HAL open science

A categorical framework for congruence of applicative bisimilarity in higher-order languages

Tom Hirschowitz, Ambroise Lafont

To cite this version:

Tom Hirschowitz, Ambroise Lafont. A categorical framework for congruence of applicative bisimilarity in higher-order languages. 2022. hal-02966439v5

HAL Id: hal-02966439

https://hal.science/hal-02966439v5

Preprint submitted on 10 Feb 2022 (v5), last revised 14 Jun 2022 (v6)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A CATEGORICAL FRAMEWORK FOR CONGRUENCE OF APPLICATIVE BISIMILARITY IN HIGHER-ORDER LANGUAGES

TOM HIRSCHOWITZ AND AMBROISE LAFONT

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

UNSW, Sydney, Australia

Abstract. Applicative bisimilarity is a coinductive characterisation of observational equivalence in call-by-name lambda-calculus, introduced by Abramsky (1990). Howe (1996) gave a direct proof that it is a congruence, and generalised the result to all languages complying with a suitable format. We propose a categorical framework for specifying operational semantics, in which we prove that (an abstract analogue of) applicative bisimilarity is automatically a congruence. Example instances include standard applicative bisimilarity in call-by-name, call-by-value, and call-by-name non-deterministic λ-calculus, and more generally all languages complying with a variant of Howe's format.

1. Introduction

1.1. **Motivation.** This paper is a contribution to the search for efficient and high-level mathematical tools to specify and reason about programming languages. This search arguably goes back at least to Turi and Plotkin [54], who coined the name "Mathematical Operational Semantics", and proved a general congruence theorem for bisimilarity. This approach has been deeply investigated, notably for quantitative languages [9]. However, as of today, attempts to apply it to higher-order (e.g., functional) languages have failed.

In previous work [33, 32], the first author has proposed an alternative approach to the problem, dropping the coalgebraic notion of bisimulation used by Turi and Plotkin in favour of a notion based on factorisation systems, similar to Joyal et al.'s [36]. Furthermore, congruence of bisimilarity is notably obtained by assuming that syntax induces a *familial* monad, in the sense of Diers [17, 15, 56].

However, the new approach has only been applied to simple, first-order languages like the π-calculus [44, 51], and Positive GSOS specifications [12]. In this paper, we extend it to functional languages, notably covering the paradigmatic case of *applicative* bisimilarity [1] in call-by-name and call-by-value λ-calculus, as well as in a simple, non-deterministic λ-calculus [50, 35, §7]. We even show that our framework subsumes the general, syntactic format proposed by Howe [35, Lemma 6.1]. We thus obtain for the first time a generic, categorical congruence result for applicative bisimilarity in functional languages.

Key words and phrases: Programming languages; categorical semantics; operational semantics; Howe's method.

LOGICAL METHODS
IN COMPUTER SCIENCE

DOI:10.2168/LMCS-???

© T. Hirschowitz and A. Lafont
Creative Commons

1.2. **Overview.** A bit more precisely:

- We propose a simple notion of signature for programming languages.
- Each signature has a category of models, including an initial one, intuitively its operational semantics.
- An abstract analogue of applicative bisimilarity, called *substitution-closed* bisimilarity, may be defined in any model, and in particular in the initial one.
- Under suitable hypotheses, we show that substitution-closed bisimilarity is a congruence in the initial model.

Categorically, this unfolds as follows.

(i) We define an abstract notion of (labelled) transition systems, as objects of a category $\mathbf{C}$, in such a way that

- there is a forgetful functor $\mathbf{C} \to \mathbf{C}_0$, intuitively returning the (potentially structured) set of states of a transition system;
- bisimulation and bisimilarity may be defined for any transition system.

(ii) Adopting Fiore, Plotkin, and Turi's seminal framework [24, 23], we then assume that $\mathbf{C}_0$ is monoidal, and define models of the syntax to be *monoid algebras* for a given *pointed strong* endofunctor Σ_0 on $\mathbf{C}_0$. Monoid algebras, a.k.a. Σ_0-monoids, are Σ_0-algebras equipped with compatible monoid structure, which models capture-avoiding substitution. The category Σ_0-**Mon** of Σ_0-monoids has an initial object $\mathbf{Z}_0$, whose carrier is the free Σ_0-algebra on the monoidal unit I, as we prove in Coq [38]. In all examples, $\mathbf{Z}_0$ is precisely the syntax.

(iii) This category Σ_0-**Mon** induces by pullback a category Σ_0-**Trans** of transition systems whose states are equipped with Σ_0-monoid structure. We call these *transition monoid algebras*, or *transition* Σ_0*-monoids*, or even simply *transition monoids* when Σ_0 is the identity. The relevant notions of bisimulation and bisimilarity for such objects are defined as in *(i)*, but for *substitution-closed* relations.

(iv) We then define models of the dynamics to be certain algebras, called *vertical*, for an endofunctor on Σ_0-**Trans**. There is an initial vertical algebra $\mathbf{Z}$, which in examples is the syntactic transition system. (Furthermore, standard applicative bisimilarity coincides with substitution-closed bisimilarity.)

(v) Finally, following an abstract analogue of Howe's method, we show that, under suitable hypotheses, substitution-closed bisimilarity on $\mathbf{Z}$ is a congruence. One crucial hypothesis is *cellularity*, in a sense closely related to [25].

1.3. **Related work.** Plotkin and Turi's *bialgebraic semantics* [54] and its few variants [16, 52] prove abstract congruence theorems for bisimilarity. However, they do not cover higher-order languages like the λ-calculus, let alone applicative bisimilarity. This was one of the main motivations for our work. Among more recent work, quite some inspiration was drawn from Ahrens et al. [5, 31], notably in the use of vertical algebras. However, a difference is that we do not insist that transitions be stable under substitution. In a different direction, Dal Lago et al. [40] prove a general congruence theorem for applicative bisimilarity, for a λ-calculus with algebraic effects. As briefly discussed in the conclusion, our framework does not yet account for such results. However, it places the generality in a different direction: namely, it is not tied to any particular language (like the λ-calculus in [40]). It would of course be useful to find a common generalisation.

Links with other relevant work by, e.g., Bodin et al. [13], 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 [33]. It would be instructive to better understand potential links between the two. Finally, let us mention recent work which, just like ours, strives to establish abstract versions of standard constructions and theorems in programming language theory like type soundness [6] or gluing [18, 21, 22].

1.4. **Relation to conference version.** This paper is a bit more than a journal version of our previous work [14]. Here is a brief summary of changes.

(a) In [14], we work with a non-trivial generalisation of monoid algebras to *skew monoidal* categories [53] and *structurally strong* functors. Here, by giving a better type to Σ_1, the endofunctor for specifying the dynamics, we manage to work with standard monoid algebras. This has the additional advantages of

- avoiding a slightly *ad hoc* compositionality assumption of [14], and
- relaxing the requirement that the tensor product should be familial.

(b) In [14], because Howe's closure operates only at the level of states, we work mostly with *prebisimulations*, in the sense of [33, §5.1]. This notion is designed to detect when the state part of a relation underlies a bisimulation, regardless of what it does on transitions. However, it feels more *ad hoc* than the standard definition of bisimulation by lifting [36]. In this paper, we extend Howe's closure to transitions, thus avoiding prebisimulations entirely.

(c) In [14], we rely on directed unions of relations, which leads to quite a few, rather painful proofs by induction. Here, we use higher-level methods to construct Howe's closure, essentially through categorification and algebraicisation. Namely:

(1) We define bisimilarity as the final object not in some partially-ordered set of relations as usual, but in some category of spans (see also [10]).
(2) Furthermore, we define Howe's closure directly as a free monoid algebra for a suitable pointed strong endofunctor on spans.
(3) More generally, we systematically rely on universal properties, which simplifies a significant number of proofs.

(d) We put less emphasis on cellularity, viewing it only as a sufficient condition for a perhaps more natural hypothesis, which already appeared in a slightly different form in [52], namely the fact that Σ_1 preserves functional bisimulations.

(e) We obtain a congruence theorem of similar scope (Theorem 6.15), and cover three new, detailed applications (§8): call-by-value, big-step λ-calculus (which was covered but too naively in [14], as we explain), a call-by-name λ-calculus with unary, erratic choice from [50, §7], and a general format proposed by Howe [35, Lemma 6.1].

(f) We fill a gap in the proof of [14, Lemma 5.13], by requiring the endofunctor Σ_0 for specifying the dynamics to preserve sifted colimits (see Remark 5.13).

1.5. **Plan.** In §2, we start by briefly recalling call-by-name λ-calculus and applicative bisimilarity. We then explain how to view the latter as substitution-closed bisimilarity, and sketch Howe's method. In §3, we then give a brief overview of the new framework by example, including a recap on monoid algebras and a statement of the main theorem (in the considered case). We then dive into the technical core of the paper by presenting our framework for transition systems and bisimilarity (§4), operational semantics (§5), and then substitution-closed bisimilarity and the main result (Theorem 6.15), together with a high-level proof sketch (§6). In §7, we reformulate the main hypothesis of Theorem 6.15 using cellularity,

which allows us to use well-known results from weak factorisation systems as sufficient conditions. We then apply our results to examples in §8. The full proof of Theorem 6.15 is given in §9. Finally, we conclude and give some perspectives on future work in §10.

1.6. **Notation and preliminaries.** In this subsection, we fix some basic notation, and review some preliminaries.

1.6.1. *Basic notation.* We often conflate natural numbers $n \in \mathbb{N}$ with the corresponding sets $\{1, \dots, n\}$. For all sets X and objects C of a given category, we denote by $X \cdot C$ the X-fold coproduct of C with itself, i.e., $\sum_{x \in X} C$. Let **Gph** denote the category of (directed, multi) graphs, **Cat** the category of small categories, and **CAT** the category of locally small categories.

1.6.2. *Comma categories and lax limits.* Given functors $F \colon \mathbf{A} \to \mathbf{C}$ and $G \colon \mathbf{B} \to \mathbf{C}$, the *comma* category F/G has

- as objects all triples (A, B, φ), where $A \in \mathbf{A}$, $B \in \mathbf{B}$, and $\varphi \colon F(A) \to G(B)$, and
- as morphisms $(A, B, \varphi) \to (A', B', \varphi')$ all pairs of morphisms $u \colon A \to A'$ and $v \colon B \to B'$ making the following square commute.

$$\begin{tikzcd} F(A) \arrow[r, "F(u)"] \arrow[d, "\varphi"'] & F(A') \arrow[d, "\varphi'"] \\ G(B) \arrow[r, "G(v)"'] & G(B') \end{tikzcd}$$

We have the following well-known fact:

Proposition 1.1. *If* $\mathbf{A}$ *and* $\mathbf{B}$ *have, and* F *preserves colimits of any given shape, then the projection functor* $F/G \to \mathbf{A} \times \mathbf{B}$ *creates them.*

Symmetrically, if $\mathbf{A}$ *and* $\mathbf{B}$ *have, and* G *preserves limits of any given shape, then the projection functor* $F/G \to \mathbf{A} \times \mathbf{B}$ *creates them.*

The comma category F/G is well-known [37, 57] to be the universal category equipped with projections to $\mathbf{A}$ and $\mathbf{B}$ and a natural transformation as in the following diagram.

$$\begin{tikzcd} F/G \arrow[r] \arrow[d] \arrow[dr, phantom, "\Nearrow"] & \mathbf{B} \arrow[d, "G"] \\ \mathbf{A} \arrow[r, "F"'] & \mathbf{C} \end{tikzcd}$$

Kelly [37] explains that the comma category is a kind of lax limit of F and G. When F is an identity, we call the comma category a *lax limit* of G.

1.6.3. *Presheaves.* Let $\widehat{\mathbb{C}}$ denote the category of (contravariant) presheaves on $\mathbb{C}$, and $\mathbf{y} \colon \mathbb{C} \to \widehat{\mathbb{C}}$ the Yoneda embedding, mapping c to $\mathbb{C}(-, c)$. Given a presheaf $F \in \widehat{\mathbb{C}}$, an element $x \in F(c)$, and a morphism $c \xrightarrow{f} c'$, we sometimes denote $F(f)(x)$ by $x \cdot f$. Given two categories $\mathbb{C}_1$ and $\mathbb{C}_2$, we denote by $[\mathbb{C}_1, \mathbb{C}_2]$ the functor category between them.

1.6.4. *Spans and relations.* In a category $\mathbf{C}$ with binary products, we interchangeably use spans $X \leftarrow R \rightarrow Y$ and their pairings $R \rightarrow X \times Y$, sometimes also calling the latter spans. Spans from X to Y are the objects of a category $Span(\mathbf{C})(X,Y)$, which is isomorphic to the slice category $\mathbf{C}/X \times Y$ in the presence of binary products. When $\mathbf{C}$ has pullbacks, these categories are the hom-categories of a bicategory [11] $Span(\mathbf{C})$, in which composition of morphisms is given by pullback. A *relation* from X to Y is merely a span whose pairing $R \rightarrow X \times Y$ is monic.

1.6.5. *Images.* Let us now recall a few elements about images.

Definition 1.2. An *image* of a morphism $f\colon A \rightarrow B$ is a factorisation $A \xrightarrow{e} M \overset{m}{\hookrightarrow} B$ with m a monomorphism, which is initial in the sense that for any factorisation $A \xrightarrow{e'} M' \overset{m'}{\hookrightarrow} B$ there is a (unique by monicness) morphism $k\colon M \rightarrow M'$ making both triangles commute in the following diagram.

$$\begin{tikzcd} A \arrow[r, "e'"] \arrow[d, "e"'] & M' \arrow[d, "m'"] \\ M \arrow[r, "m"'] \arrow[ur, dashed, "k"] & B \end{tikzcd}$$

Definition 1.3. A *strong epimorphism* is a morphism with the strong left lifting property w.r.t. all monomorphisms, i.e., a morphism $e\colon A \rightarrow B$ such that for all (solid) commuting squares

$$\begin{tikzcd} A \arrow[r] \arrow[d, "e"'] & X \arrow[d, "m"] \\ B \arrow[r] \arrow[ur, dashed, "k"] & Y \end{tikzcd}$$

with m monic there exists a unique lifting k making both triangles commute.

The terminology is justified by the following result.

Lemma 1.4. *In a category with equalisers, any strong epimorphism is an epimorphism.*

Proof. Let us assume that $e\colon A \rightarrow B$ is a strong epi and $fe = ge$, with $f, g\colon B \rightarrow C$. Then, let $k\colon A' \rightarrow B$ denote the equaliser of f and g. Because e equalises f and g, it factors as kh, for some unique $h\colon A \rightarrow A'$. But now k is monic, so by lifting there is a unique l making both triangles commute in the following diagram.

$$\begin{tikzcd} A \arrow[r, "h"] \arrow[d, "e"'] & A' \arrow[d, "k"] \\ B \arrow[r, equal] \arrow[ur, dashed, "l"] & B \end{tikzcd}$$

We thus have

$$f = f \circ \mathrm{id}_B = f \circ k \circ l = g \circ k \circ l = g \circ \mathrm{id}_B = g.$$

The morphism e is thus epi, as claimed. □

Corollary 1.5. *Factoring a morphism as a strong epi followed by a mono yields an image.*

Proof. Initiality is directly given by the lifting property. □

Proposition 1.6. *In any locally finitely presentable category, images always exist, and may be computed as (strong epi, mono)-factorisations.*

Proof. This is (part of) [2, Proposition 1.61]. □

Let us finally observe:

Proposition 1.7. *In locally presentable category, unions of subobjects exist, and may be computed by taking the cotupling of all considered subobjects, and then their (strong epi, mono)-factorisation.*

Proof. Straightforward. □

1.6.6. *Initial algebras.* Any finitary endofunctor F on any cocomplete category admits by [49, Theorem 2.1] an initial algebra, which we denote by $\mathbf{Z}_F$. Although this is detailed below, we prefer to avoid confusion and warn the reader that we also use $\mathbf{Z}_F$ for the initial F-monoid, for any *pointed strong* endofunctor F on any nice monoidal category (which is incidentally the initial $(I+F)$-algebra). Throughout the paper, when not explicitly attached to any F, $\mathbf{Z}$ is shorthand for $\mathbf{Z}_{\check{\Sigma}_1}$ (see, e.g., Proposition 3.21 or Theorem 5.18).

1.6.7. *Weak factorisation systems.* Finally, let us fix some notation about weak factorisation systems. In any category $\mathscr{C}$, we say that a morphism $f\colon A \to B$ has the (weak) left lifting property w.r.t. $g\colon C \to D$ when for all commuting squares as on the right there is a lifting k as shown that makes both triangles commute. Equivalently, we say that g has the right lifting property w.r.t. f, and write $f \pitchfork g$. Given a fixed set $\mathbb{J}$ of morphisms, the set of morphisms g such that $j \pitchfork g$ for all $j \in \mathbb{J}$ is denoted by $\mathbb{J}^{\pitchfork}$. Similarly, the set of morphisms f such that $f \pitchfork j$ for all $j \in \mathbb{J}$ is denoted by ${}^{\pitchfork}\mathbb{J}$. In particular, if $f \in {}^{\pitchfork}(\mathbb{J}^{\pitchfork})$ and $g \in \mathbb{J}^{\pitchfork}$, then $f \pitchfork g$. If $\mathscr{C}$ is locally presentable [2], then $({}^{\pitchfork}(\mathbb{J}^{\pitchfork}), \mathbb{J}^{\pitchfork})$ forms a *weak factorisation system*, in the sense that additionally any morphism $f\colon X \to Y$ factors as $X \xrightarrow{l} Z \xrightarrow{r} Y$ with $l \in {}^{\pitchfork}(\mathbb{J}^{\pitchfork})$ and $r \in \mathbb{J}^{\pitchfork}$ (see [34, Theorem 2.1.14]). Morphisms in $\mathbb{J}^{\pitchfork}$ are generically called *fibrations*, while morphisms in ${}^{\pitchfork}(\mathbb{J}^{\pitchfork})$ are called *cofibrations*.

$$\begin{tikzcd} A \arrow[r, "u"] \arrow[d, "f"'] & C \arrow[d, "g"] \\ B \arrow[r, "v"'] \arrow[ur, dashed, "k"] & D \end{tikzcd}$$

Let us conclude with the following easy, yet helpful result.

Lemma 1.8. *For any locally finitely presentable category and set $\mathbb{J}$ of maps therein, if the domains and codomains of maps in $\mathbb{J}$ are finitely presentable, then fibrations are closed under filtered colimits in the arrow category.*

Proof. Let us consider any given filtered diagram $(f_i\colon A_i \to B_i)_{i\in\mathbb{D}}$ of fibrations, and a colimit $f\colon A \to B$ in the arrow category, say $\mathbf{C}^{\to}$. We must show that $j \pitchfork f$ for all $j \in \mathbb{J}$. Let us thus consider any given commuting square

$$\begin{tikzcd} X \arrow[r, "u"] \arrow[d, "j"'] & A \arrow[d, "f"] \\ Y \arrow[r, "v"'] & B \end{tikzcd}$$

with $j \in \mathbb{J}$. Colimits in the arrow category are pointwise, so $A = \operatorname{colim}_i A_i$ and $B = \operatorname{colim}_i B_i$. Thus, by finite presentability of X and Y, and by filteredness of the diagram $\mathbb{D} \to \mathbf{C}$, u and v factor through some A_{i_0} and B_{i_1}, respectively. By filteredness of the diagram again, w.l.o.g., we may take $i_0 = i_1$, such that $(u, v) \colon j \to f$ factors through f_{i_0}. But because f_{i_0} is a fibration, we find a lifting as in

$$\begin{tikzcd}
X \arrow[r, "u'"] \arrow[rr, bend left, "u"] \arrow[d, "j"'] & A_{i_0} \arrow[r] \arrow[d, "f_{i_0}"] & A \arrow[d, "f"] \\
Y \arrow[r, "v'"] \arrow[ur, dashed] \arrow[rr, bend right, "v"'] & B_{i_0} \arrow[r] & B,
\end{tikzcd}$$

which provides the desired lifting for the original square. □

2. A brief review of Howe's method

2.1. **Applicative bisimilarity.** Let us consider the standard, big-step presentation of call-by-name λ-calculus:

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

Standardly, the evaluation relation $\Downarrow$ is considered between closed terms only.

Applicative bisimilarity is an important notion of program equivalence in this language. Indeed, it is coinductive, so one may prove that any two given programs are applicative bisimilar merely by exhibiting an applicative bisimulation. Furthermore, it is sound and complete w.r.t. (i.e., it coincides with) standard contextual equivalence.

Let us briefly recall the definition. Applicative bisimilarity is standardly introduced in two stages, which we now recall.

Definition 2.1 ([1, Definition 2.3])**.** A relation R over closed λ-terms is an *applicative simulation* iff $e_1\ R\ e_2$ and $e_1 \Downarrow \lambda x.e_1'$ entail the existence of e_2' such that $e_2 \Downarrow \lambda x.e_2'$ and, for all terms e, $e_1'[x \mapsto e]\ R\ e_2'[x \mapsto e]$.

An *applicative bisimulation* is an applicative simulation R whose converse, say $R^\dagger$, is also an applicative simulation.

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

Let us readily notice the following alternative characterisation of open extension.

Definition 2.3. A relation S on open terms is *substitution-closed* iff for all $e\ S\ e'$ and (potentially open) substitutions σ, we have $e[\sigma]\ S\ e'[\sigma]$.

Lemma 2.4. *The open extension of any relation R is the greatest* substitution-closed *relation contained in R on closed terms.*

Proof. Let us first show that $R^\otimes$ is substitution-closed. For any $e_1 \; R^\otimes \; e_2$ and σ, we want to show $e_1[\sigma] \; R^\otimes \; e_2[\sigma]$. For this, we in turn need to show that for all closing substitutions γ, we have $e_1[\sigma][\gamma] \; R \; e_2[\sigma][\gamma]$. But $e_i[\sigma][\gamma] = e_i[\sigma[\gamma]]$, where by definition and $\sigma[\gamma](x) = \sigma(x)[\gamma]$. Furthermore, $\sigma[\gamma]$ is closing. So, because we have $e_1 \; R^\otimes \; e_2$, by definition of open extension, we get $e_1[\sigma][\gamma] \; R \; e_2[\sigma][\gamma]$ as desired.

Let us now show that $R^\otimes$ is the greatest substitution-closed relation contained in R on closed terms. For this, consider any substitution-closed R' contained in R on closed terms: for all $e \; R' \; e'$, by substitution-closedness, we have $e[\sigma] \; R' \; e'[\sigma]$ for all closing σ. So because R' is contained in R on closed terms, we further have $e[\sigma] \; R \; e'[\sigma]$. This proves $e \; R^\otimes \; e'$, and thus $R' \subseteq R^\otimes$ as desired. □

The result we wish to abstract over is the following (see [47] for a historical account).

Theorem 2.5. *The open extension $\sim^\otimes$ of applicative bisimilarity is a congruence: it is an equivalence relation, and furthermore it is context-closed, i.e.,*

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

Proving that $\sim^\otimes$ is an equivalence relation is in fact straightforward. In the following, we focus on the context-closedness property.

2.2. **Howe's method.** Howe's method for proving Theorem 2.5 consists in considering a suitable relation $\sim^\bullet$, closed under substitution and context, and containing $\sim^\otimes$ by construction. He then shows that this relation $\sim^\bullet$ is an applicative bisimulation. By maximality of $\sim^\otimes$, we thus also have $\sim^\bullet \subseteq \sim^\otimes$ hence both relations coincide and $\sim^\otimes$ is context-closed as desired. However, as explained in [14, §5.1], the presence of a substitution in the premises of a transition rule seems to require $\sim^\bullet$ to be closed under *heterogeneous* substitution, in the sense that, e.g., if $e_1 \sim^\bullet e_1'$ and $e_2 \sim^\bullet e_2'$ (for open terms), then $e_1[x \mapsto e_2] \sim^\bullet e_1'[x \mapsto e_2']$. The problem is that building this into the definition of $\sim^\bullet$ leads to difficulties in the proof that it is an applicative bisimulation. Howe's workaround consists in requiring $\sim^\bullet$ to be closed under sequential composition with $\sim^\otimes$ from the outset. Coupling this right action with context closedness, he thus defines $\sim^\bullet$ as the smallest context-closed relation satisfying the rules

$$\frac{}{x \sim^\bullet x} \qquad\qquad \frac{e \sim^\bullet e' \qquad e' \sim^\otimes e''}{e \sim^\bullet e''}\,.$$

By construction, $\sim^\bullet$ is reflexive and context-closed. By a simple induction, it also substitution-closed. Furthermore, by reflexivity and the second rule, it also contains $\sim^\otimes$, and finally the second rule clearly entails $\sim^\bullet ; \sim^\otimes \subseteq \sim^\bullet$, where ";" denotes relational (or sequential) composition. It takes an induction to prove stability under heterogeneous substitution, but to give a feel for it, in the basic case where $e_1 \sim^\otimes e_1'$, we have

$$e_1[x \mapsto e_2] \sim^\bullet e_1[x \mapsto e_2'] \sim^\otimes e_1'[x \mapsto e_2']$$

by context closedness of $\sim^\bullet$ and substitution closedness of $\sim^\otimes$, so we conclude by $\sim^\bullet ; \sim^\otimes \subseteq \sim^\bullet$.

The initial plan was to show that $\sim^\bullet$ is an applicative bisimulation and deduce that it coincides with $\sim^\otimes$. It can in fact be slightly optimised by first showing that $\sim^\bullet$ is an

applicative simulation, and then that its transitive closure $(\sim^\bullet)^+$ is symmetric. The relation $(\sim^\bullet)^+$ is also an applicative simulation, hence by symmetry an applicative bisimulation. This entails the last inclusion in the chain $\sim^\otimes \subseteq \sim^\bullet \subseteq (\sim^\bullet)^+ \subseteq \sim^\otimes$, showing that all relations coincide. Finally, because $\sim^\bullet$ is context-closed, so is $\sim^\otimes$, as desired.

2.3. **Non-standard presentation.** The above, standard evaluation rules for call-by-name λ-calculus are not directly compatible with our framework. We thus adopt a slightly different presentation, where the evaluation relation relates closed terms to terms with just one potential free variable. The problem and its solution should become clear in §8.4, where we investigate Howe's general format [35, Lemma 6.1]. There, we show that any language complying with Howe's format may be covered by our framework, up to suitable encoding. The present, non-standard presentation is a slight variant of this encoding, optimised for λ-calculus. The new transition rules are as follows.

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

Here $e_1'[e_2]$ denotes substitution of the unique potential free variable in e_1' by e_2. We will see below that, with this transition system, the essentially standard notion of bisimulation coupled with the substitution-closedness requirement yields applicative bisimilarity.

3. Overview by example

In this section, we describe one particular instance of our framework, which models call-by-name λ-calculus.

3.1. **Syntax.** Let us first define the syntax of λ-calculus, following [24], as an initial[1] object in a suitable category of models. Very roughly, a model of λ-calculus syntax should be something equipped with operations modelling abstraction and application, but also with substitution. Furthermore, certain natural compatibility axioms should be satisfied, e.g.,

$$(e_1\ e_2)[\sigma] = e_1[\sigma]\ e_2[\sigma]. \tag{3.1}$$

A natural setting for specifying such operations is the functor category $\mathbf{C}_0 := [\mathbb{F}, \mathbf{Set}]$, where $\mathbb{F} \hookrightarrow \mathbf{Set}$ denotes the full subcategory spanning all sets of the form n (i.e., $\{1, \ldots, n\}$, recalling notation from §1). For any $X \in [\mathbb{F}, \mathbf{Set}]$ and $n \in \mathbb{F}$, we think of $X(n)$ as a set of 'terms' with n potential free variables, e.g., in $\{1, \ldots, n\}$ or, if the reader prefers, $\{x_1, \ldots, x_n\}$. The action of X on morphisms $n \to n'$ is thought of as variable renaming. Returning to operations, being equipped with abstraction is the same as being a Σ_0-algebra, where $\Sigma_0 \colon \mathbf{C}_0 \to \mathbf{C}_0$ is defined by $\Sigma_0(X)(n) = X(n+1)$. An algebra structure on any X thus consists of a family of maps $X(n+1) \to X(n)$, natural in n. Similarly, for specifying both application and abstraction, we consider

$$\Sigma_\Lambda(X)(n) = X(n+1) + X(n)^2. \tag{3.2}$$

Let us now consider substitution. The idea here is to equip $\mathbf{C}_0$ with monoidal structure $(\otimes, I)$, such that

[1]This pattern is advocated by the approach of *initial algebra semantics* [26], where initiality provides a recursion principle.

- elements of $(X \otimes Y)(n)$ are like *explicit substitutions* $x(\!|\sigma|\!)$, where $x \in X(p)$ and $\sigma\colon p \to Y(n)$ for some p, considered equivalent up to some standard equations[2];
- elements of $I(n) := \{1, \ldots, n\}$ are merely variables.

Being equipped with substitution (and variables) is thus the same as being a *monoid* for this tensor product:

- the multiplication $m_X\colon X \otimes X \to X$ maps any formal, explicit substitution $x(\!|\sigma|\!)$ to an actual substitution $x[\sigma]$, and
- the unit $e_X\colon I \to X$ injects variables into terms.

Finally, how do we enforce equations such as (3.1)? This goes in two stages:

- we first collect the way substitution is supposed to commute with each operation, by providing a *pointed strength*, i.e., a natural transformation with components $st_{X,Y}\colon \Sigma_\Lambda(X) \otimes Y \to \Sigma_\Lambda(X \otimes Y)$, where $X \in \mathbf{C}_0$ and $Y \in I/\mathbf{C}_0$, satisfying some equations [23, §I.1.2];
- we then use the pointed strength to enforce all equations in one go, by requiring models to have compatible Σ_Λ-algebra and substitution structure, in a suitable sense.

Let us first explain the notion of pointed strength.

Application: For modelling Equation (3.1) for application, we would in particular define $st_{X,Y}$ to map any $(in_2(x_1, x_2))(\!|\sigma|\!)$ to $in_2(x_1(\!|\sigma|\!), x_2(\!|\sigma|\!))$, for all $x_1, x_2 \in X(p)$ and $\sigma\colon p \to Y(n)$. (The coproduct injection in_2 here acts as a formal application, recalling $\Sigma_\Lambda(X)(n) = X(n+1) + X(n)^2$.)

Abstraction: For abstraction, let us start by first stating the corresponding equation. We will then define the pointed strength accordingly. Supposing that Y is equipped with a point $e_Y : I \to Y$, we define $\sigma^\uparrow\colon p+1 \to Y(n+1)$ by copairing

$$p \xrightarrow{\sigma} Y(n) \xrightarrow{Y(in_1)} Y(n+1) \qquad \text{and} \qquad 1 = I(1) \xrightarrow{(e_Y)_1} Y(1) \xrightarrow{Y(in_2)} Y(n+1).$$

The equation is then

$$\lambda(e)[\sigma] = \lambda(e[\sigma^\uparrow]). \tag{3.3}$$

Accordingly, we define the pointed strength to map any $in_1(x)(\!|\sigma|\!)$, where $x \in X(p+1)$ and $\sigma\colon p \to Y(n)$, to $in_1(x(\!|\sigma^\uparrow|\!))$.

Let us now go through the second stage of how we impose the desired equations: a model of syntax will be a monoid X equipped with Σ_Λ-algebra structure $\nu_X : \Sigma_\Lambda(X) \to X$, such that the following diagram commutes.

$$\begin{tikzcd} \Sigma_\Lambda(X) \otimes X \arrow[r, "st_{X,X}"] \arrow[d, "\nu_X \otimes X"'] & \Sigma_\Lambda(X \otimes X) \arrow[r, "\Sigma_\Lambda(m_X)"] & \Sigma_\Lambda(X) \arrow[d, "\nu_X"] \\ X \otimes X \arrow[rr, "m_X"'] & & X \end{tikzcd} \tag{3.4}$$

Indeed, suppose given, e.g., $in_1(e)(\!|\sigma|\!) \in \Sigma_\Lambda(X) \otimes X$, by applying the left then bottom morphisms we obtain $\lambda(e)[\sigma]$, while applying the top then right morphisms we obtain $\lambda(e[\sigma^\uparrow])$, as desired.

All in all, we have:

Definition 3.1. For any finitary, pointed strong endofunctor Σ_0, a *monoid algebra* for Σ_0, or a Σ_0-*monoid*, is a Σ_0-algebra $(X, \nu_X : \Sigma_0(X) \to X)$, equipped with monoid structure

[2] In [14], we instead considered a skew-monoidal variant where the tensor product does not enforce any standard equation.

$(m_X \colon X \otimes X \to X, e_X \colon I \to X)$, such that (3.4) commutes. A Σ_0-monoid *morphism* is a morphism in $\mathbf{C}_0$ which is both a monoid morphism and a Σ_0-algebra morphism.

Let Σ_0-**Mon** denote the category of Σ_0-monoids and morphisms between them.

Let us conclude by (slightly informally) stating the result exhibiting standard syntax as the initial model [24, 20, 14]. See Proposition 5.4 below for a general and rigorous statement.

Proposition 3.2. *For any finitary, pointed strong endofunctor Σ_0, under mild hypotheses, the forgetful functor Σ_0-**Mon** $\to \mathbf{C}_0$ is monadic, and the free Σ_0-algebra over I (equivalently the initial $(I + \Sigma_0)$-algebra) is an initial Σ_0-monoid.*

Example 3.3. In the case of λ-calculus, the initial Σ_Λ-monoid is thus the least fixed point $\mathbf{Z}_0 := \mu A.(I + \Sigma_\Lambda(A))$, which is isomorphic to the standard, low-level construction of syntax.

From this, one may deduce a characterisation of not only the initial Σ_Λ-monoid, but all *free* Σ_Λ-monoids, or in other words an explicit formula for the left adjoint to the forgetful functor. Namely:

Proposition 3.4. *For any finitary, pointed strong endofunctor Σ_0, under the same hypotheses as in Proposition 3.2, the free Σ_0-monoid, say $\mathscr{L}_0(K)$, over any $K \in \mathbf{C}_0$ is*

$$\mu A.(I + \Sigma_0(A) + K \otimes A).$$

Syntactically, when $\Sigma_0 = \Sigma_\Lambda$, letting $n \vdash_K e$ mean that $e \in \mathscr{L}_0(K)(n)$, $\mathscr{L}_0(K)$ is inductively generated by the following rules [29, §3.1],

$$\frac{}{n \vdash_K x}\ (x \in n) \qquad \frac{n \vdash_K e_1 \quad \dots \quad n \vdash_K e_p}{n \vdash_K k(\!|e_1, \dots, e_p|\!)}\ (k \in K(p))$$

$$\frac{n \vdash_K e_1 \qquad n \vdash_K e_2}{n \vdash_K e_1\ e_2} \qquad \frac{n + 1 \vdash_K e}{n \vdash_K \lambda(e)}$$

modulo the equivalence

$$(f \cdot k)(\!|e_1, \dots, e_q|\!) \sim k(\!|e_{f(1)}, \dots, e_{f(p)}|\!),$$

for all $f \colon p \to q$, $k \in K(p)$, and $n \vdash_K e_1, \dots, e_q$, or perhaps more synthetically

$$(f \cdot k)(\!|\sigma|\!) \sim k(\!|\sigma \circ f|\!),$$

where $\sigma \colon q \to \mathscr{L}_0(K)(n)$ denotes the cotupling of $e_1, \dots, e_q$ viewed as maps $1 \to \mathscr{L}_0(K)(n)$.

The first rule is the standard rule for variables, while the second one is for "constants", i.e., elements of K. It corresponds to the term $K \otimes A$ in the above fixed-point formula. When $p = 0$, we sometimes shorten the notation from $k(\!|\,|\!)$ to k. The last two rules are the standard rules for application and abstraction, and they correspond to the term $\Sigma_0(A)$ in the formula. The Σ_Λ-monoid structure is syntactically straightforward; notably substitution satisfies $k(\!|e_1, \dots, e_p|\!)[\sigma] = k(\!|e_1[\sigma], \dots, e_p[\sigma]|\!)$.

3.2. **Transition systems and bisimilarity.** The appropriate notion of transition system for λ-calculus is as follows.

Definition 3.5. A *transition system* X consists of

- a *state object* $X_0 \in \mathbf{C}_0 = [\mathbb{F}, \mathbf{Set}]$,
- a set X_1 of *transitions*, and

- maps $X_0(0) \xleftarrow{s_X} X_1 \xrightarrow{t_X} X_0(1)$ giving the source and target of transitions (cf. §2.3).

