congruence. However, we meet a last significant difficulty, namely that because of explicit substitution, compositionality fails. In fact, a slightly weaker property holds, essentially compositionality w.r.t. operations from Σ_0 (as opposed to explicit substitution). As a **third and final contribution**, under an additional cellularity hypothesis for Σ_1, we follow Howe's construction prove abstractly that substitution-closed bisimilarity is a congruence (Theorem 5.19). We show that the result applies to call-by-value and call-by-name variants of big-step, pure λ-calculus.

Altogether, under suitable hypotheses, our contributions provide a systematic construction, from the basic Σ_0 and Σ_1, of a syntactic transition system whose substitution-closed bisimilarity is a congruence.

1.4 Related work

The main framework meeting the above criteria *(i)* and *(ii)* is *bialgebraic semantics* [33], including a few variants [9, 31]. As far as we know, these approaches do not cover higher-order languages like the λ-calculus, which was one of the motivations for our work. Among more recent work, quite some inspiration was drawn from Ahrens et al. [3, 18], notably in the use of vertical algebras. However, a difference is that we do not insist that transitions be stable under substitution. Links with other relevant work, e.g., Bodin et al. [5], though desirable, remain unclear, perhaps because of the very different methods used. Furthermore, the cellularity used here is close to but different from the $\mathbf{T}_s^\vee$-familiality of [19]. It would be instructive to better understand potential links between the two. Finally, in unpublished work, Fiore and Saville have considered the skew-monoidal case as a technical, intermediate setting tool [14]. Our proof of Theorem 2.15 is inspired by their notes.

1.5 Plan

We start in §2 by recalling Fiore et al.'s standard framework for syntax with binding and then familiality, explaining our move to the skew-monoidal setting, and proving our initiality and familiality results. We then continue in §3 with a reminder and a reformulation of standard applicative bisimulation, which then guides the design of our abstract framework in §4, where we prove our initiality result for vertical algebras. Finally, in §5, after briefly recalling Howe's method, we present the abstract Howe theorem and its proof, and devote §6 to a conclusion and some perspectives.

1.6 Notation

The category of (contravariant) presheaves on a category $\mathbb{C}$ is denoted by $\widehat{\mathbb{C}}$, the Yoneda embedding by $\mathbf{y}$, and $[A, B]$ is shorthand for the hom-set, or hom-category depending on the context, of morphisms $A \to B$.

2 Syntax: familiality and substitution

In this section, we explain Fiore et al.'s approach to specifying syntax with variable binding, on the particular case of pure λ-calculus. We then show that the obtained monad is not familial, hence move to a non-standard base category. This requires us to prove our generalised initiality result, together with familiality of the obtained monad.

2.1 The standard setting

The first step is to recall Fiore et al.'s standard theory. The relevant category for this, say $\mathscr{C}_0$, is the functor category $[\mathbf{Set}_f, \mathbf{Set}]$ from finite sets to sets (let us in fact assume that $\mathbf{Set}_f$ is a small category equivalent to finite sets, e.g., finite ordinals with arbitrary maps between them), or in other words the presheaf category $\widehat{\mathbb{C}_0}$, where $\mathbb{C}_0 = \mathbf{Set}_f^{op}$. An object X is thus in particular a $\mathbf{Set}_f$-indexed family of sets, and we think of $X(m)$ as the set of states with *support* m. If X consists of terms, then $X(m)$ is typically the set of terms with free variables in m. The action of X on morphisms $f : m \to n$ accounts for variable renaming: we think of $X(f) : X(m) \to X(n)$ as renaming the free variables of terms in $X(m)$ according to f.

The basic ingredient for presenting our monad is the 'substitution' monoidal structure on $\mathscr{C}_0$: the unit I is the presheaf of variables, defined by $I(m) = m$, and elements of $(X \otimes Y)(m)$ are pairs of some $x \in X(n)$ and a *substitution* $\sigma : n \to Y(m)$, modulo the relation

$$(\sigma, f \cdot x') \sim (\sigma \circ f, x'), \tag{1}$$

for any map $f : n' \to n$ and $x' \in X(n')$, where $f \cdot x'$ denotes $X(f)(x')$. We denote by $x(\!|\sigma|\!)$ the equivalence class of (σ, x).

A *monoid* is then an object $X \in \mathscr{C}_0$ equipped with morphisms $e : I \to X$ and $m : X \otimes X \to X$, satisfying standard associativity and unitality axioms.

Fiore et al.'s theory then tells us that λ-calculus syntax is the free Σ_0^Λ-monoid, i.e., the free monoid equipped with a compatible algebra structure for the *pointed strong* endofunctor

$$\Sigma_0^\Lambda(X)(m) = X(m)^2 + X(m+1). \tag{2}$$

An algebra for this endofunctor $\Sigma_0^\Lambda : \mathscr{C}_0 \to \mathscr{C}_0$ is a presheaf X equipped with application and λ-abstraction

$$(-_1 \, -_2)_m : X(m)^2 \to X(m) \quad \text{and} \quad \lambda_m : X(m+1) \to X(m),$$

and the pointed strength specifies how standard, capture-avoiding substitution should commute with both operations. Indeed, a pointed strength is a natural transformation

$$\Sigma_0^\Lambda(X) \otimes Y \to \Sigma_0^\Lambda(X \otimes Y),$$

where Y ranges over *pointed* objects, i.e., objects in the coslice $I/\mathscr{C}$ under the monoidal unit I.

Example 2.1. The component of the pointed strength of Σ_0^Λ at X and Y maps any pair $in_r(e)(\!|\sigma|\!) \in (\Sigma_0^\Lambda(X) \otimes Y)(m)$ with $e \in X(n+1)$ (so $in_r(e) \in \Sigma_0^\Lambda(X)(n)$) and $\sigma : n \to Y(m)$, to $in_r(e(\!|\sigma^\uparrow|\!))$, where $\sigma^\uparrow$ denotes the composite

$$n+1 \xrightarrow{\sigma + in_r} Y(m) + (m+1) \xrightarrow{[Y(in_l), e_{m+1}]} Y(m+1) \tag{3}$$

where $e : I \to Y$ is the point of Y – recalling that $I(m+1) = m+1$ by definition. So, the pointed strength specifies that the given renaming σ commutes with λ-abstraction, preserving the fresh variable.

By work of Fiore [15], the forgetful functor Σ_0^Λ-**mon** $\to \mathscr{C}_0$ from Σ_0^Λ-monoids is monadic, and the induced monad $\mathscr{T}_0^\Lambda$ on $\mathscr{C}_0$ involves both operations, plus explicit substitution. It may be presented as a term language in which all explicit substitutions are pushed down towards the leaves, as in the grammar

$$\mathscr{T}_0^\Lambda(X)(n) \ni e ::= i \mid e_1 \, e_2 \mid \lambda.e' \mid o(\!|e_1, \ldots, e_m|\!),$$

where $i \in n$, $o \in X(m)$ for some $m, e_1, e_2, \ldots, e_m \in \mathscr{T}_0^\Lambda(X)(n)$, and $e' \in \mathscr{T}_0^\Lambda(X)(n+1)$.

Terminology 2.2. There are injections $n \hookrightarrow \mathscr{T}_0^\Lambda(X)(n)$ and $X(n) \hookrightarrow \mathscr{T}_0^\Lambda(X)(n)$, which are both close in spirit to injections of variables into terms. In order to distinguish them, we think of the former as an injection of variables, and of the latter as an injection of *constants*.

2.2 Failure of familiality

As announced in the introduction, the characterisation of syntax with variable binding through Σ_0-monoids is problematic for us because the obtained monad $\mathscr{T}_0^{\Lambda}$ is not familial, as we now explain. Let us first start by briefly recalling familiality [7, 10, 35, 36]. One way to understand it is as providing an abstract counterpart to multi-hole contexts, or linear terms, in the following sense.

Example 2.3. Consider the 'free monoid' monad M on **Set**, which maps any set X to the set $\sum_n X^n$ of finite sequences of elements. Any sequence in $M(X)$, say $(x_1, \ldots, x_n)$, viewed as a map $1 \to M(X)$, decomposes as the corresponding linear sequence $(1, \ldots, n)$ over $\{1, \ldots, n\}$, and the renaming mapping any i to x_i. Equivalently, using n as shorthand for $\{1, \ldots, n\}$, it factors as $1 \xrightarrow{(1,\ldots,n)} M(n) \xrightarrow{M[x_i]_i} M(X)$. In fact, linear sequences enjoy the following 'genericness' property.

Definition 2.4. Given any functor $F\colon \mathscr{A} \to \mathscr{B}$, a morphism $\xi\colon B \to F(A)$ is *F-generic* (or *generic* for short) whenever any square of the form below (solid) admits a unique lifting k (dashed) such that $F(k) \circ \xi = \chi$ and $g \circ k = f$.

$$\begin{tikzcd} B \arrow[r, "\chi"] \arrow[d, "\xi"'] & F(C) \arrow[d, "F(g)"] \\ F(A) \arrow[r, "F(f)"'] \arrow[ur, dashed, "F(k)"] & F(D) \end{tikzcd}$$

F is *familial* iff any morphism $f : B \to F(X)$ admits a *generic* factorisation, i.e., factors as $B \xrightarrow{\xi} F(A) \xrightarrow{F(h)} F(X)$ with ξ generic. Any morphism of the form $F(h)$ is deemed *free.*

We have the following important alternative characterisations in the case of presheaf categories, recalling from Paré [26] that a functor preserves connected limits iff it preserves wide pullbacks.

Theorem 2.5 (Weber [35, Theorem 8.1])**.** *For any accessible endofunctor F on any presheaf category $\widehat{\mathbb{C}}$, the following are equivalent:*

(i) *F is familial;*
(ii) *F preserves wide pullbacks;*
(iii) *there is a functor $E\colon$ $\mathrm{el}(F(1)) \to \widehat{\mathbb{C}}$ such that*

$$F(X)(c) \cong \sum_{x \in F(1)(c)} \widehat{\mathbb{C}}(E(c,x), X),$$

naturally in X and c.

Recall from MacLane and Moerdijk [25] that the *category of elements* $\mathrm{el}(X)$ of a presheaf X over any category $\mathbb{C}$ has pairs (c, x) with $x \in X(c)$ as objects, and as morphisms $(c, x) \to (c', x')$ all morphisms $f : c \to c'$ such that $X(f)(x') = x$.

Proposition 2.6. *The functor Σ_0^{Λ} is familial, but the monad $\mathscr{T}_0^{\Lambda}$ is not.*

Proof sketch. Familiality of Σ_0^{Λ} is easy by the theorem, since we have $\Sigma_0^{\Lambda}(X)(n) \cong [\mathbf{y}_n + \mathbf{y}_n, X] + [\mathbf{y}_{n+1}, X]$.

For non-familiality of $\mathscr{T}_0^{\Lambda}$, the proof is not particularly illuminating, but here is a hopefully helpful intuitive argument.

For any closed term e, viewed by action of $0 \to \mathbf{y}_1$ as an element of $\mathscr{T}_0^{\Lambda}(\mathbf{y}_1)(0)$, a natural candidate generic is the term $id_1(\!|e|\!)$, viewed by Yoneda as a morphism $\mathbf{y}_0 \to \mathscr{T}_0^{\Lambda}(\mathbf{y}_1)$, where we think of id_1 as a unary constant, to which we feed e as argument by explicit substitution.

