Axioms for Asynchronous Processes

Eleftherios Matsikoudis

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2011-4
http://www.eecs.berkeley.edu/Pubs/TechRpts/2011/EECS-2011-4.html

January 20, 2011

Axioms for Asynchronous Processes

by

Eleftherios Matsikoudis

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Electrical Engineering and Computer Sciences

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Edward A. Lee, Chair
Professor Christos H. Papadimitriou
Professor Leo A. Harrington

Fall 2010

Axioms for Asynchronous Processes

Abstract

Axioms for Asynchronous Processes

by

Eleftherios Matsikoudis

Doctor of Philosophy in Electrical Engineering and Computer Sciences

University of California, Berkeley

Professor Edward A. Lee, Chair

From classical computability theory to modern programming language design, the mathematical concept of function has dominated our perception of sequential computation. But as soon as we venture into the realm of concurrent interaction, it is well understood that this concept has to be abandoned. What are we to replace it with?

This question is considered too general to admit a definitive answer. If we want such an answer, we must be willing to narrow our scope, and impose some constraint on the form of concurrent interaction that we choose to consider. Here, we derive such a constraint solely from the intuitive notion of asynchrony. And under this constraint, we propose a mathematical concept of *sequential asynchronous process*, which we define axiomatically, and put forward as the sought replacement to the classical function.

Our theory is an interleaving theory. And traditionally, interleaving theories have failed to integrate a satisfactory treatment of what is known as the finite delay property, according to which, if a process can make progress, then it will eventually do so, but after an arbitrary amount of time. This failure is generally attributed to the so-called expansion law of such theories, which reduces parallel execution to indeterminate serialization. But in truth, the problem is deeply rooted in the concept of labelled transition system, which is the pervasive mathematical object underlying such theories.

To solve this problem, we introduce a new type of system, in which, instead of labelled transitions, we have, essentially, sequences of labelled transitions. We call systems of this type *labelled execution systems*. We use a coalgebraic representation to obtain a proper concept of bisimilarity among such systems, and study the conditions under which that concept agrees with the intuitive notion of branching equivalence that one has for them. Finally, we examine the difference in expressive power and branching complexity between labelled execution systems and labelled transition systems.

The intended interpretation of our concept of asynchronous process is a state of what we

may think of as a very large labelled execution system, and the role of our axioms is to fix the shape of that system.

There are two groups of axioms. The first group is used to specify the form of the executions of the system, and the way in which they branch off one another, in a manner consistent with our intuitive notion of behaviour of an asynchronous process. The second group is deduced from a single extremal axiom asserting the finality of the system in a covariety of coalgebras relating to the first group of axioms, and is used to guarantee that every behaviour is accounted for exactly once.

Contents

1 Introduction 1

2 Transition Systems 3
- 2.1 Basic definitions . . . 3
- 2.2 From systems to coalgebras . . . 6
- 2.3 More on homomorphisms and bisimulations . . . 17
- 2.4 Behaviour modelling and final coalgebras . . . 28
- 2.5 Existence of final coalgebras . . . 45

3 Execution Systems 50
- 3.1 From transitions to executions . . . 50
- 3.2 From systems to coalgebras and back . . . 52
- 3.3 Abrahamson systems . . . 58
- 3.4 Generable systems . . . 61
- 3.5 Behaviour modelling and covarieties . . . 86
- 3.6 Execution systems in the literature . . . 89

4 Sequential Asynchronous Processes 93
- 4.1 Our intuitive notion of asynchronous process . . . 93
- 4.2 Asynchronous process types . . . 94
- 4.3 Systems and coalgebras . . . 95
- 4.4 Axiomatization . . . 104
- 4.5 The Postulate of Delegated Output . . . 105

Bibliography 108

Chapter 1

Introduction

From classical computability theory to modern programming language design, the mathematical concept of function has dominated our perception of sequential computation. But as soon as we venture into the realm of concurrent interaction, it is well understood that this concept has to be abandoned. What are we to replace it with?

This question is considered too general to admit a definitive answer. If we want such an answer, we must be willing to narrow our scope, and impose some constraint on the form of concurrent interaction that we choose to consider. Here, we derive such a constraint solely from the intuitive notion of asynchrony. And under this constraint, we propose a mathematical concept of *sequential asynchronous process*, which we define axiomatically, and put forward as the sought replacement to the classical function.

Our theory is an interleaving theory. And traditionally, interleaving theories have failed to integrate a satisfactory treatment of what is known as the finite delay property, according to which, if a process can make progress, then it will eventually do so, but after an arbitrary amount of time. This failure is generally attributed to the so-called expansion law of such theories, which reduces parallel execution to indeterminate serialization. But in truth, the problem is deeply rooted in the concept of labelled transition system, which is the pervasive mathematical object underlying such theories.