Transition systems form a category $\mathbf{C}$, whose morphisms $X \to Y$ consist of compatible morphisms $f_0 \colon X_0 \to Y_0$ and $f_1 \colon X_1 \to Y_1$, in the sense that both of the following squares commute.

$$\begin{tikzcd} X_1 \arrow[r, "f_1"] \arrow[d, "s_X"'] & Y_1 \arrow[d, "s_Y"] \\ X_0(0) \arrow[r, "f_0"'] & Y_0(0) \end{tikzcd} \qquad \begin{tikzcd} X_1 \arrow[r, "f_1"] \arrow[d, "t_X"'] & Y_1 \arrow[d, "t_Y"] \\ X_0(1) \arrow[r, "f_0"'] & Y_0(1) \end{tikzcd}$$

Notation 3.6. *We write $r \colon e \Downarrow f$ for $r \in X_1$ such that $s_X(r) = e$ and $t_X(r) = f$.*

Example 3.7. The *syntactic* transition system has $\mathbf{Z}_0 \in \mathbf{C}_0$ from Example 3.3 as state object, and as transitions all derivations following the transition rules. We will come back to this case in Proposition 3.21.

Our next goal is to introduce bisimulation, for which it is convenient to characterise $\mathbf{C}$ as a presheaf category. This characterisation may be established by abstract means, but let us describe it concretely first. It is clear from the definition that transitions systems are glorified graphs. And they form a presheaf category for essentially the same reason as graphs do. Here is the base category:

Definition 3.8. Let $\mathbb{F}[\Downarrow]$ denote the category obtained by augmenting $\mathbb{F}$ with an object $\Downarrow$, together with morphisms $0 \xleftarrow{s_\Downarrow} \Downarrow \xrightarrow{t_\Downarrow} 1$, and their formal composites with non-identity morphisms from $\mathbb{F}$.

More concretely:

- There is exactly one morphism $0 \to n$ in $\mathbb{F}$ for all n, which is an identity when $n = 0$, so for all $n \neq 0$ we have a morphism $s_{\Downarrow,n} \colon \Downarrow \to n$ making the following triangle commute.

$$\begin{tikzcd} & \Downarrow \arrow[dl, "s_\Downarrow"'] \arrow[dr, "s_{\Downarrow,n}"] & \\ 0 \arrow[rr, "!"'] & & n \end{tikzcd}$$

- There are exactly n morphisms $1 \to n$ in $\mathbb{F}$ for all $n \notin \{0, 1\}$ (and no morphisms $1 \to 0$), so for all such n and $i \in n$ we have a morphism $t_{\Downarrow,n,i} \colon \Downarrow \to n$ making the following triangle commute.

$$\begin{tikzcd} & \Downarrow \arrow[dl, "t_\Downarrow"'] \arrow[dr, "t_{\Downarrow,n,i}"] & \\ 1 \arrow[rr, "i"'] & & n \end{tikzcd}$$

Proposition 3.9. *Transition systems are isomorphic to covariant presheaves on $\mathbb{F}[\Downarrow]$.*

Notation 3.10. *We often implicitly convert from transition systems to covariant presheaves, and conversely.*

Proof sketch. This will be proved below by abstract means, but for intuition let us sketch the correspondence. Given a transition system $\langle s_X, t_X \rangle \colon X_1 \to X_0(0) \times X_0(1)$, we construct a presheaf X' by setting

- $X'(n) = X_0(n)$ for all $n \in \mathbb{F}$,

- $X'(\Downarrow) = X_1$,
- with the action of $s_\Downarrow, t_\Downarrow \in \mathbb{F}$ given by s_X and t_X,
- inducing the action of all $s_{\Downarrow,n}$ and $t_{\Downarrow,n,i}$ by composition.

Conversely, for any presheaf Y, we construct a transition system Y' defined as follows:

- the state object $Y'_0 \in [\mathbb{F}, \mathbf{Set}]$ is given by restriction of Y;
- the set Y'_1 of transitions is $Y(\Downarrow)$;
- and $s_{Y'}$ and $t_{Y'}$ are $Y(s_\Downarrow)$ and $Y(t_\Downarrow)$, respectively. □

The correspondence yields basic, graph-like examples of transition systems.

Example 3.11.
(a) The representable presheaf $\mathbf{y}_0$ associated to $0 \in \mathbb{F}$ has a single closed state k_0 and its renamings (i.e., $(\mathbf{y}_0)_0(n) = 1$ for all n and for transitions $(\mathbf{y}_0)_1 = \emptyset$).
(b) The representable presheaf $\mathbf{y}_\Downarrow$ consists of a closed state k_0, a state k_1 with one free variable, their renamings, and a transition $e\colon k_0 \Downarrow k_1$.
(c) Let $\mathbf{y}_{s_\Downarrow} : \mathbf{y}_0 \to \mathbf{y}_\Downarrow$ denote the morphism mapping k_0 to k_0.

Using these basic examples, we may define bisimulation and bisimilarity by lifting following [36]:

Definition 3.12. A morphism $X \to Y$ in $\mathbf{C}$ is a *functional bisimulation* when it has the right lifting property w.r.t. the source map $\mathbf{y}_{s_\Downarrow} : \mathbf{y}_0 \to \mathbf{y}_1$. A span $X \leftarrow R \to Y$ is a *simulation* when its left leg $R \to X$ is a functional bisimulation, and a *bisimulation* when both legs are.

Remark 3.13. In this case, the Yoneda lemma says that $\mathbf{C}(\mathbf{y}_0, X) \cong X_0(0)$ and $\mathbf{C}(\mathbf{y}_\Downarrow, Y) \cong Y_1$. The right lifting property for a morphism $f\colon X \to Y$ thus says that given any $e \in Y_1$ whose source $e \cdot s_\Downarrow$ is $f(x)$ for some $x \in X_0(0)$, there exists $e' \in X_1$ such that $f(e') = e$ and $e' \cdot s_\Downarrow = x$, which matches the usual definition of functional bisimulation. The following diagram might help readability.

$$\begin{tikzcd} \mathbf{y}_0 \arrow[r, "x"] \arrow[d, "s_\Downarrow"'] & X \arrow[d, "f"] \\ \mathbf{y}_\Downarrow \arrow[r, "e"'] \arrow[ur, dashed, "e'"] & Y \end{tikzcd}$$

Definition 3.14. Let $\mathbf{Bisim}(X, Y)$ denote the category of bisimulations, with span morphisms between them.

Proposition 3.15. $\mathbf{Bisim}(X, Y)$ *has a terminal object, called* bisimilarity *and denoted by* $\sim_{X,Y}$.

Example 3.16. Bisimilarity on the syntactic transition system merely amounts to simultaneous convergence, because evaluation returns an open term, which does not have any further transition. In this case, a more relevant behavioural equivalence is substitution-closed bisimilarity, which we will define below.

3.3. **Operational semantics.** Just as we have defined the syntax as an initial Σ_0-monoid (Example 3.3), let us now define the dynamics by initiality, again starting by finding the right notion of model. First of all, models will be found among transition systems X whose underlying presheaf $X_0 \in [\mathbb{F}, \mathbf{Set}]$ is a Σ_0-monoid. Let us give these a name.

Definition 3.17. A *transition* Σ_0*-monoid* is a transition system X, together with Σ_0-monoid structure on its vertices (a.k.a. states) object X_0. When $\Sigma_0 = \mathrm{id}$, we call transition Σ_0-monoids simply *transition monoids*. (Any transition Σ_0-monoid is thus in particular a transition monoid.)

Let Σ_0-**Trans** denote the category of transition Σ_0-monoids, with as morphisms all transition system morphisms which induce Σ_0-monoid morphisms on vertices objects.

The idea is to model the transition rules as an endofunctor on transition Σ_0-monoids, leaving the underlying Σ_0-monoid untouched, i.e., a functor making the triangle

$$\begin{tikzcd} \Sigma_0\text{-}\mathbf{Trans} \arrow[rr, "\Sigma_1"] \arrow[dr, "\mathscr{D}"'] & & \Sigma_0\text{-}\mathbf{Trans} \arrow[dl, "\mathscr{D}"] \\ & \Sigma_0\text{-}\mathbf{Mon} & \end{tikzcd}$$

commute, where $\mathscr{D}$ denotes the forgetful functor (i.e., $\mathscr{D}(X) = X_0$).

For call-by-name λ-calculus, the functor $\Sigma_1 \colon \Sigma_0\text{-}\mathbf{Trans} \to \Sigma_0\text{-}\mathbf{Trans}$ modelling the non-standard rules at the end of §2 is defined as follows.

- On states, commutation of the above triangle imposes $\Sigma_1(X)_0 = X_0$.
- On transitions, let
$$\Sigma_1(X)_1 = X_0(1) + A_\beta(X),$$
where $A_\beta(X)$ denotes the set of valid premises for the second rule in §2.3, i.e., triples (r_1, e_2, r_2) such that
  - $r_1, r_2 \in X_1$ are transitions,
  - $e_2 \in X_0(0)$ is a state, and
  - $r_2 \cdot s_\Downarrow = (r_1 \cdot t_\Downarrow)[e_2]$, i.e., the source $r_2 \cdot s_\Downarrow$ of r_2 is obtained by substituting e_2 for the unique free variable in the target of r_1.

  In other words:
$$\frac{\dfrac{r_1}{e_1 \Downarrow e_1'} \qquad \dfrac{r_2}{e_1'[e_2] \Downarrow e_3}}{e_1\ e_2 \Downarrow e_3}\ .$$
  Let us notice that substitution here follows from the monoid structure of X.
- We then define the source and target maps:
  - for the first term $X_0(1)$,
    - the source of any $in_1(e)$ is $\lambda_1(e)$, where $\lambda_n \colon X_0(n+1) \to X_0(n)$ follows from the Σ_0-algebra structure of X_0;
    - the target is e itself;
  - for the second term $A_\beta(X)$,
    - the source of any $in_2(r_1, e_2, r_2)$ is $(r_1 \cdot s_\Downarrow)\ e_2$, i.e., the application of the source of r_1 to e_2 (again using the Σ_0-algebra structure of X_0);
    - the target is $r_2 \cdot t_\Downarrow$.

Accordingly, our notion of model is the following.

Definition 3.18. A *vertical* Σ_1*-algebra* is a transition Σ_0-monoid X equipped with a morphism $\nu_X \colon \Sigma_1(X) \to X$ such that $\mathscr{D}(\nu_X) = \mathrm{id}_{X_0}$, or equivalently a map $(\nu_X)_1$ making the following triangle commute.

$$\begin{tikzcd}
\Sigma_1(X)_1 \arrow[rr, "(\nu_X)_1"] \arrow[dr, "\langle s_{\Sigma_1(X)}, t_{\Sigma_1(X)}\rangle"'] & & X_1 \arrow[dl, "\langle s_X, t_X\rangle"] \\
& X_0(0) \times X_0(1) &
\end{tikzcd}$$

In the case of call-by-name λ-calculus, it should be clear that such a vertical algebra is indeed a model of the rules.

However, in order to ensure that the rules are syntax-directed, we want to distinguish, for each rule, the head operator of the source of the conclusion (abstraction for the first rule; application for the second one). Instead of demanding that $(\Sigma_1^\partial)_X$ have the form $\Sigma_1(X)_1 \to X_0(0) \times X_0(1)$, we thus rather require something of the form $\Sigma_1(X)_1 \to \Sigma_0(X_0)(0) \times X_0(1)$:

Definition 3.19 (Dynamic signatures and vertical algebras)**.**

- A *dynamic signature* Σ_1 consists of
  - a finitary functor $\Sigma_1^F : \Sigma_0\text{-}\mathbf{Trans} \to \mathbf{Set}$, and
  - a natural transformation $(\Sigma_1^\partial)_X : \Sigma_1^F(X) \to \Sigma_0(X_0)(0) \times X_0(1)$.
- The endofunctor $\check{\Sigma}_1$ *induced* by a dynamic signature maps any X to the composite $\Sigma_1^F(X) \xrightarrow{(\Sigma_1^\partial)_X} \Sigma_0(X_0)(0) \times X_0(1) \xrightarrow{\nu_{X_0,0} \times X_0(1)} X_0(0) \times X_0(1)$, where $\nu_{X_0} : \Sigma_0(X_0) \to X_0$ denotes the Σ_0-algebra structure of X_0.
- A *vertical algebra* of a dynamic signature is a vertical algebra of the induced endofunctor, in the sense of Definition 3.18.

Concretely, a vertical algebra is a dashed map making the following diagram commute.

$$\begin{tikzcd}
\Sigma_1^F(X) \arrow[r, dashed] \arrow[d, "(\Sigma_1^\partial)_X"'] & X_1 \arrow[d, "\langle s_X, t_X\rangle"] \\
\Sigma_0(X_0)(0) \times X_0(1) \arrow[r, "\nu_{X_0,0} \times X_0(1)"'] & X_0(0) \times X_0(1)
\end{tikzcd}$$

Example 3.20. For call-by-name λ-calculus, we only need to modify the source components of the above definition of Σ_1, replacing actual operations by formal ones, like so:

- the source of any $in_1(e) \in X_0(1) + A_\beta(X)$ is $in_1(e) \in \Sigma_0(X_0)(0) = X_0(1) + X_0(0)^2$;
- the source of any $in_2(r_1, e_2, r_2)$ is $in_2((r_1 \cdot s_\Downarrow), e_2) \in \Sigma_0(X_0)(0)$.

This successfully captures the syntactic transition system:

Proposition 3.21. *The initial $\check{\Sigma}_1$-algebra $\mathbf{Z}_{\check{\Sigma}_1}$, or $\mathbf{Z}$ for short, is an initial vertical algebra, and is isomorphic to the transition system of Example 3.7.*

3.4. **Substitution-closed bisimilarity.** There is an obvious notion of bisimulation for transition Σ_0-monoids:

Definition 3.22. A morphism is $\Sigma_0\text{-}\mathbf{Trans}$ is a *functional bisimulation* iff its underlying morphism in $\mathbf{C}$ is.

However, as foreshadowed by Example 3.16, the relevant notion in this case combines bisimulation with substitution-closedness, in the following sense.

Definition 3.23. For any monoid M in $\mathbf{C}_0$, an *M-module* is an object X equipped with algebra structure $X \otimes M \to X$ for the monad $- \otimes M$. A module morphism is an algebra morphism.

Example 3.24. The monoid M is itself an M-module by multiplication, and M-modules are closed under limits in $\mathbf{C}_0$, so in particular M^2 is an M-module, with action given by the composite $M^2 \otimes M \xrightarrow{\langle \pi_1 \otimes M, \pi_2 \otimes M \rangle} (M \otimes M)^2 \xrightarrow{m_M^2} M^2$.

Definition 3.25. For any transition monoid M, a span of the form $R \to M^2$ in $\mathbf{C}$ is *substitution-closed* iff R_0 may be equipped with M_0-module structure making the morphism $R_0 \to M_0^2$ into an M_0-module morphism.

Example 3.26. To see what this definition has to do with substitution-closedness, let us observe that if R is a relation in $[\mathbb{F}, \mathbf{Set}]$, an element of $(R \otimes M)(n)$ is an explicit substitution $r(\!|\sigma|\!)$ with $r \in R(p)$ for some p, and $\sigma \colon p \to M(n)$. Now, substitution-closedness amounts to a morphism $m \colon R \otimes M \to R$ commuting with projections, so if R is a relation, then r is merely a pair $e \; R \; e'$, and the morphism m ensures that $e[\sigma] \; R \; e'[\sigma]$.

Proposition 3.27. *For any transition Σ_0-monoid M, there is a terminal substitution-closed bisimulation $\sim_M^{\otimes}$, called* substitution-closed bisimilarity*.*

Proof. See Proposition 6.10 for a proof in the general case. □

Remark 3.28. One may prove that substitution-closed bisimilarity is a relation.

Proposition 3.29. *Substitution-closed bisimilarity $\sim_{\mathbf{Z}}^{\otimes}$ on the syntactic transition system $\mathbf{Z}$ coincides with applicative bisimilarity.*

Proof. Let us denote the open extension of applicative bisimilarity by $\sim_{std}^{\otimes}$, and recall that applicative bisimilarity is denoted by $\sim$. Using Lemma 2.4, $\sim_{std}^{\otimes}$ is straightforwardly a substitution-closed bisimulation, so we have $\sim_{std}^{\otimes} \subseteq \sim_{\mathbf{Z}}^{\otimes}$. But conversely any substitution-closed bisimulation relation R (hence $\sim_{\mathbf{Z}}^{\otimes}$) is in particular a substitution-closed relation contained in $\sim$ on closed terms. It is thus globally contained in $\sim_{std}^{\otimes}$ by Lemma 2.4. □

Finally, our main result instantiates to the following.

Theorem 3.30. *Substitution-closed bisimilarity is context-closed. More precisely, it is a transition Σ_0-monoid, and $\sim_{\mathbf{Z}}^{\otimes} \to \mathbf{Z}^2$ is a transition Σ_0-monoid morphism.*

In particular, there exists a span morphism $\Sigma_0((\sim_{\mathbf{Z}}^{\otimes})_0) \to (\sim_{\mathbf{Z}}^{\otimes})_0$.

4. Transition systems and bisimilarity

In this section, we start to abstract over the situation of §3, by introducing a general framework for transition systems and bisimilarity. In §4.1, we first introduce the ambient setting for this, *pre-Howe contexts*, and construct a category of transition systems, for any pre-Howe context. Then, in §4.2, we show that transition systems form a presheaf category. We then exploit this in §4.3 to define bisimulation and bisimilarity.

4.1. Pre-Howe contexts and transition systems.

Definition 4.1. A *pre-Howe context*[3] consists of

[3]The Howe contexts of [14] may be defined similarly. The difference is that for them, $\mathbf{s}$ and $\mathbf{t}$ are not necessarily functorial, but $c_1 \mapsto (\mathbf{s}(c_1), \mathbf{t}(c_1))$ defines a functor $\mathbb{C}_1 \to \mathbb{C}_0 \underline{\times} \mathbb{C}_0$, where $\mathbb{C}_0 \underline{\times} \mathbb{C}_0$ denotes the category whose objects are pairs of elements of $\mathbb{C}_0$, and where a morphism $(a_1, a_2) \to (b_1, b_2)$ consists of some indices $i, j \in \{1, 2\}$, together with a pair of morphisms $a_1 \to b_i$ and $a_2 \to b_j$.

- a small category $\mathbb{C}_0$ of *state types*,
- a small category $\mathbb{C}_1$ of *transition types*, and
- two *source* and *target* functors $\mathbf{s}, \mathbf{t} \colon \mathbb{C}_1 \to \mathbb{C}_0$.

Precomposition by $\mathbf{s}$ and $\mathbf{t}$ yields functors $\Delta_{\mathbf{s}}, \Delta_{\mathbf{t}} \colon \widehat{\mathbb{C}_0} \to \widehat{\mathbb{C}_1}$ mapping any $X \in \widehat{\mathbb{C}_0}$ to $X \circ \mathbf{s}$ and $X \circ \mathbf{t}$, respectively. Let Δ denote the pointwise product $\Delta_{\mathbf{s}} \times \Delta_{\mathbf{t}}$.

Intuitively, objects of $\mathbb{C}_0$ may be thought of as typings (typically sequents $A_1, \ldots, A_n \vdash A$, or merely natural numbers, as in §3), and objects of $\mathbb{C}_1$ as transition types. The source and target functors associate to every transition type the corresponding typings. We now use these functors to define transition systems.

Definition 4.2. Given any pre-Howe context, a *transition system* X consists of

- a *state* presheaf $X_0 \in \widehat{\mathbb{C}_0}$,
- a *transition* presheaf $X_1 \in \widehat{\mathbb{C}_1}$, and
- two *source* and *target* natural transformations $X_0 \circ \mathbf{s} \leftarrow X_1 \to X_0 \circ \mathbf{t}$, or equivalently a natural transformation $X_1 \to \Delta(X_0)$.

Proposition 4.3. *In any pre-Howe context, transition systems are precisely the objects of the lax limit category $\widehat{\mathbb{C}_1}/\Delta$ of the functor $\widehat{\mathbb{C}_0} \xrightarrow{\Delta} \widehat{\mathbb{C}_1}$ in* **CAT***, or equivalently the comma category* $\mathrm{id}_{\widehat{\mathbb{C}_1}}/\Delta$.

Proof. An object of the lax limit is by construction a triple (X_1, X_0, ∂), where $\partial \colon X_1 \to \Delta(X_0) = X_0\mathbf{s} \times X_0\mathbf{t}$. □

Notation 4.4. *In any pre-Howe context, we let $\mathbf{C} := \widehat{\mathbb{C}_1}/\Delta$, and denote the projections by $pr_1 \colon \mathbf{C} \to \widehat{\mathbb{C}_1}$ and $pr_0 \colon \mathbf{C} \to \widehat{\mathbb{C}_0}$, respectively.*

Proposition 4.5. *The projection functor $-_0 \colon \mathbf{C} \to \widehat{\mathbb{C}_0}$ has a left adjoint mapping any object X_0 to $\emptyset \to X_0\mathbf{s} \times X_0\mathbf{t}$, where $\emptyset$ denotes the initial presheaf on $\mathbb{C}_1$. For any X_0, we call this object the* discrete *transition system on X_0.*

Proof. Straightforward. □

Example 4.6. We can get $\mathbf{C} \to \widehat{\mathbb{C}_0}$ to be the forgetful functor $\mathbf{Gph} \to \mathbf{Set}$ by taking

- $\mathbb{C}_0 = 1$, so that $\widehat{\mathbb{C}_0} = \widehat{1} \cong \mathbf{Set}$,
- $\mathbb{C}_1 = 1$, so that $\mathbf{C} = \mathbf{Set}$, and
- $\mathbf{s}, \mathbf{t} \colon 1 \to 1$ to be the unique such functor, i.e., the identity.

A transition system thus consists of sets V and E together with a map $E \to V^2$, i.e., a graph.

Example 4.7. A proof-relevant variant of standard labelled transition systems (over any set $\mathbb{A}$ of labels) may be obtained as follows. We take

- $\mathbb{C}_0 = 1$ again,
- $\mathbb{C}_1 = \mathbb{A}$ viewed as a discrete category, and
- $\mathbf{s}, \mathbf{t} \colon \mathbb{C}_1 \to \mathbb{C}_0$ the unique such functor.

Thus, a transition system X consists of a set X_0 and sets X_a for all $a \in \mathbb{A}$, together with maps $X_a \to X_0^2$ returning the source and target of each a-labelled edge.

More generally, given any graph $\mathbb{L}$, taking $\mathbf{s}, \mathbf{t} \colon \mathbb{C}_1 \to \mathbb{C}_0$ to be the source and target maps $\mathbb{L}_1 \to \mathbb{L}_0$ viewed as functors between discrete categories, we obtain for $\mathbf{C} \to \widehat{\mathbb{C}_0}$ a functor equivalent to $\mathbf{Gph}/\mathbb{L} \to \mathbf{Set}/\mathbb{L}_0$.

Example 4.8. Let $\mathbb{C}_0 = \mathbb{F}^{op}$ and $\mathbb{C}_1 = 1$, with $\mathbf{s}$ and $\mathbf{t}$ picking respectively 0 and 1. In particular, $\widehat{\mathbb{C}_1} \cong \mathbf{Set}$. Then, $\Delta(X_0) = X_0(0) \times X_0(1)$ and we recover the category $\mathbf{C}$ of §3.2, and its forgetful functor to $\widehat{\mathbb{C}_0} = [\mathbb{F}, \mathbf{Set}]$.

4.2. **Transition systems as presheaves.** Before introducing bisimulation, let us establish an alternative characterisation of the category $\mathbf{C}$ of transition systems.

Proposition 4.9. *The lax limit category $\widehat{\mathbb{C}_1}/\Delta$ of transition systems is isomorphic to a presheaf category $\widehat{\mathbb{C}_{\mathbf{s},\mathbf{t}}}$.*

Proof. Let $\mathbb{C}_{\mathbf{s},\mathbf{t}}$ denote the lax colimit in $\mathbf{Cat}$ of the parallel pair $\mathbf{s}, \mathbf{t}$. By definition, it is the universal category with functors and natural transformations as in

$$\begin{tikzcd}
\mathbb{C}_1 \arrow[rr, bend left, "\mathbf{s}"] \arrow[rr, bend right, "\mathbf{t}"'] \arrow[dr, "in_1"'] & & \mathbb{C}_0 \arrow[dl, "in_0"] \\
& \mathbb{C}_{\mathbf{s},\mathbf{t}}. \arrow[u, Rightarrow, "s"] \arrow[u, Rightarrow, "t"'] &
\end{tikzcd}$$

It thus consists of the coproduct $\mathbb{C}_1 + \mathbb{C}_0$, augmented with arrows $s_L \colon \mathbf{s}(L) \to L$ and $t_L \colon \mathbf{t}(L) \to L$ for all $L \in \mathbb{C}_1$, naturally in L. Presheaves on $\mathbb{C}_{\mathbf{s},\mathbf{t}}$ coincide with $\widehat{\mathbb{C}_1}/\Delta$ because the presheaf construction turns lax colimits into lax limits. □

Notation 4.10. *We often omit the isomorphism $\widehat{\mathbb{C}_1}/\Delta \cong \widehat{\mathbb{C}_{\mathbf{s},\mathbf{t}}}$, considering it as an implicit coercion. E.g., for any $P \in \mathbb{C}_0$, $\mathbf{y}_P$ may be used to denote the transition system $\underline{P}$ with $\underline{P}_1 = \emptyset$ and $\underline{P}_0 = \mathbf{y}_P$.*

Similarly, $\mathbf{y}_L$ may be used to denote the 'minimal' transition system with one transition over L, say $\underline{L}$, i.e., $\underline{L}_1 = \mathbf{y}_L$, $\underline{L}_0 = \mathbf{y}_{\mathbf{s}(L)} + \mathbf{y}_{\mathbf{t}(L)}$, and the map $\underline{L}_1 \to \underline{L}_0\mathbf{s} \times \underline{L}_0\mathbf{t}$ uniquely determined by the element $(in_1(\mathrm{id}_{\mathbf{s}(L)}), in_2(\mathrm{id}_{\mathbf{t}(L)})) \in \underline{L}_0(\mathbf{s}(L)) \times \underline{L}_0(\mathbf{t}(L))$.

Finally, $\mathbf{y}_{s_L} \colon \mathbf{y}_{\mathbf{s}(L)} \to \mathbf{y}_L$ and $\mathbf{y}_{t_L} \colon \mathbf{y}_{\mathbf{t}(L)} \to \mathbf{y}_L$ denote the Yoneda embedding of the canonical morphisms s_L and t_L from the proof of Proposition 4.9.

By Yoneda, we thus have:

Corollary 4.11. *For all X, we have $\mathbf{C}(\mathbf{y}_L, X) \cong X_1(L)$ and $\mathbf{C}(\mathbf{y}_P, X) \cong X_0(P)$.*

Notation 4.12. *In the case of call-by-name λ-calculus, we call $\Downarrow$ the unique object coming from $\mathbb{C}_1 = 1$.*

Remark 4.13. Presheaves on $\mathbb{C}_{\mathbf{s},\mathbf{t}}$ intuitively have two dimensions, 0 and 1; the projection functor forgets dimension 1, while the left adjoint (Proposition 4.5) adds an empty dimension 1, thus lifting its 0-dimensional argument to a 1-dimensional object.

This dimensional intuition leads to the following useful observation on the forgetful functor.

Proposition 4.14. *The forgetful functor $\mathbf{C} \to \widehat{\mathbb{C}_0}$ preserves all limits and colimits, as well as image (in the sense of strong epi, mono) factorisations.*

Proof. The forgetful functor $\mathbf{C} \to \widehat{\mathbb{C}_0}$ is equivalent to the restriction functor $\widehat{\mathbb{C}_{\mathbf{s},\mathbf{t}}} \to \widehat{\mathbb{C}_0}$, which is both a left and right adjoint, hence preserves all limits and colimits. Finally, image factorisations are computed pointwise in presheaf categories (see, e.g., [2, §0]), hence are preserved by restriction functors. □

4.3. **Bisimulation and bisimilarity.** Morphisms in $\mathbf{C}$ are a generalisation of graph morphisms, which are a proof-relevant version of functional simulations. The analogue of functional bisimulations is as follows.

Definition 4.15. A morphism $f\colon X \to Y$ in $\widehat{\mathbb{C}_{\mathbf{s},\mathbf{t}}}$ is a *functional bisimulation*, or a *fibration*, iff it enjoys the (weak) right lifting property w.r.t. $\mathbf{y}_{s_L}\colon \mathbf{y}_{\mathbf{s}(L)} \to \mathbf{y}_L$, for all $L \in \mathbb{C}_1$.

Remark 4.16. This definition is strongly inspired by Joyal et al.'s [36].

Here is a characterisation of fibrations which will be important. Let us recall that a weak pullback satisfies the same universal property as a pullback, albeit without uniqueness.

Proposition 4.17. *A morphism $f\colon X \to Y$ is a functional bisimulation iff the following diagram is a pointwise weak pullback.*

$$\begin{tikzcd} X_1 \arrow[r, "f_1"] \arrow[d, "s_X"'] & Y_1 \arrow[d, "s_Y"] \\ X_0\mathbf{s} \arrow[r, "f_0\mathbf{s}"] & Y_0\mathbf{s} \end{tikzcd}$$

Remark 4.18. Being a *pointwise* weak pullback means that all squares

$$\begin{tikzcd} X_1(L) \arrow[r, "(f_1)_L"] \arrow[d, "(s_X)_L"'] & Y_1(L) \arrow[d, "(s_Y)_L"] \\ X_0(\mathbf{s}(L)) \arrow[r, "(f_0)_{\mathbf{s}(L)}"] & Y_0(\mathbf{s}(L)) \end{tikzcd}$$

should be weak pullbacks, for $L \in \mathbf{ob}(\mathbb{C}_1)$. This is weaker than being a weak pullback.

Proof of Proposition 4.17. By Yoneda, a lifting problem in $\mathbf{C}$ as below left is the same as a cone in $\widehat{\mathbb{C}_1}$ as below right, and a lifting is the same as a mediating morphism to X_1.

$$\begin{tikzcd} \mathbf{y}_{\mathbf{s}(L)} \arrow[r, "x"] \arrow[d, "\mathbf{y}_{s_L}"'] & X \arrow[d, "f"] \\ \mathbf{y}_L \arrow[r, "e"'] \arrow[ur, dashed] & Y \end{tikzcd} \qquad \begin{tikzcd} \mathbf{y}_L \arrow[rr, bend left, "e"] \arrow[dr, dashed] \arrow[ddr, bend right, "x"'] & & Y_1 \\ & X_1 \arrow[ur, "f_1"'] \arrow[d, "s_X"] & \\ & X_0\mathbf{s} \arrow[r, "f_0\mathbf{s}"'] & Y_0\mathbf{s} \arrow[uu, "s_Y"'] \end{tikzcd}$$

(On the left L and $\mathbf{s}(L)$ are viewed as objects of $\mathbb{C}_{\mathbf{s},\mathbf{t}}$, hence should technically by written $in_1(L)$ and $in_0(\mathbf{s}(L))$, respectively.) □

We now define general bisimulations, based on functional bisimulations. Usually, one considers bisimulation relations. Here, we generalise this a bit and consider arbitrary spans:

Definition 4.19. A *simulation* is a span $X \leftarrow R \to Y$ whose left leg is a fibration. A *bisimulation* is a span of fibrations (equivalently, a simulation whose converse span is also a simulation).

A *simulation (resp. bisimulation) relation* is a relation which is a simulation (resp. bisimulation).

Remark 4.20. Of course, the relevant notion in our applications is substitution-closed bisimulation, to which we will come below.

Lemma 4.21. *Simulation relations and bisimulation relations are stable under unions.*

Proof. By symmetry, it is enough to deal with the case of simulation relations. Consider any family $(R_i \hookrightarrow X \times Y)_{i \in I}$ of simulation relations. By Proposition 1.7, their union is the image of their cotupling. But because the domain $\mathbf{y}_{\mathbf{s}(L)}$ of $\mathbf{y}_{s_L}$ is representable for all $L \in \mathbb{C}_1$, any lifting problem $\mathbf{y}_{s_L} \to \bigcup_i R_i$ lifts to a lifting problem $\mathbf{y}_{s_L} \to \sum_i R_i$, which in turn lifts to a lifting problem $\mathbf{y}_{s_L} \to R_{i_0}$, for some $i_0 \in I$. We then find a lifting for the latter because R_{i_0} is a simulation by hypothesis, which yields a lifting for the original. □

Proposition 4.22. *For all $X, Y \in \mathbf{C}$, the full subcategory* $\mathbf{Bisim}(X, Y)$ *of spans between X and Y which are bisimulations admits a terminal object $\sim_{X,Y}$, called* bisimilarity*.*

Proof. As a presheaf category by Proposition 4.9, $\mathbf{C}$ is well-powered, so we may consider the union $\sim_{X,Y}$ of all bisimulation relations, which is again a bisimulation by Lemma 4.21. Finally, $\sim_{X,Y}$ is terminal, because any bisimulation R factors through its image $im(R)$, which is again a bisimulation; as a bisimulation relation, $im(R)$ thus embeds into $\sim_{X,Y}$, hence we obtain a morphism $R \twoheadrightarrow im(R) \hookrightarrow \sim_{X,Y}$, which is unique by monicity of $\sim_{X,Y} \hookrightarrow X \times Y$. □

5. Howe contexts for operational semantics

Operational semantics is a combination of syntax and transition systems, in the sense that it is about transition systems whose states form a model of a certain syntax. Our framework for operational semantics thus combines the frameworks of Fiore et al. [24] for syntax with variable binding, and of §4 for transition systems.

In §5.1, we introduce the ambient setting for our framework, *Howe contexts*, which are pre-Howe contexts equipped with structure modelling substitution. Furthermore, for any Howe context and pointed strong endofunctor Σ_0 on $\widehat{\mathbb{C}_0}$, we introduce the category Σ_0 -**Trans** of *transition Σ_0-monoids*, which are transition systems whose states form a Σ_0-monoid. We prove that the forgetful functor Σ_0 -**Trans** $\to \mathbf{C}$ is monadic.

In §5.2, we then introduce *dynamic signatures* over Σ_0, which specify the dynamics of a transition system. The (dependent) pair (Σ_0, Σ_1) then forms what we call an *operational semantics signature*. We then define the models of any such signature, called *vertical* Σ_1-algebras. They form a category Σ_1 -$\mathbf{alg}_{\mathrm{v}}$, and we prove that the forgetful functor Σ_1 -$\mathbf{alg}_{\mathrm{v}} \to \Sigma_0$ -**Trans** is monadic. We also prove that a suitably constrained construction of the initial Σ_1-algebra is in fact vertical, yielding an initial vertical Σ_1-algebra. Finally, we prove that, although both components are monadic, the composite functor Σ_1 -$\mathbf{alg}_{\mathrm{v}} \to \Sigma_0$ -**Trans** $\to \mathbf{C}$ is not.

5.1. Transition monoid algebras.

In this section, we introduce Howe contexts, and introduce transition Σ_0-monoids, for any suitable endofunctor Σ_0.

Definition 5.1. A *Howe context* consists of a pre-Howe context $\mathbf{s}, \mathbf{t} \colon \mathbb{C}_1 \to \mathbb{C}_0$, together with a monoidal structure on $\widehat{\mathbb{C}_0}$, such that the tensor preserves all colimits on the left and filtered colimits on the right.

Notation 5.2. *As for pre-Howe contexts, we let* $\mathbf{C} := \widehat{\mathbb{C}_1}/\Delta$.

Let us assume that some syntax has been specified by a finitary, pointed strong endofunctor Σ_0 on $\widehat{\mathbb{C}_0}$. We then define transition Σ_0-monoids just as in §3.

Definition 5.3. The category Σ_0 -**Trans** of *transition* Σ_0*-monoids* is the following pullback in **CAT**.

$$\begin{tikzcd} \Sigma_0\text{-}\mathbf{Trans} \arrow[r, "\mathscr{D}"] \arrow[d, "\mathscr{U}"'] \arrow[dr, phantom, "\lrcorner", very near start] & \Sigma_0\text{-}\mathbf{Mon} \arrow[d, "\mathscr{U}_0"] \\ \mathbf{C} \arrow[r, "(-)_0"'] & \widehat{\mathbb{C}_0} \end{tikzcd}$$

When Σ_0 is the constantly empty endofunctor, we speak of *transition monoids*: they consist of objects X equipped with monoid structure on X_0.