Let us show that it is in fact not generic. Indeed, under the action of $\mathbf{y}_!\colon \mathbf{y}_1 \to \mathbf{y}_0$, id_1 is mapped to $!\colon 0 \to 1$ viewed as an element of $\mathbf{y}_0(1) = \mathbf{Set}(0, 1)$, so by (1) $id_1(\!|e|\!)$ is mapped by $\mathscr{T}_0^{\Lambda}(\mathbf{y}_!)$ to

$$!(\!|e|\!) = id_0(\!|e \circ !|\!) = id_0(\!|\,|\!).$$

Of course, this would hold for any closed $e' \neq e$, so that we get a commuting square

$$\begin{tikzcd} \mathbf{y}_0 \arrow[r, "id_1(\!|e'|\!)"] \arrow[d, "id_1(\!|e|\!)"'] & \mathscr{T}_0^{\Lambda}(\mathbf{y}_1) \arrow[d, "\mathscr{T}_0^{\Lambda}(\mathbf{y}_!)"] \\ \mathscr{T}_0^{\Lambda}(\mathbf{y}_1) \arrow[r, "\mathscr{T}_0^{\Lambda}(\mathbf{y}_!)"'] & \mathscr{T}_0^{\Lambda}(\mathbf{y}_0) \end{tikzcd}$$

with no filler. In fact, because there are infinitely many distinct closed terms, we get infinitely many factorisations of the diagonal $id_0(\!|\,|\!)$, for which no candidate generic can provide enough fillers. □

2.3 Familial syntax

So the standard monad for λ-calculus syntax is not familial, which prevents us from applying the methods of [19]. From the proof of Proposition 2.6, clearly, the problem comes from quotienting by (1), so we move to the more rigid category $\mathscr{C}_0 = [\mathbb{N}, \mathbf{Set}]$ of $\mathbb{N}$-indexed families. A first difficulty is that it does not fit Fiore et al.'s general framework. Indeed, the natural tensor product on $[\mathbb{N}, \mathbf{Set}]$, defined by

$$(A \otimes B)(n) = \sum_m A(m) \times B(n)^m,$$

is only associative and unital up to non-invertible arrows, which makes $\mathscr{C}_0$ a *skew-monoidal* category.

Notation 2.7. An element of $(A \otimes B)(n)$ is a triple (m, a, β) with $a \in A(m)$ and $\beta\colon m \to B(n)$. Leaving m implicit and thinking of the triple as an explicit substitution, we denote it again by $a(\!|\beta|\!)$.

Example 2.8. Associativity fails because elements of $(A \otimes B) \otimes C$ have the form $(a(\!|b_1, \ldots, b_m|\!))(\!|\sigma|\!)$, while those of $A \otimes (B \otimes C)$ have the form $a(\!|b_1(\!|\sigma_1|\!), \ldots, b_m(\!|\sigma_m|\!)|\!)$: we can map the former to the latter by pushing σ into the substitution $(\!|b_1, \ldots, b_m|\!)$, but not conversely, because $\sigma_1, \ldots,$ and σ_m may not all be the same. In $[\mathbf{Set}_f, \mathbf{Set}]$, $a(\!|b_1(\!|\sigma_1|\!), \ldots, b_m(\!|\sigma_m|\!)|\!)$ may be given the desired form thanks to (1), by forming the compound substitution $[\sigma_1, \ldots, \sigma_m]\colon p_1 + \ldots + p_m \to C(n)$.

So we should generalise Fiore et al.'s theory to skew-monoidal categories. But in fact, moving to the category $\mathscr{C}_0$ generates a second difficulty, namely that pointed strong endofunctors become inadequate. Indeed, e.g., the pointed strength $\Sigma_0^\Lambda(X) \otimes Y \to \Sigma_0^\Lambda(X \otimes Y)$ of Σ_0^Λ on $[\mathbf{Set}_f, \mathbf{Set}]$ relies both on variables and renaming in Y (as shown by the presence of e_{m+1} and $Y(in_l)$ in the definition (3) of $\sigma^\uparrow$). So the notion of strength we need for endofunctors on $\mathscr{C}_0$ should assume that Y comes equipped with variables and renaming. Variables are given by a point as before, while renaming is taken care of by I-module structure, in the following sense.

Definition 2.9. In any skew-monoidal category $\mathscr{C}$, for any monoid X, the category X **-mod** of *(right) X-modules* has as objects all $M \in \mathscr{C}$ equipped with an action $r\colon M \otimes X \to M$, satisfying two standard coherence conditions. A module morphism is a morphism commuting with action.

As desired, an action $Y \otimes I \to Y$ yields for all n a map $(Y \otimes I)(n) = \sum_m Y(m) \times n^m \to Y(n)$ giving the action of morphisms $m \to n$ on $Y(m)$. The unit I is canonically an I-module, and a point for an I-module Y is a morphism $I \to Y$ in I **-mod**. Thus, the appropriate category for Y is the following coslice category.

Definition 2.10. Let the category I **-mod**$_I$ of *pointed I-modules* be the coslice I/I **-mod**.

Let us now define the appropriate notion of strength, similarly to [22] (generalising [15, I.1.2]). We first equip I **-mod**$_I$ with skew-monoidal structure. Following [21, (8.1)], tensor product of (pointed) I-modules is given by the following coequaliser in $\mathscr{C}$, where $r_X\colon X \otimes I \to X$ is the I-module structure on X.

$$\begin{tikzcd}
 & X \otimes (I \otimes Y) \arrow[dr, "X\otimes\lambda_Y"] & & \\
(X \otimes I) \otimes Y \arrow[ur, "\alpha"] \arrow[rr, bend right, "r_X\otimes Y"'] & & X \otimes Y \arrow[r, two heads, "\kappa"] & X \boxtimes Y
\end{tikzcd} \tag{4}$$

By [21, Theorem 8.1], I **-mod** is a skew-monoidal category (with invertible right unit), and the forgetful functor is monoidal and creates monoids. In fact, this extends to I **-mod**$_I$, and we define:

Definition 2.11. A *structural strength* on a functor $F\colon \mathscr{C}^n \to \mathscr{C}$ is a natural transformation st with components

$$st_{A,Y}\colon F(A) \otimes Y \to F(A \otimes Y)$$

where $Y \in I$ **-mod**$_I$, $A = (X_1, \dots, X_n)$, and $A \otimes Y := (X_1 \otimes Y, \dots, X_n \otimes Y)$, making the following diagrams commute.

$$\begin{tikzcd}
 & F(A) \arrow[dl, "\rho_{F(A)}"'] \arrow[dr, "F(\rho_A)"] & \\
F(A) \otimes I \arrow[rr, "st_{A,I}"'] & & F(A \otimes I)
\end{tikzcd}$$

$$\begin{tikzcd}
F(A) \otimes X \otimes Y \arrow[r, "st_{A,X\otimes Y}"] \arrow[d, "\alpha'_{F(A),X,Y}"'] & F(A \otimes X) \otimes Y \arrow[r, "st_{A\otimes X,Y}"] & F(A \otimes X \otimes Y) \arrow[d, "F(\alpha'_{A,X,Y})"] \\
F(A) \otimes (X \boxtimes Y) \arrow[rr, "st_{A,X\boxtimes Y}"'] & & F(A \otimes (X \boxtimes Y))
\end{tikzcd}$$

where $\alpha'_{A,B,C}$ is $(A \otimes \kappa) \circ \alpha_{A,B,C}$.

Example 2.12. The endofunctor $\Sigma_0^\Lambda\colon \widehat{\mathbb{N}} \to \widehat{\mathbb{N}}$ for the syntax of pure λ-calculus is defined by the same formula (2) as before. Let us now construct its structural strength. For any pointed I-module $(Y, e\colon I \to Y, r\colon Y \otimes I \to Y)$ and map $f: m \to n$, the map $\bar{r}_n\colon \sum_m Y(m) \times n^m \to Y(n)$ specialises to $Y(f) := \bar{r}_n(-, f)\colon Y(m) \to Y(n)$. We may thus define the desired structural strength by

$$\begin{array}{rrcl}
st_{X,Y,n}\colon & (\Sigma_0^\Lambda(X) \otimes Y)(n) & \to & \Sigma_0^\Lambda(X \otimes Y)(n) \\
 & (in_l(y,z))(\!|v|\!) & \mapsto & in_l(y(\!|v|\!), z(\!|v|\!)) \\
 & (in_r(x))(\!|v|\!) & \mapsto & in_r(x(\!|v^\uparrow|\!))
\end{array}$$

where we assume $v\colon m \to Y(n)$, $x \in X(m+1)$, and $v^\uparrow : m + 1 \to Y(n+1)$ is as in (3).

Observing that any monoid is in particular a pointed I-module, we may define F-monoids in the new setting.

Definition 2.13. For any structurally strong endofunctor F, an *F-monoid* is an object X equipped with F-algebra and monoid structures $a\colon F(X) \to X$ and $I \xrightarrow{e} X \xleftarrow{m} X \otimes X$, such that the following diagram commutes.

$$\begin{tikzcd}
F(X) \otimes X \arrow[r, "st_{X,X}"] \arrow[d, "a\otimes X"'] & F(X \otimes X) \arrow[r, "F(m)"] & F(X) \arrow[d, "a"] \\
X \otimes X \arrow[rr, "m"'] & & X
\end{tikzcd}$$

Proposition 2.14. *For any structurally strong endofunctor F, F-monoids form a category F* **-mon**, *whose morphisms are morphisms of underlying objects that respect both the monoid and algebra structure.*

Our first main result is:

Theorem 2.15. *For any finitary, structurally strong endofunctor F on a cocomplete skew-monoidal category $\mathscr{C}$, if the tensor preserves colimits on the left and directed colimits on the right, then the forgetful functor $\mathscr{U}^F : F$* **-mon** *$\to \mathscr{C}$ is monadic, and the free F-monoid on any $X \in \mathscr{C}$ has carrier $\mu A.(I + F(A) + X \otimes A)$, i.e., the colimit of the chain*

$$F_X^0(0) \xrightarrow{\partial_X^0} F_X^1(0) \to \dots \to F_X^n(0) \xrightarrow{\partial_X^n} F_X^{n+1}(0) \to \dots,$$

where

- *$F_X : \mathscr{C} \to \mathscr{C}$ maps any A to $I + F(A) + X \otimes A$;*
- *$L^0 = Id$ and $L^{n+1} = L \circ L^n$, for any endofunctor L on any category;*

- $\partial^0_X \colon 0 = F^0_X(0) \to F^1_X(0)$ *is the unique such map;*
- $\partial^{n+1}_X \colon F_X(F^n_X(0)) \to F_X(F^{n+1}_X(0))$ *denotes* $F_X(\partial^n_X)$.

Notation 2.16. Let $F^{\circledast}$ denote the 'free F-monoid' monad. We sometimes abbreviate $F^{\circledast}(X)$ as $X^{\circledast}$ when F is clear.

Let us now prove that $F^{\circledast}$ is familial as desired, and furthermore that the unit $\eta_F \colon F \to F^{\circledast}$ preserves familiality, in the sense of preserving generics. In order to establish this, we rely on:

Proposition 2.17 ([35, Proposition 5.10(2)])**.** *Any* cartesian *natural transformation, i.e., one whose naturality squares are pullbacks, preserves generic morphisms.*

So we want to show that η_F is cartesian. This will rely on properties of $\mathscr{C}$, that we first show are satisfied by $\widehat{\mathbb{N}}$.

Example 2.18. As a presheaf category, $\widehat{\mathbb{N}}$ is of course *extensive* [8]. Furthermore, each $\rho_X \colon X \to X \otimes I$ is obviously monic, and ρ will be cartesian if each square of the form

$$\begin{tikzcd} X(n) \arrow[r, "x \mapsto (n,x,id_n)"] \arrow[d, "!"'] & \sum_m X(m) \times n^m \arrow[d, "\sum_m ! \times n^m"] \\ 1 \arrow[r, "* \mapsto (n,id_n)"'] & \sum_m n^m \end{tikzcd}$$

is a pullback. But if $(\sum_m ! \times n^m)(x(\!|f|\!)) = (n, id_n)$ for some $x(\!|f|\!)$, then $(m, f) = (n, id_n)$, so x is the desired unique element of $X(n)$ with specified projections.