To solve this problem, we introduce a new type of system, in which, instead of labelled transitions, we have, essentially, sequences of labelled transitions. We call systems of this type *labelled execution systems*. We use a coalgebraic representation to obtain a proper concept of bisimilarity among such systems, and study the conditions under which that concept agrees with the intuitive notion of branching equivalence that one has for them. Finally, we examine the difference in expressive power and branching complexity between labelled execution systems and labelled transition systems.

The intended interpretation of our concept of asynchronous process is a state of what we may think of as a very large labelled execution system, and the role of our axioms is to fix the shape of that system.

There are two groups of axioms. The first group is used to specify the form of the executions of the system, and the way in which they branch off one another, in a manner consistent with our intuitive notion of behaviour of an asynchronous process. The second group is deduced from a single extremal axiom asserting the finality of the system in a covariety of coalgebras relating to the first group of axioms, and is used to guarantee that every behaviour is accounted for exactly once.

Chapter 2

Transition Systems

2.1 Basic definitions

Definition 2.1.1. A *transition system* is an ordered pair $\langle S, T\rangle$ such that the following are true:

(a) S is a set;

(b) T is a binary relation[1] on S.

Assume a transition system $\langle S, T\rangle$.

We write $s \longrightarrow_T s'$ if and only if $s\ T\ s'$.

We call any $s \in S$ a *state* of $\langle S, T\rangle$, and any $\langle s, s'\rangle \in \mathsf{graph}\, T$ a *transition* of $\langle S, T\rangle$.

The concept of transition system is ubiquitous in computer science: *Turing machines*, *rewriting systems*, *Kripke structures* are only but a few examples. But as versatile as it is, one cannot use it to model anything more about an individual transition than a change of state in the system. And we will need more than that.

Assume a non-empty set L of *labels*.

Definition 2.1.2. An *L-labelled transition system* is an ordered pair $\langle S, T\rangle$ such that the following are true:

[1] A *binary relation* R is an ordered triple $\langle D, C, G\rangle$ such that D is a set, C is a set, and $G \subseteq D \times C$. We write $\mathsf{dom}\, R$ for D, $\mathsf{cod}\, R$ for C, and $\mathsf{graph}\, R$ for G. We call $\mathsf{dom}\, R$ the *domain* of R, $\mathsf{cod}\, R$ the *codomain* of R, and $\mathsf{graph}\, R$ the *graph* of R.

(a) S is a set;

(b) T is a binary relation between S and $L \times S$.

Assume an L-labelled transition system $\langle S, T \rangle$.

We write $s \xrightarrow{l}_T s'$ if and only if $s \; T \; \langle l, s' \rangle$.

We call any $s \in S$ a *state* of $\langle S, T \rangle$, and any $\langle s, \langle l, s' \rangle \rangle \in \mathsf{graph}\, T$ a *transition* of $\langle S, T \rangle$.

Labelled transition systems have been around at least since Moore's work on finite automata in [43], where they appeared in tabular as well as pictorial form. In their present form, they were first introduced by Keller in [32], where they were called *named transition systems*. And although Keller used them to model parallel computation, it was apparently Milner who first saw labels as shared vehicles of interaction, and labelled transition systems as models of communicating behaviour, paving the way for [37] and the advent of process algebra.

Assume L-labelled transition systems $\langle S_1, T_1 \rangle$ and $\langle S_2, T_2 \rangle$.

Definition 2.1.3. A *bisimulation between* $\langle S_1, T_1 \rangle$ *and* $\langle S_2, T_2 \rangle$ is a binary relation $B : S_1 \leftrightarrow S_2$ such that for any s_1 and s_2 such that $s_1 \; B \; s_2$, the following are true:

(a) if $s_1 \xrightarrow{l}_{T_1} s_1'$, then there is s_2' such that $s_2 \xrightarrow{l}_{T_2} s_2'$ and $s_1' \; B \; s_2'$;

(b) if $s_2 \xrightarrow{l}_{T_2} s_2'$, then there is s_1' such that $s_1 \xrightarrow{l}_{T_1} s_1'$ and $s_1' \; B \; s_2'$.