We are now interested in computing initial Σ_0-monoids. For this, we abstract over the concrete Proposition 3.2, as follows, replacing $[\mathbb{F}, \mathbf{Set}]$ with any suitable category $\mathscr{C}_0$.

Proposition 5.4 (see [24, 20, 14])**.** *For any finitary, pointed strong endofunctor Σ_0 on a monoidal, cocomplete category $\mathscr{C}_0$ such that the tensor preserves all colimits on the left and filtered colimits on the right, the forgetful functor $\mathscr{U}_0 \colon \Sigma_0\text{-}\mathbf{Mon} \to \mathscr{C}_0$ is monadic, and the free Σ_0-algebra over I (equivalently the initial $(I + \Sigma_0)$-algebra) is an initial Σ_0-monoid.*

Proof. This has been proved in Coq [14]. □

Notation 5.5. *We denote the initial Σ_0-monoid by $\mathbf{Z}_{\Sigma_0}$, or $\mathbf{Z}_0$ for short.*

Using this, we obtain the following.

Proposition 5.6. *The adjunction between $\mathbf{C}$ and $\widehat{\mathbb{C}_0}$ (Proposition 4.5) lifts to an adjunction*

$$\begin{tikzcd} \Sigma_0\text{-}\mathbf{Mon} \arrow[r, bend left, "\mathscr{M}"] \arrow[r, phantom, "\perp"] & \Sigma_0\text{-}\mathbf{Trans} \arrow[l, bend left, "\mathscr{D}"] \end{tikzcd}$$

with

- $\mathscr{D}(\mathscr{M}(X_0)) = X_0$ *and*
- *the left adjoint $\mathscr{M}$ maps any Σ_0-monoid M to the discrete transition system on M, equipped with the original Σ_0-monoid structure on M. (In particular, we have $\mathscr{M}(X_0)_1 = \emptyset$.)*

Proof. This directly follows from the next lemma. □

Lemma 5.7. *Let us consider any pullback*

$$\begin{tikzcd} \mathbf{A} \arrow[r, "S"] \arrow[d, "V"'] \arrow[dr, phantom, "\lrcorner", very near start] & \mathbf{C} \arrow[d, "U"] \\ \mathbf{B} \arrow[r, "R"'] & \mathbf{D} \end{tikzcd}$$

in **CAT** *such that U is monadic, say with left adjoint $J \colon \mathbf{D} \to \mathbf{C}$, and R has a left adjoint L with identity unit $\eta^R_D = \mathrm{id}_D \colon D \to RLD$.*

Then, S admits a left adjoint K with identity unit, such that the canonical natural transformation $LU \to VK$ is an identity.

Proof. First of all by the triangle identities

$$\begin{tikzcd} RB \arrow[rr, equal, "\eta^R_{RB}"] \arrow[dr, equal] & & RLRB \arrow[dl, "R(\varepsilon^R_B)"] \\ & RB, & \end{tikzcd} \qquad \begin{tikzcd} LD \arrow[rr, equal, "L\eta^R_D"] \arrow[dr, equal] & & LRLD \arrow[dl, "\varepsilon^R_{LD}"] \\ & LD & \end{tikzcd}$$

we have

$$R(\varepsilon^R_B) = \mathrm{id}_{RB} \qquad \text{and} \qquad \varepsilon^R_{LD} = \mathrm{id}_{LD} \tag{5.1}$$

for all C and D.

Similarly, we have

$$RL = \mathrm{id}_{\mathbf{D}} \tag{5.2}$$

not only on objects but also on morphisms, since by naturality of η we have for any $f\colon D \to D'$:

$$RLf = RLf \circ \eta_D = \eta_{D'} \circ f = f.$$

Let us furthermore assume w.l.o.g. that the pullback is constructed in the standard way, using compatible pairs.

We then define $K(C) = (LU(C), C)$, which is legitimate since $RLU(C) = U(C)$ by hypothesis. To prove the universal property, assume given (B', C') such that $R(B') = U(C')$, and a morphism $f\colon C \to S(B', C') = C'$ in $\mathbf{C}$. Then, letting $\widetilde{U(f)}\colon LU(C) \to B'$ denote the transpose of $U(C) \xrightarrow{U(f)} U(C') = R(B')$, we have by (5.1) and (5.2)

$$R(\widetilde{U(f)}) = R(\varepsilon^R_{B'} \circ LUf) = RLUf = Uf,$$

so $(\widetilde{U(f)}, f)\colon (LUC, C) \to (B', C')$ in the pullback $\mathbf{A}$. Furthermore, the desired triangle

$$\begin{tikzcd} C \arrow[r, equal] \arrow[dr, "f"'] & S(LUC, C) \arrow[d, "S(\widetilde{U(f)}, f) = f"] \\ & S(B', C') = \mathrm{C'} \end{tikzcd}$$

commutes as desired, trivially. Finally, any $(g, h)\colon (LUC, C) \to (B', C')$ making it commute must satisfy $h = f$ and $Rg = Uh = Uf$. But $Rg\colon UC = RLUC \to RB'$ is the transpose of g, so g must conversely be the transpose of $Rg = Uf$, and hence $(g, h) = (\widetilde{U(f)}, f)$, proving the desired uniqueness property.

It remains to prove that the canonical natural transformation

$$LUC = LUSKC = LRVKC \xrightarrow{\varepsilon^R_{VKC}} VKC$$

is an identity. But by construction $VKC = V(LUC, C) = LUC$, and $\varepsilon^R_{LUC} = \mathrm{id}_{LUC}$ by (5.1), hence the result. □

Remark 5.8. The names, $\mathscr{M}$ and $\mathscr{D}$, stand for "monter" and "descendre", "go up" and "go down" in French.

Proposition 5.9. *The forgetful functor $\mathscr{U}\colon \Sigma_0\text{-}\mathbf{Trans} \to \mathbf{C}$ is finitary and monadic.*

Proof. This follows from the fact that transition Σ_0-monoids are the algebras of an equational system over $\mathbf{C}$ in the sense of Fiore and Hur, to which [19, Theorem 6.1] applies. □

Notation 5.10. *We denote by $\mathscr{L}$ the left adjoint to $\mathscr{U}$.*

5.2. **Operational semantics signatures.** Similarly, we define abstract dynamic signatures, which abstract over those of Definition 3.19:

Definition 5.11. Given a Howe context $\mathbf{s}, \mathbf{t} \colon \mathbb{C}_1 \to \mathbb{C}_0$ and a pointed strong $\Sigma_0 \colon \widehat{\mathbb{C}_0} \to \widehat{\mathbb{C}_0}$, a *dynamic signature* $\Sigma_1 = (\Sigma_1^F, \Sigma_1^\partial)$ over Σ_0 consists of a finitary functor $\Sigma_1^F : \Sigma_0\text{-}\mathbf{Trans} \to \widehat{\mathbb{C}_1}$, together with a natural transformation Σ_1^∂ with components $\Sigma_1^F(X) \to \Sigma_0(X_0)\mathbf{s} \times X_0\mathbf{t}$.

Let us pack up the static and dynamic notions of signature.

Definition 5.12. An *operational semantics signature* (Σ_0, Σ_1) on a given Howe context $\mathbf{s}, \mathbf{t} \colon \mathbb{C}_1 \to \mathbb{C}_0$ consists of a pointed strong endofunctor Σ_0 preserving sifted colimits, together with a dynamic signature Σ_1 over it.

Remark 5.13. Preservation of sifted colimits [3] is stronger than finitarity for Σ_0. We need it for Lemma 9.54 below. In a presheaf category like $\widehat{\mathbb{C}_0}$, if Σ_0 preserves pullbacks (for example, by familiality), it is equivalent to being finitary and preserving all epis, as seen from the proof of [4, Theorem 18.1]. In [14], we mistakenly only require Σ_0 to be finitary, which yields a gap in the proof of [14, Lemma 5.13].

Example 5.14. The endofunctor $\Sigma_0(X)(n) = X(n+1) + X(n)^2$ on $[\mathbb{F}, \mathbf{Set}]$ preserves sifted colimits. This easily follows from the fact that sifted colimits are the ones commuting with products in sets.

Let us now introduce the models of a dynamic signature. We start by fixing, for the rest of this section, an operational semantics signature (Σ_0, Σ_1) on a Howe context $\mathbf{s}, \mathbf{t} \colon \mathbb{C}_1 \to \mathbb{C}_0$.

Definition 5.15. Let $\check{\Sigma}_1 \colon \Sigma_0\text{-}\mathbf{Trans} \to \Sigma_0\text{-}\mathbf{Trans}$ map any transition Σ_0-monoid X to the composite

$$\Sigma_1^F(X) \xrightarrow{(\Sigma_1^\partial)_X} \Sigma_0(X_0)\mathbf{s} \times X_0\mathbf{t} \xrightarrow{\nu_{X_0}\mathbf{s} \times X_0\mathbf{t}} X_0\mathbf{s} \times X_0\mathbf{t},$$

where ν_{X_0} denotes the Σ_0-algebra structure of X_0.

Proposition 5.16. *The endofunctor $\check{\Sigma}_1$ is finitary and makes the following triangle commute.*

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

Proof. Commutativity of the triangle holds by construction, and finitarity follows from finitarity of Σ_1^F. □

Definition 5.17. A $\check{\Sigma}_1$-algebra structure $\check{\Sigma}_1(X) \to X$ on an object $X \in \Sigma_0\text{-}\mathbf{Trans}$ is *vertical* when its image under the forgetful functor $\Sigma_0\text{-}\mathbf{Trans} \to \Sigma_0\text{-}\mathbf{Mon}$ is the identity. Let $\Sigma_1\text{-}\mathbf{alg}_\mathrm{v}$ denote the full subcategory of $\check{\Sigma}_1\text{-}\mathbf{alg}$ spanned by all vertical algebras.

Theorem 5.18. *The forgetful functor $\Sigma_1\text{-}\mathbf{alg}_\mathrm{v} \to \Sigma_0\text{-}\mathbf{Trans}$ is monadic, and furthermore the initial $\check{\Sigma}_1$-algebra $\mathbf{Z}_{\check{\Sigma}_1}$, or $\mathbf{Z}$ for short, may be chosen to be vertical, hence is also initial in $\Sigma_1\text{-}\mathbf{alg}_\mathrm{v}$.*

Proof. For the first statement, vertical algebras may be specified as an equational system, in the sense of [19], so the result follows by [19, Theorem 6.1]. For the second statement, $\mathbf{Z}$ is the colimit of the initial chain

$$\mathbf{Z}_0 \xrightarrow{!} \check{\Sigma}_1(\mathbf{Z}_0) \xrightarrow{\check{\Sigma}_1(!)} \dots \xrightarrow{\check{\Sigma}_1^{n-1}(!)} \check{\Sigma}_1^n(\mathbf{Z}_0) \to \dots$$

(where $\mathbf{Z}_0$ is shorthand for $\mathscr{M}(\mathbf{Z}_0)$, for readability, which is initial in Σ_0-**Trans**, by Proposition 5.6). The image of this chain in Σ_0-**Mon** is the everywhere-identity chain on $\mathbf{Z}_0$. □

By construction, we have:

Proposition 5.19. *We have* $\mathscr{D}(\mathbf{Z}) = \mathbf{Z}_0$.

Let us readily annihilate any hope that vertical Σ_1-algebras are monadic over $\mathbf{C}$.

Lemma 5.20. *Consider the operational semantics signature of λ-calculus with Σ_0 from* (3.2) *and Σ_1 from Example 3.20. Then, the composite forgetful functor Σ_1-**alg**$_\mathrm{v}$ $\to$ Σ_0-**Trans** $\to$ **C** is not monadic.*

Proof. Let us draw inspiration from the classical example of a non-monadic composite of monadic functors [8, p. 107], namely the composite **Cat** $\to$ **RGph** $\to$ **Set**, where **RGph** denotes the category of reflexive graphs, and the forgetful functor to sets returns the set of arrows. One may show that both functors are monadic, but that their composite is not. By Beck's monadicity theorem [42, Theorem VI.7.1], it suffices to find a parallel pair $X \rightrightarrows Y$ in **Cat** whose image in sets admits an absolute coequaliser, and show that this coequaliser is not created by the composite forgetful functor. The idea is to take Y to consist of two arrows

$$1 \xrightarrow{a} 2 \qquad\qquad 3 \xrightarrow{b} 4,$$

and $X = 1 + Y$ to have an additional object, say 0. We then define $u, v \colon X \to Y$ to be the identity on Y, and respectively map 0 to 2 and 3. The coequalisers in **Cat** (top) and **Set** (bottom) look as follows, abbreviating each id_C to just C for readability.

$$\begin{tikzcd}
0 \quad 1 \arrow[r, "a"] & 2 \quad 3 \arrow[r, "b"] & 4 \arrow[r, bend left, "u"] \arrow[r, bend right, "v"'] & 1 \arrow[r, "a"] & 2 \quad 3 \arrow[r, "b"] & 4 \arrow[r, two heads] & 1 \arrow[r, "a"] \arrow[rr, bend right, "b\circ a"'] & \{2,3\} \arrow[r, "b"] & 4 \arrow[d, mapsto] \\
0 \quad 1 \quad a & 2 \quad 3 & b \quad 4 \arrow[r, bend left] \arrow[r, bend right] & 1 \quad a & 2 \quad 3 & b \quad 4 \arrow[r, two heads] & 1 \quad a & \{2,3\} & b \quad 4
\end{tikzcd}$$

One easily proves that the one in sets is split, hence absolute. Because of the composite arrow $b \circ a$ that the coequaliser in **Cat** must have but the one in sets does not, the coequaliser is not created by the forgetful functor, thus contradicting monadicity.

For proving the lemma, we rely on the β-rule to mimick composition in constructing the following parallel pair $X \rightrightarrows Y$ in Σ_1-**alg**$_\mathrm{v}$.

- Y is the vertical Σ_1-algebra defined as the λ-calculus extended with two constants a and b, unary operations k and l, and an axiom $b \Downarrow l(\!|x|\!)$;
- X is the vertical Σ_1-algebra extending Y with a constant c;
- the Σ_1-algebra morphisms $u, v \colon X \to Y$ respectively map c to b and $k(\!|a|\!)$.

The coequaliser E of these two morphisms computed in the presheaf category $\mathbf{C}$ is a quotient of Y by the equation $b = k(\!|a|\!)$. It thus has as reductions $[e] \Downarrow [f]$ between equivalence classes, for all reductions $e' \Downarrow f'$ between representatives $e' \in [e]$ and $f' \in [f]$. E.g., it has

a reduction $[k(\!|a|\!)] \Downarrow [l(\!|x|\!)]$, since $b \in [k(\!|a|\!)]$ and $b \Downarrow l(\!|x|\!)$. However, E lacks a reduction $[(\lambda x.k(\!|x|\!))\ a] \Downarrow [l(\!|x|\!)]$. Indeed, $[(\lambda x.k(\!|x|\!))\ a]$ has a unique representative, namely $(\lambda x.k(\!|x|\!))\ a$, whose evaluation in Y gets stuck at $k(\!|a|\!)$. However, the coequaliser in Σ_1-$\mathbf{alg}_{\mathrm{v}}$ does have such a reduction by applying the β-rule:

$$\frac{[\lambda x.k(\!|x|\!)] \Downarrow [k(\!|x|\!)] \qquad [k(\!|a|\!)] = [b] \Downarrow [l(\!|x|\!)]}{[(\lambda x.k(\!|x|\!))\ a] \Downarrow [l(\!|x|\!)]}\ .$$

Thus, the coequaliser is not created by the forgetful functor.

Finally, this coequaliser is split (hence absolute): the morphism $v\colon X \to Y$ mapping c to $k(\!|a|\!)$ has a section $f\colon Y \to X$, which maps $k(\!|a|\!)$ to c, and the coequaliser arrow $e\colon Y \to E$ has a section $g\colon E \to Y$ mapping $k(\!|a|\!)$ to b. It is straightforward to check that this indeed defines a split coequaliser, i.e., that $g \circ e = u \circ f$. □

Remark 5.21. A crucial point is that g may (and does) map a and k to themselves (the latter being necessary to preserve the target of $(\lambda x.k(\!|x|\!)) \Downarrow k(\!|x|\!)$), and $k(\!|a|\!)$ to $b \neq k(\!|a|\!)$. Indeed, g lives in $\mathbf{C}$, as opposed to Σ_0-$\mathbf{Trans}$, hence need not preserve substitution. All that is required is naturality. But $k(\!|a|\!)$, being closed, cannot be the target of any transition, and by induction cannot be the source of any transition either, hence the result.

Also, let us stress that in Y, $(\lambda x.k(\!|x|\!))\ a$ does not evaluate, since $k(\!|a|\!)$ itself does not. Similarly, in E, which is not saturated by the evaluation rules, we do have the premises $(\lambda x.k(\!|x|\!)) \Downarrow k(\!|x|\!)$ and $k(\!|a|\!) = b \Downarrow l(\!|x|\!)$, but not the conclusion $(\lambda x.k(\!|x|\!))\ a \Downarrow l(\!|x|\!)$.

6. Substitution-closed bisimilarity

We have now introduced our notion of syntactic transition system, given by Σ_0-transition monoids in a Howe context, and explained how to generate such systems from operational semantics signatures. In this section, we incorporate substitution into the notions of bisimulation and bisimilarity introduced in §4.3, which yields substitution-closed bisimilarity. We then state our main theorem, for which we include a high-level proof sketch.

In order to introduce substitution-closed bisimilarity, we first lift the notion of bisimulation to Σ_0-monoids, generalising §3.4:

Definition 6.1. A morphism in Σ_0-$\mathbf{Trans}$ is a *functional bisimulation* iff its underlying morphism in $\mathbf{C}$ is. A span is a *simulation* iff its left leg is a functional bisimulation, and a *bisimulation* iff both of its legs are functional bisimulations.

Let us readily prove the following characterisation by lifting, recalling from Notation 5.10 that $\mathscr{L}\colon \mathbf{C} \to \Sigma_0$-$\mathbf{Trans}$ is left adjoint to the forgetful functor.

Proposition 6.2. *A morphism in* Σ_0-$\mathbf{Trans}$ *is a functional bisimulation iff it has the right lifting property w.r.t.* $\mathscr{L}(\mathbf{y}_{s_L})\colon \mathscr{L}(\mathbf{y}_{\mathbf{s}(L)}) \to \mathscr{L}(\mathbf{y}_L)$*, for all* $L \in \mathbb{C}_1$*.*

Proof. By adjunction. □

As seen in Example 3.16, we now want to go beyond bisimilarity, and introduce abstract versions of substitution-closed bisimulation and bisimilarity. For this, let us give the general definition of modules over a monoid.

Definition 6.3. For any monoid M in a monoidal category $\mathscr{C}$, let M-**Mod** denote the category of algebras for the monad $- \otimes M$.

An M-module thus consists of an object X equipped with an action $X \otimes M \to X$ of M satisfying straightforward coherence conditions.

Example 6.4. M itself is an M-module, with action given by multiplication.

Before going into substitution-closed bisimulation, let us record the following useful properties of modules in a Howe context.

Proposition 6.5. *In any Howe context, for all monoids M in $\widehat{\mathbb{C}_0}$,*

- *the forgetful functor M-**Mod** $\to \widehat{\mathbb{C}_0}$ creates all limits and colimits, and furthermore*
- *the category M-**Mod** is regular and the forgetful functor M-**Mod** $\to \widehat{\mathbb{C}_0}$ creates image factorisations.*

Proof. For creation of limits and colimits:

- As algebras for the monad $- \otimes M$, M-modules are closed under limits.
- They are also closed under all types of colimits preserved by $- \otimes M$, i.e., all of them by definition of Howe contexts.

Thus, M-**Mod** is complete and cocomplete, hence regularity reduces to showing that the pullback of any regular epi is again a regular epi. So let us consider any pullback square

$$\begin{tikzcd} A \arrow[r, "u"] \arrow[d, "v"'] \arrow[dr, phantom, "\lrcorner", very near start] & B \arrow[d, "f"] \\ C \arrow[r, "g"'] & D \end{tikzcd}$$

in M-**Mod**, with f a regular epi, and show that v must also be a regular epi. By creation, hence preservation, of limits and colimits, the given pullback square is also a pullback in $\widehat{\mathbb{C}_0}$ and f is a regular epi there too. So by regularity of the presheaf category $\widehat{\mathbb{C}_0}$, v is a regular epi in $\widehat{\mathbb{C}_0}$. Equivalently, it is a coequaliser of its kernel pair. But by creation of limits the kernel pair uniquely lifts to a kernel pair in M-**Mod**, and by creation of colimits v is a coequaliser there too. This shows that M-**Mod** is regular.

Finally, given $X, Z \in M$-**Mod**, let us consider any image factorisation $X \overset{e}{\twoheadrightarrow} Y \overset{m}{\hookrightarrow} Z$ in $\widehat{\mathbb{C}_0}$ of a morphism $f\colon X \to Z$ in M-**Mod**, i.e., e is a regular epi and m is a mono in $\widehat{\mathbb{C}_0}$. In this situation, e is the coequaliser of its kernel pair in $\widehat{\mathbb{C}_0}$, but, as we just saw, this kernel pair lifts to a kernel pair in M-**Mod**, whose coequaliser is created by the forgetful functor, hence e is a coequaliser, hence a regular epi in M-**Mod**. Finally, f also coequalises the kernel pair, hence the existence of a unique mediating morphism $Y \to Z$ in M-**Mod**, which must be m by faithfulness of the forgetful functor M-**Mod** $\to \widehat{\mathbb{C}_0}$. Thus, m is also a morphism in M-**Mod**. Finally, its monicity follows again by faithfulness of the forgetful functor. $\square$

Let us now introduce substitution-closed spans, first in an arbitrary monoidal category, and then in a Howe context. This then leads us to substitution-closed bisimulation.

Definition 6.6. In a monoidal category $\mathscr{C}$ with binary products, given a monoid M and M-modules X and Y, a *substitution-closed* span is a span $p\colon R \to X \times Y$ equipped with M-module structure $\rho\colon R \otimes M \to R$ on R, such that $p\colon R \to X \times Y$ is an M-module morphism.

The last condition is equivalent to commutation of the following diagram.

$$\begin{tikzcd} R \otimes M \arrow[rr, "\rho"] \arrow[d, "p \otimes M"'] & & R \arrow[d, "p"] \\ (X \times Y) \otimes M \arrow[r, "\langle \pi_1 \otimes M, \pi_2 \otimes M \rangle"'] & (X \otimes M) \times (Y \otimes M) \arrow[r, "a_X \times a_Y"'] & X \times Y \end{tikzcd}$$

Definition 6.7. Consider any Howe context $\mathbf{s}, \mathbf{t}\colon \mathbb{C}_1 \to \mathbb{C}_0$ and transition monoid $M \in \mathbf{C}$. Let $X, Y \in \mathbf{C}$ be equipped with M_0-module structure on X_0 and Y_0. A *substitution-closed* span is a span $R \to X \times Y$ equipped with substitution-closed structure on $R_0 \to X_0 \times Y_0$.

Definition 6.8. For any Howe context $\mathbf{s}, \mathbf{t}\colon \mathbb{C}_1 \to \mathbb{C}_0$ and transition monoid $M \in \mathbf{C}$, a *substitution-closed simulation (resp. bisimulation)* is a substitution-closed span $R \to M^2$ (viewing M_0 itself as an M_0-module by Example 6.4) which is a simulation (resp. bisimulation). Let $\mathbf{Bisim}^{\otimes}(M)$ denote the full subcategory of $\mathbf{C}/M^2$ spanned by substitution-closed bisimulations.

Let us now prove the existence of substitution-closed bisimilarity.

Lemma 6.9. *Substitution-closed simulation and bisimulation relations are stable under unions.*

Proof. By Lemma 4.21, the union of a family of substitution-closed simulation (resp. bisimulation) relations is again a simulation (resp. bisimulation) relation. But by Proposition 6.5, the union in $\mathbf{C}$ is again substitution-closed, which concludes the proof. □

Proposition 6.10. *For any Howe context* $\mathbf{s}, \mathbf{t}\colon \mathbb{C}_1 \to \mathbb{C}_0$ *and monoid* $M \in \mathbf{C}$*, the category* $\mathbf{Bisim}^{\otimes}(M)$ *of substitution-closed bisimulations over* M *admits a terminal object* $\sim^{\otimes}_{M}$*, called* substitution-closed bisimilarity*.*

Proof. Straightforward generalisation of the proof of Proposition 4.22 using the lemma. □

Notation 6.11. *When* $M = \mathbf{Z}$*, we abbreviate* $\sim^{\otimes}_{\mathbf{Z}}$ *to just* $\sim^{\otimes}$*.*

We now want to state the abstract version of our main theorem, but we need an additional hypothesis, which we now introduce. The idea is essentially that Σ_1 should preserve functional bisimulations, which does not quite make sense, because the codomain of Σ_1^F is $\widehat{\mathbb{C}_1}$, where no notion of functional bisimulation has been defined yet. Recalling Proposition 4.17, we work around this as follows.

Definition 6.12. A dynamic signature $\Sigma_1 = (\Sigma_1^F, \Sigma_1^\partial)$ *preserves functional bisimulations* iff for any functional bisimulation $f\colon R \to X$ in Σ_0-**Trans**, the following square is a pointwise weak pullback.

$$\begin{tikzcd} \Sigma_1^F(R) \arrow[r, "\Sigma_1^F(f)"] \arrow[d, "\pi_1 \circ (\Sigma_1^\partial)_R"'] & \Sigma_1^F(X) \arrow[d, "\pi_1 \circ (\Sigma_1^\partial)_X"] \\ \Sigma_0(R_0)\mathbf{s} \arrow[r, "\Sigma_0(f_0)\mathbf{s}"'] & \Sigma_0(X_0)\mathbf{s} \end{tikzcd} \tag{6.1}$$

Remark 6.13. It may not be obvious that the dynamic signature for call-by-name λ-calculus preserves functional bisimulations. We will come back to this in §7 by showing that it satisfies a sufficient condition, cellularity.

Remark 6.14. It may seem linguistically inappropriate to say that Σ_1 preserves functional bisimulations, since Σ_1 is not merely a functor, and we have not even defined fibrations in the codomain category $\widehat{\mathbb{C}_1}$ anyway. We will justify the terminology in Proposition 7.8, but for now let us move on directly to the main result.

Theorem 6.15. *If Σ_1 preserves functional bisimulations, then substitution-closed bisimilarity is context-closed. More precisely, $\sim^{\otimes}$ is a transition Σ_0-monoid, and $\sim^{\otimes} \to \mathbf{Z}^2$ is a transition Σ_0-monoid morphism.*

Proof sketch (see §9 for the full proof). The proof takes inspiration from Howe's original method.

1. We first define the *Howe closure* H_0 of substitution-closed bisimilarity $\sim_0^{\otimes}$ on states as the initial Σ_0^H-monoid for the pointed strong endofunctor Σ_0^H on $\widehat{\mathbb{C}_0}/\mathbf{Z}^2$ defined by $\Sigma_0^H(R) = \Sigma_0(R) + (R; \sim_0^{\otimes})$. We then show that, by construction, H_0 is a Σ_0-monoid and both projections are Σ_0-monoid morphisms.

2. We then define the *transition Howe closure* H of (the full) substitution-closed bisimilarity $\sim^{\otimes}$, as an initial algebra for an endofunctor Σ_1^H on a suitable category $\mathbf{C}_{\mathbf{Z}}^H$. Very roughly, $\mathbf{C}_{\mathbf{Z}}^H$ is the category of spans $R \to \mathbf{Z}^2$ whose projection is precisely $H_0 \to \mathbf{Z}_0^2$, and $\Sigma_1^H(R) = \check{\Sigma}_1(R) + (R; \sim^{\otimes})$. We show:

Lemma 6.16. *There exists a span morphism $\mathbf{i}^H : \sim^{\otimes} \to H$.*

3. Next comes the key lemma:

Lemma 6.17. *If Σ_1 preserves functional bisimulations, then the transition Howe closure H is a substitution-closed simulation.*

Remark 6.18. Since H_0 is a Σ_0-monoid by construction, H is easily seen to be substitution-closed, so the lemma really is about it being a simulation.

The key lemma is proved by characterising H as an initial algebra for a different endofunctor on a different category, whose initial chain involves iterated applications of Σ_1 (preserving simulations by hypothesis) to $\pi_1 \colon \mathscr{M}(H_0) \to \mathscr{M}(\mathbf{Z}_0)$, which is trivially a simulation.

4. In the standard proof method, the next step is to prove that the transitive closure of H_0 is symmetric. But in our case H_0 is a general span, not a relation. In order to avoid some coherence issues, we introduce a suitable notion of *relational* transitive closure for general spans, denoted by $-^{\overline{+}}$, which is equipped with a canonical map $\mathbf{i}_R^{\overline{+}} \colon R \to R^{\overline{+}}$ for each span R. We then show:

Lemma 6.19. *The relational transitive closure ${H_0}^{\overline{+}}$ of the Howe closure H_0 on states is symmetric.*

As substitution-closed simulations are closed under transitive closure, we obtain

Corollary 6.20. *$H^{\overline{+}}$ is a substitution-closed simulation which is symmetric on states.*

We then use the following lemma (proved in §9.7).

Lemma 6.21. *For any substitution-closed simulation R such that R_0 is symmetric, there exists a substitution-closed bisimulation R' and a span morphism $\mathbf{i}'_R \colon R \to R'$.*

By terminality of $\sim^{\otimes}$, we thus get a unique morphism $!_{H^{\overline{+}\prime}} \colon H^{\overline{+}\prime} \to \sim^{\otimes}$ over $\mathbf{Z}^2$.

5. From the chain

$$\sim^{\otimes} \xrightarrow{\mathbf{i}^H} H \xrightarrow{\mathbf{i}^{\overline{+}}_H} H^{\overline{+}} \xrightarrow{\mathbf{i}'_R} H^{\overline{+}\prime} \xrightarrow{!_{H^{\overline{+}\prime}}} \sim^{\otimes}$$

we get by terminality that $\sim^{\otimes}$ is a retract of a transition Σ_0-monoid, namely H. The result then readily follows from monadicity of transition Σ_0-monoids (Proposition 5.9) and the following result, taking $X = H$, $Y = \sim^{\otimes}$, and $Z = \mathbf{Z}^2$. □

Lemma 6.22. *Consider a monad $T \colon \mathscr{C} \to \mathscr{C}$ on any category $\mathscr{C}$, T-algebras X and Z, and morphisms $X \overset{e}{\twoheadrightarrow} Y \overset{m}{\hookrightarrow} Z$ in $\mathscr{C}$ such that the composite is a T-algebra morphism, e is a split epi, and m is monic. Then there is a unique T-algebra structure on Y such that e and m both are T-algebra morphisms.*

Proof. Let $s \colon Y \to X$ denote any section of e. The desired structure is given by

$$T(Y) \xrightarrow{T(s)} T(X) \to X \overset{e}{\twoheadrightarrow} Y,$$

where the middle morphism is the given T-algebra structure on X, and the rest follows by monicity of m. □

7. Preservation of functional bisimulations, and cellularity

Let us now consider the main hypothesis of Theorem 6.15, preservation of functional bisimulations. In §7.1, we rephrase the condition in a way that makes more sense linguistically, i.e., by an actual preservation condition. We then work towards a characterisation in terms of cellularity. In §7.2, we first briefly recall *familial* functors [17, 15, 56], and show that the operational semantics signature for call-by-name λ-calculus gives rise to two familial functors, in a suitable sense. In §7.3, we restrict attention to the case where both components of the dynamic signature give rise to familial functors in this sense, and show that preservation of functional bisimulations is then equivalent to a *cellularity* condition [25, 14], which itself comes with a useful sufficient condition.

7.1. **An alternative characterisation.**

Let us first give an alternative definition of dynamic signatures.

Definition 7.1. Let $\widehat{\mathbb{C}_1}/\Delta_{\mathbf{s}}$ denote the following lax limit category.

$$\begin{tikzcd} & \widehat{\mathbb{C}_1}/\Delta_{\mathbf{s}} \arrow[dl, "\pi_2"'] \arrow[dr, "\pi_1"] \arrow[d, phantom, "\Longleftarrow"] & \\ \widehat{\mathbb{C}_0} \arrow[rr, "\Delta_{\mathbf{s}}"'] & & \widehat{\mathbb{C}_1} \end{tikzcd}$$

Concretely, an object consists of presheaves X_1 and X_0, together with a morphism $X_1 \to X_0\mathbf{s}$. Just as $\mathbf{C}$, $\widehat{\mathbb{C}_1}/\Delta_{\mathbf{s}}$ is in fact a presheaf category:

Proposition 7.2. *The category $\widehat{\mathbb{C}_1}/\Delta_{\mathbf{s}}$ is isomorphic to the presheaf category over the lax colimit $\mathbb{C}_{\mathbf{s}}$ of the functor $\mathbf{s} \colon \mathbb{C}_1 \to \mathbb{C}_0$, as in*

$$\begin{tikzcd} \mathbb{C}_1 \arrow[rr, "\mathbf{s}"] \arrow[dr, "in_1"'] & \arrow[d, phantom, "\Leftarrow s =="] & \mathbb{C}_0 \arrow[dl, "in_0"] \\ & \mathbb{C}_{\mathbf{s}}. & \end{tikzcd}$$

Proof. Similar to Proposition 4.9. □

Remark 7.3. Concretely, $\mathbb{C}_{\mathbf{s}}$ is the coproduct of $\mathbb{C}_0$ and $\mathbb{C}_1$, augmented with morphisms $s_L : \mathbf{s}(L) \to L$ for all $L \in \mathbb{C}_1$, naturally in L.

Notation 7.4. *In the case of call-by-name λ-calculus, as in* **C** *(Notation 4.12), we call* $\Downarrow$ *the unique object coming from* $\mathbb{C}_1 = 1$.

Definition 7.5. For any operational semantics signature (Σ_0, Σ_1), let $\Sigma_1^{\mathbf{s}} : \Sigma_0\text{-}\mathbf{Trans} \to \widehat{\mathbb{C}_1}/\Delta_{\mathbf{s}}$ map any $X_1 \to X_0\mathbf{s} \times X_0\mathbf{t}$ to the first leg $\Sigma_1^F(X) \to \Sigma_0(X_0)\mathbf{s}$ of $(\Sigma_1^\partial)_X$.

Proposition 7.6. *For any operational semantics signature* (Σ_0, Σ_1)*, the functor* $\Sigma_1^{\mathbf{s}} : \Sigma_0\text{-}\mathbf{Trans} \to \widehat{\mathbb{C}_1}/\Delta_{\mathbf{s}}$ *is finitary, and makes the following diagram commute,*

$$\begin{tikzcd} \Sigma_0\text{-}\mathbf{Trans} \arrow[r, "\Sigma_1^{\mathbf{s}}"] \arrow[d, "\mathscr{D}"'] & \widehat{\mathbb{C}_1}/\Delta_{\mathbf{s}} \arrow[d, "\pi_2"] \\ \Sigma_0\text{-}\mathbf{Mon} \arrow[r, "\mathscr{U}_0"'] & \widehat{\mathbb{C}_0} \arrow[r, "\Sigma_0"'] & \widehat{\mathbb{C}_0} \end{tikzcd} \tag{7.1}$$

where π_2*, as in Definition 7.1, maps any object* $X_1 \to X_0\mathbf{s}$ *to* X_0.

Proof. Finitarity holds because it does pointwise, by assumption. The diagram commutes by construction. □

In $\widehat{\mathbb{C}_{\mathbf{s}}}$ (or, through the isomorphism of Proposition 7.2, in $\widehat{\mathbb{C}_1}/\Delta_{\mathbf{s}}$), we may define functional bisimulations by analogy with Definition 4.15.

Definition 7.7. A morphism $f : X \to Y$ in $\widehat{\mathbb{C}_{\mathbf{s}}}$ is a *functional bisimulation*, or a *fibration*, iff it enjoys the (weak) right lifting property w.r.t. $\mathbf{y}_{s_L} : \mathbf{y}_{\mathbf{s}(L)} \to \mathbf{y}_L$, for all $L \in \mathbb{C}_1$.