Tensor product moreover preserves wide pullbacks, hence is familial, on both sides. On the left, this holds because coproducts commute with connected limits in presheaf categories in general, hence wide pullbacks. On the right, it holds for the same reason, plus the fact that each functor $X \mapsto X^n$ in **Set** also preserves wide pullbacks.

Abstracting over this situation, we obtain:

Theorem 2.19. *In the situation of Theorem 2.15, if*

- $\mathscr{C}$ *is extensive,*
- ρ *is cartesian and monic, and*
- $\otimes$ *and* F *are familial,*

then $F^{\circledast}$ *is familial and the natural transformation* $\eta_F \colon F \to F^{\circledast}$ *is monic and cartesian.*

3 Evaluation and applicative bisimulation

In the previous section, we have amended Fiore et al.'s standard framework to make it compatible with the familial approach to operational semantics. The next step is to deal with transitions. For this, we start in this section by analysing standard, syntactic applicative bisimulation. Guided by our findings, we will design our abstract setting in the next section.

3.1 Substitution-closed bisimulation

Standardly, evaluation is inductively defined by the rules

$$\frac{}{\lambda x.e \Downarrow \lambda x.e} \qquad \frac{e_1 \Downarrow \lambda x.e_1' \qquad e_1'[x \mapsto e_2] \Downarrow e_3}{e_1\, e_2 \Downarrow e_3}$$

and applicative bisimilarity is introduced in two stages. First, one defines applicative bisimulation on closed terms.

Definition 3.1. A relation R over closed λ-terms is an *applicative bisimulation* iff $e_1 \ R \ e_2$ and $e_1 \Downarrow \lambda x.e_1'$ entails the existence of e_2' such that $e_2 \Downarrow \lambda x.e_2'$ and, for all terms e, $e_1'[x \mapsto e] \ R \ e_2'[x \mapsto e]$, and symmetrically.

Applicative bisimulations are closed under unions, and so there is a largest applicative bisimulation called *applicative bisimilarity* and denoted by $\sim$. Then comes the second stage:

Definition 3.2. The *open extension* of a relation R on closed terms is the relation R° on potentially open terms such that $e \ R^\circ \ e'$ iff for all closed substitutions σ covering all involved free variables we have $e[\sigma] \ R \ e'[\sigma]$.

Lemma 3.3. *The open extension of any relation R is equivalently the greatest relation R' on potentially open terms such that if $e \ R' \ e'$, then for all closed substitutions σ covering all involved free variables we have $e[\sigma] \ R \ e'[\sigma]$.*

Proof. By definition, $R\Box$ satisfies the condition. To see that it is the greatest such relation, consider any R' satisfying it: for all $e \ R' \ e'$, we have $e[\sigma] \ R \ e'[\sigma]$ for all closing σ, hence $e \ R\Box \ e'$ by definition; thus $R' \subseteq R\Box$ as desired. ◻

The result we wish to prove in the abstract setting is:

Theorem 3.4 (See [27] for a historical account)**.** *The open extension $\sim^\circ$ of applicative bisimilarity is a congruence, i.e., it is an equivalence relation, and furthermore*

- $e_1 \sim^\circ e_2$ *entails* $\lambda x.e_1 \sim^\circ \lambda x.e_2$ *for all* x;
- $e_1 \sim^\circ e_2$ *and* $e_1' \sim^\circ e_2'$ *entail* $e_1\, e_1' \sim^\circ e_2\, e_2'$.

We take a slightly different viewpoint here, starting from the following observation.

Lemma 3.5. *The open extension of applicative bisimilarity is equivalently the greatest substitution-closed* opening bisimulation*, i.e., the greatest relation R on potentially open terms such that*

- $e_1 \ R \ e_2$ *entails* $e_1[x \mapsto e] \ R \ e_2[x \mapsto e]$ *for any* e;
- *if* e_1 *and* e_2 *are closed,* $e_1 \ R \ e_2$*, and* $e_1 \to^\star \lambda x.e_1'$ *then* $e_2 \to^\star \lambda x.e_2'$ *with* $e_1' \ R \ e_2'$*, and symmetrically.*

Proof. The relation $\sim^\circ$ is straightforwardly a substitution-closed opening bisimulation. But any substitution-closed opening bisimulation R restricts to an applicative bisimulation on closed terms, which is thus included in $\sim$. On open terms, if $e_1 \ R \ e_2$, then for all closing substitutions σ we have by substitution-closure of R that $e_1[\sigma] \ R \ e_2[\sigma]$, hence $e_1[\sigma] \sim e_2[\sigma]$ and so $e_1 \sim^\circ e_2$ by definition. ◻

When designing our abstract framework, we should thus be able to model two things: opening bisimulation and substitution-closed relations.

The latter is easy, remembering that tensor product on $\widehat{\mathbb{N}}$ is a sort of explicit substitution. Indeed, substitution-closed relations on any object X are relations in the category of X-modules (in the sense of Definition 2.9), i.e., objects R equipped with a map $R \otimes X \to R$ over X^2.

For modelling opening bisimulation, we should extend $\widehat{\mathbb{N}}$, which is only good at modelling the syntax of λ-calculus: in order to also model transitions, let us consider the following category.

Definition 3.6. Let $\mathbb{C}$ denote the free category on the graph with a vertex n for all $n \in \mathbb{N}$, plus a vertex $\Downarrow$, with only edges $0 \xrightarrow{s} \Downarrow \xleftarrow{t} 1$. Let $\mathscr{C} = \widehat{\mathbb{C}}$.

Objects $X \in \widehat{\mathbb{C}}$ are $\mathbb{N}$-indexed families, together with a set $X(\Downarrow)$ of *transitions*, the source and target of any $r \in X(\Downarrow)$ are given by $X(s)\colon X(\Downarrow) \to X(0)$ and $X(t)\colon X(\Downarrow) \to X(1)$.

The full embedding $\mathbf{i}\colon \mathbb{N} \hookrightarrow \mathbb{C}$ induces by restriction and left Kan extension a coreflection

$$\begin{tikzcd} \widehat{\mathbb{N}} \arrow[r, bend left, "\mathscr{M}"] \arrow[r, phantom, "\perp"] & \widehat{\mathbb{C}} \arrow[l, bend left, "\mathscr{D}"] \end{tikzcd}$$

(where $\mathscr{M}$ stands for 'monter', get up in French, and $\mathscr{D}$ for 'descendre', get down).

Because targets of transitions in any $X \in \widehat{\mathbb{C}}$ are in $X(1)$, we may define the transition system for big-step λ-calculus so as to make opening bisimulation the natural notion of bisimulation: we take it to be the presheaf X on $\mathbb{C}$, with $X(\Downarrow)$ the set of evaluation proofs $r\colon e \Downarrow \lambda x.e'$, in the standard sense, $X(s)(r) = e \in X(0)$, and $X(t)(r) = e' \in X(1)$. This way, any term evaluates to the body of its value, which, combined with substitution-closedness, achieves the desired effect. We thus consider the modified evaluation rules

$$\frac{e_1 \Downarrow e_1' \qquad e_1'[e_2] \Downarrow e_3}{e_1\ e_2 \Downarrow e_3} \qquad\qquad \frac{}{\lambda.e \Downarrow e} \tag{5}$$

where $e_1', e_3 \in X(1)$ and $e_1'[e_2]$ denotes standard capture-avoiding substitution of the unique free variable of e_1'.

Following the *open maps* approach to bisimulation [19, 20], functional opening bisimulations may be defined as arrows $f: R \to X$ such that for all commuting squares as the solid part of

$$\begin{tikzcd} 0 \arrow[r, "r"] \arrow[d, "s"'] & R \arrow[d, "f"] \\ \Downarrow \arrow[r, "e"'] \arrow[ur, dashed, "k"] & X, \end{tikzcd} \tag{6}$$

there is a filling k as shown (dashed), making both triangles commute. Opening bisimulations are then relations $R \hookrightarrow X^2$ whose projections are functional opening bisimulations.

Proposition 3.7. Substitution-closed *bisimulations, or* X-bisimulations, *i.e., bisimulations of* X*-modules, are closed under unions and hence admit a greatest element, called* X*-bisimilarity, or* substitution-closed *bisimilarity.*

With this definition, we have:

Proposition 3.8. *Substitution-closed bisimilarity is precisely the open extension of applicative bisimilarity.*

3.2 Vertical algebras

Naively, following [19], the next step should be to describe the syntax and evaluation rules of big-step λ-calculus as the initial algebra for some monad $\mathscr{T}$ on $\widehat{\mathbb{C}}$. The idea is that $\mathscr{T}(A)(n)$ should consist of terms with n free variables and constants in all $A(m)$'s, while $\mathscr{T}(A)(\Downarrow)$ should consist of transition proofs with constants in $A(\Downarrow)$ (remembering Terminology 2.2). As explained in §1, the problem lies in the β-rule (modified or not), whose premises use substitution. In order for the monad $\mathscr{T}$ to make sense, this requires the argument A to feature some notion of term substitution. In other situations, one could even imagine requiring A to be a model of the syntax. We will thus define $\mathscr{T}$ as a monad on the pullback category

$$\begin{tikzcd} \Sigma_0^{\Lambda}\text{-}\mathbf{Mon} \arrow[r, "\mathscr{D}'"] \arrow[d, "\mathscr{U}"'] \arrow[dr, phantom, "\lrcorner", very near start] & \Sigma_0^{\Lambda}\text{-}\mathbf{mon} \arrow[d, "\mathscr{U}_0"] \\ \widehat{\mathbb{C}} \arrow[r, "\mathscr{D}"'] & \widehat{\mathbb{N}}, \end{tikzcd} \tag{7}$$

whose objects we call *transition* Σ_0^{Λ}*-monoids*, are presheaves A on $\mathbb{C}$, equipped with Σ_0^{Λ}-monoid structure on the underlying presheaf $\mathscr{D}(A)$ on $\mathbb{N}$.

Let us define $\mathscr{T}$ through a generating endofunctor: given any Σ_0^{Λ}-monoid $A \in \widehat{\mathbb{C}}$, we let

$$\Sigma_1^{\Lambda}(A)(n) \quad = \quad (\Sigma_0^{\Lambda})^{\circledast}(\mathscr{D}(A))(n) \tag{8}$$

$$\Sigma_1^{\Lambda}(A)(\Downarrow) \quad = \quad A(1) + ar^{\beta}(A), \tag{9}$$

for all $n \in \mathbb{N}$, where

- $(\Sigma_0^{\Lambda})^{\circledast}$ is as in Notation 2.16,
- $A(1)$ accounts for Rule (5, right), and
- $ar^{\beta}(A)$ accounts for Rule (5, left): it denotes the set of triples $(r_1, e_2, r_2) \in A(\Downarrow)\times A(0)\times A(\Downarrow)$, such that $r_2 \cdot s = (r_1 \cdot t)[e_2]$ (where '$\cdot$' is as in (1) but for contravariant presheaves, e.g., $r_2 \cdot s = A(s)(r_2)$).

The source and target maps are defined as expected. E.g., any (r_1, e_2, r_2) with $r_1\colon e_1 \Downarrow e_1'$ and $r_2\colon e_1'[e_2] \Downarrow e_3$ has source and target $e_1 (\!|\,|\!)\ e_2 (\!|\,|\!)$ and $e_3 (\!|1|\!)$, respectively.