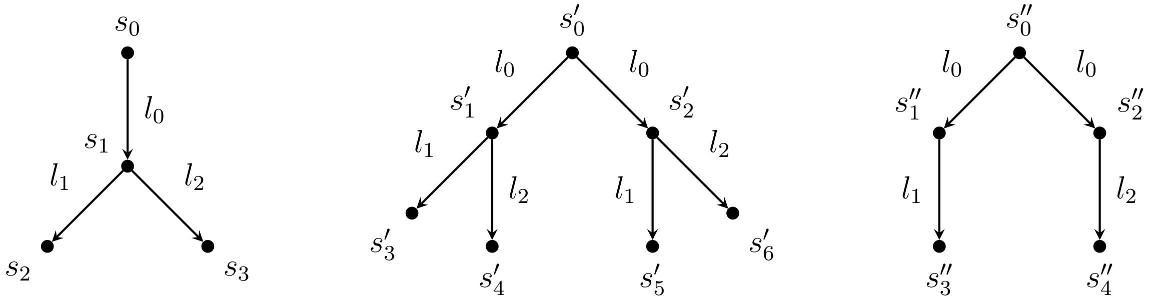
We say that B *is a bisimulation on* $\langle S, T \rangle$ if and only if B is a bisimulation between $\langle S, T \rangle$ and $\langle S, T \rangle$.

We say that B *is a bisimulation equivalence on* $\langle S, T \rangle$ if and only if B is a bisimulation on $\langle S, T \rangle$, and an equivalence relation on S.

We say that s_1 *and* s_2 *are bisimilar among* $\langle S_1, T_1 \rangle$ *and* $\langle S_2, T_2 \rangle$ if and only if there is a bisimulation B between $\langle S_1, T_1 \rangle$ and $\langle S_2, T_2 \rangle$ such that $s_1 \; B \; s_2$.

We say that s_1 *and* s_2 *are bisimilar in* $\langle S, T \rangle$ if and only if s_1 and s_2 are bisimilar among $\langle S, T \rangle$ and $\langle S, T \rangle$.

For example, consider the following three diagrams, which are of course pictures of labelled transition systems:



s_0 and s'_0 are bisimilar among the first two diagrams; neither of them is bisimilar to s''_0.

The idea of bisimilarity is that for any path branching out of either one of the two states, there is a path branching out of the other one, that carries the same labels in the same order, and goes through states that are again related to the corresponding states of the first path in the same way. This last piece of recursion is what separates bisimilarity from trace equivalence, making the former sensitive to the branching potential of each state.

Notice that a transition system can be thought of as a labelled transition system, every transition of which is decorated with a single fixed label. This induces a concept of bisimulation between transition systems: simply erase every instance of l in the above definition.[2]

The concept of bisimulation is due to David Park (see [46]), and is without doubt the most significant contribution of the theory of concurrency to the broader arena of computer science and mathematics at large.[3] After learning about Milner's work in [39], where bisimulation was worked out in the context of a calculus for the first time, and prompted by the perception of an analogy between the mathematical notion of a set and that of a process with just one kind of action, Peter Aczel used transition systems to model a theory of sets that need not be well founded, and the concept of bisimulation to strengthen, in a sensible and pleasing way, the *Axiom of Extensionality* therein (see [2]).[4] But then he went further. He noticed that transition systems could be viewed as coalgebras for a certain endofunctor, and models of his axiom as final objects in a suitable category of such coalgebras. Work on a generalization of this result culminated in [5] to bear a final coalgebra theorem, asserting the existence of final coalgebras for a wide range of common endofunctors, and a categorical definition of bisimulation, generalizing the latter as a viable

[2] This will be a common theme throughout this and the next chapter: we will often state definitions and prove statements only for the labelled case; a proper "unlabelling" will immediately carry these over to the non-labelled case.

[3] To be fair, the concept of bisimulation has been independently discovered in the fields of modal logic and set theory as well. See [58] for a comprehensive historical account.

[4] Forti and Honsell had already discovered and used the concept of bisimulation to that end in [22]. But here, we are not so much interested in non-well-founded set theory, but rather in the developments following Aczel's work on it.

coalgebraic dual to the algebraic notion of congruence. This led eventually to a general theory of *universal coalgebra* (see [56]).

We will now begin to recast things in this coalgebraic framework. This will be of great technical as well as conceptual use, especially in the next chapter, when we replace the concept of transition system with another one, more apt to our purpose. Working with coalgebras, we will need to make reference to a few common concepts from category theory, and some familiarity with the latter will at times be useful. But such familiarity is not assumed, and we will make an effort to supplement each category-theoretic definition or argument with a concrete explanation.

2.2 From systems to coalgebras

Consider once more the concept of transition system. We have formalized this as a set of states together with a binary relation on that set. But there is another way: to look at this binary relation as a set-valued function.