Proposition 7.8 (Price for our linguistic mischief)**.** *A dynamic signature preserves functional bisimulations (Definition 6.12) iff the induced functor* $\Sigma_1^{\mathbf{s}} : \Sigma_0\text{-}\mathbf{Trans} \to \widehat{\mathbb{C}_1}/\Delta_{\mathbf{s}}$ *does.*

Proof. The functor $\Sigma_1^{\mathbf{s}}$ maps any functional bisimulation $f : R \to Y$ to the square (6.1), and just as in Proposition 4.17 a morphism in $\widehat{\mathbb{C}_{\mathbf{s}}}$ is a functional bisimulation iff the corresponding square in $\widehat{\mathbb{C}_1}$ is a pointwise weak pullback. □

7.2. **Familiality.** In the previous sections, we have seen that functional bisimulations may be defined by lifting both in $\Sigma_0\text{-}\mathbf{Trans}$ and $\widehat{\mathbb{C}_1}/\Delta_{\mathbf{s}}$. We now want to exploit this to obtain a characterisation of preservation of functional bisimulations, which will then lead us to useful sufficient conditions.

For this, let us briefly recall familial functors, and show that the functors Σ_0 and $\Sigma_1^{\mathbf{s}} : \Sigma_0\text{-}\mathbf{Trans} \to \widehat{\mathbb{C}_1}/\Delta_{\mathbf{s}}$ induced by the dynamic signature for call-by-name λ-calculus are familial.

Familial functors are a generalisation of polynomial functors on sets, i.e., functors of the form $F(X) = \sum_{o \in O} X^{n_o}$, where O is a set of 'operations', and $n_o \in \mathbb{N}$ is the 'arity' of any $o \in O$. We want to generalise this to presheaf categories.

Example 7.9. Consider for example the 'free category' monad T on **Gph**. Analysing and abstracting over the definition of T, we will arrive at the notion of familial functor. Let us first recall that graphs are presheaves over the category

$$[0] \underset{t}{\overset{s}{\rightrightarrows}} [1].$$

T does not change the vertex set, and an edge of $T(G)$ is merely a path in G. Indexing this by the length of the path, we obtain

$$T(G)[0] \cong \mathbf{Gph}(\mathbf{y}_{[0]}, G) \qquad \text{and} \qquad T(G)[1] \cong \textstyle\sum_n \mathbf{Gph}([n], G),$$

where $[n]$ denotes the filiform graph $\bullet \to \bullet \ldots \to \bullet$ with n edges (which is consistent with $[0]$ and $[1]$ through the Yoneda embedding). Furthermore, the source of a path $[n] \to G$ is obtained as the composite

$$[0] \xrightarrow{s_n} [n] \to G,$$

where the first morphism selects the first vertex of the path. Similarly the target is obtained by precomposition with the morphism, say t_n, selecting the last vertex.

From these observations, let us now explain how the whole of T may be derived from

- the graph $T(1)$, which generalises the set of operations, and
- a functor $\mathrm{el}(T(1)) \to \mathbf{Gph}$, morally describing the arity of each operation,

where we recall from MacLane and Moerdijk [43]:

Definition 7.10. 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 a morphism $f \upharpoonright x' \colon (c, x) \to (c', x')$ for all $f \colon c \to c'$ such that $X(f)(x') = x$.

The graph $T(1)$ has a single vertex, and as many paths as we can derive from a single endo-edge on this vertex: $\mathbb{N}$, because there is one for each length. The category of elements of $T(1)$ thus looks like the following,

$$\begin{tikzcd}
{([1],0)} & {([1],1)} & \ldots & {([1],n)} & \ldots \\
 & & {([0],\star)} \arrow[ull, bend right, "t\upharpoonright 0"'] \arrow[ull, bend right, "s\upharpoonright 0"'] \arrow[ul, "s\upharpoonright 1"] \arrow[ul, "t\upharpoonright 1"'] \arrow[ur, bend left, "s\upharpoonright n"] \arrow[ur, "t\upharpoonright n"'] & &
\end{tikzcd}$$

and the assignments

$$([0], \star) \mapsto [0] \qquad\qquad ([1], n) \mapsto [n]$$

extend to a functor $E \colon \mathrm{el}(T(1)) \to \mathbf{Gph}$ by mapping each source or target map $([0], \star) \to ([1], n)$ to the corresponding map $[0] \to [n]$. This functor may be visualised as

$$\begin{tikzcd}
{[0]} & {[1]} & \ldots & {[n]} & \ldots \\
 & & {[0].} \arrow[ull, bend right, "t_0"'] \arrow[ull, bend right, "s_0"'] \arrow[ul, "s_1"] \arrow[ul, "t_1"'] \arrow[ur, bend left, "t_n"] \arrow[ur, "s_n"'] & &
\end{tikzcd}$$

The promised expression of T in terms of $T(1)$ and E is:

$$T(G)(c) \cong \sum_{o \in T(1)(c)} \mathbf{Gph}(E(c, o), G).$$

Definition 7.11. A functor $F\colon \mathscr{A} \to \widehat{\mathbb{C}}$ to some presheaf category is *familial* iff we have a natural isomorphism

$$F(X)(c) \cong \sum_{o \in O(c)} \mathscr{A}(E(c,o), X),$$

for some presheaf $O \in \widehat{\mathbb{C}}$ and functor $E\colon \mathrm{el}(O) \to \mathscr{A}$. The presheaf O is called the presheaf of *operations*, or the *spectrum* [17] of F, while E is called the *arity*, or *exponent* functor.

Remark 7.12. This definition is a bit elliptic, so let us make functoriality more explicit.

- Functoriality in X is by post-composition.
- For functoriality in c, for any $f\colon c \to d$ in $\mathbb{C}$ and $o \in O(d)$, letting $o' = o \cdot f$, we get a morphism $f \upharpoonright o\colon (c, o') \to (d, o)$ in $\mathrm{el}(O)$, hence a morphism $E(f \upharpoonright o)\colon E(c, o') \to E(d, o)$. Precomposition by this morphisms yields a map

  $$\mathscr{A}(E(d,o), X) \to \mathscr{A}(E(c,o'), X).$$

  Postcomposing with the obvious coproduct injections, and cotupling, we get the desired map

  $$\sum_{o \in O(d)} \mathscr{A}(E(d,o), X) \to \sum_{o' \in O(c)} \mathscr{A}(E(c,o'), X).$$

Remark 7.13. If $\mathscr{A}$ has a terminal object, we always have $O \cong F(1)$.

Example 7.14. Let us show that the endofunctor $\Sigma_0\colon \widehat{\mathbb{C}_0} \to \widehat{\mathbb{C}_0}$ from §3.1 for λ-calculus is familial, where we recall that $\mathbb{C}_0 = \mathbb{F}^{op}$. Indeed, we then have

$$\begin{array}{rcl} \Sigma_0(X)(n) & = & X(n+1) + X(n)^2 \\ & \cong & \widehat{\mathbb{C}_0}(\mathbf{y}_{n+1}, X) + \widehat{\mathbb{C}_0}(2 \cdot \mathbf{y}_n, X) \end{array}$$

Thus, we choose:

$$\begin{array}{rcl} O(n) & = & \{\mathrm{abs}, \mathrm{app}\} \end{array} \qquad \begin{array}{rcl} E(n, \mathrm{abs}) & = & \mathbf{y}_{n+1} \\ E(n, \mathrm{app}) & = & 2 \cdot \mathbf{y}_n. \end{array}$$

These definitions can be straightforwardly upgraded to functors $O \in \widehat{\mathbb{C}_0}$ and $E\colon \mathrm{el}(O) \to \widehat{\mathbb{C}_0}$, and we get the desired isomorphism.

Example 7.15. Let us now show that the functor $\Sigma_1^{\mathrm{s}}\colon \Sigma_0\text{-}\mathbf{Trans} \to \widehat{\mathbb{C}_1}/\Delta_{\mathrm{s}}$ for call-by-name λ-calculus is familial. By definition, it maps any X to the set-map $X_0(1) + A_\beta(X) \to \Sigma_0(X_0)(0)$ defined in Example 3.20. Let us transfer this across the isomorphism $\widehat{\mathbb{C}_1}/\Delta_{\mathrm{s}} \cong \widehat{\mathbb{C}_{\mathrm{s}}}$ (recalling Remark 7.3), and show that Σ_1^{s} may be expressed as in Definition 7.11, first for $c \in \mathbb{F}$ and then for $c = \Downarrow$.

For $c = n \in \mathbb{F}$, we almost may proceed as for Σ_0, except that the domain category has changed (from $\widehat{\mathbb{C}_0}$ to $\Sigma_0\text{-}\mathbf{Trans}$). But recalling that $\mathscr{L}\colon \mathbf{C} \to \Sigma_0\text{-}\mathbf{Trans}$ denotes the left adjoint to the forgetful functor $\mathscr{U}$, we have

$$\begin{array}{rcl} \Sigma_1^{\mathrm{s}}(X)(n) & = & \Sigma_0((\mathscr{U}(X))_0)(n) \\ & \cong & \mathbf{C}(\mathbf{y}_{n+1}, \mathscr{U}(X)) + \mathbf{C}(2 \cdot \mathbf{y}_n, \mathscr{U}(X)) \\ & \cong & \Sigma_0\text{-}\mathbf{Trans}(\mathscr{L}(\mathbf{y}_{n+1}), X) + \Sigma_0\text{-}\mathbf{Trans}(\mathscr{L}(2 \cdot \mathbf{y}_n), X), \end{array}$$

so we may (partially) define

$$\begin{array}{ll} O(n) = \{\mathrm{abs}, \mathrm{app}\} & E(n, \mathrm{abs}) = \mathscr{L}(\mathbf{y}_{n+1}) \\ & E(n, \mathrm{app}) = \mathscr{L}(2 \cdot \mathbf{y}_n). \end{array} \tag{7.2}$$

Now, for $c = \Downarrow$, remembering from Notation 7.4 that we call $\Downarrow \in \mathbb{C}_{\mathbf{s}}$ the unique object of $\mathbb{C}_{\mathbf{s}}$ coming from $\mathbb{C}_1 = 1$, on transitions, we have:

$$\begin{aligned}\Sigma_1^{\mathbf{s}}(X)(\Downarrow) &= X_0(1) + A_\beta(X)\\ &\cong \mathbf{C}(\mathbf{y}_1, \mathscr{U}(X)) + A_\beta(X)\\ &\cong \Sigma_0\text{-}\mathbf{Trans}(\mathscr{L}(\mathbf{y}_1), X) + A_\beta(X).\end{aligned}$$

We thus need to find E_β such that $A_\beta(X) \cong \Sigma_0\text{-}\mathbf{Trans}(E_\beta, X)$, and then we would complete equations (7.2) with: $O(\Downarrow) = \{\lambda\text{-val}, \beta\text{-red}\}$ $\quad E(\Downarrow, \lambda\text{-val}) = \mathscr{L}(\mathbf{y}_1)$, $E(\Downarrow, \beta\text{-red}) = E_\beta$.

Definition 7.16. The morphism $\bar{\chi}\colon \mathscr{L}(\mathbf{y}_0) \to \mathscr{L}(\mathbf{y}_1 + \mathbf{y}_0)$ is defined as follows.

- We start from $k_1(\!|k_0|\!) \in \mathscr{U}(\mathscr{L}(\mathbf{y}_1 + \mathbf{y}_0))(0)$, where we recall from the rules below Proposition 3.4 that the presheaf $\mathscr{U}(\mathscr{L}(\mathbf{y}_1 + \mathbf{y}_0))(0)$ has as states λ-terms over a closed constant k_0, and a unary constant k_1.
- We then let $\bar{\chi}$ denote the mate of the morphism $\mathbf{y}_0 \to \mathscr{U}(\mathscr{L}(\mathbf{y}_1 + \mathbf{y}_0))$ corresponding to $k_1(\!|k_0|\!)$ by the Yoneda lemma.

Let now E_β denote the following pushout,

$$\begin{tikzcd}
\mathscr{L}(\mathbf{y}_0) \arrow[r, "\bar{\chi}"] \arrow[d, "\mathscr{L}(\mathbf{y}_{s_\Downarrow})"'] & \mathscr{L}(\mathbf{y}_1 + \mathbf{y}_0) \arrow[r, "\mathscr{L}(\mathbf{y}_{t_\Downarrow} + \mathbf{y}_0)"] & \mathscr{L}(\mathbf{y}_\Downarrow + \mathbf{y}_0) \arrow[d, "in_2"] \\
\mathscr{L}(\mathbf{y}_\Downarrow) \arrow[rr, "in_1"'] & & E_\beta
\end{tikzcd}$$

which exists because by Proposition 5.9 $\Sigma_0\text{-}\mathbf{Trans}$ is the category of algebras for a finitary monad on a presheaf category, hence a locally finitely presentable category by [2, Remark in §2.78], hence cocomplete.

Let us now show that $A_\beta(X) \cong \Sigma_0\text{-}\mathbf{Trans}(E_\beta, X)$ for any X: as $\Sigma_0\text{-}\mathbf{Trans}(-, X)$ turns colimits into limits, we have the pullback

$$\begin{tikzcd}
{[E_\beta, X]} \arrow[rr] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & & {[\mathscr{L}(\mathbf{y}_\Downarrow), X]} \arrow[d, "{[\mathscr{L}(\mathbf{y}_{s_\Downarrow}), X]}"] \\
{[\mathscr{L}(\mathbf{y}_\Downarrow + \mathbf{y}_0), X]} \arrow[r, "{[\mathscr{L}(\mathbf{y}_{t_\Downarrow} + \mathbf{y}_0), X]}"'] & {[\mathscr{L}(\mathbf{y}_1 + \mathbf{y}_0), X]} \arrow[r, "{[\bar{\chi}, X]}"'] & {[\mathscr{L}(\mathbf{y}_0), X]},
\end{tikzcd}$$

where $\bar{\chi}$ is as in Definition 7.16, and we abbreviate $\Sigma_0\text{-}\mathbf{Trans}(-_1, -_2)$ to $[-_1, -_2]$ for readability. By Yoneda, this reduces to

$$\begin{tikzcd}
{[E_\beta, X]} \arrow[rr] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & & X(\Downarrow) \arrow[d, "X(s_\Downarrow)"] \\
X(\Downarrow) \times X(0) \arrow[r, "X(t_\Downarrow) \times X(0)"'] & X(1) \times X(0) \arrow[r, "(t,u) \mapsto t[u]"'] & X(0),
\end{tikzcd}$$

which shows that we have $A_\beta(X) \cong [E_\beta, X]$ as desired.

We have thus defined the actions of the functors $O \in \widehat{\mathbb{C}_{\mathbf{s}}}$ and $E\colon \mathrm{el}(O) \to \Sigma_0\text{-}\mathbf{Trans}$ on objects. On morphisms, the only non-obvious point is the image of $s_\Downarrow \upharpoonright \lambda\text{-val}$ and $s_\Downarrow \upharpoonright \beta\text{-red}$. The former morphism is mapped to the identity on $E(0, \mathrm{abs}) = \mathscr{L}(\mathbf{y}_1) = E(\Downarrow, \lambda\text{-val})$. The latter is mapped to the composite

$$\mathscr{L}(2 \cdot \mathbf{y}_0) \xrightarrow{\mathscr{L}(\mathbf{y}_{s_\Downarrow} + \mathbf{y}_0)} \mathscr{L}(\mathbf{y}_\Downarrow + \mathbf{y}_0) \xrightarrow{in_2} E_\beta. \tag{7.3}$$

This achieves the desired isomorphism $\Sigma_1^{\mathbf{s}}(X)(c) \cong \sum_{o \in O(c)} \Sigma_0\text{-}\mathbf{Trans}(E(c, o), X)$.

7.3. **Cellularity.** We now want to exploit familiality to obtain an alternative characterisation of preservation of functional bisimulations. The starting point is the observation that when a functor $F\colon \mathscr{A} \to \widehat{\mathbb{C}}$ is familial, say as $F(A)(c) = \sum_{o\in O(c)} \mathscr{A}(E(c,o),A)$, then any morphism of the form $f\colon \mathbf{y}_c \to F(A)$, corresponding by Yoneda and familiality to some pair (o,ϕ) with $\phi\colon E(c,o) \to A$, factors as

$$\mathbf{y}_c \xrightarrow{(o,\mathrm{id}_{E(c,o)})} F(E(c,o)) \xrightarrow{F(\phi)} F(A).$$

Furthermore, the first component $(o,\mathrm{id}_{E(c,o)})$ is easily seen to be *generic*, in the following sense.

Definition 7.17. 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 for all χ, f, and g making the square below (solid) commute,

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

there is a unique lifting k (dashed) such that $F(k)\circ\xi = \chi$ and $g\circ k = f$.

In fact, we have the following important alternative characterisation of familial functors to presheaf categories.

Theorem 7.18. *For any functor $F\colon \mathscr{A} \to \widehat{\mathbb{C}}$ such that $\mathscr{A}$ has a terminal object, F is familial iff all morphisms $f\colon X \to F(A)$ factor as*

$$X \xrightarrow{\xi} F(U) \xrightarrow{F(\phi)} F(A),$$

with ξ generic.

Proof. This is [25, Proposition 3.8], using the remark just before it. □

Remark 7.19. The factorisation is essentially unique.

The characterisation of familiality in terms of generic morphisms allows us to characterise familial functors which preserve fibrations, as cellular functors, which we now introduce.

Definition 7.20. A *category with generating cofibrations* is a category $\mathscr{C}$ equipped with a set $\mathbb{J}$ of morphisms.

For any such $(\mathscr{C},\mathbb{J})$, as in §1.6, we call *fibrations* all morphisms in $\mathbb{J}^{\pitchfork}$, and *cofibrations* all morphisms in ${}^{\pitchfork}(\mathbb{J}^{\pitchfork})$.

Example 7.21. In Σ_0**-Trans**, with generating cofibrations consisting of all morphisms $\mathscr{L}(\mathbf{y}_{s_L})$, fibrations are precisely functional bisimulations, by Proposition 6.2.

Example 7.22. In $\widehat{\mathbb{C}_s}$, with generating cofibrations consisting of all morphisms $\mathbf{y}_{s_L}$, fibrations are precisely functional bisimulations, by Definition 7.7.

Definition 7.23. For any categories with generating cofibrations $(\mathscr{A},\mathbb{J})$ and $(\widehat{\mathbb{C}},\mathbb{K})$, such that $\mathbb{C}$ is small and $\mathscr{A}$ has a terminal object, a familial functor $F\colon \mathscr{A} \to \widehat{\mathbb{C}}$ is *cellular* iff for all commuting squares

$$\begin{tikzcd} C \arrow[r, "k"] \arrow[d, "\xi"'] & D \arrow[d, "\chi"] \\ F(X) \arrow[r, "F(\delta)"'] & F(Y) \end{tikzcd} \tag{7.4}$$

with $k \in \mathbb{K}$ and ξ and χ generic, δ is a cofibration (i.e., $\delta \in {}^{\pitchfork}(\mathbb{J}^{\pitchfork})$).

Before exploiting cellularity as promised, let us briefly pause to give an equivalent characterisation of cellularity in suitably nice cases.

Definition 7.24. Let us consider any familial functor $F\colon \mathscr{A} \to \widehat{\mathbb{C}}$, say as $F(A)(c) = \sum_{o \in O(c)} \mathscr{A}(E(c,o), A)$, such that $\mathbb{C}$ is small and $\mathscr{A}$ has a terminal object. Then, for any operation $o \in F(1)(d)$ and morphism $s\colon c \to d$ in $\mathbb{C}$, the *s-arity* of o, or its *boundary arity* when s is clear from context, is the morphism $E(d, o \upharpoonright s)\colon E(c, o \cdot s) \to E(d,o)$ in $\mathscr{A}$.

Notation 7.25. *Let χ_o denote the generic morphism $\mathbf{y}_d \to F(E(d,o))$ induced by any operation $o \in F(1)(d)$.*

By construction, for any o and s as in the definition, if $\mathbf{y}_s$ is a generating cofibration (i.e., is in $\mathbb{K}$), then the diagram

$$\begin{tikzcd} \mathbf{y}_c \arrow[r, "\mathbf{y}_s"] \arrow[d, "\chi_{o\cdot s}"'] & \mathbf{y}_d \arrow[d, "\chi_o"] \\ F(E(c, o\cdot s)) \arrow[r, "F(E(d,o\upharpoonright s))"'] & F(E(d,o)) \end{tikzcd}$$

is of the form (7.4) by construction. We thus have by definition:

Proposition 7.26. *Let us consider any cellular $F\colon \mathscr{A} \to \widehat{\mathbb{C}}$ between categories with generating cofibrations $(\mathscr{A}, \mathbb{J})$ and $(\widehat{\mathbb{C}}, \mathbb{K})$ such that $\mathbb{C}$ is small and $\mathscr{A}$ has a terminal object. Then, for any representable morphism $\mathbf{y}_s\colon \mathbf{y}_c \to \mathbf{y}_d$ in $\mathbb{K}$, the s-arity of any operation $o \in F(1)(d)$ is a cofibration.*

When all domains and codomains of generating cofibrations $k \in \mathbb{K}$ are representable, say as $\mathbf{y}_s\colon \mathbf{y}_c \to \mathbf{y}_d$, then all generic morphisms $\chi\colon D \to F(Y)$, with D a codomain of some generating cofibration, are isomorphic to some $\mathbf{y}_d \to F(E(d,o))$, for some operation $o \in F(1)(d)$, and the morphism δ must then be isomorphic to $E(d, o \upharpoonright s)\colon E(c, o\cdot s) \to E(d,o)$. We thus get a partial converse to Proposition 7.26:

Proposition 7.27. *For any categories with generating cofibrations $(\mathscr{A}, \mathbb{J})$ and $(\widehat{\mathbb{C}}, \mathbb{K})$, such that $\mathbb{C}$ is small, $\mathscr{A}$ has a terminal object, and all domains and codomains of generating cofibrations $k \in \mathbb{K}$ are representable, a familial functor $F\colon \mathscr{A} \to \widehat{\mathbb{C}}$ is cellular iff the boundary arities of all operations are cofibrations.*

Returning to the general case, let us now prove a first characterisation of preservation of fibrations in terms of cellularity.

Lemma 7.28. *For any categories with generating cofibrations $(\mathscr{A}, \mathbb{J})$ and $(\widehat{\mathbb{C}}, \mathbb{K})$ such that $\mathbb{C}$ is small and $\mathscr{A}$ has a terminal object, a familial functor $F\colon \mathscr{A} \to \widehat{\mathbb{C}}$ preserves fibrations iff it is cellular.*

Proof. Let us first prove the 'if' direction. We must show that for any $f\colon A \to B$ in $\mathbb{J}^{\pitchfork}$, $F(f)$ is in $\mathbb{K}^{\pitchfork}$, i.e., that any commuting square

$$\begin{tikzcd} C \arrow[r, "u"] \arrow[d, "k"'] & F(A) \arrow[d, "F(f)"] \\ D \arrow[r, "v"'] & F(B) \end{tikzcd} \tag{7.5}$$

with $k \in \mathbb{K}$ and $f \in \mathbb{J}^{\pitchfork}$ admits a lifting. But taking generic factorisations of both horizontal morphisms and using genericness, any such square factors as the solid part of

$$\begin{tikzcd}
C \arrow[r, "\xi"] \arrow[d, "k"'] & F(X) \arrow[r, "F(\phi)"] \arrow[d, "F(\delta)"] & F(A) \arrow[d, "F(f)"] \\
D \arrow[r, "\chi"'] & F(Y) \arrow[r, "F(\psi)"'] \arrow[ur, dashed, "F(l)"] & F(B).
\end{tikzcd}$$

By cellularity, we have $\delta \in {}^{\pitchfork}(\mathbb{J}^{\pitchfork})$. We thus find a lifting l as shown, which makes $F(l) \circ \chi$ into a lifting for the original square.

Conversely, let us assume that F preserves fibrations, and consider any square of the form (7.4) with $k \in \mathbb{K}$. We need to show $\delta \in {}^{\pitchfork}(\mathbb{J}^{\pitchfork})$. But for any commuting square as below left

$$\begin{tikzcd}
X \arrow[r, "\phi"] \arrow[d, "\delta"'] & A \arrow[d, "f"] \\
Y \arrow[r, "\psi"'] & B
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
C \arrow[r, "\xi"] \arrow[d, "k"'] & F(X) \arrow[r, "F(\phi)"] \arrow[d, "F(\delta)"] & F(A) \arrow[d, "F(f)"] \\
D \arrow[r, "\chi"'] \arrow[ur, dashed, "\gamma"] & F(Y) \arrow[r, "F(\psi)"'] \arrow[ur, dashed, "F(l)"] & F(B)
\end{tikzcd}$$

with $f \in \mathbb{J}^{\pitchfork}$, by pasting this square with our generic square (7.4), we obtain the solid part above right. Finally, because F preserves fibrations, we find a lifting γ as shown, which by genericness of χ (and then ξ) yields the desired lifting l. □

Corollary 7.29. *For any categories with generating cofibrations $(\mathscr{A}, \mathbb{J})$ and $(\widehat{\mathbb{C}}, \mathbb{K})$, such that $\mathbb{C}$ is small, $\mathscr{A}$ has a terminal object, and all domains and codomains of generating cofibrations $k \in \mathbb{K}$ are representable, a familial functor $F \colon \mathscr{A} \to \widehat{\mathbb{C}}$ preserves fibrations iff the boundary arities of all operations are cofibrations.*

Proof. By Proposition 7.27 and Lemma 7.28. □

Here is the announced characterisation of preservation of functional bisimulations, which follows directly from Lemma 7.28 and Corollary 7.29.

Corollary 7.30. *In any Howe context, for any operational semantics signature (Σ_0, Σ_1), let $\Sigma_1^{\mathbf{s}} \colon \Sigma_0\text{-}\mathbf{Trans} \to \widehat{\mathbb{C}_{\mathbf{s}}}$ be familial with exponent $E \colon \mathrm{el}(\Sigma_1^{\mathbf{s}}(1)) \to \Sigma_0\text{-}\mathbf{Trans}$. Then the following are equivalent:*

(i) *$\Sigma_1^{\mathbf{s}}$ preserves functional bisimulations;*
(ii) *$\Sigma_1^{\mathbf{s}}$ is cellular;*
(iii) *the boundary arities of all operations are cofibrations, i.e., for all $L \in \mathbb{C}_1$ and $o \in \Sigma_1^{\mathbf{s}}(1)(L)$, the morphism*

$$E(s_L \upharpoonright o) \colon E(\mathbf{s}(L), o \cdot s_L) \to E(L, o)$$

is a cofibration.

This characterisation of preservation of functional bisimulations in terms of cofibrations is easier to prove in practice, since cofibrations in turn admit the following well-known characterisation.

Definition 7.31. Consider any set $\mathbb{J}$ of maps in a given category.

- A *basic relative $\mathbb{J}$-cell complex* is any morphism f obtained by pushing out some morphism from $\mathbb{J}$ along any morphism, as in

$$\begin{tikzcd}
A \arrow[r, "j \in \mathbb{J}"] \arrow[d] & B \arrow[d] \\
C \arrow[r, "f"'] & D.
\end{tikzcd}$$

- A *relative $\mathbb{J}$-cell complex* is a (potentially transfinite) composite of basic relative $\mathbb{J}$-cell complexes.

Proposition 7.32 ([34, Lemma 2.1.10])**.** *For any set $\mathbb{J}$ of maps in a locally presentable category, all relative $\mathbb{J}$-cell complexes are cofibrations in the generated weak factorisation system $({}^{\pitchfork}(\mathbb{J}^{\pitchfork}), \mathbb{J}^{\pitchfork})$.*

8. Applications

In this section, we apply our results to show that substitution-closed bisimilarity is a congruence in concrete examples.

8.1. **Call-by-name.** We have already specified the syntax (§3.1) and transitions (§3.3) of call-by-name λ-calculus. We have also seen (Example 7.15) that the induced functor $\Sigma_1^{\mathbf{s}} : \Sigma_0\text{-}\mathbf{Trans} \to \widehat{\mathbb{C}_1}/\Delta_{\mathbf{s}}$ is familial. By Theorem 6.15, Corollary 7.30, and Proposition 7.32, congruence of substitution-closed bisimilarity will follow if we prove that the boundary arities $E(s_{\Downarrow} \upharpoonright o)$ corresponding to both transition rules are relative cell complexes.

The boundary arity $E(s_{\Downarrow} \upharpoonright \lambda\text{-val})$ is an identity, hence trivially a relative cell complex.

For the β-rule, it is not entirely trivial that the boundary arity $E(s_{\Downarrow} \upharpoonright \beta\text{-red})$ of the second transition rule, defined as the composite (7.3), is a cofibration. But, as (essentially) noted in [14, Example 5.21], it is a relative cell complex by construction, as both components are pushouts of $\mathscr{L}(\mathbf{y}_{s_{\Downarrow}})$, as should be clear from the following diagram, where again $\bar{\chi}$ is as in Definition 7.16.

$$\begin{tikzcd}
 & \mathscr{L}(\mathbf{y}_0) \arrow[r, "\mathscr{L}(\mathbf{y}_{s_{\Downarrow}})"] \arrow[d, "\mathscr{L}(\mathbf{y}_{t_{\Downarrow}}+\mathbf{y}_0)\circ\bar{\chi}"'] & \mathscr{L}(\mathbf{y}_{\Downarrow}) \arrow[d] \\
\mathscr{L}(\mathbf{y}_0+\mathbf{y}_0) \arrow[r, "\mathscr{L}(\mathbf{y}_{s_{\Downarrow}}+\mathbf{y}_0)"'] & \mathscr{L}(\mathbf{y}_{\Downarrow}+\mathbf{y}_0) \arrow[r] \arrow[ur, phantom, "\ulcorner"] & E_\beta
\end{tikzcd}$$

8.2. **Call-by-value.** Let us now treat the call-by-value variant of untyped λ-calculus, essentially as in [46, 47]. In this setting, it is important to distinguish substitution by values and by terms. Indeed, letting I denote the identity $\lambda x.x$, the terms $e = \lambda x.I$ and $e' = \lambda x.((\lambda y.I)\ x)$ are contextually equivalent, since during evaluation in any context, the bound variable x will only be replaced with a value. However, if one defines applicative bisimulation naively, i.e., requiring it to be closed under arbitrary substitution, then e and e' are not bisimilar, as I is not bisimilar to $(\lambda y.I)\ \Omega$ – which diverges. We thus want to restrict to value substitution – which our treatment in [14] overlooks!

Here is one way of doing this. The idea is to have two sorts, one for values and the other for general terms. We thus would like $\widehat{\mathbb{C}_0}$ to be equivalent to the category $[\mathbf{Set}^2, \mathbf{Set}^2]_f$ of finitary endofunctors on $\mathbf{Set}^2$. But

$$[\mathbf{Set}^2, \mathbf{Set}^2]_f \simeq [2 \cdot \mathbf{Set}^2_f, \mathbf{Set}] \simeq [2 \cdot \mathbb{F}^2, \mathbf{Set}],$$

so we take $\mathbb{C}_0$ to be the opposite of $2 \cdot \mathbb{F}^2$. By the equivalence, composition of finitary endofunctors equips $\widehat{\mathbb{C}_0}$ with monoidal structure, and we denote the two sorts by $\mathbf{p}$ and $\mathbf{v}$, respectively for "programs" and "values". In the presheaf point of view, we denote by $(m,n)_{\mathbf{p}}$ and $(m,n)_{\mathbf{v}}$, respectively, objects in the first and second term of $2 \times \mathbb{F}^2 = \mathbb{F}^2 + \mathbb{F}^2$, so that $X(m,n)(c)$ in the finitary endofunctor world corresponds to $X(m,n)_c$ in the presheaf world,

for $c \in \{\mathbf{v}, \mathbf{p}\}$. We think of $X(m,n)_{\mathbf{v}}$ (resp. $X(m,n)_{\mathbf{p}}$) as a set of values (resp. programs) with m potential free program variables, and n potential free value variables.

Since abstraction should bind a value variable, the syntax

$$\begin{array}{rcl} e, f & ::= & e\ f \mid v \\ v & ::= & x \mid \lambda x.e \end{array}$$

is thus specified by

$$\Sigma_0(X)(m,n)(\mathbf{p}) = X(m,n)(\mathbf{p})^2 + X(m,n)(\mathbf{v}) \qquad \text{and} \qquad \Sigma_0(X)(m,n)(\mathbf{v}) = X(m,n+1)(\mathbf{p}).$$

A model X of the syntax should thus in particular feature operations

$$\mathrm{abs}_{m,n} \colon X(m,n+1)(\mathbf{p}) \to X(m,n)(\mathbf{v}) \qquad \mathrm{app}_{m,n} \colon X(m,n)(\mathbf{p})^2 \to X(m,n)(\mathbf{p})$$
$$\mathrm{val}_{m,n} \colon X(m,n)(\mathbf{v}) \to X(m,n)(\mathbf{p}),$$

where the last operation requires that values should embed into programs.

Notation 8.1. *We implicitly view any $X \in [2 \cdot \mathbb{F}^2, \mathbf{Set}]$ as (some fixed, global choice of) the corresponding functor $\mathbf{Set}^2 \to \mathbf{Set}^2$. In particular, we write $X(K)$, for any $K \in \mathbf{Set}^2$. Accordingly, we view pairs (m,n) as objects of $\mathbf{Set}^2$. E.g., in this sense, $(m,n+1)$ is isomorphic to $(m,n) + \mathbf{y}_{\mathbf{v}}$, where $\mathbf{y}_{\mathbf{v}}$ denotes the Yoneda embedding of $\mathbf{v}$ along $2 \to \mathbb{F}^2 \hookrightarrow \mathbf{Set}^2$. So the arity for abstraction in fact yields an operation $X(K + \mathbf{y}_{\mathbf{v}})(\mathbf{p}) \to X(K)(\mathbf{v})$.*

Denoting composition of finitary endofunctors by $\otimes$, we define a pointed strength for Σ_0 as follows. For any $X \in \widehat{\mathbb{C}_0}$, $Y \in I/\widehat{\mathbb{C}_0}$, and $K \in \mathbb{F}^2$:

- at $\mathbf{p}$, we have
$$(\Sigma_0(X) \otimes Y)(K)(\mathbf{p}) = X(Y(K))(\mathbf{p})^2 + X(Y(K))(\mathbf{v}) = \Sigma_0(X \otimes Y)(K)(\mathbf{p}),$$
so we take $st_{X,Y,K,\mathbf{p}}$ to be the identity;
- at $\mathbf{v}$, we have
$$(\Sigma_0(X) \otimes Y)(K)(\mathbf{v}) = X(Y(K) + \mathbf{y}_{\mathbf{v}})(\mathbf{p}) \quad \text{and} \quad \Sigma_0(X \otimes Y)(K)(\mathbf{v}) = X(Y(K + \mathbf{y}_{\mathbf{v}}))(\mathbf{p}),$$
so we define $st_{X,Y,K,\mathbf{p}}$ by applying $X(-)(\mathbf{p})$ to the copairing of $Y(K) \to Y(K + \mathbf{y}_{\mathbf{v}})$ and
$$\mathbf{y}_{\mathbf{v}} \to I(K + \mathbf{y}_{\mathbf{v}}) \to Y(K + \mathbf{y}_{\mathbf{v}}).$$

We then specify transitions, which we first recall:

$$\frac{}{\lambda x_1.e \Downarrow e} \qquad\qquad \frac{e_1 \Downarrow e_1' \qquad e_2 \Downarrow e_2' \qquad e_1'[\lambda x_1.e_2'] \Downarrow e_3}{e_1\ e_2 \Downarrow e_3}\ .$$

Remark 8.2. Here, we adopt the same convention as in the categorical picture, where terms are implicitly considered as indexed over sets, say $\{x_1, \ldots, x_n\}$, of potential free variables. Furthermore, the evaluation relation again relates closed terms to terms over one potential free variable x_1.