Remark 3.9. Substitution $e_1'[e_2]$ follows from the monoid structure of A. We should also prove that $\Sigma_1^{\Lambda}(A)$ is a transition Σ_0^{Λ}-monoid, which holds because $(\Sigma_0^{\Lambda})^{\circledast}(\mathscr{D}(A))$ is a Σ_0^{Λ}-monoid by construction. In fact, we have $\mathscr{D}'(\Sigma_1^{\Lambda}(A)) = \mathscr{K}_0(\mathscr{D}'(A))$, where $\mathscr{K}_0$ is the comonad induced by the adjunction $\mathscr{L}_0 \dashv \mathscr{U}_0$.

We have thus organised the operational rules into an endofunctor on transition Σ_0^Λ-monoids. However, we do not yet have any monad, or, worse, any notion of model. Indeed, any Σ_1^Λ-algebra A has two underlying algebra structures for the endofunctor $(\Sigma_0^\Lambda)^\circledast$: one from the Σ_0^Λ-algebra structure of $\mathscr{D}'(A)$, and the other from the fact that Σ_1^Λ coincides with $(\Sigma_0^\Lambda)^\circledast$ on the syntactic level.

But in fact, we may construct the following modified variant $\check{\Sigma}_1^\Lambda$ of Σ_1^Λ by setting

$$\check{\Sigma}_1^\Lambda(A)(n) = A(n) \qquad \text{and} \qquad \check{\Sigma}_1^\Lambda(A)(\Downarrow) = \Sigma_1^\Lambda(A)(\Downarrow),$$

with source and target maps given by composition with the $(\Sigma_0^\Lambda)^\circledast$-algebra structure of A. The relevant models are just $\check{\Sigma}_1^\Lambda$-algebras whose structure map is sent to the identity by the forgetful functor Σ_0^Λ **-Mon** $\to$ Σ_0^Λ **-mon**. Following Ahrens et al. [3, 18], we deem such algebras *vertical*.

As we will prove below (Theorem 4.20), free vertical algebras may be characterised inductively, which in the present case means that the free vertical algebra over any transition Σ_0^Λ-monoid is inductively defined by the modified rules (5), augmented with an axiom

$$\frac{o \Downarrow o' \text{ in } A}{o(\!|\,|\!) \Downarrow o'(\!|1|\!)} \;. \tag{10}$$

Lifting the adjunction of Theorem 2.15 from $\widehat{\mathbb{C}_0}$ to $\widehat{\mathbb{C}}$, we obtain a chain of monadic adjunctions

$$\begin{tikzcd} \widehat{\mathbb{C}} \arrow[r, bend left] \arrow[r, phantom, "\perp"] & \Sigma_0^\Lambda\textbf{-Mon} \arrow[l, bend left] \arrow[r, bend left] \arrow[r, phantom, "\perp"] & \check{\Sigma}_1^\Lambda\text{-alg}_v, \arrow[l, bend left] \end{tikzcd}$$

where $\check{\Sigma}_1^\Lambda\text{-alg}_v$ is the category of vertical $\check{\Sigma}_1^\Lambda$-algebras.

We at last arrive at a setting in which to study the question we started with, in its new guise: is substitution-closed bisimilarity a congruence in the initial vertical algebra?

However, there is a last issue that prevents us from directly applying the techniques of [19]: calling $\mathscr{T}^\Lambda$ the obtained monad over $\mathscr{C} = \widehat{\mathbb{C}}$, compositionality fails, i.e., the multiplication μ for $\mathscr{T}^\Lambda$ is not a functional bisimulation.

Example 3.10. The problem is essentially as follows. Monad multiplication takes a term of terms (i.e., constants are terms), and performs the substitution: typically, for any such terms of terms $E_1, \ldots, E_n$, we have

$$\mu(e(\!|E_1, \ldots, E_m|\!)) = e[\mu(E_1), \ldots, \mu(E_m)],$$

i.e., e where each variable $j \in m$ is replaced with $\mu(E_j)$. E.g., over $m = 2 = \{x, y\}$, the term of terms $(x\ y)(\!|\lambda z.z, \lambda z.z|\!)$ is mapped to the closed redex $e := (\lambda z.z)\ (\lambda z.z)$. Since we have $e \Downarrow z$, compositionality would require the existence of a transition $(x\ y)(\!|\lambda z.z, \lambda z.z|\!) \Downarrow E'$, for some E'. However, the only transition rule whose conclusion has as source a term of the form $o(\!|\ldots|\!)$ is (10), in which o is nullary, so it cannot apply here.

Although we cannot hope free vertical algebras to be compositional, we in fact only need a restricted form of compositionality, demanding that for any commuting square as the solid part below,

$$\begin{tikzcd} 0 \arrow[r] \arrow[d] & \Sigma_0^\Lambda(A) \arrow[r] & \Sigma_1^\Lambda(A) \arrow[d] \\ \Downarrow \arrow[rr] \arrow[urr, dashed, "k"] & & A \end{tikzcd} \tag{11}$$

there exists a lifting k as shown. In this case, we call A a *weakly compositional* algebra.

Remark 3.11. The endofunctor Σ_0^Λ really acts on $\mathscr{C}_0$, not $\mathscr{C}$, so some implicit casting is going on: the whole diagram takes place in $\mathscr{C}$, so $\Sigma_1^\Lambda(A)$ and A should be considered as shorthand for their images through the forgetful functor $\mathscr{U}\colon \Sigma_0^\Lambda$ **-Mon** $\to \mathscr{C}$, while $\Sigma_0^\Lambda(A)$ is in fact shorthand for $\mathscr{M}(\Sigma_0^\Lambda(\mathscr{D}(\mathscr{U}(A))))$.

Intuitively, by Yoneda, weak compositionality says that for any transition $r \in A(\Downarrow)$ whose source is obtained by evaluating a term e of depth 1, r is obtained by evaluating a transition term $R \in \Sigma_1^\Lambda(A)(\Downarrow)$, whose source is e, all as in

$$\begin{tikzcd} e \arrow[r, mapsto] \arrow[d, dashed, "R"'] & \llbracket e \rrbracket_A \arrow[d, "r = \llbracket R \rrbracket_A"] \\ e' \arrow[r, mapsto] & x', \end{tikzcd}$$

where $\llbracket - \rrbracket_A\colon \Sigma_1^\Lambda(A) \to A$ denotes the algebra structure. In other terms, $\llbracket - \rrbracket_A$ has the functional bisimulation property restricted to terms of depth 1 without explicit substitution.

4 An abstract setting for Howe's method

We are now ready for abstracting over big-step λ-calculus.

4.1 Howe contexts

We start by axiomatising the ambient setting for transition systems, notably their layered nature. Let us recall that any fully faithful functor $F\colon \mathbb{C} \to \mathbb{D}$ induces a full coreflection

$$\sum_F\colon \quad \begin{tikzcd} \widehat{\mathbb{C}} \arrow[r, bend left] \arrow[r, phantom, "\perp"] & \widehat{\mathbb{D}} \arrow[l, bend left] \end{tikzcd} \quad : \Delta_F,$$

where $\sum_F$ and Δ_F respectively denote left Kan extension and restriction along F^{op}. By the triangle identities, $\Delta_F(\varepsilon_X)$ and $\varepsilon_{\sum_F(C)}$ are isomorphisms for all $X \in \widehat{\mathbb{D}}$ and $C \in \widehat{\mathbb{C}}$.

Furthermore, let $\mathbb{2}$ denote the poset $\{0 < 1\}$ viewed as a category.

Definition 4.1. A *Howe context* consists of a small category $\mathbb{C}$ together with a functor $\mathbf{p}\colon \mathbb{C} \to \mathbb{2}$, equipped with skew-monoidal structure on the presheaf category $\widehat{\mathbb{C}_0}$, where $\mathbb{C}_0$ denotes the fibre of $\mathbf{p}$ over 0, satisfying

(H1) each functor $- \otimes C$ is familial and preserves colimits,
(H2) each functor $C \otimes -$ is familial and preserves filtered colimits, and

(H3) denoting by $\mathbf{i}\colon \mathbb{C}_0 \hookrightarrow \mathbb{C}$ the canonical embedding, the counit $\varepsilon_L\colon \sum_{\mathbf{i}} \Delta_{\mathbf{i}}(L) \to L$ is a copairing $[s,t]\colon P + Q \to L$ with $P, Q \in \mathbb{C}_0$, for all $L \in \mathbb{C} \setminus \mathbb{C}_0$.

In particular, each $\Delta_{\mathbf{i}}[s,t]$ is an isomorphism, so any morphism from some $R \in \mathbb{C}_0$ to L factors uniquely through either s or t.

Notation 4.2. We respectively call objects of $\mathbb{C}$, $\mathbb{C}_0$, and $\mathbb{C} \setminus \mathbb{C}_0$ *basic objects*, *state types*, and *transition types*, and let $\mathscr{C}_0 = \widehat{\mathbb{C}_0}$, $\mathscr{C} = \widehat{\mathbb{C}}$, $\mathscr{M} = \sum_{\mathbf{i}}$, and $\mathscr{D} = \Delta_{\mathbf{i}}$. We often omit $\mathscr{M}$ and $\mathscr{D}$ for readability.

Let us note that $\mathscr{M}$ extends a presheaf $X \in \widehat{\mathbb{C}_0}$ by mapping any transition type to the empty set.

Example 4.3. For pure λ-calculus, $\varepsilon_{\Downarrow}$ is the copairing $0 + 1 \xrightarrow{[s,t]} \Downarrow$ (or rather its image under the Yoneda embedding). We have already explained familiality in Example 2.18. Regarding colimits, for $- \otimes X$, the considered colimits should merely commute with coproducts, hence $- \otimes X$ commutes with all colimits. For $X \otimes -$, the considered colimits should commute with coproducts and finite products in sets, so the best we can say in general is that $X \otimes -$ commutes with sifted colimits, hence in particular with filtered colimits.

Let us conclude this section with two crucial properties.

Lemma 4.4. *The functor $\mathscr{D}\colon \mathscr{C} \to \mathscr{C}_0$ is a bifibration, for which opcartesian liftings are isos at transitions types.*

Proof. The opcartesian lifting $f \cdot X$ of any given $X \in \mathscr{C}$ along $f : \mathscr{D}(X) \to C$ is given by taking $(f \cdot X)(P) = C(P)$ for all $P \in \mathbb{C}_0$ and, $(f \cdot X)(L) = X(L)$ for all transition types $L \leftarrow P + Q\colon [s,t]$, with $(f \cdot X)(s)$ and $(f \cdot X)(t)$ given by composition, e.g., $X(L) \to X(P) \to C(P)$.

The cartesian lifting $X \cdot f$ of any $X \in \mathscr{C}$ along $f : C \to \mathscr{D}(X)$ is given by taking $(X \cdot f)(P) = C(P)$ for all $P \in \mathbb{C}_0$ and, for all transition types $P + Q \to L$, $(X \cdot f)(L)$ to be the pullback

$$\begin{tikzcd}
(X \cdot f)(L) \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & X(L) \arrow[d, "\langle X(s), X(t)\rangle"] \\
C(P) \times C(Q) \arrow[r, "f_P \times f_Q"'] & X(P) \times X(Q)
\end{tikzcd}$$ □

Corollary 4.5. *Any morphism in $\mathscr{C}$ factors as a* vertical *morphism l (= such that $\mathscr{D}(l) = id$), followed by a cartesian one.*

4.2 Substitution-closed bisimulation

With the setting of Howe contexts in place, we now abstract over substitution-closed bisimulations.

Plain, functional bisimulations are defined by lifting against all maps s from distinguished copairings $[s,t]$, e.g., as in (6). Similarly, bisimulation relations are defined as relations whose projections are functional bisimulations.