In order to specify such transitions, we take $\mathbb{C}_1 = 1$, as in the call-by-name case, with $\mathbf{s}$ and $\mathbf{t}$ mapping the unique object to $(0,0)_{\mathbf{p}}$ and $(0,1)_{\mathbf{p}}$, respectively: transitions will relate a closed program to a program with one value variable (morally the body of the obtained abstraction). For any $X \in \mathbf{C} = \widehat{\mathbb{C}_1}/\Delta \cong \mathbf{Set}/\Delta$, We take $\Sigma_1^F(X)$ to be the coproduct

$$\Sigma_1^F(X) = X_0(0,1) + A_{\beta_v}(X),$$

where $A_{\beta_v}(X)$ denotes the set of valid premises for the second rule, i.e., triples $(r_1, r_2, r_3) \in X_1$ such that

$$r_3 \cdot s_{\Downarrow} = (r_1 \cdot t_{\Downarrow})[\lambda(r_2 \cdot t_{\Downarrow})].$$

Familiality of the induced functor $\Sigma_1^{\mathrm{s}} \colon \Sigma_0\text{-}\mathbf{Trans} \to \widehat{\mathbb{C}_1}/\Delta_{\mathrm{s}}$ follows similarly to the call-by-name case, from the fact that $A_{\beta_v}(X)$ is isomorphic to $[E_{\beta_v}, X]$, where E_{β_v} denotes the following pushout.

$$\begin{tikzcd} \mathscr{L}(\mathbf{y}_0) \arrow[r, "\mathscr{L}(\mathbf{y}_{s_{\Downarrow}})"] \arrow[d, "\xi"'] & \mathscr{L}(\mathbf{y}_{\Downarrow}) \arrow[d] \\ \mathscr{L}(\mathbf{y}_{\Downarrow} + \mathbf{y}_{\Downarrow}) \arrow[r] \arrow[ur, phantom, "\ulcorner"] & E_{\beta_v} \end{tikzcd}$$

Here, ξ corresponds by adjunction and Yoneda to the closed term

$$(r_1 \cdot t_{\Downarrow})[\lambda(r_2 \cdot t_{\Downarrow})] \in \mathscr{U}(\mathscr{L}(\mathbf{y}_{\Downarrow} + \mathbf{y}_{\Downarrow}))(0),$$

where r_1 and r_2 denote the two transition constants generating $\mathbf{y}_{\Downarrow} + \mathbf{y}_{\Downarrow}$.

By Theorem 6.15, Corollary 7.30, and Proposition 7.32, congruence of substitution-closed bisimilarity will follow if we can prove that boundary arities $E(s_L \restriction o)$ of both transition rules are relative cell complexes. This is again trivial for the first rule, while for the second one we obtain the following morphism, which is a relative cell complex by construction.

$$\begin{tikzcd} & \mathscr{L}(\mathbf{y}_0) \arrow[r, "\mathscr{L}(\mathbf{y}_{s_{\Downarrow}})"] \arrow[d, "\xi"'] & \mathscr{L}(\mathbf{y}_{\Downarrow}) \arrow[d] \\ \mathscr{L}(\mathbf{y}_0 + \mathbf{y}_0) \arrow[r, "\mathscr{L}(\mathbf{y}_{s_{\Downarrow}} + \mathbf{y}_{s_{\Downarrow}})"'] & \mathscr{L}(\mathbf{y}_{\Downarrow} + \mathbf{y}_{\Downarrow}) \arrow[r] \arrow[ur, phantom, "\ulcorner"] & E_{\beta_v} \end{tikzcd}$$

8.3. **Erratic non-determinism.** In this section, we consider the non-deterministic λ-calculus investigated in [50, §7]. Its syntax is that of pure λ-calculus, augmented with a unary operation $\uplus$, and its reduction rules [50, on pages 125 and 142] are

$$\frac{}{\uplus e \Rightarrow e} \qquad \frac{}{\uplus e \Rightarrow \Omega} \qquad \frac{}{(\lambda x.e)\ f \Rightarrow e[f]}$$

$$\frac{e_1 \Rightarrow e_3}{e_1\ e_2 \Rightarrow e_3\ e_2} \qquad \frac{}{e \Rightarrow e} \qquad \frac{e_1 \Rightarrow e_2 \qquad e_2 \Rightarrow e_3}{e_1 \Rightarrow e_3},$$

where Ω denotes any diverging term.

Let us start by giving a big-step presentation of this language. We consider a labelled transition relation, i.e., we have two transition relations $\Downarrow_\lambda$ (between a closed term and a term with one free variable, as before) and $\Downarrow_\tau$ (between closed terms), inductively generated by the following rules.

$$\frac{}{\uplus e \Downarrow_\tau e} \qquad \frac{}{\uplus e \Downarrow_\tau \Omega} \qquad \frac{}{\lambda x.e \Downarrow_\lambda e} \qquad \frac{e_1 \Downarrow_\lambda e_3 \qquad e_3[e_2] \Downarrow_\lambda e_4}{e_1\ e_2 \Downarrow_\lambda e_4} \qquad \frac{e_1 \Downarrow_\lambda e_3 \qquad e_3[e_2] \Downarrow_\tau e_4}{e_1\ e_2 \Downarrow_\tau e_4}$$

$$\frac{e_1 \Downarrow_\tau e_3}{e_1\ e_2 \Downarrow_\tau e_3\ e_2} \qquad \frac{}{e \Downarrow_\tau e} \qquad \frac{e_1 \Downarrow_\tau e_2 \qquad e_2 \Downarrow_\tau e_3}{e_1 \Downarrow_\tau e_3} \qquad \frac{e_1 \Downarrow_\tau e_2 \qquad e_2 \Downarrow_\lambda e_3}{e_1 \Downarrow_\lambda e_3}.$$

We have:

Proposition 8.3. *A relation is an applicative bisimulation in Sangiorgi's sense, say a* Sangiorgi bisimulation*, iff its open extension is a substitution-closed bisimulation with the new rules.*

Proof. This is an easy corollary of the following lemma. □

Lemma 8.4. *For all closed e_1 and e_2, and e_3 with one potential free variable, we have*

- $e_1 \Rightarrow e_2$ *iff* $e_1 \Downarrow_\tau e_2$, *and*
- $e_1 \Rightarrow \lambda(e_3)$ *iff* $e_1 \Downarrow_\lambda e_3$.

Proof.

- All reduction rules also occur as some new rule for $\Downarrow_\tau$, except the β-rule, which is easily derivable. We thus have $\Rightarrow \subseteq \Downarrow_\tau$.
- Similarly, if $e_1 \Rightarrow \lambda(e_3)$, then by the previous point $e_1 \Downarrow_\tau \lambda(e_3)$, hence by the last rule $e_1 \Downarrow_\lambda e_3$.
- Finally, let us prove both converse statements in one go by induction on the transition proof.
  - The first two axioms also are axioms in the original rules, hence easy.
  - For the third axiom, we are given $\lambda(e) \Downarrow_\lambda e$, hence clearly $\lambda(e) \Rightarrow \lambda(e)$ as desired.
  - For the first β rule, $e_1 = e_4\ e_5$, and we know by induction hypothesis that $e_4 \Rightarrow \lambda(e_6)$ and $e_6[e_5] \Rightarrow \lambda(e_3)$. We thus get $e_4\ e_5 \Rightarrow \lambda(e_6)\ e_5 \Rightarrow e_6[e_5] \Rightarrow \lambda(e_3)$, hence the desired result.
  - The second β rule is similar, except that we get a chain $e_4\ e_5 \Rightarrow \lambda(e_6)\ e_5 \Rightarrow e_6[e_5] \Rightarrow e_2$.
  - The next three rules also occur as original rules, hence are easily dealt with by induction hypothesis.
  - For the last rule, we have by induction hypothesis $e_1 \Rightarrow e_4$ and $e_4 \Rightarrow \lambda(e_3)$, hence $e_1 \Rightarrow \lambda(e_3)$, as desired. □

The syntax is easily modelled by taking $\Sigma_0(X)(n) = X(n+1) + X(n)^2 + X(n)$, and the new rules are easily seen to fit into a Σ_1 such that Σ_1^{s} preserves functional bisimulations by Corollary 7.30 and Proposition 7.32. Finally, substitution-closed bisimilarity in the initial model coincides with Sangiorgi's applicative bisimilarity, so we again deduce congruence of applicative bisimilarity by Theorem 6.15.

8.4. **Howe's format.** In this section, we show that, up to suitable encoding, our framework covers languages complying with a format proposed by Howe [35, Lemma 6.1] — or rather a slight variant thereof (see Remark 8.27 below). We first introduce the format, and then explain how to embed it into our framework. The format illustrates in which sense cellularity is in particular an acyclicity condition. Furthermore, it would be easy, but much more verbose, to extend the format to a simply-typed setting; we refrain from doing so in this already quite long paper.

8.4.1. *Recalling Howe's format.* Regarding syntax, Howe's framework is parameterised by the choice between call-by-name and call-by-value. His signatures are then much like standard binding signatures [48]. We model this in a setting similar to the call-by-value setting of §8.2, except that this time we build into the base category the fact that values embed into terms. Specifically, we work with the monoidal category $[\mathbf{Set}^2, \mathbf{Set}^2]_f \simeq [2 \times \mathbb{F}^2, \mathbf{Set}]$, where 2 denotes the walking arrow category $\mathbf{v} \xrightarrow{\iota} \mathbf{p}$. We think of an object of the base category $2 \times \mathbb{F}^2$, which is a pair $(s, f\colon m \to n)$ with $s \in \{\mathbf{v}, \mathbf{p}\}$, as the index of terms of sort s, with m value variables and n program variables, each $f(i)$ being thought as the program counterpart of $i \in m$.

Let us first explain how the syntactic part of Howe's format may be understood in terms of pointed strong endofunctors. Howe's notion of syntactic signature encodes a choice between call-by-value and call-by-name, through the choice of a distinguished sort, $\mathbf{v}$ or $\mathbf{p}$.

Notation 8.5.

- *Given $m, n \in \mathbb{F}$ and $s \in \{\mathbf{v}, \mathbf{p}\}$, we denote any object $(s, (m \xrightarrow{f} n))$ of $\mathbb{2} \times \mathbb{F}^{\mathbb{2}}$ by $(m \xrightarrow{f} n)_s$, and the particular morphism $(\iota, \mathrm{id}) \colon (m \xrightarrow{f} n)_{\mathbf{v}} \to (m \xrightarrow{f} n)_{\mathbf{p}}$ by ι.*
- *Given $m, n \in \mathbb{F}$, we denote the object $(m \xrightarrow{in_1} m+n)$ of $\mathbb{F}^{\mathbb{2}}$ by (m, n). Otherwise said, we treat the corresponding embedding $\mathbb{F}^2 \hookrightarrow \mathbb{F}^{\mathbb{2}}$ as an implicit coercion.*
- *Accordingly, for any $m, n \in \mathbb{F}$ and $s \in \{\mathbf{v}, \mathbf{p}\}$, we denote $(m \xrightarrow{in_1} m+n)_s$ by $(m, n)_s$.*
- *Given $T \colon \mathbb{2} \times \mathbb{F}^{\mathbb{2}} \to \mathbf{Set}$ and $s \in \{\mathbf{v}, \mathbf{p}\}$, we denote by $T_s \colon \mathbb{F}^{\mathbb{2}} \to \mathbf{Set}$ the functor mapping $(m \xrightarrow{f} n)$ to $T(m \xrightarrow{f} n)_s$.*

Remark 8.6. Through the equivalence $[\mathbf{Set}^{\mathbb{2}}, \mathbf{Set}^{\mathbb{2}}]_f \simeq [\mathbb{2} \times \mathbb{F}^{\mathbb{2}}, \mathbf{Set}]$, composition of endofunctors becomes a monoidal product defined by

$$(F \otimes G)(m \xrightarrow{f} n)_s = \mathrm{Lan}_J(F_s)(G(m \xrightarrow{f} n)_{\mathbf{v}} \xrightarrow{G\iota} G(m \xrightarrow{f} n)_{\mathbf{p}}),$$

where $J : \mathbb{F}^2 \to \mathbf{Set}^{\mathbb{2}}$ is the canonical embedding. The unit I is defined by

$$I(n_{\mathbf{v}} \xrightarrow{f} n_{\mathbf{p}})_s = n_s.$$

for any object $(n_{\mathbf{v}} \xrightarrow{f} n_{\mathbf{p}})_s$. Intuitively, I merely returns the set of variables of each sort.

Howe distinguishes value operations[4] from program operations:

Definition 8.7. A *Howe binding signature* consists of

- a *binding sort* $s_v \in \{\mathbf{v}, \mathbf{p}\}$,
- a set $O_{\mathbf{p}}$ of *program operations*, equipped with a map $N^{\mathbf{p}} \colon O_{\mathbf{p}} \to \mathbb{N}$ and a family $d^{\mathbf{p}} \in \prod_{o \in O_{\mathbf{p}}} \mathbb{N}^{N^{\mathbf{p}}_o}$, and
- a set $O_{\mathbf{v}}$ of *value operations*, equipped with two maps $N^+ \colon O_{\mathbf{v}} \to \mathbb{N}$ and $N^- \colon O_{\mathbf{v}} \to \mathbb{N}$, and a family $d^- \in \prod_{o \in O_{\mathbf{v}}} \mathbb{N}^{N^-_o}$.

Terminology 8.8. For any operation $o \in O_{\mathbf{p}}$ (resp. $O_{\mathbf{v}}$), we call the sequence $(d^{\mathbf{p}}_{o,1}, \ldots, d^{\mathbf{p}}_{o,N^{\mathbf{p}}_o})$ (resp. $(0, \ldots, 0, d^-_{o,1}, \ldots, d^-_{o,N^-_o})$, with N^+_o leading 0s) the *(binding) arity* of o. Typically, λ-abstraction and pairing are value operations. The numbers N^+_o and N^-_o respectively count

- *active* arguments which should be evaluated before reaching a value, as both arguments of the pairing operation, and
- *passive* arguments which are not evaluated until the operation is destroyed, as the argument of a λ-abstraction.

Active arguments are not allowed to bind any variable, hence the absence of a family d^+ in signatures.

Despite their name, value operations may or may not return values, depending on the status of their arguments: a pair $\langle e_1, e_2 \rangle$ is only a value when both e_1 and e_2 are. In order to generate the right syntax in our strongly-sorted setting, we need to introduce two incarnations of each value operation, one for constructing values, the other for constructing

[4]Value operations correspond to canonical operators in [35].

programs. Any Howe binding signature generates a pointed strong endofunctor, as follows. We first define an auxiliary operation for adding bound variables in the right component — as prescribed by the binding sort s_v.

Notation 8.9. *For any* $(m \xrightarrow{f} n) \in \mathbb{F}^2$, $p \in \mathbb{F}$, *and* $s \in \{\mathbf{v}, \mathbf{p}\}$ *let* $(m \xrightarrow{f} n) +_s p$ *denote*

- $(m + p \xrightarrow{f + \mathrm{id}_p} n + p)$ *if* $s = \mathbf{v}$, *and*
- $(m \xrightarrow{f} n \hookrightarrow n + p)$ *if* $s = \mathbf{p}$.

Definition 8.10. Given any Howe binding signature B, the *generated* endofunctor Σ_0^B on $[2 \times \mathbb{F}^2, \mathbf{Set}]$ is defined by

$$\begin{array}{rcl} \Sigma_0^B(F)(m \xrightarrow{f} n)_\mathbf{v} & = & \sum_{o \in O_\mathbf{v}} F((m \xrightarrow{f} n)_\mathbf{v})^{N_o^+} \times \prod_{i \in N_o^-} F((m \xrightarrow{f} n) +_{s_v} d_{o,i}^-)_\mathbf{p}, \\ \Sigma_0^B(F)(m \xrightarrow{f} n)_\mathbf{p} & = & \sum_{o \in O_\mathbf{v}} F((m \xrightarrow{f} n)_\mathbf{p})^{N_o^+} \times \prod_{i \in N_o^-} F((m \xrightarrow{f} n) +_{s_v} d_{o,i}^-)_\mathbf{p} \\ & & + \sum_{o \in O_\mathbf{p}} \prod_{i \in N_o^\mathbf{p}} F((m \xrightarrow{f} n) +_{s_v} d_{o,i}^\mathbf{p})_\mathbf{p} \end{array}$$

for all $F \in [2 \times \mathbb{F}^2, \mathbf{Set}]$ and $m, n \in \mathbb{F}$, with obvious action on morphisms.

We now want to express Σ_0^B as a familial functor. For this, we start by defining

- a functor $A_o^\mathbf{p} \colon \mathbb{F}^{2\,op} \to [2 \times \mathbb{F}^2, \mathbf{Set}]$ for each program operation $o \in O_\mathbf{p}$,
- a functor $A_o^\mathbf{v} \colon (2 \times \mathbb{F}^2)^{op} \to [2 \times \mathbb{F}^2, \mathbf{Set}]$ for each value operation $o \in O_\mathbf{v}$,

in such a way that $\Sigma_0^B(F)(m \xrightarrow{f} n)_s = \sum_o [A_o^s(m \xrightarrow{f} n), F]$.

Definition 8.11. Given any Howe binding signature,

- for any program operation $o \in O_\mathbf{p}$, let the *arity* of o be the functor $A_o^\mathbf{p} \colon (\mathbb{F}^2)^{op} \to [2 \times \mathbb{F}^2, \mathbf{Set}]$ mapping $(m \xrightarrow{f} n)$ to the coproduct of representable presheaves
$$A_o^\mathbf{p}(m \xrightarrow{f} n) = \sum_{i \in N_o^\mathbf{p}} \mathbf{y}_{((m \xrightarrow{f} n) +_{s_v} d_{o,i}^\mathbf{p})_\mathbf{p}};$$
- for any value operation $o \in O_\mathbf{v}$, let the *arity* of o be the functor $A_o^\mathbf{v} \colon (2 \times \mathbb{F}^2)^{op} \to [2 \cdot \mathbb{F}, \mathbf{Set}]$ mapping any object $(m \xrightarrow{f} n)_s$ to the coproduct of representable presheaves
$$A_o^\mathbf{v}(m \xrightarrow{f} n)_s = N_o^+ \cdot \mathbf{y}_{(m \xrightarrow{f} n)_s} + \sum_{i \in N_o^-} \mathbf{y}_{((m \xrightarrow{f} n) +_{s_v} d_{o,i}^-)_\mathbf{p}}.$$

Remark 8.12. The arity functors are contravariant because the Yoneda embedding is, for covariant presheaves.

By construction:

Proposition 8.13. *The endofunctor* Σ_0^B *is familial, with*

- *as spectrum the functor*
$$S^B \cong \Sigma_0^B(1) \cong O_\mathbf{p} \cdot \mathbf{y}_{(0 \to 0)_\mathbf{p}} + O_\mathbf{v} \cdot \mathbf{y}_{(0 \to 0)_\mathbf{v}},$$
i.e.,
$$\begin{array}{rcl} S^B(m \xrightarrow{f} n)_\mathbf{p} & = & O_\mathbf{p} + O_\mathbf{v} \\ S^B(m \xrightarrow{f} n)_\mathbf{v} & = & O_\mathbf{v}, \end{array}$$

with obvious action on morphisms, so that

$$\mathrm{el}(S^B) = O_{\mathbf{p}} \cdot (\mathbb{F}^2)^{op} + O_{\mathbf{v}} \cdot (2 \times \mathbb{F}^2)^{op},$$

- *and as exponent the functor* $E^B \colon \mathrm{el}(S^B) \to [2 \times \mathbb{F}^2, \mathbf{Set}]$ *defined as the cotupling* $[[A^{\mathbf{p}}_o]_{o \in O_{\mathbf{p}}}, [A^{\mathbf{v}}_o]_{o \in O_{\mathbf{v}}}]$.

Proof. By mere calculation, using the Yoneda lemma. □

Corollary 8.14. *We have, for all* $F \in [2 \times \mathbb{F}^2, \mathbf{Set}]_f$ *and* $(m \xrightarrow{f} n) \in \mathbb{F}^2$,

$$\Sigma_0^B(F)((m \xrightarrow{f} n)_{\mathbf{p}}) \cong \sum_{o \in O_{\mathbf{p}}} [A^{\mathbf{p}}_o(m \xrightarrow{f} n), F] + \sum_{o \in O_{\mathbf{v}}} [A^{\mathbf{v}}_o(m \xrightarrow{f} n)_{\mathbf{p}}, F]$$

and

$$\Sigma_0^B(F)(m \xrightarrow{f} n)_{\mathbf{v}} \cong \sum_{o \in O_{\mathbf{v}}} [A^{\mathbf{v}}_o(m \xrightarrow{f} n)_{\mathbf{v}}, F].$$

We now describe the syntax generated by B.

Let $\mathcal{T}_0^B$ be the monad induced by the monadic forgetful functor Σ_0^B-$\mathbf{Mon} \to [2 \times \mathbb{F}^2, \mathbf{Set}]$.

In order to define the format, we need to unfold $\mathcal{T}_0^B(K)$, for K a coproduct of representable presheaves of the form $\mathbf{y}_{(n_{\mathbf{v}}, n_{\mathbf{p}})_s}$. For a single such presheaf, by Proposition 3.4, $\mathcal{T}_0^B(\mathbf{y}_{(n_{\mathbf{v}}, n_{\mathbf{p}})_s}) = \mu A.(I + \Sigma_0^B(A) + \mathbf{y}_{(n_{\mathbf{v}}, n_{\mathbf{p}})_s} \otimes A)$. Unfolding the definition of tensor product (Remark 8.6), noticing that the hom-set $[s, s']$

- is empty iff $s = \mathbf{p}$ and $s' = \mathbf{v}$, and
- is otherwise a singleton,

we obtain the following result.

Proposition 8.15. *We have, for all* $n_{\mathbf{v}}, n_{\mathbf{p}}, m, n \in \mathbb{N}$, $s, s' \in \{\mathbf{v}, \mathbf{p}\}$, $A \colon 2 \times \mathbb{F}^2 \to \mathbf{Set}$, *and* $f \colon m \to n$:

$$\begin{aligned}(\mathbf{y}_{(n_{\mathbf{v}}, n_{\mathbf{p}})_s} \otimes A)(m \xrightarrow{f} n)_{s'} &\cong [s, s'] \times A^{n_{\mathbf{p}}}(m \xrightarrow{f} n)_{\mathbf{p}} \times A^{n_{\mathbf{v}}}(m \xrightarrow{f} n)_{\mathbf{v}} \\ &\cong \begin{cases} \emptyset & \text{if } s = \mathbf{p} \text{ and } s' = \mathbf{v} \\ A^{n_{\mathbf{p}}}(m \xrightarrow{f} n)_{\mathbf{p}} \times A^{n_{\mathbf{v}}}(m \xrightarrow{f} n)_{\mathbf{v}} & \text{otherwise.} \end{cases}\end{aligned}$$

Remark 8.16. Intuitively, this will entail that $\mathcal{T}_0^B(\mathbf{y}_{(n_{\mathbf{v}}, n_{\mathbf{p}})_s})$ extends the syntax generated by Σ_0^B with a new operation taking $n_{\mathbf{p}}$ programs and $n_{\mathbf{v}}$ values as arguments, and returning a program or a value, depending on s. As usual, if the output is a value, by action of ι, there is a corresponding program operation.

We now describe $\mathcal{T}_0^B(K)$, for K any coproduct of representable presheaves of the form $\mathbf{y}_{(n_{\mathbf{v}}, n_{\mathbf{p}})_s}$. Let us first introduce some notations.

Definition 8.17. A **family of operation arities** is a family $c = (m_i, n_i, s_i)_{i \in I}$ of triples $(m_i, n_i, s_i) \in \mathbb{N}^2 \times \{\mathbf{v}, \mathbf{p}\}$.

Notation 8.18. *We think of any such triple* $c_i = (m_i, n_i, s_i)$ *as an operation* $c_i \colon \mathbf{v}^{m_i} \times \mathbf{p}^{n_i} \to s_i$, *hence denote any such family by* $(c_i \colon \mathbf{v}^{m_i} \times \mathbf{p}^{n_i} \to s_i)_{i \in I}$. *We in fact extend this notation by*

- *writing* $\mathbf{v}^m \times \mathbf{p}^n \times s^q$, *to denote either* $\mathbf{v}^{m+q} \times \mathbf{p}^n$, *when* $s = \mathbf{v}$, *or* $\mathbf{v}^m \times \mathbf{p}^{n+q}$, *when* $s = \mathbf{p}$,

- *omitting* $-^n$ *if* $n = 0$ *in the above product, e.g.,* $\mathbf{v}^m \to s$ *denotes* $\mathbf{v}^m \times \mathbf{p}^0 \to s$*, and finally*
- *writing* $c\colon s$ *for* $c\colon \mathbf{v}^0 \times \mathbf{p}^0 \to s$*.*

We furthermore denote the disjoint union of families of operation arities K *and* L *by* $K+L$*. Given a family* $K = (c_i\colon \mathbf{v}^{m_i} \times \mathbf{p}^{n_i} \to s_i)_i$ *of operation arities, we also denote by* K *the coproduct of representable presheaves* $\sum_i \mathbf{y}_{(m_i,n_i)_{s_i}} : \mathbb{2} \times \mathbb{F}^{\mathbb{2}} \to \mathbf{Set}$*.*

Remark 8.19. By Notation 8.9, for $m, n, p \in \mathbb{F}$, $(m,n) +_s p$ denotes $(m+p, n)$ if $s = \mathbf{v}$, and $(m, n+p)$ otherwise.

We now give an inductive description of $\mathcal{T}_0^B(K)$ on objects of the form $(m,n)_s$ (Notation 8.9). Letting $m, n \vdash_K e : s$ mean that $e \in \mathcal{T}_0^B(K)(m,n)_s$, for a family K of operation arities, the free Σ_0^B-monoid over K is inductively generated on objects of the form $(m,n)_s$ by the following rules.

$$\frac{}{m,n \vdash_K x_i^s : s}\ (i \in m) \qquad\qquad \frac{}{m,n \vdash_K a_i : \mathbf{p}}\ (i \in n)$$

$$\frac{\begin{array}{ccc} m,n \vdash_K e_1 : \mathbf{v} & \dots & m,n \vdash_K e_{m'} : \mathbf{v} \\ m,n \vdash_K f_1 : \mathbf{p} & \dots & m,n \vdash_K f_{n'} : \mathbf{p} \end{array}}{m,n \vdash_K k(e_1,\dots,e_{m'}; f_1,\dots,f_{n'}) : s}\ ((k\colon \mathbf{v}^{m'} \times \mathbf{p}^{n'} \to s) \in K)$$

$$\frac{(m,n) +_{s_v} d^{\mathbf{p}}_{o,1} \vdash_K e_1 : \mathbf{p} \quad \dots \quad (m,n) +_{s_v} d^{\mathbf{p}}_{o,N_o^{\mathbf{p}}} \vdash_K e_{N_o^{\mathbf{p}}} : \mathbf{p}}{m,n \vdash_K o(e_1,\dots,e_{N_o^{\mathbf{p}}}) : \mathbf{p}}\ (o \in O_{\mathbf{p}})$$

$$\frac{\begin{array}{ccc} m,n \vdash_K e_1 : s & \dots & m,n \vdash_K e_{N_o^+} : s \\ (m,n) +_{s_v} d^-_{o,1} \vdash_K f_1 : \mathbf{p} & \dots & (m,n) +_{s_v} d^-_{o,N_o^-} \vdash_K f_{N_o^-} : \mathbf{p} \end{array}}{m,n \vdash_K o_s(e_1,\dots,e_{N_o^+}; f_1,\dots,f_{N_o^-}) : s}\ (o \in O_{\mathbf{v}})$$

Remark 8.20. Program operations $o \in O_{\mathbf{p}}$ only have one list of arguments. Furthermore, they always return programs, so there is no need to annotate them. By contrast, value operations $o' \in O_{\mathbf{v}}$ have two lists of arguments (active and passive arguments, see Terminology 8.8), and may return values or programs, depending on the status of their active arguments. Thus, e.g., any unannotated operation application $o(e_1,\dots,e_n)$ must be a program operation application, while any annontated operation application $o_s(e_1,\dots,e_m; f_1,\dots,f_n)$ must be a value operation application.

The action on morphisms is straightforward: for renaming, we rename (value and program) variables accordingly; for ι, we replace each $x_i^{\mathbf{v}}$ (resp. $o_{\mathbf{v}}$) with $x_i^{\mathbf{p}}$ (resp. $o_{\mathbf{p}}$). For morphisms $K \to L$, we proceed similarly.

Terminology 8.21. We think of elements k from K as *metavariables*, while terms of the form a_i or x_i^s are mere *variables*.

Notation 8.22.

- *Following* [29]*, for any* $(k\colon \mathbf{v}^m \times \mathbf{p}^n \to s) \in K$*, we abbreviate* $k(x_1^{\mathbf{v}},\dots,x_m^{\mathbf{v}}; a_1,\dots,a_n)$ *to just* k *when* (m,n) *is clear from context.*
- *For value operations* o*, we sometimes omit the subscript* s *in* $o_s(\dots)$*, when the expected sort is clear from context.*
- *Similarly, we sometimes omit the exponent* s *in variables* x_i^s*.*

- *In metavariable application $k(e_1, \ldots, e_{m'}; f_1, \ldots, f_{n'})$ and value operation application $o(e_1, \ldots, e_p; f_1, \ldots, f_q)$, when one sequence is empty we omit it altogether, writing, e.g., $k(e_1, \ldots, e_{m'})$ or $o(f_1, \ldots, f_q)$. When both lists are empty we simply write k, resp. o.*

Thus, e.g., k may denote a nullary metavariable, or a non-nullary metavariable with identity substitution.

Let us now introduce Howe's notion of signature for evaluation rules, restricting attention to signatures satisfying the syntactic condition of [35, Lemma 6.1]. Evaluation is a binary relation between closed programs and closed values. By default, all value operations o are considered as coming equipped with their *canonical* evaluation rule

$$\frac{e_1 \Downarrow v_1 \qquad \ldots \qquad e_{N_o^+} \Downarrow v_{N_o^+}}{o_{\mathbf{p}}(e_1, \ldots, e_{N_o^+}; f_1, \ldots, f_{N_o^-}) \Downarrow o_{\mathbf{v}}(v_1, \ldots, v_{N_o^+}; f_1, \ldots, f_{N_o^-})} \ . \tag{8.1}$$

Howe's signatures thus only need to specify evaluation of program operations. We now successively introduce notions of premises, rules, and signatures.

We start by fixing a global choice of finite coproducts in both $[2 \cdot \mathbb{N}^2, \mathbf{Set}]$ (families of operation arities) $[2 \cdot \mathbb{F}^2, \mathbf{Set}]$ (presheaves).

Definition 8.23. Given any families K and L of operation arities, a *premise* $K \to L$ consists of a *source* program $0, 0 \vdash_K e : \mathbf{p}$ and a *target* value $0, 0 \vdash_L v : \mathbf{v}$, where L and v take one of the following two forms:

- either $L = K + (c \colon \mathbf{v})$, extending K with one closed value metavariable, in which case $v = c$,
- or $L = K + (\alpha_i \colon \mathbf{v})_{i \in \{1, \ldots, N_o^+\}} + (\beta_i \colon s_v^{d_{o,i}^-} \to \mathbf{p})_{i \in \{1, \ldots, N_o^-\}}$ for some value operation $o \in O_{\mathbf{v}}$, in which case the target is $o_{\mathbf{v}}(\alpha_1, \ldots, \alpha_{N_o^+}; \beta_1, \ldots, \beta_{N_o^-})$.

Notation 8.24. *We write any premise $(e, v) \colon K \to L$ as $K \xrightarrow{e \Downarrow v} L$.*

Definition 8.25. A *rule* consists of

- a *head* program operation o,
- a composable sequence of premises $K_0 \xrightarrow{e_1 \Downarrow v_1} K_1 \to \ldots \to K_{q-1} \xrightarrow{e_q \Downarrow v_q} K_q$, where $K_0 = (k_i \colon s_v^{d_{o,i}^{\mathbf{p}}} \to \mathbf{p})_{i \in \{1, \ldots, N_o^{\mathbf{p}}\}}$, and
- a *tail* metavariable $(k_v \colon \mathbf{v})$ in K_q.

A *Howe dynamic signature* over a Howe binding signature is a family of rules.

Notation 8.26. *Such a rule is denoted by*

$$\frac{e_1 \Downarrow v_1 \qquad \ldots \qquad e_q \Downarrow v_q}{o(k_1, \ldots, k_{N_o^{\mathbf{p}}}) \Downarrow k_v} \ .$$

The families of operation arities $K_0, \ldots, K_q$ are left implicit.

Remark 8.27. There are some discrepancies w.r.t. Howe's original format [35, Lemma 6.1].

- We restrict to rules with a finite number of premises[5].
- We have a slightly different treatment of value vs. program variables and metavariables.

[5]We expect that our setting can be extended to account for rules with an infinite number of premises by replacing the finitarity condition on Σ_1^F with a weaker accessibility requirement.

Definition 8.28. For any Howe binding signature B, and Howe dynamic signature D over it, the *evaluation*, denoted by $\Downarrow^{B,D}$, or $\Downarrow$ for short, is the binary relation between programs and values, obtained inductively by instantiating the given rules together with the canonical rules (8.1).

Let us now recall Howe's general definition of applicative bisimulation, and its open extension.

Definition 8.29. For any Howe binding signature B, and Howe dynamic signature D over it, an *applicative simulation* is a binary relation R in $\mathbf{Set}^2$ over the injection $\mathcal{T}_0^B(\emptyset)(0,0)_{\mathbf{v}} \hookrightarrow \mathcal{T}_0^B(\emptyset)(0,0)_{\mathbf{p}}$, such that for any closed programs e and e' such that $e\ R_{\mathbf{p}}\ e'$, and any transition $e \Downarrow o_{\mathbf{v}}(v_1,\ldots,v_{N_o^+};e_1,\ldots,e_{N_o^-})$, there exist $v'_1,\ldots,v'_{N_o^+},e'_1,\ldots,e'_{N_o^-}$ and a transition $e' \Downarrow o_{\mathbf{v}}(v'_1,\ldots,v'_{N_o^+};e'_1,\ldots,e'_{N_o^-})$, such that $v_i\ R_{\mathbf{v}}\ v'_i$ for all $i \in N_O^+$, and $e_j[\sigma]\ R_{\mathbf{p}}\ e'_j[\sigma]$, for all $j \in N_o^-$ and closing substitutions σ.

An *applicative bisimulation* is an applicative simulation whose converse relation also is one.

Applicative bisimulations are closed under union, and we let $\sim_{B,D}$, or $\sim$ for short, denote the largest one.

Finally, the open extension R° of a relation R is the largest substitution-closed relation contained in R on closed programs and values.

Remark 8.30. By definition, being a subobject of the injection $\mathcal{T}_0^B(\emptyset)(0,0)_{\mathbf{v}} \hookrightarrow \mathcal{T}_0^B(\emptyset)(0,0)_{\mathbf{p}}$ in $\mathbf{Set}^2$, any applicative simulation is such that for any related values $v\ R_{\mathbf{v}}\ v'$, v and v' should also be related as programs, i.e., $\iota \cdot v\ R_{\mathbf{p}}\ \iota \cdot v'$.

8.4.2. *Congruence by encoding in our framework.* We now want to prove using Theorem 6.15 that $\sim^\circ$ is a congruence. For this, we could try to naively model evaluation rules as a dynamic signature Σ_1 over Σ_0^B. However, substitution-closed bisimilarity in the obtained system would not coincide with applicative bisimilarity in Howe's sense. Furthermore, the induced functor $\Sigma_1^{\mathbf{s}} \colon \Sigma_0^B\text{-}\mathbf{Trans} \to \widehat{\mathbb{C}_{\mathbf{s}}}$ would not be cellular. Indeed, consider for instance the usual, call-by-name rule for application. In Howe's format, it gives the following:

$$\frac{k_1 \Downarrow \lambda(k_3) \qquad k_3(k_2) \Downarrow k_4}{k_1\ k_2 \Downarrow k_4},$$

where all k_i are metavariables.