Now, substitution-closed bisimulations should live in a category of transition systems whose underlying object in $\mathscr{C}_0$ is substitution-closed. As we will use this idea of transition systems with structured underlying object several times, let us factor out the construction:

Lemma 4.6. *For any monadic functor $U_0\colon \mathscr{E} \to \mathscr{C}_0$, consider the pullback*

$$\begin{tikzcd}
\mathscr{D}^*(\mathscr{E}) \arrow[r, "\mathscr{D}\restriction\mathscr{E}"] \arrow[d, "\mathscr{D}^*(U_0)"'] \arrow[dr, phantom, "\lrcorner", very near start] & \mathscr{E} \arrow[d, "U_0"] \\
\mathscr{C} \arrow[r, "\mathscr{D}"'] & \mathscr{C}_0.
\end{tikzcd}$$

If the monad induced by U_0 is accessible, then $\mathscr{D} \restriction \mathscr{E}$ has a fully-faithful left adjoint and $\mathscr{D}^*(U_0)$ has a left adjoint.

Terminology 4.7. If objects of $\mathscr{E}$ are called a certain name, say, *things*, then objects of $\mathscr{D}^*(\mathscr{E})$ will often be called *transition things*.

Definition 4.8. Let *transition monoids* be the objects of $\mathbf{Mon}(\mathscr{C}) := \mathscr{D}^*(\mathbf{mon}(\mathscr{C}))$, where $\mathbf{mon}(\mathscr{C})$ is the category of monoids in $\mathscr{C}_0$.

Furthermore, let the category of *transition X-modules* be $X\text{-}\mathbf{Mod} := \mathscr{D}^*(\mathscr{D}(X)\text{-}\mathbf{mod})$, for any $X \in \mathbf{Mon}(\mathscr{C})$.

We let $\mathscr{F} : \mathscr{D}(X)\text{-}\mathbf{mod} \to \mathscr{C}_0$ denote the forgetful functor, and $\mathscr{F}' = \mathscr{D}^*(\mathscr{F})\colon X\text{-}\mathbf{Mod} \to \mathscr{C}$.

So transition X-modules are objects $M \in \mathscr{C}$ whose underlying $\mathscr{D}(M) \in \mathscr{C}_0$ is a $\mathscr{D}(X)$-module.

Definition 4.9. For any transition monoid X, a *functional X-bisimulation* is a map of transition X-modules whose image under $\mathscr{F}'$ is a functional bisimulation. An *X-bisimulation* is a relation of transition X-modules whose projections are both functional X-bisimulations.

We are now interested in defining the largest X-bisimulation. This requires the following few intermediate results, leading to Corollary 4.12.

Proposition 4.10. *The forgetful functor $\mathscr{F}'\colon X\text{-}\mathbf{Mod} \to \mathscr{C}$ creates unions.*

Lemma 4.11. *X-bisimulations are closed under unions.*

Proof. X-bisimulations are in particular plain bisimulations, so any union, which is computed as in $\mathscr{C}$ by Proposition 4.10, is again a bisimulation, as desired. □

Corollary 4.12. *For any X, the union $\sim^{\otimes}_X$ of all X-bisimulations over X, called X*-bisimilarity*, is an X-bisimulation.*

4.3 Signatures, models, and initiality

In this section, in order to specify syntax and transition rules in the abstract setting, we introduce the notion of 2-signature and its models. We furthermore show that under mild hypotheses any 2-signature admits free models.

Let us first incorporate syntax into the transition monoids of Definition 4.8. For a general Howe context and structurally strong endofunctor Σ_0 on $\mathscr{C}_0$, we first define the category

$\Sigma_0\text{-}\mathbf{Mon} := \mathscr{D}^*(\Sigma_0\text{-}\mathbf{mon})$ of *transition* Σ_0*-monoids* with notation as below left.

$$\begin{tikzcd} \Sigma_0\text{-}\mathbf{Mon} \arrow[r, "\mathscr{D}'"] \arrow[d, "\mathscr{U}"'] \arrow[dr, phantom, "\lrcorner", very near start] & \Sigma_0\text{-}\mathbf{mon} \arrow[d, "\mathscr{U}_0"] \\ \mathscr{C} \arrow[r, "\mathscr{D}"'] & \mathscr{C}_0 \end{tikzcd} \qquad \begin{tikzcd} \Sigma_0\text{-}\mathbf{Mon} \arrow[r, "\Sigma_1"] \arrow[d, "\mathscr{D}'"'] & \Sigma_0\text{-}\mathbf{Mon} \arrow[d, "\mathscr{D}'"] \\ \Sigma_0\text{-}\mathbf{mon} \arrow[r, "\mathscr{K}_0"'] & \Sigma_0\text{-}\mathbf{mon} \end{tikzcd} \tag{12}$$

Notation 4.13. We often omit $\mathscr{D}'$ and $\mathscr{U}$ for readability.

An endofunctor Σ_1 on $\Sigma_0\text{-}\mathbf{Mon}$ will be called *syntactically free* iff the Σ_0-monoid structure of any $\mathscr{D}'(\Sigma_1(X))$ is precisely the free structure on $\mathscr{U}_0(\mathscr{D}'(X))$, i.e., $\mathscr{D}'(\Sigma_1(X)) = \mathscr{L}_0(\mathscr{U}_0(\mathscr{D}'(X)))$, where $\mathscr{L}_0$ denotes the left adjoint to $\mathscr{U}_0$. Otherwise said, letting $\mathscr{K}_0$ denote the induced comonad $\mathscr{L}_0 \circ \mathscr{U}_0$, the diagram above right commutes.

Notation 4.14. Let $\mathscr{T}_0 = \mathscr{U}_0 \circ \mathscr{L}_0$, the induced monad.

Definition 4.15. A *2-signature* on a Howe context $\mathbb{C} \to \mathbb{2}$ consists of a structurally strong, finitary endofunctor Σ_0 on $\mathscr{C}_0$ preserving wide pullbacks, and a syntactically free endofunctor Σ_1 on transition Σ_0-monoids.

Remark 4.16. Any 2-signature induces a signature in the sense of Hirschowitz et al. [18, §4.1].

Let us now define the category of models of a 2-signature, by mimicking the concrete notion of vertical algebra from §3.2. For this, we observe that passing from Σ_1^Λ to $\check{\Sigma}_1^\Lambda$ may be done by opcartesian lifting in the sense of Lemma 4.4. Indeed, bifibrations are stable under pullback, so $\mathscr{D}'$ is again a bifibration, and we may state:

Definition 4.17. Let $\check{\Sigma}_1 \colon \Sigma_0\text{-}\mathbf{Mon} \to \Sigma_0\text{-}\mathbf{Mon}$ be defined by letting $\check{\Sigma}_1(X)$ be a choice of oplifting of $\Sigma_1(X)$ along the counit $\varepsilon^0_{\mathscr{D}'(X)} \colon \mathscr{D}'(\Sigma_1(X)) = \mathscr{K}_0(\mathscr{D}'(X)) \to \mathscr{D}'(X)$.

We obtain by construction:

Proposition 4.18. *The triangle below commutes.*

$$\begin{tikzcd} \Sigma_0\text{-}\mathbf{Mon} \arrow[rr, "\check{\Sigma}_1"] \arrow[dr, "\mathscr{D}'"'] & & \Sigma_0\text{-}\mathbf{Mon} \arrow[dl, "\mathscr{D}'"] \\ & \Sigma_0\text{-}\mathbf{mon} & \end{tikzcd}$$

Definition 4.19. Let the category $\check{\Sigma}_1\text{-alg}_v$ be the full subcategory of $\check{\Sigma}_1\text{-alg}$ on *vertical* $\check{\Sigma}_1$-algebras, i.e., ones whose structure maps $\check{\Sigma}_1(X) \to X$ are sent by $\mathscr{D}'$ to identities.

The functor $\check{\Sigma}_1$ restricts to an endofunctor $(\check{\Sigma}_1)_{|X}$ on the fibre over any Σ_0-monoid X, and we call Σ_1 *vertically finitary* iff each such restriction $(\check{\Sigma}_1)_{|X}$ is finitary.

Theorem 4.20. *If Σ_1 is vertically finitary, then vertical $\check{\Sigma}_1$-algebras are monadic over transition Σ_0-monoids, and the free vertical $\check{\Sigma}_1$-algebra over any given X is $\mu A.(X + (\check{\Sigma}_1)_{|\mathscr{D}'(X)}(A))$, i.e., $\mathrm{colim}_{n\in\omega} X_n$ with $X_0 = X$ and $X_{n+1} = X + (\check{\Sigma}_1)_{|\mathscr{D}'(X)}(X_n)$. In particular, the initial vertical $\check{\Sigma}_1$-algebra is the colimit (in the fibre) of the chain $0^\circledast \to (\check{\Sigma}_1)_{|0^\circledast}(0^\circledast) \to (\check{\Sigma}_1)^2_{|0^\circledast}(0^\circledast) \to \ldots$, where $0^\circledast$ denotes the initial Σ_0-monoid.*

4.4 Weak compositionality

We conclude this section by showing *weak compositionality* of the initial vertical algebra, say $\mathbf{Z} = \mu A.(\check{\Sigma}_1)_{|0^\circledast}(A)$, under a suitable hypothesis. The only thing we will need to know is that the source term of a transition in $\Sigma_1(X)$ has depth one. In order to state this properly, let us observe that by (12) we have $\mathscr{D}\mathscr{U}\Sigma_1 = \mathscr{T}_0\mathscr{D}\mathscr{U}$, so by universal property of counit, as $\mathscr{D}(\mathscr{M}(P)) \cong P$, for any transition $r \colon L \to \mathscr{U}(\Sigma_1(X))$, the source $r \circ s$ admits a unique lifting $r \setminus s$ as shown below.

$$\begin{tikzcd} P \arrow[rr, "s"] \arrow[d, dashed, "r\backslash s"'] & & L \arrow[d, "r"] \\ \mathscr{M}(\mathscr{T}_0(\mathscr{D}(\mathscr{U}(X)))) = \mathscr{M}(\mathscr{D}(\mathscr{U}(\Sigma_1(X)))) \arrow[rr, "\varepsilon_{\mathscr{U}(\Sigma_1(X))}"'] & & \mathscr{U}(\Sigma_1(X)) \end{tikzcd}$$

Definition 4.21. The 2-signature (Σ_0, Σ_1) is *layered* iff for any $r \colon L \to \mathscr{U}(\Sigma_1(X))$, $r \setminus s$ lifts through $\mathscr{M}(\eta_{\Sigma_0, \mathscr{D}(\mathscr{U}(X))})$, as in

$$\begin{tikzcd} & \mathscr{M}(\Sigma_0(\mathscr{D}(\mathscr{U}(X)))) \arrow[d, "\mathscr{M}(\eta_{\Sigma_0, \mathscr{D}(\mathscr{U}(X))})"] \\ P \arrow[ur, dashed] \arrow[r, "r\backslash s"'] & \mathscr{M}(\mathscr{T}_0(\mathscr{D}(\mathscr{U}(X)))). \end{tikzcd}$$

Informally, this means that the source of the conclusion of a transition rule must be an operation applied to some "metavariables": given any transition $r \in \Sigma_1(X)(L)$, there exists $m \in \Sigma_0(X)(P)$ such that $r \cdot s = \eta(m)$, with $\eta \colon \Sigma_0 \to \mathscr{T}_0$.

Remark 4.22. As η is monic by Theorem 2.19, and $\mathscr{M}$ preserves monos, the lifting is unique.

Proposition 4.23. *If (Σ_0, Σ_1) is layered, then any commuting square as the solid part below admits a lifting k as shown.*

$$\begin{tikzcd} P \arrow[r, "e"] \arrow[d, "s"'] & \Sigma_0(\mathbf{Z}) \arrow[r, "\iota_X"] & \Sigma_1(\mathbf{Z}) \arrow[d] \\ L \arrow[urr, dashed, "k"] \arrow[rr, "r"'] & & \mathbf{Z} \end{tikzcd} \tag{13}$$

4.5 Congruence statement

We have now characterised the initial vertical algebra for a 2-signature as our abstract model of syntactic transition systems from operational semantics. Let us now state the desired congruence result in this abstract setting.

Definition 4.24. A relation $R \hookrightarrow \mathscr{U}(X)^2$ over a transition Σ_0-monoid X is a *congruence* iff it is an equivalence relation and furthermore, omitting $\mathscr{U}$, the composite

$$\mathscr{T}_0(\mathscr{D}(R)) \to \mathscr{T}_0(\mathscr{D}(X^2)) \to \mathscr{T}_0(\mathscr{D}(X))^2 \to \mathscr{D}(X)^2$$

factors through $\mathscr{D}(R) \to \mathscr{D}(X)^2$.

Let us directly reduce this to the following easier form:

Lemma 4.25. *An equivalence relation $R \hookrightarrow \mathscr{U}(X)^2$ is a congruence iff $\mathscr{D}(R)$ is both a submonoid and a subalgebra of $\mathscr{D}(\mathscr{U}(X))^2$, i.e., omitting $\mathscr{U}$ and $\mathscr{D}$, both composites*

$$R \otimes R \to X^2 \otimes X^2 \to (X \otimes X)^2 \to X^2 \tag{14}$$

$$\Sigma_0(R) \to \Sigma_0(X^2) \to \Sigma_0(X)^2 \to X^2 \tag{15}$$

factor through $R \to X^2$.

5 Howe's method

In the previous section, we have introduced 2-signatures and shown that under suitable hypotheses they admit initial models (= initial vertical algebras) which are weakly compositional. We now want to adapt Howe's method to this abstract setting and show that substitution-closed bisimilarity on the initial vertical algebra is a congruence. We thus fix any vertically finitary, layered 2-signature (Σ_0, Σ_1), and let $\mathbf{Z}$ denote its initial vertical algebra. We start in §5.1 by giving a brief introduction to Howe's method. In §5.2, we continue with some preparatory work: we introduce *prebisimulation*, which is very close to lifting-based bisimulation, except that, unlike for plain bisimulations, it makes sense to take the context closure of a prebisimulation. We exploit this in §5.3, where we introduce our abstract version of the *Howe closure* $R^\bullet$ of a relation R. We furthermore reduce congruence of bisimilarity $\sim^{\otimes}_{\mathbf{Z}}$ for the initial vertical Σ_1-algebra $\mathbf{Z}$ to the fact that the Howe closure $(\sim^{\vee,\otimes}_{\mathbf{Z}})^\bullet$ of the associated prebisimulation is itself a presimulation. Finally, in §5.4, we introduce *cospan-cellularity* and show that if Σ_1 is cospan-cellular, then $(\sim^{\vee,\otimes}_{\mathbf{Z}})^\bullet$ is indeed a presimulation, which entails the main result.

5.1 A subjective introduction to Howe's method

Let us start by briefly recalling Howe's method, loosely following Pitts [27]. Let us return to our running example, big-step, pure λ-calculus, and naively attempt to prove that bisimilarity is context-closed. The idea is to consider some context-closed relation $\sim^{\bullet}_{\mathbf{Z}}$, containing $\sim^{\otimes}_{\mathbf{Z}}$ by construction, and then to show that it is a bisimulation. By maximality of $\sim^{\otimes}_{\mathbf{Z}}$, we then also have $\sim^{\bullet}_{\mathbf{Z}} \subseteq \sim^{\otimes}_{\mathbf{Z}}$ hence both relations coincide and $\sim^{\otimes}_{\mathbf{Z}}$ is context-closed as desired.

In order for $\sim^{\bullet}_{\mathbf{Z}}$ to be context-closed, it should at least contain all pairs $(C[e_1, \ldots, e_n], C[e'_1, \ldots, e'_n])$ with $e_i \sim^{\otimes}_{\mathbf{Z}} e'_i$ for all $i \in n$. Typically, let us consider $e_1 \sim^{\otimes}_{\mathbf{Z}} e'_1$ and $e_2 \sim^{\otimes}_{\mathbf{Z}} e'_2$, and try to prove the simulation property for $e_1\ e_2 \sim^{\bullet}_{\mathbf{Z}} e'_1\ e'_2$. We thus assume $e_1 \Downarrow \lambda x.e_3$ and $e_3[e_2] \Downarrow e_4$, and try to find e'_3 such that $e'_1 \Downarrow \lambda x.e'_3$ and e'_4 such that $e'_3[e'_2] \Downarrow e'_4$ with $e_4 \sim^{\bullet}_{\mathbf{Z}} e'_4$. Because $e_1 \sim^{\otimes}_{\mathbf{Z}} e'_1$, we find $e'_1 \Downarrow \lambda x.e'_3$ such that for all e, $e_3[e] \sim^{\otimes}_{\mathbf{Z}} e'_3[e]$. This e'_3 is a natural candidate to suit our needs. But then how do we find e'_4? Because $\sim^{\bullet}_{\mathbf{Z}}$ is context-closed and $\sim^{\otimes}_{\mathbf{Z}}$ is substitution-closed, we have

$$e_3[e_2] \sim^{\bullet}_{\mathbf{Z}} e_3[e'_2] \sim^{\otimes}_{\mathbf{Z}} e'_3[e'_2]. \tag{16}$$

Assuming we are reasoning by induction over the considered transition proof, we find e''_4 (by induction hypothesis) and e'_4 (by definition of bisimulation) as in

$$\begin{tikzcd} e_3[e_2] \arrow[d, Rightarrow] & \sim^{\bullet}_{\mathbf{Z}} & e_3[e'_2] \arrow[d, Rightarrow, dashed] & \sim^{\otimes}_{\mathbf{Z}} & e'_3[e'_2] \arrow[d, Rightarrow, dashed] \\ e_4 & \sim^{\bullet}_{\mathbf{Z}} & e''_4 & \sim^{\otimes}_{\mathbf{Z}} & e'_4. \end{tikzcd}$$

This suggests that $\sim^{\bullet}_{\mathbf{Z}}$ should be closed under action of $\sim^{\otimes}_{\mathbf{Z}}$, i.e., $\sim^{\bullet}_{\mathbf{Z}}; \sim^{\otimes}_{\mathbf{Z}} \subseteq \sim^{\bullet}_{\mathbf{Z}}$, or otherwise said: for all $e \sim^{\bullet}_{\mathbf{Z}} e' \sim^{\otimes}_{\mathbf{Z}} e''$ we have $e \sim^{\bullet}_{\mathbf{Z}} e''$. Howe's idea is to take this as a defining property of $\sim^{\bullet}_{\mathbf{Z}}$. Coupling this with context closure, we define $\sim^{\bullet}_{\mathbf{Z}}$ as the smallest context-closed relation satisfying the rules

$$\frac{}{x \sim^{\bullet}_{\mathbf{Z}} x} \qquad \frac{e \sim^{\bullet}_{\mathbf{Z}} e' \qquad e' \sim^{\otimes}_{\mathbf{Z}} e''}{e \sim^{\bullet}_{\mathbf{Z}} e''}.$$

By construction, $\sim^{\bullet}_{\mathbf{Z}}$ is reflexive and context-closed. By reflexivity and the second rule, it also contains $\sim^{\otimes}_{\mathbf{Z}}$.

Remark 5.1. In §5.3, we use an equivalent (because $\sim^{\otimes}_{\mathbf{Z}}$ is reflexive and transitive), inductive, and perhaps more compact definition.

The initial plan was to show that $\sim^{\bullet}_{\mathbf{Z}}$ is a bisimulation and deduce that it coincides with $\sim^{\otimes}_{\mathbf{Z}}$. We can in fact optimise this slightly by first showing that $\sim^{\bullet}_{\mathbf{Z}}$ is a simulation, and then that its transitive closure $(\sim^{\bullet}_{\mathbf{Z}})^+$ is symmetric. The relation $(\sim^{\bullet}_{\mathbf{Z}})^+$ is thus a symmetric simulation, hence a bisimulation. This entails the last inclusion in the chain $\sim^{\otimes}_{\mathbf{Z}} \subseteq \sim^{\bullet}_{\mathbf{Z}} \subseteq (\sim^{\bullet}_{\mathbf{Z}})^+ \subseteq \sim^{\otimes}_{\mathbf{Z}}$, showing that all relations coincide. Finally, because $\sim^{\bullet}_{\mathbf{Z}}$ is context-closed, so is $\sim^{\otimes}_{\mathbf{Z}}$, as desired.

5.2 Prebisimulations

In order to adapt Howe's method to the abstract setting, we often need to consider the context closure $\Sigma_0(R)$ of relations, which lives in $\mathscr{C}_0$, not $\mathscr{C}$. It is thus more convenient to resort to the following variant of bisimulations, adapted from [19].

Definition 5.2. For any $X, Y \in \mathscr{C}$ and relations $R, R' \subseteq \mathscr{D}(X) \times \mathscr{D}(Y)$, we say that R *left-progresses* to R', and write $R \rightsquigarrow_l R'$, when any commuting square of the form below left, where $\mathscr{M}(R) \to X$ is obtained by transposition, may be embedded into some commuting diagram as below right.

$$\begin{tikzcd} P \arrow[r] \arrow[d, "s"'] & \mathscr{M}(R) \arrow[d] \\ L \arrow[r] & X \end{tikzcd} \qquad \begin{tikzcd} P \arrow[rr] \arrow[dd, "s"'] & & \mathscr{M}(R) \arrow[dl] \arrow[ddr] & \\ & X & & \\ L \arrow[ur] \arrow[rrr, dashed] & & & Y \\ Q \arrow[u, "t"] \arrow[rr, dashed] & & \mathscr{M}(R') \arrow[uul] \arrow[ur] & \end{tikzcd} \tag{17}$$

A *presimulation* is a relation R such that $R \rightsquigarrow_l R$, and a *prebisimulation* is a presimulation whose converse relation is also a presimulation.

Intuitively, left-progression is one half of the standard notion of progression [29]: given any transition $x_1 \xrightarrow{L} x_2$ with $x_1\ R\ y_1$, we find a transition $y_1 \xrightarrow{L} y_2$ such that $x_2\ R'\ y_2$.

Remark 5.3. This is equivalent to [19, Definition 5.1].

The advantage of prebisimulations over bisimulations is that they live in $\mathscr{C}_0$, hence morally only involve terms. However, they tell us essentially the same thing as bisimulations, as shown in [19, §5.2]. In particular, one can prove that prebisimulations are closed under union.

Definition 5.4. The union of all prebisimulations over any fixed X is called *prebisimilarity* and denoted by $\sim^{\vee}_{X}$.

Furthermore, this lifts to substitution-closed prebisimulations: let an X*-prebisimulation* be a relation $R \hookrightarrow \mathscr{D}(M)^2$ in X-**mod**, for any $M \in X$-**Mod**, such that $\mathscr{F}(R)$ is a prebisimulation; we have that X-prebisimulations are closed under unions, whence:

Definition 5.5. Let $\sim^{\vee,\otimes}_{X}$ denote the union of all X-prebisimulations over X, called X*-prebisimilarity*.

Proposition 5.6. *We have* $\mathscr{D}(\sim^{\otimes}_{X}) = \sim^{\vee,\otimes}_{X}$.