Notation 8.31. *For any $n \in \mathbb{N}$, we use the abbreviation $n_s := ((0,0) +_{s_v} n)_s$ — because we will mainly need tuples $(n_{\mathbf{p}}, n_{\mathbf{v}})$ with only $n_{s_v} \neq 0$ from now on.*

Letting E denote the exponent of the induced $\Sigma_1^{\mathbf{s}}$, and o denote the element corresponding to this rule in the spectrum, the boundary arity $E(s_\Downarrow \restriction o) \colon E(\mathbf{s}(\Downarrow), o \cdot s_\Downarrow) \to E(\Downarrow, o)$ is the bottom composite in the following diagram,

$$\begin{tikzcd}
 & & \mathscr{L}(\mathbf{y}_{0_\mathbf{p}}) \arrow[r, "\mathscr{L}(\mathbf{y}_{s_\Downarrow})"] \arrow[d, "\bar{\chi}"'] & \mathscr{L}(\mathbf{y}_\Downarrow) \arrow[dd] \\
 & \mathscr{L}(\mathbf{y}_{0_\mathbf{v}} + \mathbf{y}_{0_\mathbf{p}}) \arrow[r, "\lambda(k_1)+\mathrm{id}_{\mathscr{L}(\mathbf{y}_{0_\mathbf{p}})}"] \arrow[d, "\mathscr{L}(\mathbf{y}_{t_\Downarrow}+\mathbf{y}_{0_\mathbf{p}})"'] & \mathscr{L}(\mathbf{y}_{1_\mathbf{p}} + \mathbf{y}_{0_\mathbf{p}}) & \\
\mathscr{L}(\mathbf{y}_{0_\mathbf{p}} + \mathbf{y}_{0_\mathbf{p}}) \arrow[r, "\mathscr{L}(\mathbf{y}_{s_\Downarrow}+\mathbf{y}_{0_\mathbf{p}})"'] & \mathscr{L}(\mathbf{y}_\Downarrow + \mathbf{y}_{0_\mathbf{p}}) \arrow[rr] & & E(\Downarrow, o)
\end{tikzcd}$$

where

- $\bar{\chi}$ is analogous to Definition 7.16;
- we implicitly use transposition, the Yoneda lemma, and canonical isomorphisms $\mathscr{L}(A) + \mathscr{L}(B) \cong \mathscr{L}(A + B)$.

This bottom composite is not a relative cellular complex, essentially because $\lambda(k_1) \colon \mathscr{L}(\mathbf{y}_{0_\mathbf{v}}) \to \mathscr{L}(\mathbf{y}_{1_\mathbf{p}})$ is not of the form $\mathscr{L}(\mathbf{y}_{s_\Downarrow})$.

The solution to both problems (wrong substitution-closed bisimilarity and non-cellularity of $\Sigma_1^\mathbf{s}$) is to rectify this last point. Namely, we construct a new signature from the evaluation rules, in such a way that we may observe each argument of a value. For this, we first need to generalise the notions of premise, rule, and signature, as well as the evaluation relation induced by a signature, which in turn requires us to introduce the Howe context induced by a Howe dynamic signature over a Howe binding signature.

We fix a Howe binding signature B and a Howe dynamic signature D over it for the rest of this section.

Definition 8.32. We define the *Howe context* induced by (B, D), as follows.

- For state types, we take $\mathbb{C}_0 = (2 \times \mathbb{F}^2)^{op}$, with the monoidal structure on $\hat{\mathbb{C}}_0 = [2 \times \mathbb{F}^2, \mathbf{Set}]$ specified in Remark 8.6.
- For transition types, we take
$$\mathbb{C}_1 = \{\Downarrow\} \uplus \sum_{o \in O_\mathbf{v}, i \in N_o^+} \{\Downarrow_{o,i}^+\} \uplus \sum_{o \in O_\mathbf{v}, j \in N_o^-} \{\Downarrow_{o,j}^-\}.$$
- We define source and target as follows
$$\begin{array}{rcl} \mathbf{s} \colon \mathbb{C}_1 & \to & \mathbb{C}_0 \\ \Downarrow & \mapsto & 0_\mathbf{p} \\ \Downarrow_{o,i}^+ & \mapsto & 0_\mathbf{v} \\ \Downarrow_{o,j}^- & \mapsto & 0_\mathbf{v} \end{array} \qquad \begin{array}{rcll} \mathbf{t} \colon \mathbb{C}_1 & \to & \mathbb{C}_0 & \\ \Downarrow & \mapsto & 0_\mathbf{v} & \\ \Downarrow_{o,i}^+ & \mapsto & 0_\mathbf{v} & (o \in O_\mathbf{v},\ i \in N_o^+) \\ \Downarrow_{o,j}^- & \mapsto & (d_{o,j}^-)_\mathbf{p} & (o \in O_\mathbf{v},\ j \in N_o^-) \end{array}$$

We now introduce the new notions of premise, rule, and dynamic signature, which we deem *rigid* to avoid confusion with the original.

Definition 8.33. A *rigid premise* consists of a family of operation arities K, called the *source type*, a transition type $\alpha \in \mathbb{C}_1$ and a *source* term $0, 0 \vdash_K e : \mathbf{s}(\alpha)$.

The *target type* of a rigid premise (K, α, e) is the family $L := K + (k \colon s_v^n \to s)$, where $\mathbf{t}(\alpha) = n_s$, and its *target* is the fresh metavariable $k \colon s_v^n \to s \in L$.

We denote such a premise by $K \xrightarrow{e\alpha k} L$.

Definition 8.34. A *rigid rule* consists of

- a *head* program operation o,

- a composable sequence of rigid premises $K_0 \xrightarrow{e_1 \alpha_1 k_1} K_1 \to \ldots \to K_{q-1} \xrightarrow{e_p \alpha_q k_q} K_q$, where $K_0 = (k_i \colon s_v^{d^{\mathbf{p}}_{o,i}} \to \mathbf{p})_{i \in \{1,\ldots,N^{\mathbf{p}}_o\}}$, and
- a *tail* metavariable $(k_v \colon \mathbf{v})$ in K_q.

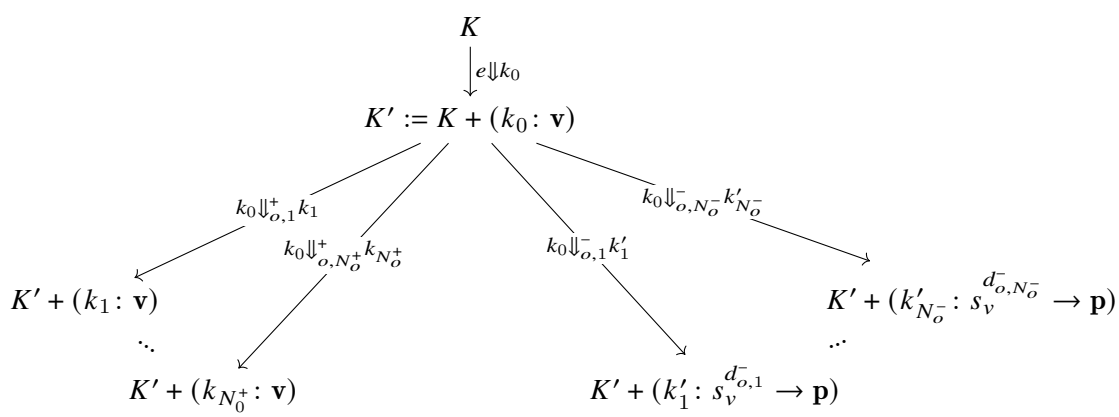
A *rigid dynamic signature* over a Howe binding signature is a family of rigid rules.

Definition 8.35. The labelled transition system induced by B any the rigid dynamic signature D' is defined by instantiating the given rigid rules, together with the canonical rules (8.1), and the following new rules.

$$\frac{}{o_{\mathbf{v}}(v_1, \ldots, v_{N^+_o}; v'_1, \ldots, v'_{N^-_o}) \Downarrow^+_{o,i} v_i} \qquad \frac{}{o_{\mathbf{v}}(v_1, \ldots, v_{N^+_o}; v'_1, \ldots, v'_{N^-_o}) \Downarrow^-_{o,j} v'_j} \,. \tag{8.2}$$

We may now define the rigid signature induced by (B, D).

Definition 8.36. Let the rigid dynamic signature $R(B, D)$ induced by (B, D) be obtained as follows. For each rule in D, $R(B, D)$ has a rule for the same program operation, whose premises are obtained by replacing each original premise $K \xrightarrow{e \Downarrow o_{\mathbf{v}}(k_1,\ldots,k_{N^+_o};k'_1,\ldots,k'_{N^-_o})} L$ with any linearisation (in the straightforward, suitable sense) of the following tree.



Let us show that the rigid signature $R(B, D)$ is adequate.

Proposition 8.37. *The open extension of applicative bisimilarity in the sense of (B, D) coincides with substitution-closed bisimilarity in the labelled transition system generated by $R(B, D)$.*

Proof. Let $\sim'$ denote substitution-closed bisimilarity in the sense of $R(B, D)$.

First of all, $\sim^\circ$ is a substitution-closed bisimulation; indeed, $\Downarrow$ is the same relation in both systems — by a straightforward induction — and by definition of applicative bisimulation, for any $o_{\mathbf{v}}(v_1, \ldots, v_{N^+_o}, e_1, \ldots, e_{N^-_o}) \sim o_{\mathbf{v}}(v'_1, \ldots, v'_{N^+_o}, e'_1, \ldots, e'_{N^-_o})$, we have $v_i \sim v'_i$ and $e_j \sim^\circ e'_j$, for all i and j, which ensures that the new transitions $\Downarrow^+_{o,i}$ and $\Downarrow^-_{o,j}$ are matched. Thus, we have $\sim^\circ \subseteq \sim'$.

Conversely, the new transitions ensure that $\sim'$ is an applicative bisimulation, hence is contained in $\sim$ on closed terms. But it is substitution-closed, so we get $\sim' \subseteq \sim^\circ$, as desired. $\square$

Finally, we show that the rigid signature $R(B,D)$ straightforwardly gives rise to a dynamic signature $\Sigma_1^{B,D}$, such that the initial vertical $\Sigma_1^{B,D}$-algebra is isomorphic to the generated labelled transition system, and furthermore that the induced functor $(\Sigma_1^{B,D})^{\mathbf{s}}$ is cellular, hence that by Proposition 8.37, Corollary 7.30, and Theorem 6.15, $\sim^\circ$ is a congruence.

Notation 8.38. *We tend to abbreviate $\Sigma_1^{B,D}$ to Σ_1, for readability.*

Let us present the dynamic signature $\Sigma_1^{B,D}$, which will consist of various components: one dynamic signature $\Sigma_{1,o}$ for each value operation $o \in O_{\mathbf{v}}$, describing both kinds of canonical rules (8.1) and (8.2), plus one dynamic signature $\Sigma_{1,r}$ for each other rule.

- For each value operation $o \in O_{\mathbf{v}}$, we define

$$\begin{array}{rcl} \Sigma_{1,o}^F(X)(\Downarrow) & = & X(\Downarrow)^{N_o^+} \times \prod_{j\in N_o^-} X((d_{o,j}^-)_{\mathbf{p}}) \\ \Sigma_{1,o}^F(X)(\Downarrow_{o,i}^+) & = & X(0_{\mathbf{v}})^{N_o^+} \times \prod_{j\in N_o^-} X((d_{o,j}^-)_{\mathbf{p}}) \\ \Sigma_{1,o}^F(X)(\Downarrow_{o,i}^-) & = & X(0_{\mathbf{v}})^{N_o^+} \times \prod_{j\in N_o^-} X((d_{o,j}^-)_{\mathbf{p}}) \end{array}$$

  with as $\Sigma_{1,o}^F(X) \to \Sigma_0^B(X)\mathbf{s} \times X\mathbf{t}$:
  - at $\Downarrow$: $(r_1,\ldots,r_{N_o^+},e_1,\ldots,e_{N_o^-}) \mapsto \begin{array}{l}(in_{o_{\mathbf{p}}}(r_1\cdot s_{\Downarrow},\ldots,r_{N_o^+}\cdot s_{\Downarrow},e_1,\ldots,e_{N_o^-}),\\ o_{\mathbf{v}}(r_1\cdot t_{\Downarrow},\ldots,r_{N_o^+}\cdot t_{\Downarrow},e_1,\ldots,e_{N_o^-}))\end{array}$,
  - at $\Downarrow_{o,i}^+$: $(v_1,\ldots,v_{N_o^+},e_1,\ldots,e_{N_o^-}) \mapsto (in_{o_{\mathbf{v}}}(v_1,\ldots,v_{N_o^+},e_1,\ldots,e_{N_o^-}),v_i)$,
  - at $\Downarrow_{o,j}^-$: $(v_1,\ldots,v_{N_o^+},e_1,\ldots,e_{N_o^-}) \mapsto (in_{o_{\mathbf{v}}}(v_1,\ldots,v_{N_o^+},e_1,\ldots,e_{N_o^-}),e_j)$.
- For each rigid premise $p = (K \xrightarrow{e\alpha k} L)$, we define the cospan induced by p to be

$$\mathscr{L}(K) \xrightarrow{\ell} E_p \xleftarrow{r} \mathscr{L}(L),$$

  where the left-hand morphism ℓ is defined by the pushout

$$\begin{tikzcd} & \mathscr{L}(\mathbf{y}_{\mathbf{s}(\alpha)}) \arrow[r, "e"] \arrow[d, "\mathscr{L}(\mathbf{y}_{s_\alpha})"'] & \mathscr{L}(K) \arrow[d, "\ell"] \\ \mathscr{L}(\mathbf{y}_{\mathbf{t}(\alpha)}) \arrow[r, "\mathscr{L}(\mathbf{y}_{t_\alpha})"'] & \mathscr{L}(\mathbf{y}_\alpha) \arrow[r] & E_p, \end{tikzcd} \tag{8.3}$$

  and the right-hand morphism

$$r\colon \mathscr{L}(L) = \mathscr{L}(K + \mathbf{y}_{\mathbf{t}(\alpha)}) \cong \mathscr{L}(K) + \mathscr{L}(\mathbf{y}_{\mathbf{t}(\alpha)}) \to E_p$$

  is obtained by copairing ℓ and the bottom composite in (8.3).
- For each rigid rule

$$\frac{(k_i\colon s_v^{d_{o,i}^{\mathbf{p}}} \to \mathbf{p})_{i\in\{1,\ldots,N_o^{\mathbf{p}}\}} = K_0 \xrightarrow{e_1\alpha_1 k_1} K_1 \to \ldots \to K_{n-1} \xrightarrow{e_n\alpha_n k_n} K_n}{o(k_1,\ldots,k_{N_o^{\mathbf{p}}}) \Downarrow k''},$$

  say r, we define

$$\Sigma_{1,r}^F(X)(\Downarrow) = [E_{p_1},X] \times_{[\mathscr{L}(K_1),X]} \ldots \times_{[\mathscr{L}(K_{p-1}),X]} [E_{p_n},X], \tag{8.4}$$

  where for all $i \in n$, $\mathscr{L}(K_{i-1}) \xrightarrow{s_i} E_{p_i} \xleftarrow{t_i} \mathscr{L}(K_i)$ denotes the cospan induced by the ith premise, with $\Sigma_{1,r}^F(X)(\Downarrow_{o',i}^+) = \Sigma_{1,r}^F(X)(\Downarrow_{o',j}^-) = \emptyset$ for all $o' \in O_{\mathbf{v}}$, $i \in N_{o'}^+$, $j \in N_{o'}^-$, and again with the morphism $\Sigma_{1,r}^F(X) \to \Sigma_0^B(X)\mathbf{s} \times X\mathbf{t}$ mapping any compatible tuple $(\rho_1,\ldots,\rho_n) \in \Sigma_{1,r}^F(X)(\Downarrow)$ to $(in_o(a),b)$, where $a \in \prod_{i\in N_o^{\mathbf{p}}} X((d_{o,i}^{\mathbf{p}})_{\mathbf{p}})$ corresponds by Yoneda and adjunction to the morphism

$$\mathscr{L}(K_0) \xrightarrow{s_1} E_1 \xrightarrow{\rho_1} X,$$

and $b \in X(0_v)$ corresponds to

$$\mathscr{L}(\mathbf{y}_{0_v}) \xrightarrow{\mathscr{L}(in_2)} \mathscr{L}(K_n) \xrightarrow{t_n} E_n \xrightarrow{\rho_n} X.$$

- We let $(\Sigma_1^{B,D})^F = \sum_{o \in O_v} \Sigma_{1,o}^F + \sum_r \Sigma_{1,r}^F$, with morphism to $\Sigma_0^B(-)\mathbf{s} \times (-)\mathbf{t}$ given by cotupling.

By construction, we have:

Proposition 8.39. *The transition Σ_0^B-monoid generated by the dynamic signature $\Sigma_1^{B,D}$ is isomorphic to the rigid transition system $R(B,D)$.*

Furthermore, we observe:

Lemma 8.40. *The induced functor $(\Sigma_1^{B,D})^{\mathbf{s}}$ is cellular.*

Proof. The functor is familial by construction. By Corollary 7.30*(iii)* and Proposition 7.32, it suffices to show that for any rule r whose conclusion is a transition of type α, the boundary arity

$$E(s_\alpha \restriction r) \colon E(\mathbf{s}(\alpha), r \cdot s_\alpha) \to E(\alpha, r)$$

is a relative cell complex. The arity of each canonical rule (8.1) is clearly a coproduct of generating cofibrations. The arities of canonical rules (8.2) are identities, hence trivially relative cell complexes. For any other rule r, by (8.4) the arity is the left-hand leg of cospan obtained by composing all $K_{i-1} \to E_i \leftarrow K_i$, which is thus by construction a composite of pushouts of generating cofibrations, hence a relative cell complex. □

We at last obtain:

Theorem 8.41. *For any Howe binding signature B and dynamic signature D, the open extension of applicative bisimilarity on the generated transition system is a congruence.*

Proof. The open extension of applicative bisimilarity for (B,D) coincides with substitution-closed bisimilarity for $R(B,D)$ by Proposition 8.37, which further coincides with substitution-closed for the transition system generated by $\Sigma_1^{B,D}$ by Proposition 8.39. Finally, the latter is a congruence by Lemmas 7.28 and 8.40, and Theorem 6.15. □

9. Congruence of substitution-closed bisimilarity

In this section, we elaborate on the proof sketch of Theorem 6.15 given in §6. The overall structure remains the same, and the final part of the proof sketch is complete, so we mainly elaborate on items *(1)–(4)*.

9.1. **Preliminaries on spans.** In this section, we fix a locally finitely presentable category $\mathscr{C}$, recall some known tools about spans, and develop a few new ones, including categorified notions of reflexivity, transitivity, symmetry, and transitive closure. As announced in §1.6, we freely switch from spans $X \leftarrow S \to Y$ to their pairings $S \to X \times Y$ in $\mathscr{C}/X \times Y$. Furthermore,

Definition 9.1. Given any spans $R \xrightarrow{\langle p_1, p_2 \rangle} X \times Y$ and $S \xrightarrow{\langle q_1, q_2 \rangle} Y \times Z$, their *span*, or *sequential*, composition, denoted by $R; S \to X \times Z$, is the (or, rather, some global choice of) following span obtained by pullback.

$$\begin{tikzcd}
 & & R;S \arrow[dl] \arrow[dr] & & \\
 & R \arrow[dl, "p_1"'] \arrow[dr, "p_2"] & & S \arrow[dl, "q_1"'] \arrow[dr, "q_2"] & \\
X & & Y & & Z
\end{tikzcd}$$

Definition 9.2. A span $p\colon S \to X^2$ (a.k.a. a graph $S \rightrightarrows X$) is *reflexive* if there is a morphism

$$\begin{tikzcd}
X \arrow[rr, dashed] \arrow[dr, "\langle \mathrm{id}_X, \mathrm{id}_X \rangle"'] & & S \arrow[dl, "p"] \\
 & X^2 &
\end{tikzcd}$$

from the diagonal to S in $\mathscr{C}/X^2$. It is *transitive* if there is a morphism $S;S \to S$ in $\mathscr{C}/X^2$. Finally, it is *symmetric* if there is a morphism $S^\dagger \to S$, where $(-)^\dagger$ denotes the functor swapping projections, i.e., mapping any $R \xrightarrow{\langle p_1, p_2 \rangle} X^2$ to $R \xrightarrow{\langle p_2, p_1 \rangle} X^2$.

For potentially non-reflexive spans, we will use the following reflexive transitive closure.

Definition 9.3. The *reflexive transitive closure* S^* of any span $S \to X^2$ is the coproduct $\sum_{n \in \mathbb{N}} S^{;n}$, or for short $\sum_{n \in \mathbb{N}} S^n$ when the context is clear, where $S^{;n}$ denotes iterated span composition of S with itself, inductively defined by $S^{;0} = X$ and $S^{;n+1} = S^{;n};S$.

At some point, we will also use a more "relational" notion of transitive closure, which we now introduce.

Definition 9.4. Given a span $S \to X^2$, we denote by $\overline{S}$ the relation on X induced by the image factorisation of $S \to X \times X$.

Definition 9.5. The *relational transitive closure* $S^{\overline{+}}$ of a span S on X is the union $\bigcup_{n>0} \overline{S^{;n}}$.

Remark 9.6. Unions of relations exist by Proposition 1.7, as we have assumed the ambient category $\mathscr{C}$ to be locally finitely presentable.

We immediately observe the following.

Lemma 9.7. *For all $n \in \mathbb{N}$, spans $S \to X^2$ and $R \to X^2$, and families $(S_i \to X^2)_{i \in I}$ of relations spans, we have*

(1) $\overline{S^\dagger} \cong \overline{S}^\dagger$,
(2) $(R;S)^\dagger \cong S^\dagger; R^\dagger$, *hence* $(S^\dagger)^{;n} \cong (S^{;n})^\dagger$, *and*
(3) $\bigcup_{i \in I} S_i^\dagger \cong (\bigcup_{i \in I} S_i)^\dagger$.

Proof. Let $\sigma := \langle \pi_2, \pi_1 \rangle\colon X^2 \to X^2$.

- For (1), by definition each side of the isomorphism corresponds to one side of the exterior of the following commuting pentagon.

$$\begin{tikzcd}
S \arrow[rr, two heads] \arrow[d, two heads] \arrow[dr, "\langle \pi_1, \pi_2 \rangle"] \arrow[ddr, "\langle \pi_2, \pi_1 \rangle"] & & \overline{S} \arrow[d, hook] \\
\overline{S^\dagger} \arrow[dr, hook] & & X^2 \arrow[dl, "\sigma"] \\
 & X^2 &
\end{tikzcd}$$

But, σ being an isomorphism, both sides are strong epi-mono factorisations of $\langle \pi_2, \pi_1 \rangle \colon S \to X^2$, hence are isomorphic.

- The first point of (2) is clear. For the second one, we proceed by induction. The base case is trivial, and assuming $(S^\dagger)^{;n} \cong (S^{;n})^\dagger$, we have
$$(S^\dagger)^{;n+1} \cong (S^\dagger)^{;n}; S^\dagger \cong (S^{;n})^\dagger; S^\dagger \cong (S; S^{;n})^\dagger \cong (S^{;n+1})^\dagger,$$
hence the result.
- For (3) we have $\bigcup_{i\in I} S_i^\dagger = \overline{\sum_{i\in I} S_i^\dagger}$ and $(\bigcup_{i\in I} S_i)^\dagger = \overline{(\sum_{i\in I} S_i)}^\dagger \cong \overline{(\sum_{i\in I} S_i)^\dagger}$ by (1), so it suffices to show $\sum_{i\in I} S_i^\dagger \cong (\sum_{i\in I} S_i)^\dagger$. But, letting $b_i \colon S_i \to X^2$ denote each projection, the former is (shorthand for) the morphism $[\sigma \circ b_i]_i \colon \sum_i S_i \to X^2$, and the latter is $\sigma \circ [b_i]_i$, which are in fact equal.

□

Corollary 9.8. *For all spans $S \to X^2$, we have $S^{\dagger\overline{+}} \cong S^{\overline{+}\dagger}$.*

Proof. We have $S^{\dagger\overline{+}} = \bigcup_{n>0} \overline{(S^\dagger)^{;n}} \cong (\bigcup_{n>0} \overline{S^{;n}})^\dagger \cong S^{\overline{+}\dagger}$, by the lemma. □

The following lemma will later be used to exploit preservation of sifted colimits by Σ_0.

Lemma 9.9. *If S is a reflexive span on X, then $S^{\overline{+}}$ is the (filtered) colimit of the chain*
$$X \to \overline{S} \cong \overline{S; X} \to \overline{S; S} \cong \overline{S; S; X} \to \overline{S; S; S} \to \dots$$

Proof. Consider a colimiting cone for the given diagram, say to $C \in \mathscr{C}$. Because the forgetful functor $\mathscr{C}/X^2 \to \mathscr{C}$ creates colimits, it is in fact a colimiting cocone in $\mathscr{C}/X^2$. Now, the diagram consists of monic morphisms, hence by [2, Proposition 1.62(i)], so does the colimiting cocone. Furthermore, by [2, Proposition 1.62(ii)], the mediating morphism $C \to X^2$ is again monic. The cocone thus lifts to the category $\mathbf{Rel}(X)$ of (binary) relations over X. Because the forgetful functor $\mathbf{Rel}(X) \to \mathscr{C}/X^2$ is fully faithful, the cocone remains colimiting in $\mathbf{Rel}(X)$. Finally, $\mathbf{Rel}(X)$ is a preorder category, so the colimit of the considered, directed diagram is equally a colimit of the underlying discrete diagram, which is $S^{\overline{+}}$ by definition. □

The next result will be useful to show that the relational transitive closure of the Howe closure of substitution-closed bisimilarity is symmetric on states.

Lemma 9.10. *For any reflexive span $R \to X^2$, $R^{\overline{+}}$ is symmetric if there exists a span morphism $R \to R^{\overline{+}\dagger}$.*

Proof. Given a morphism $j \colon R \to R^{\overline{+}\dagger}$, we consider the composite
$$R^{\overline{+}} = \bigcup_{n>0} \overline{R^{;n}} \to \bigcup_{n>0} \overline{(R^{\overline{+}\dagger})^{;n}} \cong \bigcup_{n>0} \overline{((R^{\overline{+}})^{;n})^\dagger} \to \bigcup_{n>0} \overline{R^{\overline{+}\dagger}} \cong \bigcup_{n>0} R^{\overline{+}\dagger} \cong R^{\overline{+}\dagger},$$
where the first morphism is obtained from j, and the second one is obtained from morphisms $(R^{\overline{+}})^{;n} \to R^{\overline{+}}$. □

9.2. **Howe closure on states.** We fix an operational semantics signature (Σ_0, Σ_1) on a Howe context $\mathbf{s}, \mathbf{t} \colon \mathbb{C}_1 \to \mathbb{C}_0$, and recall

- from Notation 5.5 that $\mathbf{Z}_0$ denotes the initial Σ_0-monoid,
- from Theorem 5.18 that $\mathbf{Z}$ denotes the initial vertical $\check{\Sigma}_1$-algebra (hence in particular that $\mathscr{D}(\mathbf{Z}) = \mathbf{Z}_0$), and
- from Notation 6.11 that $\sim^{\otimes}$ denotes substitution-closed bisimilarity on $\mathbf{Z}$ (hence in particular that $\mathscr{D}(\sim^{\otimes}) = \sim_0^{\otimes}$ is its state part).

Definition 9.11. Let $\Sigma_0^H : \widehat{\mathbb{C}_0}/\mathbf{Z}_0^2 \to \widehat{\mathbb{C}_0}/\mathbf{Z}_0^2$ map any span $X \to \mathbf{Z}_0^2$ to

$$\Sigma_0(X) + (X; \sim_0^{\otimes}) \to \Sigma_0(\mathbf{Z}_0)^2 + \mathbf{Z}_0^2 \to \mathbf{Z}_0^2.$$

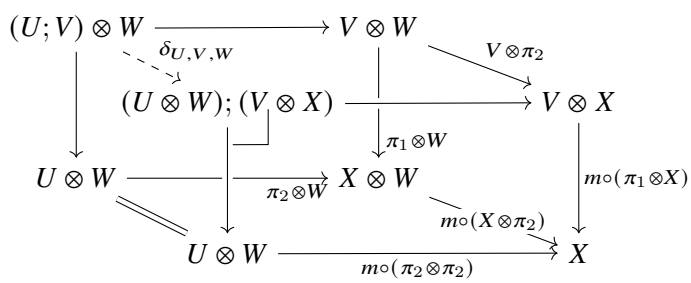
This functor Σ_0^H is clearly inspired from the standard Howe closure. We now want to prove that it is pointed strong, which requires us to equip $\widehat{\mathbb{C}_0}/\mathbf{Z}_0^2$ with monoidal structure. But $\mathbf{Z}_0^2$ is a monoid, and it is well-known [55, §2] that any slice of a monoidal category over any monoid M is again monoidal. The tensor of $X \to M$ and $Y \to M$ is simply $X \otimes Y \to M \otimes M \to M$, and the unit is $I \to M$. We may thus state the following result.

Proposition 9.12. *The functor Σ_0^H is pointed strong.*

For proving this, we first need the following.

Lemma 9.13. *For any monoid X in any monoidal category $\mathscr{C}$ with finite limits, there is a natural transformation with components $\delta_{U,V,W} : (U;V) \otimes W \to U \otimes W; V \otimes X$ in $\mathscr{C}/X^2$.*

Proof. Let $m \colon X \otimes X \to X$ denote multiplication. By tensoring the defining pullback of $U;V$ with W we obtain the back face below.



By universal property of pullback, we then get the dashed arrow making all faces commute, which gives our candidate $\delta_{U,V,W}$. Naturality follows by universal property of pullback. □

Proof of Proposition 9.12. Because the tensor preserves all colimits on the left, pointed strong endofunctors are closed under coproducts, so it suffices to show that both terms of the sum are pointed strong. The first one inherits the pointed strength of Σ_0, while the pointed strength of $-; \sim_0^{\otimes}$ follows from Lemma 9.13 and substitution-closedness of $\sim_0^{\otimes}$, like so: $(U; \sim_0^{\otimes}) \otimes V \to (U \otimes V); (\sim_0^{\otimes} \otimes \mathbf{Z}_0) \to (U \otimes V); \sim_0^{\otimes}$. □

Presheaf categories being well-known to be closed under the slice construction, we have the following.

Lemma 9.14. *The category $\widehat{\mathbb{C}_0}/\mathbf{Z}_0^2$ is a presheaf category.*

This allows us to deduce the following.

Proposition 9.15. *The endofunctor Σ_0^H is finitary.*

Proof. By commutation of filtered colimits with finite limits in presheaf categories. □

By Proposition 9.12, Lemma 9.14, and Proposition 9.15, the following is legitimate.

Definition 9.16. Let $H_0 = \mathbf{Z}_{\Sigma_0^H}$ denote the initial Σ_0^H-monoid. We denote by $\pi_1, \pi_2 \colon H_0 \to \mathbf{Z}_0$ the left and right projections.

By Proposition 5.4, we also get the following for free.

Proposition 9.17. *The object $H_0 \to \mathbf{Z}_0^2$ is an initial algebra for the endofunctor $\widehat{\mathbb{C}_0}/\mathbf{Z}_0^2 \to \widehat{\mathbb{C}_0}/\mathbf{Z}_0^2$ mapping any $X \to \mathbf{Z}_0^2$ to $I + \Sigma_0^H(X) \to \mathbf{Z}_0^2$.*

Proposition 9.18. *The underlying object H_0 is a Σ_0-monoid.*

Proof. Directly follows from the Σ_0^H-monoid structure. □

Next, we exhibit an alternative characterisation of H_0, which relies on the following result.

Lemma 9.19 (Packing lemma)**.** *Consider finitary endofunctors F and G on a cocomplete category $\mathbf{C}$, and let $G^\star(A) \cong \mu S.(A + G(S))$ denote the 'free G-algebra' monad* [49, Theorem 2.1]. *Then we have $\mu S.(F(S) + G(S)) \cong \mu S.G^\star(F(S))$.*

Proof. Indeed, we have

$$\begin{array}{rcll} \mu S.G^\star(F(S)) & \cong & \mu S.\mu U.(F(S) + G(U)) & \\ & \cong & \mu S.(F(S) + G(S)) & \text{(by the Diagonal rule [7, Theorem 16]).} \end{array}$$ □

Proposition 9.20. *The object $H_0 \to \mathbf{Z}_0^2$ is an initial algebra for the endofunctor $\Sigma_0^{H\prime} \colon \widehat{\mathbb{C}_0}/\mathbf{Z}_0^2 \to \widehat{\mathbb{C}_0}/\mathbf{Z}_0^2$ mapping any $X \to \mathbf{Z}_0^2$ to $I; \sim_0^{\otimes *} + \Sigma_0(X); \sim_0^{\otimes *} \to \mathbf{Z}_0^2$.*

Proof. Let $F(S) = I + \Sigma_0(S)$ and $G(S) = S; \sim_0^{\otimes}$. We observe that G preserves coproducts (because pullback along the first projection $\sim_0^{\otimes} \to \mathbf{Z}_0$, as a left adjoint, preserves colimits), so that $G^\star(U) \cong \sum_n G^n(U)$. By commutation of coproducts with $U; -$, we thus have

$$G^\star(U) \cong \sum_n G^n(U) \cong \sum_n U; (\sim_0^{\otimes})^n \cong U; \sum_n (\sim_0^{\otimes})^n = U; \sim_0^{\otimes *}.$$

Thus, by Lemma 9.19, we get $G^\star(F(S)) \cong (I + \Sigma_0(S)); \sim_0^{\otimes *} \cong (I; \sim_0^{\otimes *}) + (\Sigma_0(S); \sim_0^{\otimes *})$, as desired. □

9.3. **Double categorical notation.** Our next goal is to define the Howe closure on transitions. For this, we appeal to Morton's *double bicategories* [45]. They are a refinement of double categories, in which both the horizontal and vertical categories are bicategories. We rely in particular on his Theorem 4.1.3, which (when dualised) states that for any category $\mathscr{C}$ with pullbacks, there is a double bicategory $2Sp(\mathscr{C})$:

- objects are objects of $\mathscr{C}$,
- both the vertical and horizontal bicategories are $Span(\mathscr{C})$,
- cells, called *double spans*, are precisely commuting diagrams of the following form.

$$\begin{array}{ccccc} A & \longleftarrow & B & \longrightarrow & C \\ \uparrow & & \uparrow & & \uparrow \\ A' & \longleftarrow & B' & \longrightarrow & C' \\ \downarrow & & \downarrow & & \downarrow \\ A'' & \longleftarrow & B'' & \longrightarrow & C'' \end{array} \tag{9.1}$$

We will not need the rest of the structure. All we need to know is that cells compose horizontally and vertically just as in a weak double category. We will use the double bicategories $2Sp(\widehat{\mathbb{C}_0})$ and $2Sp(\widehat{\mathbb{C}_1})$.

Notation 9.21. *We use the following notational conventions.*

- *We denote cells $2Sp(\widehat{\mathbb{C}_0})$ such as* (9.1) *above by*

$$\begin{tikzcd} A \arrow[r, "B", "\circ"{marking}] \arrow[d, "A'", "\circ"{marking}] \arrow[dr, phantom, "B'"] & C \arrow[d, "C'", "\circ"{marking}] \\ A'' \arrow[r, "B''"', "\circ"{marking}] & C''. \end{tikzcd}$$

- *Furthermore, cells in $\widehat{\mathbb{C}_1}$ of the form below left will be denoted as below right.*

$$\begin{tikzcd} X_0\mathbf{s} & S_0\mathbf{s} \arrow[l] \arrow[r] & Y_0\mathbf{s} \\ X_1 \arrow[u] \arrow[d] & S_1 \arrow[l] \arrow[r] \arrow[u] \arrow[d] & Y_1 \arrow[u] \arrow[d] \\ X_0'\mathbf{t} & S_0'\mathbf{t} \arrow[l] \arrow[r] & Y_0'\mathbf{t} \end{tikzcd} \qquad \begin{tikzcd} X_0 \arrow[r, "S_0", "\circ"{marking}] \arrow[d, "X_1", "\bullet"{marking}] \arrow[dr, phantom, "S_1"] & Y_0 \arrow[d, "Y_1", "\bullet"{marking}] \\ X_0' \arrow[r, "S_0'"', "\circ"{marking}] & Y_0' \end{tikzcd} \tag{9.2}$$

  Explicitly, spans of the form $X_0\mathbf{s} \leftarrow X_1 \rightarrow X_0'\mathbf{t}$ are denoted by $X_0 \xrightarrow{X_1} X_0'$, while spans of the form $X_0 \leftarrow S_0 \rightarrow Y_0$ are still denoted by $X_0 \xrightarrow{S_0} Y_0$, but silently coerced by $\Delta_{\mathbf{s}}$ or $\Delta_{\mathbf{t}}$ depending on context.
- *For both types of cells, we collapse identity borders, as usual.*
- *When a span is trivial on one side, we use standard arrows for its borders, and a double arrow for its middle arrow, all in the relevant direction. E.g., the diagram below left may be depicted as below right.*

$$\begin{tikzcd} A & B \arrow[l] \arrow[r] & C \\ A' \arrow[u, "a"] & B' \arrow[l] \arrow[r] \arrow[u, "b"] & C' \arrow[u, "c"] \end{tikzcd} \qquad \begin{tikzcd} A \arrow[r, "B", "\circ"{marking}] & C \\ A' \arrow[u, "a"] \arrow[r, "B'"', "\circ"{marking}] \arrow[ur, phantom, "b\Uparrow"] & C' \arrow[u, "c"] \end{tikzcd}$$

Cells of the form (9.2) live in $2Sp(\widehat{\mathbb{C}_1})$, hence may be composed horizontally. Relevant examples of vertical composition will be obtained by embedding cells of the form (9.1) along $\Delta_{\mathbf{s}}$ (resp. $\Delta_{\mathbf{t}}$), and vertically composing with cells of the form (9.2) in $2Sp(\widehat{\mathbb{C}_1})$. This yields a top (resp. bottom) action of $2Sp(\widehat{\mathbb{C}_0})$, which we both denote by mere pasting.

Lemma 9.22. *Given a composable pasting diagram made of cells of both types, any two parsings agree up to isomorphism.*

Proof. By interchange of limits. □

Let us end this subsection by generalising simulations to $2Sp(\widehat{\mathbb{C}_1})$. By Proposition 4.17, the span denoted by a cell (9.2, right) in $2Sp(\widehat{\mathbb{C}_1})$ is a simulation iff the top left square in the corresponding diagram (9.2, left) is a pointwise weak pullback. Abstracting over this:

Definition 9.23. A cell in any double bicategory of the form $2Sp(\widehat{\mathbb{C}})$ is a *simulation* iff its top left square is a pointwise weak pullback.

Proposition 9.24. *Simulations are closed under horizontal and vertical composition in any double bicategory of the form* $2Sp(\widehat{\mathbb{C}})$.

In order to prove this, we need the following weak analogues of the pullback lemma.

Notation 9.25. *We denote weak pullbacks (in any category) by dashed corners.*

Lemma 9.26. *In any category (resp. presheaf category),*

(i) *for any commuting diagram*

$$\begin{tikzcd} A \arrow[r] \arrow[d] \arrow[dr, phantom, "\ulcorner", very near start] & B \arrow[r] \arrow[d] \arrow[dr, phantom, "\ulcorner", very near start] & C \arrow[d] \\ D \arrow[r] & E \arrow[r] & F, \end{tikzcd}$$

if both squares are weak pullbacks (resp. pointwise weak pullbacks), then so is the outer rectangle; and

(ii) *for any commuting diagram*

$$\begin{tikzcd} A \arrow[r] \arrow[d] \arrow[dr, phantom, "\ulcorner", very near start] \arrow[rr, bend left] & B \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & C \arrow[d] \\ D \arrow[r] & E \arrow[r] & F, \end{tikzcd}$$

if the right-hand square is a pullback and the outer rectangle is a weak pullback (resp. pointwise weak pullback), then the left-hand square is a weak pullback (resp. pointwise weak pullback).

Proof. Similar to the proof of the standard pullback lemma. □

Proof of Proposition 9.24. Straightforward, using Lemma 9.26. □

For vertical composition with cells from $2Sp(\widehat{\mathbb{C}_0})$, as in Notation 9.21, we will also need the following.

Proposition 9.27. *Precomposition with* $\mathbf{s}$ *and* $\mathbf{t}$ *yields maps* $2Sp(\widehat{\mathbb{C}_0}) \to 2Sp(\widehat{\mathbb{C}_1})$ *between cell sets, which preserve borders and simulations.*

Proof. Both precomposition functors straightforwardly preserve pointwise weak pullbacks. □

9.4. **Howe closure on transitions.** Let us now define the Howe closure on transitions. First, we delineate an ambient category $\mathbf{C}^H_{\mathbf{Z}}$. The idea is that objects of this category should be transition systems $S \to \mathbf{Z}^2$ over $\mathbf{Z}^2$ whose image under the projection $\mathbf{C}/\mathbf{Z}^2 \to \widehat{\mathbb{C}_0}/\mathbf{Z}^2_0$ is precisely $H_0 \to \mathbf{Z}^2_0$. Thus, an object of $\mathbf{C}^H_{\mathbf{Z}}$ consists of an object $S_1 \in \widehat{\mathbb{C}_1}$, equipped with a dashed cone to the outer part of the diagram below.

$$\begin{tikzcd} \mathbf{Z}_0\mathbf{s} & H_0\mathbf{s} \arrow[l] \arrow[r] & \mathbf{Z}_0\mathbf{s} \\ \mathbf{Z}_1 \arrow[u] \arrow[d] & S_1 \arrow[l, dashed] \arrow[r, dashed] \arrow[u, dashed] \arrow[d, dashed] & \mathbf{Z}_1 \arrow[u] \arrow[d] \\ \mathbf{Z}_0\mathbf{t} & H_0\mathbf{t} \arrow[l] \arrow[r] & \mathbf{Z}_0\mathbf{t} \end{tikzcd} \tag{9.3}$$

Equivalently, they are morphisms over the limit, so that we may define $\mathbf{C}^H_{\mathbf{Z}}$ as a slice category by merely stating the following.

Definition 9.28. Let $R^{\partial H}$ denote the limit of the outer part of (9.3).

Definition 9.29. $\mathbf{C}^H_{\mathbf{Z}}$ is the category of cones over the outer part of (9.3), or equivalently, it is the slice category $\widehat{\mathbb{C}_1}/R^{\partial H}$.

Furthermore, we denote by $\mathscr{U}^H_{\mathbf{Z}} : \mathbf{C}^H_{\mathbf{Z}} \to \mathbf{C}/\mathbf{Z}^2$ the forgetful functor.

Proposition 9.30. *The initial object in* $\mathbf{C}^H_{\mathbf{Z}}$ *is the span* $\mathbf{Z} \leftarrow H_0 \to \mathbf{Z}$*, i.e., the one with* $S_1 = 0$.

Definition 9.31. Let $\Sigma^H_1 : \mathbf{C}^H_{\mathbf{Z}} \to \mathbf{C}^H_{\mathbf{Z}}$ map any object $\mathbf{Z} \leftarrow S \to \mathbf{Z}$ to the coproduct of the following two pastings.

$$\begin{tikzcd}
 & \mathbf{Z}_0 \arrow[r, "H_0"] \arrow[dd, bend right=60, "\Sigma^F_1(\mathbf{Z})"'] & \mathbf{Z}_0 \arrow[dd, bend left=60, "\Sigma^F_1(\mathbf{Z})"] & \\
\mathbf{Z}_1 \arrow[ur, phantom, "\Uparrow"] & \Sigma_0(\mathbf{Z}_0) \arrow[l, Rightarrow] \arrow[u] \arrow[r, "\Sigma_0(H_0)"] \arrow[d, "\Sigma^F_1(S)"] & \Sigma_0(\mathbf{Z}_0) \arrow[u] \arrow[d, "\Sigma^F_1(\mathbf{Z})"] & \mathbf{Z}_1 \arrow[ul, phantom, "\Uparrow"] \arrow[l, Rightarrow] \\
 & \mathbf{Z}_0 \arrow[r, "H_0"'] & \mathbf{Z}_0 &
\end{tikzcd}
\qquad
\begin{tikzcd}
 & H_0 \arrow[d, phantom, "\Uparrow"] & \\
\mathbf{Z}_0 \arrow[r, "H_0"] \arrow[d, "\mathbf{Z}_1"'] \arrow[uur, bend left] & \mathbf{Z}_0 \arrow[r, "\sim_0^{\otimes}"] \arrow[d, "\mathbf{Z}_1"] \arrow[dl, phantom, "S_1"] & \mathbf{Z}_0 \arrow[d, "\mathbf{Z}_1"] \arrow[dl, phantom, "\sim_1^{\otimes}"] \\
\mathbf{Z}_0 \arrow[r, "H_0"'] \arrow[dr, bend right] & \mathbf{Z}_0 \arrow[r, "\sim_0^{\otimes}"'] \arrow[d, phantom, "\Downarrow"] & \mathbf{Z}_0 \arrow[u] \\
 & H_0 &
\end{tikzcd}
\tag{9.4}$$

Proposition 9.32. *The functor* $\Sigma^H_1 : \mathbf{C}^H_{\mathbf{Z}} \to \mathbf{C}^H_{\mathbf{Z}}$ *is finitary.*

Proof. The forgetful functor $\mathbf{C}^H_{\mathbf{Z}} \cong \widehat{\mathbb{C}_1}/R^{\partial H} \to \widehat{\mathbb{C}_1}$ creates colimits, so it suffices to show that the composite $\mathbf{C}^H_{\mathbf{Z}} \xrightarrow{\Sigma^H_1} \mathbf{C}^H_{\mathbf{Z}} \to \widehat{\mathbb{C}_1}$ is finitary. This functor maps any S to $\Sigma^F_1(S) + S_1; {\sim_1}^{\otimes}$, hence is finitary because Σ^F_1 is and $-; \sim_1^{\otimes}$ is cocontinuous. □

The last result legitimates the following definition.

Definition 9.33. Let $H_{\mathbf{Z}}$ denote the initial Σ^H_1-algebra. We call $H := \mathscr{U}^H_{\mathbf{Z}}(H_{\mathbf{Z}}) \in \mathbf{C}/\mathbf{Z}^2$ the *Howe closure* of substitution-closed bisimilarity.

We readily can prove the following.

Lemma 6.16. *There exists a span morphism* $\mathbf{i}^H : \sim^{\otimes} \to H$.

Proof. By construction, the underlying object of H is in particular a $\check{\Sigma}_1$-algebra, so by initiality we obtain a unique span morphism $\mathbf{Z} \to H$ — in other words H is reflexive. Furthermore, again by construction, H is an algebra for the endofunctor $-; \sim^{\otimes}$ on $\mathbf{C}/\mathbf{Z}^2$. We thus may form the composite $\sim^{\otimes} \cong \mathbf{Z}; \sim^{\otimes} \to H; \sim^{\otimes} \to H$. □

9.5. **Alternative characterisations of the Howe closure.** In this section, we exhibit a few alternative characterisations of the Howe closure on transitions. The definition in the previous section is convenient for proving that the transitive closure is symmetric, while our final alternative characterisation will enable a conceptual proof of the simulation property.

First of all, we have:

Lemma 9.35. *The object $H_{\mathbf{Z}} \in \mathbf{C}^H_{\mathbf{Z}}$ is (isomorphic to) the initial algebra of the endofunctor $\Sigma^H_{1,pack} : \mathbf{C}^H_{\mathbf{Z}} \to \mathbf{C}^H_{\mathbf{Z}}$ mapping any $\mathbf{Z} \leftarrow S \to \mathbf{Z}$ to the following pasting.*

$$\begin{tikzcd}
 & & & & \mathbf{Z}_0 \arrow[dll, bend right] \arrow[dr, bend left, "\sim_0^{\otimes *}"] & \\
 & \mathbf{Z}_0 \arrow[rr, "H_0"] \arrow[rrr, bend left, "H_0"] & & \mathbf{Z}_0 \arrow[r, equal] \arrow[d, "\Sigma_1^F(\mathbf{Z})"] & \mathbf{Z}_0 \arrow[r, "\sim_0^{\otimes *}"] \arrow[d] & \mathbf{Z}_0 \arrow[d] \\
\mathbf{Z}_1 & \Sigma_0(\mathbf{Z}_0) \arrow[l, Rightarrow] \arrow[u] \arrow[r, "\Sigma_0(H_0)"] \arrow[d, "\Sigma_1^F(\mathbf{Z})"'] & \Sigma_0(\mathbf{Z}_0) \arrow[u] \arrow[r, Rightarrow] \arrow[d, "\Sigma_1^F(S)"'] & \mathbf{Z}_1 \arrow[d] & \mathbf{Z}_1 \arrow[r, "\sim_1^{\otimes *}"] & \mathbf{Z}_1 \arrow[d] \\
 & \mathbf{Z}_0 \arrow[rr, "H_0"'] \arrow[rrr, bend right, "H_0"'] & & \mathbf{Z}_0 \arrow[r, equal] & \mathbf{Z}_0 \arrow[r, "\sim_0^{\otimes *}"'] & \mathbf{Z}_0 \arrow[u]
\end{tikzcd} \tag{9.5}$$

For the proof, we need an intermediate result, Corollary 9.37 below, which relies on the following lemma.

Lemma 9.36. *Consider any diagram*

$$\begin{tikzcd}
\mathscr{A} \arrow[r, "F"] & \mathscr{A} \\
\mathscr{B} \arrow[u, "K"] \arrow[r, "G"'] \arrow[ur, phantom, "\Downarrow\alpha"] & \mathscr{B} \arrow[u, "K"']
\end{tikzcd}$$

of functors and natural transformations, such that F and G are finitary, and $\mathscr{A}$ and $\mathscr{B}$ are cocomplete. Let M be the induced endofunctor on the comma category $\mathscr{A}/K$, mapping $a \xrightarrow{f} Kb$ to $Fa \xrightarrow{Ff} FKb \xrightarrow{\alpha_b} KGb$. Then, given an object $f : a \to K(b)$ of $\mathscr{A}/K$, there is a unique morphism $f^* : F^\star a \to KG^\star b$ such that the following diagram commutes,

$$\begin{tikzcd}
F(F^*(a)) \arrow[rr, "\nu_a^F"] \arrow[d, "F(f^*)"'] & & F^*(a) \arrow[d, "f^*"] & a \arrow[l, "\eta_a^F"'] \arrow[d, "f"] \\
F(K(G^*(b)) \arrow[r, "\alpha_{G^*(b)}"'] & K(G(G^*(b))) \arrow[r, "K(\nu_b^G)"'] & K(G^*(b)) & K(b) \arrow[l, "K(\eta_b^G)"]
\end{tikzcd}$$

where $\nu_a : F(F^*(a)) \to F^*(a)$ denotes the canonical F-algebra structure on $F^*(a)$, and similarly for G. Furthermore, the right-hand square above, viewed as a morphism $f \to f^*$ exhibits f^* as a free M-algebra on f.

Proof. Let us first observe that K lifts to a functor $\overline{K} : G\text{-}\mathbf{alg} \to F\text{-}\mathbf{alg}$, which maps $Gx \to x$ to $FKx \xrightarrow{\alpha_x} KGx \to Kx$. This in particular equips $K(G^*(b))$ with F-algebra structure. Let us thus define f^*, by universal property of $F^*(a)$, to be the unique F-algebra morphism $F^*(a) \to K(G^*(b))$ whose restriction to a is $K(\eta_b^G) \circ f$. This ensures in particular that the required diagram commutes.

Let us finally show that f^* is an initial M-algebra. For this, we observe that M-algebra structure on $f : a \to K(b)$ means morphisms u and v making the following diagram commute.

$$\begin{tikzcd}
F(a) \arrow[rr, "u"] \arrow[d, "F(f)"'] & & a \arrow[d, "f"] \\
F(K(b)) \arrow[r, "\alpha_b"'] & K(G(b)) \arrow[r, "K(v)"'] & K(b)
\end{tikzcd}$$

Thus, M-algebra structure (u, v) on f is exactly the same as F-algebra structure u on a, G-algebra structure v on b, and an F-algebra morphism $a \to \overline{K}(b)$. This shows that M-**alg** is isomorphic to the comma category F-**alg**$/\overline{K}$. But for any morphism, say (h, k) from f to any $f' \colon a' \to \overline{K}(b')$ in F-**alg**$/\overline{K}$, by universal property of $F^*(a)$ and $G^*(b)$, we get maps $\tilde{h}$ and $\tilde{k}$, respectively in F-**alg** and G-**alg**, making both triangles commute in the following diagram.

$$\begin{tikzcd}
F^*(a) \arrow[dr, dashed, "\tilde{h}"'] \arrow[dd, "f^*"'] & & a \arrow[ll, "\eta^F_a"'] \arrow[dl, "h"] \arrow[dd, "f"] \\
& a' \arrow[dd, "f'"] & \\
K(G^*(b)) \arrow[dr, dashed, "K(\tilde{k})"'] & & K(b) \arrow[ll, "K(\eta^G_b)"'] \arrow[dl, "k"] \\
& K(b') &
\end{tikzcd}$$

By functoriality of $\overline{K}$ and uniqueness in the universal property of $F^*(a)$, the left-hand square also commutes, so $(\tilde{h}, \tilde{k})$ is a morphism in F-**alg**$/\overline{K}$, as desired. Finally, uniqueness follows again by universal property of $F^*(a)$ and $G^*(b)$. □

Corollary 9.37. *Consider any diagram*

$$\begin{tikzcd}
\mathscr{A} \arrow[r, "U"] \arrow[d, "J"'] \arrow[dr, phantom, "\Downarrow\alpha"] & \mathscr{A} \arrow[d, "J"] \\
\mathscr{B} \arrow[r, "V"'] & \mathscr{B} \\
\mathscr{C} \arrow[u, "K"] \arrow[r, "W"'] \arrow[ur, phantom, "\Downarrow\beta"] & \mathscr{C} \arrow[u, "K"']
\end{tikzcd}$$

of functors and natural transformations such that J has a right adjoint, $\mathscr{A}$ and $\mathscr{C}$ are cocomplete, and U and W are finitary.

Furthermore, let M be the induced endofunctor on the comma category J/K, mapping $Ja \xrightarrow{f} Kc$ to $(JUa \xrightarrow{\alpha_a} VJa \xrightarrow{Vf} VKc \xrightarrow{\beta_c} KWc)$.

Then, given an object $f : Ja \to Kb$ of J/K, there is a unique morphism $f^* : JU^\star a \to KW^\star b$ making the following diagram commute.

$$\begin{tikzcd}
JUU^*a \arrow[r, "J\nu^U_a"] \arrow[d, "\alpha_{U^*a}"'] & JU^*a \arrow[dd, "f^*"] & Ja \arrow[l, "J(\eta^U_a)"'] \arrow[ddd, "f"] \\
VJU^*a \arrow[d, "Vf^*"'] & & \\
VKW^*b \arrow[d, "\beta_{W^*b}"'] & & \\
KWW^*b \arrow[r, "K(\nu^W_b)"'] & KW^*b & Kb \arrow[l, "K(\eta^W_b)"]
\end{tikzcd}$$

Furthermore, the right-hand square above exhibits f^* as a free M-algebra on f.

Proof. This follows from Lemma 9.36 with $F = U$, $G = W$, K as RK where R is the right adjoint of J, by considering the mate $\alpha' : UR \to RV$ of $\alpha : JU \to VJ$, defined as $UR \xrightarrow{\eta UR} RJUR \xrightarrow{R\alpha R} RVJR \xrightarrow{RV\varepsilon} RV$, (where η and ε denote the unit and the counit of the adjunction $J \dashv R$) and composing it with β to get a natural transformation $URK \to RKW$. $\square$

Proof of Lemma 9.35. Let us denote by M the endofunctor on $\mathbf{C}^H_{\mathbf{Z}}$ mapping an object $\mathbf{Z} \leftarrow S \to \mathbf{Z}$ to the the right pasting of Diagram 9.4:

$$M\left(\begin{tikzcd} \mathbf{Z}_0 \arrow[r, "H_0"] \arrow[d, "\mathbf{z}_1"'] \arrow[dr, phantom, "S_1"] & \mathbf{Z}_0 \arrow[d, "\mathbf{z}_1"] \\ \mathbf{Z}_0 \arrow[r, "H_0"'] & \mathbf{Z}_0 \end{tikzcd}\right) = \begin{tikzcd} \mathbf{Z}_0 \arrow[r, "H_0"] \arrow[d, "\mathbf{z}_1"'] \arrow[dr, phantom, "S_1"] & \mathbf{Z}_0 \arrow[r, "\sim_0^{\otimes}"] \arrow[d, "\mathbf{z}_1"] \arrow[dr, phantom, "\sim_1^{\otimes}"] & \mathbf{Z}_0 \arrow[d, "\mathbf{z}_1"] \\ \mathbf{Z}_0 \arrow[r, "H_0"'] & \mathbf{Z}_0 \arrow[r, "\sim_0^{\otimes}"'] & \mathbf{Z}_0 \end{tikzcd}$$

Now, by the packing lemma, it is enough to show that

$$M^\star\left(\begin{tikzcd} \mathbf{Z}_0 \arrow[r, "H_0"] \arrow[d, "\mathbf{z}_1"'] \arrow[dr, phantom, "S_1"] & \mathbf{Z}_0 \arrow[d, "\mathbf{z}_1"] \\ \mathbf{Z}_0 \arrow[r, "H_0"'] & \mathbf{Z}_0 \end{tikzcd}\right) = \begin{tikzcd} \mathbf{Z}_0 \arrow[r, "H_0"] \arrow[d, "\mathbf{z}_1"'] \arrow[dr, phantom, "S_1"] & \mathbf{Z}_0 \arrow[r, "\sim_0^{\otimes *}"] \arrow[d, "\mathbf{z}_1"] \arrow[dr, phantom, "\sim_1^{\otimes *}"] & \mathbf{Z}_0 \arrow[d, "\mathbf{z}_1"] \\ \mathbf{Z}_0 \arrow[r, "H_0"'] & \mathbf{Z}_0 \arrow[r, "\sim_0^{\otimes *}"'] & \mathbf{Z}_0 \end{tikzcd}$$

To this end, we are going to organise $\mathbf{C}^H_{\mathbf{Z}}$ as a comma category on which M acts, so as to apply Corollary 9.37. Let $\mathscr{A}$ denote the category of objects S_1 equipped with a span $\mathbf{Z}_1 \leftarrow S_1 \to \mathbf{Z}_1$ and $\mathscr{B}$ denote the product category $\mathscr{B}_{\mathbf{s}} \times \mathscr{B}_{\mathbf{t}}$, where $\mathscr{B}_\sigma$ is the category of objects S_1 equipped with a span $\mathbf{Z}_0\sigma \leftarrow S_1 \to \mathbf{Z}_0\sigma$.

Let $J : \mathscr{A} \to \mathscr{B}$ denote the functor mapping $\mathbf{Z}_1 \leftarrow S_1 \to \mathbf{Z}_1$ to $(\mathbf{Z}_0\mathbf{s} \leftarrow S_1 \to \mathbf{Z}_0\mathbf{s}, \mathbf{Z}_0\mathbf{t} \leftarrow S_1 \to \mathbf{Z}_0\mathbf{t})$, by postcomposing with the relevant morphisms. As the forgetful functor from a category of spans creates colimits, J is cocontinuous and thus has a right adjoint by [2, Theorem 1.66], since its domain is a locally presentable category. Let $K : 1 \to \mathscr{B}$ be the functor selecting the pair $(\mathbf{Z}_0\mathbf{s} \leftarrow H_0\mathbf{s} \to \mathbf{Z}_0\mathbf{s}, \mathbf{Z}_0\mathbf{t} \leftarrow H_0\mathbf{t} \to \mathbf{Z}_0\mathbf{t})$. Now, it is straightforward to check that $\mathbf{C}^H_{\mathbf{Z}}$ is isomorphic to the comma category J/K.

Next, we reconstruct M as acting on J/K through this isomorphism in order to fit the setting of Corollary 9.37.

Let

- $U : \mathscr{A} \to \mathscr{A}$ denote the functor mapping $\mathbf{Z}_1 \leftarrow S_1 \to \mathbf{Z}_1$ to $\mathbf{Z}_1 \leftarrow S_1 \leftarrow S_1; \sim_1^{\otimes} \to \sim_1^{\otimes} \to \mathbf{Z}_1$;
- $V : \mathscr{B} \to \mathscr{B}$ denote the functor $V_{\mathbf{s}} \times V_{\mathbf{t}}$, where $V_\sigma : \mathscr{B}_\sigma \to \mathscr{B}_\sigma$ maps $\mathbf{Z}_0\sigma \leftarrow S_1 \to \mathbf{Z}_0\sigma$ to $\mathbf{Z}_0\sigma \leftarrow S_1; \sim_0^{\otimes} \sigma \to \mathbf{Z}_0\sigma$;
- $W : 1 \to 1$ denote the identity endofunctor.

Now we apply Corollary 9.37 with suitable $\alpha : JU \to VJ$ and $\beta : WK \to KW$ so that M corresponds to our endofunctor through the isomorphism $\mathbf{C}^H_{\mathbf{Z}} \cong J/K$. Since $U^\star(\mathbf{Z}_1 \leftarrow$

$S_1 \to \mathbf{Z}_1) = (\mathbf{Z}_1 \leftarrow S_1; \sim_1^{\otimes *} \to \mathbf{Z}_1)$, the only thing to check is that the proposed definition for $M^\star(S_1)$ indeed defines a M-algebra, and that $S_1 \to U^\star(S_1)$ induces a morphism $S_1 \to M^\star(S_1)$, which is straightforward. □

Let us now turn to our final characterisation of H, which relies on the following category, which is a relaxation of $\mathbf{C}^H_{\mathbf{Z}}$, in which the left-hand object in (9.3) is only forced to coincide with $\mathbf{Z}$ on $\widehat{\mathbb{C}_0}$.

Definition 9.38. Let $\mathbf{C}^H_{lax}$ denote the category whose objects consist of objects X_1 and S_1 in $\widehat{\mathbb{C}_1}$, equipped with dashed arrows making the following diagram commute.

$$\begin{tikzcd}
\mathbf{Z}_0\mathbf{s} & H_0\mathbf{s} \arrow[l] \arrow[r] & \mathbf{Z}_0\mathbf{s} \\
X_1 \arrow[u, dashed] \arrow[d, dashed] & S_1 \arrow[l, dashed] \arrow[r, dashed] \arrow[u, dashed] \arrow[d, dashed] & \mathbf{Z}_1 \arrow[u] \arrow[d] \\
\mathbf{Z}_0\mathbf{t} & H_0\mathbf{t} \arrow[l] \arrow[r] & \mathbf{Z}_0\mathbf{t}
\end{tikzcd} \tag{9.6}$$

Remark 9.39. Using the notation of §9.3, an object of $\mathbf{C}^H_{lax}$ is a cell of the form

$$\begin{tikzcd}
\mathbf{Z}_0 \arrow[r, "H_0", "\circ"{marking}] \arrow[d, "X_1"', "\bullet"{marking}] \arrow[dr, phantom, "S_1"] & \mathbf{Z}_0 \arrow[d, "\mathbf{Z}_1", "\bullet"{marking}] \\
\mathbf{Z}_0 \arrow[r, "H_0"', "\circ"{marking}] & \mathbf{Z}_0.
\end{tikzcd}$$

Such an object may also be viewed as a span of the form $X \leftarrow S \to \mathbf{Z}$ in $\mathbf{C}$, which projects down to $\mathbf{Z}_0 \leftarrow H_0 \to \mathbf{Z}_0$ in $\widehat{\mathbb{C}_0}$.

Let us briefly relate $\mathbf{C}^H_{lax}$ to other useful categories.

Definition 9.40. Let $\mathbf{2}$ denote the free category on the graph $0 \to 1$, and $\mathbf{C^2/Z}$ denote the comma category

$$\begin{tikzcd}
\mathbf{C^2/Z} \arrow[r] \arrow[d] \arrow[dr, phantom, "\Uparrow"] & 1 \arrow[d, "\mathbf{Z}"] \\
\mathbf{C^2} \arrow[r, "\mathrm{dom}"'] & \mathbf{C},
\end{tikzcd}$$

whose objects are spans of the form $X \leftarrow S \to \mathbf{Z}$.

Definition 9.41. We define a commuting diagram

$$\begin{tikzcd}
\mathbf{C}^H_{\mathbf{Z}} \arrow[r, hook, "\mathscr{I}^H"] \arrow[d, "\mathscr{U}^H_{\mathbf{Z}}"'] & \mathbf{C}^H_{lax} \arrow[d, "\mathscr{U}^H_{lax}"] \arrow[ddr, bend left, "\mathscr{V}"] & \\
\mathbf{C}/\mathbf{Z}^2 \arrow[r, hook, "\mathscr{J}^H"'] \arrow[drr, bend right, "\mathscr{W}"'] & \mathbf{C^2/Z} \arrow[dr, "\mathrm{dom}"] & \\
& & \mathbf{C^2}
\end{tikzcd} \tag{9.7}$$

of functors:

- $\mathscr{U}^H_{\mathbf{Z}}$ and $\mathscr{U}^H_{lax}$ are the obvious forgetful functors,
- $\mathscr{I}^H$ and $\mathscr{J}^H$ are the obvious embeddings, and
- $\mathscr{V}$ and $\mathscr{W}$ are defined by composition with the domain functor dom.

Proposition 9.42. *The initial object in* $\mathbf{C}^H_{lax}$ *is the span* $\mathbf{Z}_0 \leftarrow H_0 \rightarrow \mathbf{Z}$*, i.e., the one with* $X_1 = S_1 = 0$*.*

Let us now introduce the endofunctor of which our characterisation of H will be an initia algebra.

Definition 9.43. Let $\Sigma^H_{1,lax} \colon \mathbf{C}^H_{lax} \rightarrow \mathbf{C}^H_{lax}$ map any object $X \leftarrow S \rightarrow \mathbf{Z}$ to the following pasting.

$$\begin{tikzcd}
 & & & H_0 \arrow[dl, phantom, "\Uparrow"] & & \\
 & \mathbf{Z}_0 \arrow[rr, "H_0"] & & \mathbf{Z}_0 \arrow[r, equal] & \mathbf{Z}_0 \arrow[r, "\sim_0^{\otimes *}"] \arrow[d] & \mathbf{Z}_0 \arrow[d] \\
\check{\Sigma}_1(X)_1 \arrow[r, equal] \arrow[dd, bend right] & \Sigma_0(\mathbf{Z}_0) \arrow[u] \arrow[d, "\Sigma_1^F(X)"'] \arrow[rr, "\Sigma_0(H_0)"] \arrow[urr, phantom, "\Uparrow"] & & \Sigma_0(\mathbf{Z}_0) \arrow[u] \arrow[d, "\Sigma_1^F(S)"'] \arrow[r, equal] & \mathbf{Z}_1 \arrow[d] \arrow[r, phantom, "\sim_1^{\otimes *}"] & \mathbf{Z}_1 \arrow[d] \\
 & \mathbf{Z}_0 \arrow[rr, "H_0"'] \arrow[dr, bend right, "H_0"'] & & \mathbf{Z}_0 \arrow[r, equal] \arrow[d, phantom, "\Downarrow"] & \mathbf{Z}_0 \arrow[r, "\sim_0^{\otimes *}"'] & \mathbf{Z}_0 \\
 & & & H_0 & &
\end{tikzcd} \tag{9.8}$$

Remark 9.44. The difference with $\Sigma^H_{1,pack}$ is that, X being different from $\mathbf{Z}$ in general, we cannot use any $\check{\Sigma}_1$-algebra structure on the left.

Proposition 9.45. *The functor* $\Sigma^H_{1,lax} \colon \mathbf{C}^H_{lax} \rightarrow \mathbf{C}^H_{lax}$ *is finitary.*

Proof. Just as Proposition 9.32. □

Let us now work towards our final characterisation of H.

Definition 9.46. Let δ denote the natural transformation

$$\begin{tikzcd}
\mathbf{C}^H_{\mathbf{Z}} \arrow[r, "\mathscr{I}^H"] \arrow[d, "\Sigma^H_{1,pack}"'] \arrow[dr, phantom, "\Downarrow\delta"] & \mathbf{C}^H_{lax} \arrow[d, "\Sigma^H_{1,lax}"] \\
\mathbf{C}^H_{\mathbf{Z}} \arrow[r, "\mathscr{I}^H"'] & \mathbf{C}^H_{lax}
\end{tikzcd}$$

whose component at any $\mathbf{Z} \leftarrow S \rightarrow \mathbf{Z}$ in $\mathbf{C}^H_{\mathbf{Z}}$ is the following morphism in $\mathbf{C}^H_{lax}$.

$$\begin{tikzcd}
\check{\Sigma}_1(\mathbf{Z}) \arrow[d] & \check{\Sigma}_1(S); \sim^{\otimes *} \arrow[l] \arrow[r] \arrow[d, equal] & \mathbf{Z} \arrow[d, equal] \\
\mathbf{Z} & \check{\Sigma}_1(S); \sim^{\otimes *} \arrow[l] \arrow[r] & \mathbf{Z}
\end{tikzcd}$$

Furthermore, let $\mathscr{K}^H \colon \Sigma^H_{1,pack}\text{-}\mathbf{alg} \rightarrow \Sigma^H_{1,lax}\text{-}\mathbf{alg}$ denote the induced lifting of $\mathscr{I}^H$, as in

$$\begin{tikzcd}
\Sigma^H_{1,pack}\text{-}\mathbf{alg} \arrow[r, "\mathscr{K}^H"] \arrow[d] & \Sigma^H_{1,lax}\text{-}\mathbf{alg} \arrow[d] \\
\mathbf{C}^H_{\mathbf{Z}} \arrow[r, "\mathscr{I}^H"'] & \mathbf{C}^H_{lax},
\end{tikzcd}$$

where both vertical arrows denote forgetful functors.

Lemma 9.47. *The* $\Sigma^H_{1,lax}$*-algebra* $\mathscr{K}^H(H_{\mathbf{Z}})$ *is initial. In other words, letting* H_{lax} *denote any initial* $\Sigma^H_{1,lax}$*-algebra, we have* $\mathscr{K}^H(H_{\mathbf{Z}}) \cong H_{lax}$*.*

Proof. In order to apply Corollary 9.37, we organise $\mathbf{C}^H_{lax}$ as a comma category $J/\mathbf{C}^H_l$, decomposing its left and middle/right parts, with $J \colon \mathbf{C}^H_r \rightarrow \mathbf{C}^H_l$ defined as follows:

- $\mathbf{C}_l^H = \widehat{\mathbb{C}_1}/\Delta(\mathbf{Z}_0)$;
- $\mathbf{C}_r^H$ is the comma category

$$\begin{tikzcd} \mathbf{C}_r^H \arrow[r] \arrow[d, "pr_r"'] \arrow[dr, phantom, "\Uparrow"] & 1 \arrow[d, "\mathbf{Z}"] \\ \widehat{\mathbb{C}_1}/\Delta(H_0) \arrow[r, "\widehat{\mathbb{C}_1}/\Delta(\pi_2)"'] & \widehat{\mathbb{C}_1}/\Delta(\mathbf{Z}_0); \end{tikzcd}$$

  concretely, objects consist of a presheaf $S_1 \in \widehat{\mathbb{C}_1}$, together with dashed maps making the following diagram commute;

$$\begin{tikzcd} H_0\mathbf{s} \arrow[r, "\pi_2\mathbf{s}"] & \mathbf{Z}_0\mathbf{s} \\ S_1 \arrow[u, dashed] \arrow[d, dashed] \arrow[r, dashed] & \mathbf{Z}_1 \arrow[u] \arrow[d] \\ H_0\mathbf{t} \arrow[r, "\pi_2\mathbf{t}"'] & \mathbf{Z}_0\mathbf{t} \end{tikzcd} \tag{9.9}$$

- J is the composite

$$\mathbf{C}_r^H \xrightarrow{pr_r} \widehat{\mathbb{C}_1}/\Delta(H_0) \xrightarrow{\widehat{\mathbb{C}_1}\Delta(\pi_1)} \widehat{\mathbb{C}_1}/\Delta(\mathbf{Z}_0);$$

  concretely, J maps any object (9.9) to the following diagram.

$$\begin{tikzcd} \mathbf{Z}_0\mathbf{s} & H_0\mathbf{s} \arrow[l, "\pi_1\mathbf{s}"'] \\ & S_1 \arrow[u, dashed] \arrow[d, dashed] \\ \mathbf{Z}_0\mathbf{t} & H_0\mathbf{t} \arrow[l, "\pi_1\mathbf{t}"] \end{tikzcd}$$

We note that J is cocontinuous since colimits are computed pointwise in $\mathbf{C}_l^H$ and $\mathbf{C}_r^H$, and thus has a right adjoint R, since its domain is locally presentable, by [2, Theorem 1.66].

Furthermore, $\mathbf{C}_\mathbf{Z}^H$ is isomorphic to the comma category $J/\mathbf{Z}$[6].