5.3 Howe closure

We now want to prove that $\sim^{\otimes}_{\mathbf{Z}}$ is a congruence, where, we recall, $\mathbf{Z} = \mu A.(\check{\Sigma}_1)_{|0^{\circledast}}(A)$ is the initial vertical algebra.

Lemma 5.7. *If* $\sim^{\vee,\otimes}_{\mathbf{Z}}$ *is both a subalgebra and a submonoid of* $\mathscr{D}(\mathbf{Z})^2$*, then* $\sim^{\otimes}_{\mathbf{Z}}$ *is a congruence.*

Proof. We have $\mathscr{D}(\sim^{\otimes}_{\mathbf{Z}}) = \sim^{\vee,\otimes}_{\mathbf{Z}}$ by Proposition 5.6, so we conclude by Lemma 4.25 ◻

So we need to prove that $\sim^{\vee,\otimes}_{\mathbf{Z}}$ is both a subalgebra and a submonoid. For this, we use the following variant of Howe's construction.

Definition 5.8. For any relations R and A on $\mathscr{D}(\mathscr{U}(X))$, let $\mathscr{H}_R(A)$ denote the image (in $\mathscr{C}_0$) of $I + \Sigma_0(A); R$ in X^2, where I is viewed as a relation through the unit and diagonal maps $I \to X \to X^2$ and ; denotes sequential composition of relations. The *Howe closure* $R^{\bullet}$ of any R is the initial $\mathscr{H}_R$-algebra, i.e., the (directed) union $\bigcup_n \mathscr{H}^n_R(\varnothing)$.

Let us first show the following lemma, in case $X = \mathbf{Z}$.

Lemma 5.9. *If* $R \subseteq \mathbf{Z}^2$ *is reflexive, transitive, and a* $\mathbf{Z}$*-module, then (omitting* $im(-)$ *for readability)*

(i) $\mathbf{Z} \subseteq R^{\bullet}$ *(and so* $I \subseteq R^{\bullet}$*);*
(ii) $R^{\bullet}; R \subseteq R^{\bullet}$ *(and so* $R \subseteq R^{\bullet}$ *by (i));*
(iii) $\Sigma_0(R^{\bullet}) \subseteq R^{\bullet}$*;*
(iv) $R^{\bullet} \otimes R^{\bullet} \subseteq R^{\bullet}$ *(and so* $R^{\bullet} \otimes \mathbf{Z} \subseteq R^{\bullet}$ *by (i)).*

Corollary 5.10. *We have* $\mathscr{T}_0(R^{\bullet}); R \subseteq R^{\bullet}$.

We then narrow down the goal as follows, using $(-)^{+}$ to denote transitive closure.

Lemma 5.11. *If* $(\sim^{\vee,\otimes}_{\mathbf{Z}})^{\bullet+}$ *is a prebisimulation, then* $\sim^{\otimes}_{\mathbf{Z}}$ *is a congruence.*

Proof. By hypothesis, $(\sim^{\vee,\otimes}_{\mathbf{Z}})^{\bullet+}$ is a prebisimulation, hence $(\sim^{\vee,\otimes}_{\mathbf{Z}})^{\bullet+} \subseteq \sim^{\vee,\otimes}_{\mathbf{Z}}$. Moreover, the converse also holds by Lemma 5.9*(ii)*, so $(\sim^{\vee,\otimes}_{\mathbf{Z}})^{\bullet+} = \sim^{\vee,\otimes}_{\mathbf{Z}}$. But by Lemma 5.9 and the fact that subalgebras and submonoids are stable under transitive closure, $(\sim^{\vee,\otimes}_{\mathbf{Z}})^{\bullet+}$ is a subalgebra and a submonoid, hence so is $\sim^{\vee,\otimes}_{\mathbf{Z}}$, and we conclude by Lemma 5.7. ◻

This would leave us with the task of delineating hypotheses under which $(\sim^{\vee,\otimes}_{\mathbf{Z}})^{\bullet+}$ is indeed a prebisimulation, but we can narrow this down a bit further (this is sometimes called the transitive closure trick), using the following two intermediate lemmas.

Lemma 5.12. *If* $R \subseteq X^2$ *in* $\mathscr{C}$ *is a* presimulation up to transitivity *[29], i.e.,* $R \rightsquigarrow_l R^{+}$*, then* R^{+} *is a presimulation.*

Lemma 5.13. *If* $R \subseteq \mathbf{Z}^2$ *is an equivalence relation, then so is* $R^{\bullet+}$.

Lemma 5.14. *If* $(\sim^{\vee,\otimes}_{\mathbf{Z}})^{\bullet}$ *is a presimulation, then* $\sim^{\otimes}_{\mathbf{Z}}$ *is a congruence.*

Proof. By Lemma 5.12, $(\sim^{\vee,\otimes}_{\mathbf{Z}})^{\bullet+}$ is a presimulation. But $(\sim^{\vee,\otimes}_{\mathbf{Z}})^{\bullet+}$ is symmetric by Lemma 5.13, so it is in fact a prebisimulation, hence we conclude by Lemma 5.11. ◻

5.4 Familiality and cellularity

In the previous section, we have established that congruence of $\sim^{\otimes}_{\mathbf{Z}}$ will follow from $(\sim^{\vee,\otimes}_{\mathbf{Z}})^{\bullet}$ being a presimulation, and so are now seeking sufficient conditions for this. Our key tool will be cellularity, a special case of familiality. As we have already seen that $\Sigma^{\wedge}_0$ is familial (Proposition 2.6), let us now start by proving that $\Sigma^{\wedge}_1$ is so too.

Example 5.15. $\Sigma^{\wedge}_1$ is familial, and $\Sigma^{\wedge}_0 \to \Sigma^{\wedge}_1$ preserves generics. Using Theorem 2.5, the interesting point is that we have $\Sigma^{\wedge}_1(X)(\Downarrow) \cong [\mathbf{y}_1, X] + [E_\beta, X]$, where E_β denotes the colimit of the diagram

$$\begin{tikzcd}[column sep=small]
 & \mathscr{L}(\mathbf{y}_1) \arrow[dl, "\mathscr{L}(t)"'] \arrow[dr, "\mathscr{L}(in_l)"] & & \mathscr{L}(\mathbf{y}_0) \arrow[dl, "\tilde{\chi}"'] \arrow[dr, "\mathscr{L}(s)"] & \\
\mathscr{L}(\mathbf{y}_{\Downarrow}) & & \mathscr{L}(\mathbf{y}_1 + \mathbf{y}_0) & & \mathscr{L}(\mathbf{y}_{\Downarrow})
\end{tikzcd}$$

with $\tilde{\chi}$ denoting the transpose of the morphism $\chi \colon \mathbf{y}_0 \to \mathscr{U}\mathscr{L}(\mathbf{y}_1 + \mathbf{y}_0)$ corresponding by Yoneda to the element

$$(in_l(id_1))(\!|in_r(id_0)|\!) \in \mathscr{U}\mathscr{L}(\mathbf{y}_1 + \mathbf{y}_0)(0).$$

On the other hand, $\check{\Sigma}_1$ is not familial, because the candidate assignment $\mathrm{el}(\check{\Sigma}_1(1)) \to \mathscr{C}$ is not functorial.

Howe's approach to proving that $(\sim^{\vee,\otimes}_{\mathbf{Z}})^{\bullet}$ is a presimulation is by induction on the given transition. More generally, for any prebisimulation R satisfying the hypotheses of Lemma 5.9, we prove by induction on n that any commuting diagram as the solid part below left

$$\begin{tikzcd}
P \arrow[rrr] \arrow[dd, "s"'] & & & R^{\bullet} \arrow[dl] \arrow[ddr] & \\
 & \check{\Sigma}^n_1(0^{\circledast}) \arrow[r] & \mathbf{Z} & & \\
L \arrow[ur] \arrow[rrrr, dashed] & & & & \mathbf{Z} \\
Q \arrow[u, "t"] \arrow[rrr, dashed] & & & R^{\bullet} \arrow[uul] \arrow[ur] &
\end{tikzcd}
\qquad
\begin{tikzcd}
R^{\bullet}_n \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & R^{\bullet} \arrow[r] \arrow[d] & \mathbf{Z} \\
\check{\Sigma}^n_1(0^{\circledast}) \arrow[r] & \mathbf{Z} &
\end{tikzcd}$$

may be completed as shown. This indeed entails that $R^{\bullet}$ is a presimulation, because $\mathbf{Z}$ is the colimit of all $\check{\Sigma}^n_1(0^{\circledast})$, any morphism $L \to \mathbf{Z}$ factors through some $\check{\Sigma}_1^n(0^{\circledast}) \to \mathbf{Z}$, hence we get $R^\bullet \rightsquigarrow_l R^\bullet$ as desired. But this is equivalent to the relation $R^\bullet_n \hookrightarrow \check{\Sigma}_1^n(0^{\circledast}) \times \mathbf{Z}$ obtained by pullback above right is a presimulation, so we want to prove by induction that each $R^\bullet_n$ is a presimulation.

Let us isolate the interesting part of this inductive proof:

Lemma 5.16. *For all $n > 0$, $R^\bullet_n$ is a presimulation if, for all commuting diagrams*

$$\begin{tikzcd} P \arrow[r] \arrow[d] & L \arrow[d] & Q \arrow[l] \arrow[d] \\ \Sigma_1(A) \arrow[r, "\Sigma_1(k)"'] & \Sigma_1(B) & \Sigma_1(C), \arrow[l, "\Sigma_1(l)"] \end{tikzcd} \tag{18}$$

where $\to$ denotes generic morphisms, k and l are such that any diagram of the form below left may be completed as shown.

$$\begin{tikzcd} A \arrow[rr] \arrow[d, "k"'] & & R^\bullet_{n-1} \arrow[dl] \arrow[dr] & \\ B \arrow[r] \arrow[rrr, dashed] & \check{\Sigma}_1^{n-1}(0^{\circledast}) & & \mathbf{Z} \\ C \arrow[u, "l"] \arrow[rr, dashed] & & R^\bullet_{n-1} \arrow[ul] \arrow[ur] & \end{tikzcd} \qquad \begin{tikzcd} A \arrow[rr] \arrow[d, "k"'] & & R \arrow[dl] \arrow[dr] & \\ B \arrow[r] \arrow[rrr, dashed] & X & & Y \\ C \arrow[u, "l"] \arrow[rr, dashed] & & R \arrow[ul] \arrow[ur] & \end{tikzcd} \tag{19}$$

By induction hypothesis, we know that $R^\bullet_{n-1}$ is a presimulation, but the problem is that (k, l) may not be of the form (s, t), and will rarely be so in practice. This is precisely where cellularity comes into play.

Definition 5.17. A *copresimulation* is any cospan (k, l) such that for any presimulation $R \hookrightarrow X \times Y$, any square as in (19, right) may be completed as shown.

The functor Σ_1 is *cospan-cellular* iff it is familial and, for all commuting diagrams (18) with generic vertical arrows, the cospan (k, l) is a copresimulation.

If Σ_1 is cospan-cellular, then by induction hypothesis $R^\bullet_{n-1}$ is a presimulation, so (19, left) may be completed as shown, so by Lemma 5.16 $R^\bullet_n$ is a presimulation. We have proved:

Lemma 5.18. *If Σ_1 is cospan-cellular and $R \subseteq \mathbf{Z}^2$ is a prebisimulation satisfying the hypotheses of Lemma 5.9, the relation $R^\bullet$ is a presimulation.*

Thus, by Lemma 5.14, we obtain:

Theorem 5.19. *For any vertically finitary, layered 2-signature (Σ_0, Σ_1) with Σ_1 cospan-cellular, substitution-closed bisimilarity $\sim^{\otimes}_{\mathbf{Z}}$ on the initial vertical algebra is a congruence.*

A mysterious bit remains: what does cospan-cellularity mean? In order to demonstrate that our running example Σ_1^Λ is cospan-cellular, we first need to develop some basic results about copresimulations.