We then define functors $\Sigma_{1,r}^H$ and $(\check{\Sigma}_1)_{|\mathbf{Z}_0}$ and a natural transformation h as in

$$\begin{tikzcd} \mathbf{C}_r^H \arrow[r, "\Sigma_{1,r}^H"] \arrow[d, "J"'] \arrow[dr, phantom, "\Downarrow h"] & \mathbf{C}_r^H \arrow[d, "J"] \\ \mathbf{C}_l^H \arrow[r, "(\check{\Sigma}_1)_{|\mathbf{Z}_0}"'] & \mathbf{C}_l^H, \end{tikzcd}$$

as follows:

- $(\check{\Sigma}_1)_{|\mathbf{Z}_0}$ is $\check{\Sigma}_1$ restricted to the fibre of $\widehat{\mathbb{C}_1}/\Delta \to \widehat{\mathbb{C}_0}$ over $\mathbf{Z}_0$, as in the left part of (9.8);
- $\Sigma_{1,r}^H$ acts as the right part of (9.8);
- h connects both parts using the left projection.

Now, we apply Corollary 9.37 twice with $U = \Sigma_{1,r}^H$, $V = (\check{\Sigma}_1)_{|\mathbf{Z}_0}$ and $\alpha = h$:

(1) for $W = V$, K the identity endofunctor, β the identity natural transformation, and $f = \mathrm{id}_\emptyset \colon J(\emptyset) \to \emptyset$, the induced endofunctor M is precisely $\Sigma_{1,lax}^H$, and Corollary 9.37 yields a morphism, say f^*, making the diagram below commute;

[6]Note that this decomposition as a comma category differs from the one in the proof of Lemma 9.35.

$$\begin{tikzcd}
J(\Sigma^H_{1,r}((\Sigma^H_{1,r})^*(\emptyset))) \arrow[r, "J(\nu_\emptyset^{\Sigma^H_{1,r}})"] \arrow[d, "h_{(\Sigma^H_{1,r})^*(\emptyset)}"'] & J((\Sigma^H_{1,r})^*(\emptyset)) \arrow[dd, "f^*"] & J(\emptyset) \arrow[l, "J(!)"'] \arrow[dd, "!"] \\
(\check{\Sigma}_1)_{|\mathbf{Z}_0}(J((\Sigma^H_{1,r})^*(\emptyset))) \arrow[d, "(\check{\Sigma}_1)_{|\mathbf{Z}_0}(f^*)"'] \arrow[r, phantom, "\text{(A)}"] & & \\
(\check{\Sigma}_1)_{|\mathbf{Z}_0}((\check{\Sigma}_1)^*_{|\mathbf{Z}_0}(\emptyset)) \arrow[r, "\nu_\emptyset^{(\check{\Sigma}_1)_{|\mathbf{Z}_0}}"'] & (\check{\Sigma}_1)^*_{|\mathbf{Z}_0}(\emptyset) & \emptyset \arrow[l, "!"]
\end{tikzcd}$$

(2) for $K : 1 \to \mathbf{C}^H_l$ selecting $\mathbf{Z}_0\mathbf{s} \leftarrow \mathbf{Z}_1 \to \mathbf{Z}_0\mathbf{t}$, β induced by the V-algebra structure on it, and $g = !_{\mathbf{Z}} \colon J(\emptyset) = \emptyset \to \mathbf{Z}$, the induced endofunctor N is precisely $\Sigma^H_{1,pack}$, and Corollary 9.37 yields a unique morphism, say g^*, making the diagram below commute.

$$\begin{tikzcd}
J(\Sigma^H_{1,r}((\Sigma^H_{1,r})^*(\emptyset))) \arrow[r, "J(\nu_\emptyset^{\Sigma^H_{1,r}})"] \arrow[d, "h_{(\Sigma^H_{1,r})^*(\emptyset)}"'] & J((\Sigma^H_{1,r})^*(\emptyset)) \arrow[dd, "g^*"] & J(\emptyset) \arrow[l, "J(!)"'] \arrow[dd, "!"] \\
(\check{\Sigma}_1)_{|\mathbf{Z}_0}(J((\Sigma^H_{1,r})^*(\emptyset))) \arrow[d, "(\check{\Sigma}_1)_{|\mathbf{Z}_0}(g^*)"'] \arrow[r, phantom, "\text{(B)}"] & & \\
(\check{\Sigma}_1)_{|\mathbf{Z}_0}(\mathbf{Z}) \arrow[d, "\nu_\emptyset^{(\check{\Sigma}_1)_{|\mathbf{Z}_0}}"'] & & \\
\mathbf{Z} \arrow[r, equal] & \mathbf{Z} \arrow[r, equal] & \mathbf{Z}
\end{tikzcd}$$

Both right-hand squares are trivial, and both left-hand hand squares are in fact the same, so that $f^* = g^*$. Finally, in terms of $J/\mathbf{C}^H_l$, the statement claims that the M-algebra structure of f^* decomposes as

$$\begin{tikzcd}
(\check{\Sigma}_1)_{|\mathbf{Z}_0}((\check{\Sigma}_1)^*_{|\mathbf{Z}_0}(\emptyset)) \arrow[d, "\nu_\emptyset^{(\check{\Sigma}_1)_{|\mathbf{Z}_0}}"'] & (\check{\Sigma}_1)_{|\mathbf{Z}_0}(J((\Sigma^H_{1,r})^*(\emptyset))) \arrow[l, "(\check{\Sigma}_1)_{|\mathbf{Z}_0}(f^*)"'] & J(\Sigma^H_{1,r}((\Sigma^H_{1,r})^*(\emptyset))) \arrow[l, "h_{(\Sigma^H_{1,r})^*(\emptyset)}"'] \arrow[d, equal] \\
(\check{\Sigma}_1)^*_{|\mathbf{Z}_0}(\emptyset) \arrow[d, equal] & & J(\Sigma^H_{1,r}((\Sigma^H_{1,r})^*(\emptyset))) \arrow[ll, "(\check{\Sigma}_1)_{|\mathbf{Z}_0}(f^*)"'] \arrow[d, "J(\nu_\emptyset^{\Sigma^H_{1,r}})"] \\
(\check{\Sigma}_1)^*_{|\mathbf{Z}_0}(\emptyset) & & J((\Sigma^H_{1,r})^*(\emptyset)), \arrow[ll, "f^*"']
\end{tikzcd}$$

which in terms of spans yields exactly the desired form. □

Corollary 9.48. *We have $\mathcal{J}^H(H) \cong \mathcal{U}^H_{lax}(H_{lax})$ in $\mathbf{C^2/Z}$.*

Proof. We have $\mathcal{J}^H(H) = \mathcal{J}^H(\mathcal{U}^H_{\mathbf{Z}}(H_{\mathbf{Z}})) = \mathcal{U}^H_{lax}(\mathcal{J}^H(H_{\mathbf{Z}})) \cong \mathcal{U}^H_{lax}(H_{lax})$ in $\mathbf{C^2/Z}$. □

9.6. **Simulation property.** Our next goal is to prove the following.

Lemma 6.17. *If Σ_1 preserves functional bisimulations, then the transition Howe closure H is a substitution-closed simulation.*

The rest of this subsection is devoted to the proof.

For substitution-closedness, $\sim_0^{\otimes}$ is reflexive and by Lemma 6.16 we have a span morphism $\sim_0^{\otimes} \to H_0$, so we may form the composite

$$H_0 \otimes \mathbf{Z} \to H_0 \otimes \sim_0^{\otimes} \to H_0 \otimes H_0 \to H_0,$$

where the last morphism is the monoid multiplication of H_0, as established in Proposition 9.18.

For the simulation property, we will use the characterisation of H as $\mathbf{Z}_{\Sigma^H_{1,lax}}$. For this, we need to lift the notion of simulation from $\mathbf{C}$ to $\mathbf{C}^H_{lax}$. Recalling the functor $\mathscr{V}\colon \mathbf{C}^H_{lax} \to \mathbf{C}^{\mathbf{2}}$ from Definition 9.41, which maps each span $X \leftarrow S \to \mathbf{Z}$ in $\mathbf{C}^H_{lax}$ to its left-hand leg $X \leftarrow S$, we have:

Definition 9.50. An object $X \leftarrow S \to \mathbf{Z}$ of $\mathbf{C}^H_{lax}$ is a *simulation* iff its image by $\mathscr{V}$, i.e., $X \leftarrow S$, is a functional bisimulation.

Next, we want to show that the computation of H as an initial chain in $\mathbf{C}^H_{lax}$ is preserved by $\mathscr{V}$. We intend to use this to apply Lemma 1.8, which will reduce our goal to proving that each object of the chain is a functional bisimulation.

Lemma 9.51. *All of the following functors are cocontinuous:*

(i) *any functor from the terminal category;*
(ii) *both functors* $\widehat{\mathbb{C}_1}/\Delta(\pi_i)\colon \widehat{\mathbb{C}_1}/\Delta(H_0) \to \widehat{\mathbb{C}_1}/\Delta(\mathbf{Z}_0)$*, for* $i = 1, 2$*, which map any span* $H_0\mathbf{s} \leftarrow S_1 \to H_0\mathbf{t}$ *to*

$$\mathbf{Z}_0\mathbf{s} \xleftarrow{\pi_i} H_0\mathbf{s} \leftarrow S_1 \to H_0\mathbf{t} \xrightarrow{\pi_i} \mathbf{Z}_0 t;$$

(iii) *the projection functor* $pr_r\colon \mathbf{C}^H_r \to \widehat{\mathbb{C}_1}/\Delta(H_0)$*, mapping any object* (9.9) *to (the pairing of) its left-hand border* $H_0\mathbf{s} \leftarrow S_1 \to H_0\mathbf{t}$*;*
(iv) *the embedding* $\widehat{\mathbb{C}_1}/\Delta(\mathbf{Z}_0) \hookrightarrow \widehat{\mathbb{C}_1}/\Delta = \mathbf{C}$*;*
(v) *the embedding* $\widehat{\mathbb{C}_1}/\Delta(H_0) \hookrightarrow \widehat{\mathbb{C}_1}/\Delta = \mathbf{C}$*; and*
(vi) $\mathscr{V}\colon \mathbf{C}^H_{lax} \to \mathbf{C}^{\mathbf{2}}$.

Proof.

(i) Trivial.
(ii) As post-composition functors, these have right adjoints.
(iii) $\mathbf{C}^H_r$ is by definition the comma category

$$\begin{tikzcd} \mathbf{C}^H_r \arrow[r] \arrow[d, "pr_r"'] \arrow[dr, phantom, "\Uparrow"] & 1 \arrow[d, "\mathbf{Z}"] \\ \widehat{\mathbb{C}_1}/\Delta(H_0) \arrow[r, "\widehat{\mathbb{C}_1}/\Delta(\pi_2)"'] & \widehat{\mathbb{C}_1}/\Delta(\mathbf{Z}_0), \end{tikzcd}$$

so by *(i)*, *(ii)*, and Proposition 1.1, the projection functor

$$\langle pr_r, !\rangle\colon \mathbf{C}^H_l \to \widehat{\mathbb{C}_1}/\Delta(H_0) \times 1 \cong \widehat{\mathbb{C}_1}/\Delta(H_0)$$

to the product preserves colimits.

(iv) In the commuting square

$$\begin{tikzcd} \widehat{\mathbb{C}_1}/\Delta(\mathbf{Z}_0) \arrow[r, hook] \arrow[d, "\langle pr_1, !\rangle"'] & \widehat{\mathbb{C}_1}/\Delta \arrow[d, "\langle pr_1, pr_0\rangle"] \\ \widehat{\mathbb{C}_1} \times 1 \arrow[r, "\widehat{\mathbb{C}_1} \times \Delta(\mathbf{Z}_0)"'] & \widehat{\mathbb{C}_1} \times \widehat{\mathbb{C}_0}, \end{tikzcd}$$

both vertical functors (which are the canonical projection functors) create colimits. Furthermore, the bottom functor preserves them, because

- colimits are pointwise in a product of cocomplete categories, and
- any functor from the terminal category is cocontinuous.

Thus, the top functor is cocontinuous, as desired.

(v) Same, with the square

$$\begin{tikzcd} \widehat{\mathbb{C}_1}/\Delta(H_0) \arrow[r, hook] \arrow[d] & \widehat{\mathbb{C}_1}/\Delta \arrow[d] \\ \widehat{\mathbb{C}_1} \times 1 \arrow[r, "\widehat{\mathbb{C}_1}\times\Delta(H_0)"'] & \widehat{\mathbb{C}_1} \times \widehat{\mathbb{C}_0}. \end{tikzcd}$$

(vi) The functor $\mathscr{V}\colon \mathbf{C}^H_{lax} \to \mathbf{C^2}$ is induced by universal properties of lax limits as in

$$\begin{tikzcd} & \mathbf{C}^H_r \arrow[rr, "pr_r"] \arrow[dd] & & \widehat{\mathbb{C}_1}/\Delta(H_0) \arrow[rr, hook] \arrow[dd, "\widehat{\mathbb{C}_1}/\Delta(\pi_1)"] \arrow[dl, phantom, "\Leftarrow\alpha"] & & \widehat{\mathbb{C}_1}/\Delta \arrow[dd, equal] \\ \mathbf{C}^H_{lax} \arrow[ur] \arrow[rr] \arrow[dr] \arrow[urr, phantom, "\Rrightarrow"] \arrow[drr, phantom, "\Leftarrow"] & & \mathbf{C}^H_m \arrow[ur] \arrow[rr] \arrow[dr] \arrow[urr, phantom, "\Rrightarrow"] \arrow[drr, phantom, "\Leftarrow"] & & \mathbf{C^2} \arrow[ur] \arrow[dr] \arrow[urr, phantom, "\Rrightarrow"] \arrow[drr, phantom, "\Leftarrow"] & & \\ & \mathbf{C}^H_l \arrow[rr, equal] & & \widehat{\mathbb{C}_1}/\Delta(\mathbf{Z}_0) \arrow[rr, hook] & & \widehat{\mathbb{C}_1}/\Delta, \end{tikzcd}$$

where $\mathbf{C}^H_m$ is defined as the lax limit of $\widehat{\mathbb{C}_1}/\Delta(\pi_1)$, and α, which is induced by universal property of $\widehat{\mathbb{C}_1}/\Delta$, has as component at any $p\colon X \to \Delta(H_0)$ the morphism

$$\begin{tikzcd} X \arrow[r, equal] \arrow[d, "p"'] & X \arrow[d, "\Delta(\pi_1)\circ p"] \\ \Delta(H_0) \arrow[r, "\Delta(\pi_1)"'] & \Delta(\mathbf{Z}_0) \end{tikzcd}$$

in $\widehat{\mathbb{C}_1}/\Delta$. For each lax limit, the given functor is cocontinuous, so the projection functor to the product creates colimits. We thus get a diagram

$$\begin{tikzcd} \mathbf{C}^H_{lax} \arrow[rr] \arrow[rrr, bend left, "\mathscr{V}"] \arrow[d] & & \mathbf{C}^H_m \arrow[rr] \arrow[d] & & \mathbf{C^2} \arrow[d] \\ \mathbf{C}^H_r \times \mathbf{C}^H_l \arrow[rr, "pr_r\times\mathrm{id}"'] & & \widehat{\mathbb{C}_1}/\Delta(H_0) \times \widehat{\mathbb{C}_1}/\Delta(\mathbf{Z}_0) \arrow[rr, hook] & & (\widehat{\mathbb{C}_1}/\Delta)^2 \end{tikzcd}$$

in which all vertical functors create colimits, and both bottom functors are cocontinuous by the previous points. Thus, the top functor $\mathscr{V}$ is cocontinuous as desired. $\square$

Our next step, in order to apply Lemma 1.8, is to prove that each object of the initial chain for $H_{\mathbf{Z}}$ is a simulation in $\mathbf{C}^H_{lax}$. This will follow from the next result.

Lemma 9.52. *If Σ_1 preserves functional bisimulations and $S \in \mathbf{C}^H_{lax}$ is a simulation, then so is $\Sigma^H_{1,lax}(S)$.*

Proof. The pasting (9.8) is isomorphic to the following.

$$
\begin{tikzcd}
 & \mathbf{Z}_0 \arrow[rr, "H_0"] & & \mathbf{Z}_0 \arrow[d, equal] \\
\check{\Sigma}_1(X)_1 \arrow[r, equal] \arrow[ur, phantom, "\Uparrow"] \arrow[dd, "\Sigma_1^F(X)"'] & \Sigma_0(\mathbf{Z}_0) \arrow[r, "\Sigma_0(H_0)"] \arrow[d, "\Sigma_1^F(S)"] & \Sigma_0(\mathbf{Z}_0) \arrow[r] \arrow[d, "\Sigma_1^F(\mathbf{Z})"'] \arrow[dr, phantom, "\Rightarrow"] & \mathbf{Z}_0 \arrow[r, "\sim_0^{\otimes *}"] \arrow[d, "\mathbf{Z}_1"] & \mathbf{Z}_0 \arrow[d, "\mathbf{Z}_1"] \\
 & \mathbf{Z}_0 \arrow[r, "H_0"'] \arrow[rr, bend right, "H_0"'] & \mathbf{Z}_0 \arrow[r, equal] \arrow[d, phantom, "\Downarrow"] & \mathbf{Z}_0 \arrow[r, "\sim_0^{\otimes *}"'] & \mathbf{Z}_0 \arrow[u, phantom, "\sim_1^{\otimes *}"]
\end{tikzcd}
\tag{9.10}
$$

By Propositions 9.24 and 9.27, it suffices to show that all non-identity cells in (9.10) are simulations if S is. Let us run through them, left-to-right, top-to-bottom:

- The top cell is a simulation because
  - $\mathbf{Z}_0$ is the initial $(I + \Sigma_0)$-algebra, so $I \to \mathbf{Z}_0 \leftarrow \Sigma_0(\mathbf{Z}_0)$ is a coproduct diagram;
  - similarly, by Proposition 9.20, H_0 is the initial algebra for the endofunctor $X \mapsto (I; \sim_0^{\otimes *} + \Sigma_0(X); \sim_0^{\otimes *})$, so $I; \sim_0^{\otimes *} \to H_0 \leftarrow \Sigma_0(H_0); \sim_0^{\otimes *}$ is a coproduct diagram;
  - so, by extensivity of $\widehat{\mathbb{C}_0}$, both squares in

$$
\begin{tikzcd}
I; \sim_0^{\otimes *} \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner"] & H_0 \arrow[d] & \Sigma_0(H_0); \sim_0^{\otimes *} \arrow[l] \arrow[d] \arrow[dl, phantom, "\lrcorner"] \\
I \arrow[r] & \mathbf{Z}_0 & \Sigma_0(\mathbf{Z}_0) \arrow[l]
\end{tikzcd}
$$

    are pullbacks. The right-hand one is mapped by $\Delta_{\mathbf{s}}$ (which, as a right adjoint, preserves pullbacks) precisely to the top left square of the top cell.
- The first little cell $\Sigma_1^F(S)$ is a simulation by hypothesis.
- The middle little cell is trivially a simulation.
- The third little cell $\sim_1^{\otimes *}$ is a simulation because simulations are closed under transitive closure.
- Finally, the bottom cell is trivially a simulation. □

Finally:

Proof of Lemma 6.17. Let us start by observing that, because the domains and codomains of all s_L are representable, hence finitely presentable, functional bisimulations are closed under filtered colimits in $\mathbf{C}^{\mathbf{2}}$, by Lemma 1.8.

Now, the first projection $\mathbf{Z}_0 \leftarrow H_0$ is trivially a functional bisimulation, since there are no transitions in $\mathbf{Z}_0$, so by Proposition 9.42 the initial object $\emptyset_{\mathbf{C}^H_{lax}}$ of $\mathbf{C}^H_{lax}$ is a simulation in the sense of Definition 9.50. Hence, by induction, using Lemma 9.52, so are all objects of the initial chain of $\Sigma^H_{1,lax}$. Thus, by Definition 9.50, $\mathscr{V}$ maps this initial chain to a chain of functional bisimulations in $\mathbf{C}^{\mathbf{2}}$ and so

$$
\begin{aligned}
\mathscr{V}(H_{lax}) &:= \mathscr{V}(\mathrm{colim}_n (\Sigma^H_{1,lax})^n (\emptyset_{\mathbf{C}^H_{lax}})) \\
&\cong \mathrm{colim}_n \mathscr{V}((\Sigma^H_{1,lax})^n (\emptyset_{\mathbf{C}^H_{lax}})) \quad \text{(by Lemma 9.51)}
\end{aligned}
$$

is a functional bisimulation by closedness of functional bisimulations under filtered colimits.

Finally, this entails that the left-hand leg $\mathscr{W}(H)$ of the Howe closure is a functional bisimulation, because we have

$$\begin{array}{rcll} \mathscr{W}(H) & = & \mathrm{dom}(\mathscr{J}^H(H)) & \text{(by definition of } \mathscr{W}) \\ & = & \mathrm{dom}(\mathscr{U}^H_{lax}(H_{lax})) & \text{(by Corollary 9.48)} \\ & = & \mathscr{V}(H_{lax}) & \text{(by definition of } \mathscr{V}). \end{array}$$

□

9.7. **Symmetry of transitive closure.** In this section, we prove the remaining Lemmas 6.19 and 6.21. Let us first recall the former:

Lemma 6.19. *The relational transitive closure $H_0{}^{\overline{+}}$ of the Howe closure H_0 on states is symmetric.*

By Lemma 9.10, Lemma 6.19 will follow if we construct a span morphism $H_0 \to H_0^{\overline{+}\dagger}$. As H_0 is an initial algebra for $I + \Sigma_0^H$ (Proposition 9.17), it suffices to prove the following lemma.

Lemma 9.54. *The span $H_0^{\overline{+}\dagger}$ has an algebra structure for $(I + \Sigma_0^H)$.*

This relies on the following lemmas, used in particular with $F = \Sigma_0$. The first one is well known:

Lemma 9.55. *Given an endofunctor F on some category $\mathscr{C}$, the forgetful functor F-**alg** $\to \mathscr{C}$ creates limits, and all colimits that F preserves.*

Lemma 9.56. *Given an endofunctor F on a regular category $\mathscr{C}$, if F preserves reflexive coequalisers, then the forgetful functor from the category of F-algebras creates image factorisations.*

Proof. Suppose given an algebra morphism $A \xrightarrow{f} B$. The image factorisation is obtained as the (reflexive) coequaliser of the kernel pair $A \times_f A \rightrightarrows A$. The diagram of this reflexive coequaliser lifts to F-**alg**, hence so does the coequaliser, by the previous lemma. □

Proof of Lemma 9.54. We need to find algebra structures on $H_0^{\overline{+}\dagger}$ for I, Σ_0, and $-;\sim_0^{\otimes}$. For I, we have the morphism $I \to \mathbf{Z}_0 \to H_0^{\dagger} \to H_0^{\overline{+}\dagger}$.

For Σ_0, note that by Lemmas 9.9 and Corollary 9.8, $H_0^{\overline{+}\dagger}$ is the colimit of the chain

$$\mathbf{Z}_0 \to \overline{H_0^{\dagger}} \cong \overline{H_0^{\dagger};\mathbf{Z}_0} \to \overline{H_0^{\dagger};H_0^{\dagger}} \cong \overline{H_0^{\dagger};H_0^{\dagger};\mathbf{Z}_0} \to \overline{H_0^{\dagger};H_0^{\dagger};H_0^{\dagger}} \to \dots$$

As it is filtered and thus sifted, and Σ_0 preserves sifted colimits by hypothesis, by Lemma 9.55, it is enough to show that each $\overline{H_0^{\dagger};\dots;H_0^{\dagger}}$ has a structure of Σ_0-algebra (morphisms in the above chain are then automatically algebra morphisms because the involved spans are relations). But, Σ_0 also preserves reflexive coequalisers (which are sifted colimits), so, by Lemma 9.56, the forgetful functor from Σ_0-algebras creates image factorisations. It is thus enough to equip $H_0^{\dagger};\dots;H_0^{\dagger}$ with Σ_0-algebra structure, which is straightforward because H_0 is already an algebra and algebras are stable under pullbacks (Lemma 9.55).

It remains to find a suitable morphism $H_0^{\overline{+}\dagger};\sim_0^{\otimes} \to H_0^{\overline{+}\dagger}$, or equivalently, by applying $-^{\dagger}$, a morphism $\sim_0^{\otimes\dagger};H_0^{\overline{+}} \to H_0^{\overline{+}}$. But by symmetry of $\sim_0^{\otimes}$, we have the composite

$$\sim_0^{\otimes\dagger};H_0^{\overline{+}} \to \sim_0^{\otimes};H_0^{\overline{+}} \to H_0;H_0^{\overline{+}} \to H_0^{\overline{+}}.$$

□

Finally, it remains to prove:

Lemma 6.21. *For any substitution-closed simulation R such that R_0 is symmetric, there exists a substitution-closed bisimulation R' and a span morphism $\mathbf{i}'_R \colon R \to R'$.*

Proof. First, consider the relation $\overline{R}$ induced by R by the image factorisation $R \twoheadrightarrow \overline{R} \hookrightarrow \mathbf{Z}\times\mathbf{Z}$. $\overline{R}$ is still a substitution-closed simulation and $\overline{R}_0$ is symmetric. Now, we define R' as follows:

- $R'_0 = \overline{R}_0$
- R'_1 is the limit of the following diagram:

$$\begin{tikzcd}
\mathbf{Z}_0\mathbf{s} & R'_0\mathbf{s} \arrow[l] \arrow[r] & \mathbf{Z}_0\mathbf{s} \\
\mathbf{Z}_1 \arrow[u] \arrow[d] & R'_1 \arrow[l, dashed] \arrow[r, dashed] \arrow[u, dashed] \arrow[d, dashed] & \mathbf{Z}_1 \arrow[u] \arrow[d] \\
\mathbf{Z}_0\mathbf{t} & R'_0\mathbf{t} \arrow[l] \arrow[r] & \mathbf{Z}_0\mathbf{t}.
\end{tikzcd}$$

More concretely, an element of $R'_1(c_1)$ is a pair of transitions at c_1 with related sources and targets. The morphism $R \to R'$ is obtained by the composite $R \to \overline{R} \to R'$, where the last morphism exploits the definition of R'_1 as a limit. It is straightforward to check that R' is a substitution-closed simulation. Moreover, it is symmetric (even at the level of transitions), so it is a bisimulation. □

10. Conclusion

We have introduced the notion of Howe context, in which we have defined transition monoids, an abstract notion of labelled transition system whose states feature some sort of substitution. For them, we have introduced an abstract variant of applicative bisimilarity called substitution-closed bisimilarity.

Furthermore, we have introduced operational semantics signatures as a device for specifying syntax with variable binding and operational semantics. We have finally shown that if the dynamic part of an operational semantics signature preserves functional bisimulations, then substitution-closed bisimilarity on the generated transition monoid is a congruence.

This all follows the pattern of our previous work [14], but simplifying the framework and relaxing some hypotheses, as explained in the introduction.

We hope these simplifications pave the way for more abstract results in the same vein. To start with, we would like to generalise our approach. Indeed, it fails to directly account for some important applications of Howe's method, notably to PCF [27], algebraic effects [40], and higher-order process calculi [41]. Furthermore, methods similar to Howe's have been used for purposes other than congruence of applicative bisimilarity [47, 39, 28]: it might be useful to design abstract versions of such results using our methods.

References

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

[2] J. Adámek and J. Rosicky. *Locally Presentable and Accessible Categories*. Cambridge University Press, 1994.

[3] J. Adámek, J. Rosický, and E. Vitale. What are sifted colimits? *Theory and Applications of Categories*, 23, 2010.

[4] J. Adámek, J. Rosický, and E. M. Vitale. *Algebraic Theories: A Categorical Introduction to General Algebra*. Cambridge Tracts in Mathematics. Cambridge University Press, 2010.

[5] Benedikt Ahrens, André Hirschowitz, Ambroise Lafont, and Marco Maggesi. Reduction monads and their signatures. *PACMPL*, 4(POPL):31:1–31:29, 2020.

[6] Nathanael Arkor and Marcelo Fiore. Algebraic models of simple type theories: A polynomial approach. In Hermanns et al. [30], pages 88–101.

[7] Roland Carl Backhouse, Marcel Bijsterveld, Rik van Geldrop, and Jaap van der Woude. Categorical fixed point calculus. In David H. Pitt, David E. Rydeheard, and Peter T. Johnstone, editors, *Proceedings of the 6th International Conference on Category Theory and Computer Science*, volume 953 of *LNCS*, pages 159–179. Springer, 1995.

[8] Michael Barr and Charles Wells. Toposes, triples, and theories. *Reprints in Theory and Applications of Categories*, 12, 2005. Originally published by: Springer, 1985.

[9] Falk Bartels. *On Generalised Coinduction and Probabilistic Specification Formats*. PhD thesis, Vrije Universiteit Amsterdam, 2004.

[10] Henning Basold, Damien Pous, and Jurriaan Rot. Monoidal company for accessible functors. In Filippo Bonchi and Barbara König, editors, *Proc. 7th International Conference on Algebra and Coalgebra in Computer Science*, volume 72 of *LIPIcs*, pages 5:1–5:16. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.

[11] Jean Bénabou. Introduction to bicategories. *Lecture Notes in Mathematics*, 47:1–77, 1967.

[12] B. Bloom, S. Istrail, and A. Meyer. Bisimulation can't be traced. *Journal of the ACM*, 42:232–268, 1995.

[13] Martin Bodin, Philippa Gardner, Thomas Jensen, and Alan Schmitt. Skeletal semantics and their interpretations. *PACMPL*, 3(POPL):44:1–44:31, 2019.

[14] Peio Borthelle, Tom Hirschowitz, and Ambroise Lafont. A cellular Howe theorem. In Hermanns et al. [30].

[15] Aurelio Carboni and Peter Johnstone. Connected limits, familial representability and Artin glueing. *Mathematical Structures in Computer Science*, 5(4):441–459, 1995.

[16] Andrea Corradini, Reiko Heckel, and Ugo Montanari. Compositional SOS and beyond: a coalgebraic view of open systems. *Theoretical Computer Science*, 280(1-2):163–192, 2002.

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

[18] Marcelo Fiore. Semantic analysis of normalisation by evaluation for typed lambda calculus. In *Proc. 4th ACM SIGPLAN International Conference on Principles and Practice of Declarative Programming*, pages 26–37. ACM, 2002.

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

[20] Marcelo Fiore and Philip Saville. List objects with algebraic structure. In Dale Miller, editor, *Proc. 2nd International Conference on Formal Structures for Computation and Deduction*, volume 84 of *LIPIcs*, pages 16:1–16:18. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.

[21] Marcelo Fiore and Philip Saville. Coherence and normalisation-by-evaluation for bicategorical cartesian closed structure. In Hermanns et al. [30], pages 425–439.

[22] Marcelo Fiore and Philip Saville. Relative full completeness for bicategorical cartesian closed structure. volume 12077 of *LNCS*, pages 277–298. Springer, 2020.

[23] Marcelo P. Fiore. Second-order and dependently-sorted abstract syntax. In *LICS*, pages 57–68. IEEE, 2008.

[24] Marcelo P. Fiore, Gordon D. Plotkin, and Daniele Turi. Abstract syntax and variable binding. In *Proc. 14th Symposium on Logic in Computer Science* IEEE, 1999.

[25] Richard H. G. Garner and Tom Hirschowitz. Shapely monads and analytic functors. *Journal of Logic and Computation*, 28(1):33–83, 2018.

[26] J.A. Goguen, J.W. Thatcher, and E.G. Wagner. An initial algebra approach to the specification, correctness and implementation of abstract data types. In R. Yeh, editor, *Current Trends in Programming Methodology, IV: Data Structuring*, pages 80–144. Prentice-Hall, 1978.

[27] Andrew D. Gordon. Bisimilarity as a theory of functional programming. *Theoretical Computer Science*, 228(1-2):5–47, 1999.

[28] Jean Goubault-Larrecq, Sławomir Lasota, and David Nowak. Logical relations for monadic types. *Mathematical Structures in Computer Science*, 18(6):1169–1217, 2008.

[29] Makoto Hamana. Free Σ-monoids: A higher-order syntax with metavariables. In Wei-Ngan Chin, editor, *Proc. 2nd Asian Symposium on Programming Languages and Systems*, volume 3302 of *LNCS*, pages 348–363. Springer, 2004.

[30] Holger Hermanns, Lijun Zhang, Naoki Kobayashi, and Dale Miller, editors. *Proc. 35th ACM/IEEE Symposium on Logic in Computer Science* ACM, 2020.

[31] André Hirschowitz, Tom Hirschowitz, and Ambroise Lafont. Modules over monads and operational semantics. In Zena M. Ariola, editor, *Proc. 5th International Conference on Formal Structures for Computation and Deduction*, volume 167 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 12:1–12:23. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[32] Tom Hirschowitz. Cellular Monads from Positive GSOS Specifications. In *26th International Workshop on Expressiveness in Concurrency and 16th Workshop on Structural Operational Semantics*, volume 300, pages 1–18, Amsterdam, Netherlands, 2019.

[33] Tom Hirschowitz. Familial monads and structural operational semantics. *PACMPL*, 3(POPL):21:1–21:28, 2019.

[34] Mark Hovey. *Model Categories*, volume 63 of *Mathematical Surveys and Monographs, Volume 63, AMS (1999)*. American Mathematical Society, 1999.

[35] Douglas J. Howe. Proving congruence of bisimulation in functional programming languages. *Information and Computation*, 124(2):103–112, 1996.

[36] André Joyal, Mogens Nielsen, and Glynn Winskel. Bisimulation and open maps. In *Proc. 8th Symposium on Logic in Computer Science*, pages 418–427. IEEE, 1993.

[37] G. Maxwell Kelly. Elementary observations on 2-categorical limits. *Bulletin of the Australian Mathematical Society*, 39:301–317, 1989.

[38] Ambroise Lafont. Initial semantics for substitution skew-monoids. https://github.com/amblafont/Skew-Monoidalcategories, 2020. Coq code.

[39] Ugo Dal Lago and Francesco Gavazzo. Effectful normal form bisimulation. In Luís Caires, editor, *Proc. 28th European Symposium on Programming*, volume 11423 of *LNCS*, pages 263–292. Springer, 2019.

[40] Ugo Dal Lago, Francesco Gavazzo, and Paul Blain Levy. Effectful applicative bisimilarity: Monads, relators, and Howe's method. In *Proc. 32nd Symposium on Logic in Computer Science*, pages 1–12. IEEE, 2017.

[41] Sergueï Lenglet and Alan Schmitt. Howe's method for contextual semantics. In Luca Aceto and David de Frutos-Escrig, editors, *Proc. 26th International Conference on Concurrency Theory*, volume 42 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 212–225. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015.

[42] Saunders Mac Lane. *Categories for the Working Mathematician*. Number 5 in Graduate Texts in Mathematics. Springer, 2nd edition, 1998.

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

[44] Robin Milner, Joachim Parrow, and David Walker. A calculus of mobile processes, I/II. *Information and Computation*, 100(1):1–77, 1992.

[45] Jeffrey C. Morton. Double bicategories and double cospans. *Journal of Homotopy and Related Structures*, 4(1):389–428, 2009.

[46] C.-H. L. Ong. The concurrent lambda calculus I: A general precongruence theorem for applicative bisimulation. In *Proceedings on Seminars on Parallel Programming Systems*, pages 139–164. Department of Information Systems and Computer Science, National University of Singapore, 1992.

[47] Andrew M. Pitts. *Howe's method for higher-order languages*, chapter 5. Number 52 in Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 2011.

[48] Gordon Plotkin. An illative theory of relations. In R. Cooper et al., editors, *Situation Theory and its Applications*, number 22 in CSLI Lecture Notes, page 133–146. Stanford University, 1990.

[49] Jan Reiterman. A left adjoint construction related to free triples. *Journal of Pure and Applied Algebra*, 10:57–71, 1977.

[50] Davide Sangiorgi. The lazy lambda calculus in a concurrency scenario. *Information and Computation*, 111:120–153, 1994.

[51] Davide Sangiorgi and David Walker. *The π-calculus – A Theory of Mobile Processes*. Cambridge University Press, 2001.

[52] Sam Staton. General structural operational semantics through categorical logic. In *Proc. 23rd Symposium on Logic in Computer Science*, pages 166–177, 2008.

[53] Kornel Szlachányi. Skew-monoidal categories and bialgebroids. *Advances in Mathematics*, 231:1694–1730, 2012.

[54] Daniele Turi and Gordon D. Plotkin. Towards a mathematical operational semantics. In *Proc. 12th Symposium on Logic in Computer Science*, pages 280–291, 1997.

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

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

[57] Mark Weber. Yoneda structures from 2-toposes. *Applied Categorical Structures*, 15:259–323, 2007.