Lemma 5.20. *Copresimulations contain the basic cospans (s, t) and identity cospans. Moreover, they are closed under pointwise coproduct, precomposition of their right-hand leg with any morphism, and cospan composition.*

Example 5.21. Recalling Example 5.15, we need to show that the cospan $\mathscr{L}(\mathbf{y}_0 + \mathbf{y}_0) \to E_\beta \leftarrow \mathscr{L}(\mathbf{y}_1)$ corresponding to the β-rule is a copresimulation, which holds by Lemma 5.20, as it is the composite of

$$\mathscr{L}(\mathbf{y}_0 + \mathbf{y}_0) \to \mathscr{L}(\mathbf{y}_{\Downarrow} + \mathbf{y}_0) \leftarrow \mathscr{L}(\mathbf{y}_1 + \mathbf{y}_0) \xleftarrow{\tilde{X}} \mathscr{L}(\mathbf{y}_0)$$

and the basic cospan $\mathscr{L}(\mathbf{y}_0 \to \mathbf{y}_{\Downarrow} \leftarrow \mathbf{y}_1)$.

Example 5.22. The call-by-value λ-calculus may be treated similarly, with the same Σ_0, using the same rule for values (5, right), but changing rule (5, left) to

$$\frac{e_1 \Downarrow e_1' \qquad e_2 \Downarrow e_2' \qquad e_1'[\lambda.e_2'] \Downarrow e_3}{e_1\ e_2 \Downarrow e_3}.$$

The induced functor Σ_1 is cospan-cellular by Lemma 5.20, since the cospan (18) for the new rule may be obtained by starting with the pointwise coproduct

$$\mathscr{L}(2 \cdot \mathbf{y}_0) \xrightarrow{\mathscr{L}(2\cdot s)} \mathscr{L}(2 \cdot \mathbf{y}_{\Downarrow}) \xleftarrow{\mathscr{L}(2\cdot t)} \mathscr{L}(2 \cdot \mathbf{y}_1)$$

(where $n \cdot c$ denotes the n-fold coproduct $c + \ldots + c$), precomposing its right-hand leg with

$$\mathscr{L}(\mathbf{y}_0) \xrightarrow{\tilde{X}} \mathscr{L}(\mathbf{y}_1) + \mathscr{L}(\mathbf{y}_0) \xrightarrow{\mathscr{L}(\mathbf{y}_1) + \tilde{\lambda}} 2 \cdot \mathscr{L}(\mathbf{y}_1) \cong \mathscr{L}(2 \cdot \mathbf{y}_1),$$

and then composing with the basic cospan.

6 Conclusion and perspectives

We have presented an abstract framework, 2-signatures on Howe contexts, in which, under suitable hypotheses, substitution-closed bisimilarity is a congruence. We have proved new initiality and familiality results along the way, notably an adaptation of work by Fiore and collaborators to the skew-monoidal case. Finally, we have covered the basic examples of call-by-name and call-by-value variants of big-step λ-calculus. Future research directions include trying to generalise our framework to cover variants of Howe's method which currently seem to lie beyond its scope, e.g., so-called early-style bisimilarity in higher-order π-calculus or calculi with passivation [24]. Similarly, we plan to try and apply our techniques to variants of applicative bisimilarity, e.g., open bisimilarity [23].

Acknowledgments

Thanks to Marcelo Fiore, Richard Garner, Shin-ya Katsumata, Sergueï Lenglet, Alan Schmitt, Pawel Sobocinski, Sam Staton, and the anonymous referees for useful input.

This research has partly been funded by the CoqHoTT ERC Grant 637339.

References

[1] Martín Abadi, Luca Cardelli, Pierre-Louis Curien, and Jean-Jacques Lévy. 1990. Explicit Substitutions. In *Proc. 17th International Symposium on Principles of Programming Languages*, Frances E. Allen (Ed.). ACM, 31–46. https://doi.org/10.1145/96709.96712

[2] Samson Abramsky. 1990. The lazy lambda calculus. In *Research Topics in Functional Programming*, D. A. Turner (Ed.). Addison–Wesley.

[3] Benedikt Ahrens, André Hirschowitz, Ambroise Lafont, and Marco Maggesi. 2019. Reduction monads and their signatures. *PACMPL* 4, POPL (2019). https://doi.org/10.1145/3371099

[4] Thorsten Altenkirch, James Chapman, and Tarmo Uustalu. 2015. Monads need not be endofunctors. *Logical Methods in Computer Science* 11, 1 (2015). https://doi.org/10.2168/LMCS-11(1:3)2015

[5] Martin Bodin, Philippa Gardner, Thomas Jensen, and Alan Schmitt. 2019. Skeletal semantics and their interpretations. *PACMPL* 3, POPL (2019), 44:1–44:31. https://doi.org/10.1145/3290357

[6] Filippo Bonchi, Daniela Petrisan, Damien Pous, and Jurriaan Rot. 2014. Coinduction up-to in a fibrational setting. In *Proc. 29th Symposium on Logic in Computer Science* ACM, 20:1–20:9. https://doi.org/10.1145/2603088.2603149

[7] Aurelio Carboni and Peter Johnstone. 1995. Connected Limits, Familial Representability and Artin Glueing. *Mathematical Structures in Computer Science* 5, 4 (1995), 441–459. https://doi.org/10.1017/S0960129500001183

[8] Aurelio Carboni, Stephen Lack, and Robert F. C. Walters. 1993. Introduction to extensive and distributive categories. *Journal of Pure and Applied Algebra* 84, 2 (1993).

[9] Andrea Corradini, Reiko Heckel, and Ugo Montanari. 2002. Compositional SOS and beyond: a coalgebraic view of open systems. *Theoretical Computer Science* 280, 1-2 (2002), 163–192. https://doi.org/10.1016/S0304-3975(01)00025-1

[10] Yves Diers. 1978. Spectres et localisations relatifs à un foncteur. *Comptes rendus hebdomadaires des séances de l'Académie des sciences* 287, 15 (1978), 985–988.

[11] Marcelo Fiore and Chung-Kil Hur. 2009. On the construction of free algebras for equational systems. *Theoretical Computer Science* 410 (2009), 1704–1729.

[12] Marcelo Fiore, Gordon Plotkin, and Daniele Turi. 1999. Abstract Syntax and Variable Binding. In *Proc. 14th Symposium on Logic in Computer Science* IEEE.

[13] Marcelo Fiore and Philip Saville. 2017. List Objects with Algebraic Structure. In *Proc. 2nd International Conference on Formal Structures for Computation and Deduction (LIPIcs)*, Dale Miller (Ed.), Vol. 84. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 16:1–16:18. https://doi.org/10.4230/LIPIcs.FSCD.2017.16

[14] Marcelo Fiore and Philip Saville. 2019. List Objects with Algebraic Structure. (2019). Long version of [13].

[15] Marcelo P. Fiore. 2008. Second-Order and Dependently-Sorted Abstract Syntax. In *Proc. 23rd Symposium on Logic in Computer Science* IEEE, 57–68. https://doi.org/10.1109/LICS.2008.38

[16] Richard Garner and Tom Hirschowitz. 2018. Shapely monads and analytic functors. *Journal of Logic and Computation* 28, 1 (2018), 33–83. https://doi.org/10.1093/logcom/exx029

[17] Makoto Hamana. 2004. Free S-Monoids: A Higher-Order Syntax with Metavariables *(Lecture Notes in Computer Science)*, Wei-Ngan Chin (Ed.), Vol. 3302. Springer, 348–363. https://doi.org/10.1007/978-3-540-30477-7_23

[18] André Hirschowitz, Tom Hirschowitz, and Ambroise Lafont. 2019. Modules over monads and operational semantics. (2019). https://hal.archives-ouvertes.fr/hal-02338144 preprint.

[19] Tom Hirschowitz. 2019. Familial monads and structural operational semantics. *PACMPL* 3, POPL (2019), 21:1–21:28. https://dl.acm.org/citation.cfm?id=3290334

[20] André Joyal, Mogens Nielsen, and Glynn Winskel. 1993. Bisimulation and open maps. In *Proc. 8th Symposium on Logic in Computer Science* IEEE, 418–427. https://doi.org/10.1109/LICS.1993.287566

[21] Stephen Lack and Ross Street. 2014. On monads and warpings. *Cahiers de Topologie et Géométrie Différentielle Catégoriques* LV, 4 (2014), 244–266.

[22] Stephen Lack and Ross Street. 2015. Skew-monoidal reflection and lifting theorems. *Theory and Applications of Categories* 30, 28 (2015), 985–1000.

[23] S. Lassen. 1999. Bisimulation in Untyped Lambda Calculus: Böhm Trees and Bisimulation up to Context. In *Proc. 15th Conference on Mathematical Foundations of Progamming Semantics, MFPS 1999, Tulane University, New Orleans, LA, USA, April 28 - May 1, 1999 (Electronic Notes in Theoretical Computer Science)*, Stephen D. Brookes, Achim Jung, Michael W. Mislove, and Andre Scedrov (Eds.), Vol. 20. Elsevier, 346–374. https://doi.org/10.1016/S1571-0661(04)80083-5

[24] Sergueï Lenglet and Alan Schmitt. 2015. Howe's Method for Contextual Semantics. In *Proc. 26th International Conference on Concurrency Theory*, Luca Aceto and David de Frutos-Escrig (Eds.), Vol. 42. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 212–225. https://doi.org/10.4230/LIPIcs.CONCUR.2015.212

[25] Saunders Mac Lane and Ieke Moerdijk. 1992. *Sheaves in Geometry and Logic: A First Introduction to Topos Theory.* Springer.

[26] Robert Paré. 1990. Simply connected limits. *Canadian Journal of Mathematics* 42 (1990), 731–746.

[27] Andrew M. Pitts. 2011. *Howe's method for higher-order languages*, Chapter 5. In Sangiorgi and Rutten [30].

[28] Gordon D. Plotkin. 1981. *A structural approach to operational semantics.* DAIMI Report FN-19. Computer Science Department, Aarhus University.

[29] Damien Pous and Davide Sangiorgi. 2011. *Enhancements of the bisimulation proof method*, Chapter 6. In Sangiorgi and Rutten [30].

[30] Davide Sangiorgi and Jan Rutten (Eds.). 2011. *Advanced Topics in Bisimulation and Coinduction.* Number 52 in Cambridge Tracts in Theoretical Computer Science. Cambridge University Press.

[31] Sam Staton. 2008. General Structural Operational Semantics through Categorical Logic. In *Proc. 23rd Symposium on Logic in Computer Science* 166–177. https://doi.org/10.1109/LICS.2008.43

[32] Kornel Szlachányi. 2012. Skew-monoidal categories and bialgebroids. *Advances in Mathematics* 231 (2012), 1694–1730. https://doi.org/10.1016/j.aim.2012.06.027

[33] Daniele Turi and Gordon D. Plotkin. 1997. Towards a Mathematical Operational Semantics. In *Proc. 12th Symposium on Logic in Computer Science* 280–291. https://doi.org/10.1109/LICS.1997.614955

[34] Vladimir Voevodsky, Benedikt Ahrens, Daniel Grayson, et al. 2019. UniMath — a computer-checked library of univalent mathematics. https://github.com/UniMath/UniMath

[35] Mark Weber. 2004. Generic morphisms, parametric representations and weakly cartesian monads. *Theory and Applications of Categories* 13 (2004), 191–234.

[36] Mark Weber. 2007. Familial 2-functors and parametric right adjoints. *Theory and Applications of Categories* 18, 22 (2007), 665–732.