Assume sets S_1 and S_2.

Assume a binary relation $R : S_1 \leftrightarrow S_2$.

We write $\mathsf{fun}\, R$ for a function[5] from S_1 to $\mathscr{P}\, S_2$ such that for any $s_1 \in S_1$,

$$(\mathsf{fun}\, R)(s_1) = \{s_2 \mid s_1\ R\ s_2\}.$$

Assume a function $f : S_1 \to \mathscr{P}\, S_2$.[6]

We write $\mathsf{rel}\, f$ for a binary relation between S_1 and S_2 such that for any $s_1 \in S_1$ and any $s_2 \in S_2$,

$$s_1\ (\mathsf{rel}\, f)\ s_2 \iff s_2 \in f(s_1).$$

The following is immediate:

Proposition 2.2.1. *The following are true:*

(a) $\mathsf{rel}(\mathsf{fun}\, R) = R$;

(b) $\mathsf{fun}(\mathsf{rel}\, f) = f$.

[5] A *function* f is an ordered triple $\langle D, C, G\rangle$ such that D is a set, C is a set, $G \subseteq D \times C$, and for every d, c_1 and c_2, if $\langle d, c_1\rangle \in G$ and $\langle d, c_2\rangle \in G$, then $c_1 = c_2$. We write $\mathsf{dom}\, f$ for D, $\mathsf{cod}\, f$ for C, and $\mathsf{graph}\, f$ for G. We call $\mathsf{dom}\, f$ the *domain* of f, $\mathsf{cod}\, f$ the *codomain* of f, and $\mathsf{graph}\, f$ the *graph* of f.

[6] For every set S, we write $\mathscr{P}\, S$ for the power set of S.

By Proposition 2.2.1, a transition system $\langle S, T\rangle$ can be represented as a function from S to $\mathscr{P}\,S$, namely as $\mathsf{fun}\,T$, and conversely, a function $\tau : S \to \mathscr{P}\,S$ can be represented as a transition system, namely as $\langle S, \mathsf{rel}\,\tau\rangle$. Thus, we could alternatively define a transition system to be a function from a set to the power set of that set, or more verbosely, a pair of a set S and a function $\tau : S \to \mathscr{P}\,S$. This would be a coalgebraic definition, one we shall have more to say about after a more general introduction into the concept of coalgebra.

A coalgebra is defined relative to an endofunctor, and an endofunctor is defined relative to a category.

A *category* is just a two-sorted partial algebra of *objects* and *arrows*, with an *identity*, a *domain*, a *codomain*, and a *composition* operation, satisfying a couple of simple equational axioms. These operations assume their suggested meaning in every *concrete* category, where any object is a set, possibly with some structure, and any arrow is a function from one set to another, typically preserving structure.

For reasons that will soon become clear, we shall want to work with some very large categories, the collections of whose objects and arrows are not sets, not even proper classes.

The first such category that we will be working with is the category of all classes[7] and all class functions[8] between them.

We write **Class** for the category whose objects are all the classes, and arrows all the class functions.

It is of course only in a generalized sense that we may think of **Class** as a partial algebra. All the same, we shall not worry too much about this, or any other issue of foundational nature. Such issues can be addressed in one way or another (for example, see [35, chap. I] or [6, chap. 2]), but a thorough treatment here would only obscure the presentation of our ideas. And in any case, we will avoid impredicative constructions and comprehension principles that test the consistency of the theory.

A *functor* is a category homomorphism: a structure-preserving map from one category to another.

An *endofunctor* is a functor from a category to that same category.

For example, an endofunctor on **Class** is a map from **Class** to **Class** that maps classes to classes and class functions to class functions, preserving the identity operation on classes, and the domain, codomain, and composition operation on class functions.

[7] A collection C of sets is a *class* if and only if there is a unary formula φ of set theory such that for every set S, $S \in C$ if and only if $\varphi(S)$ is true.

[8] A *class function* f is an ordered triple $\langle D, C, G\rangle$ such that D is a class, C is a class, $G \subseteq D \times C$, and for every d, c_1 and c_2, if $\langle d, c_1\rangle \in G$ and $\langle d, c_2\rangle \in G$, then $c_1 = c_2$. We write $\mathsf{dom}\,f$ for D, $\mathsf{cod}\,f$ for C, and $\mathsf{graph}\,f$ for G. We call $\mathsf{dom}\,f$ the *domain* of f, $\mathsf{cod}\,f$ the *codomain* of f, and $\mathsf{graph}\,f$ the *graph* of f.

One such endofunctor is the functor Pow, which assigns to every class C the class

$$\mathsf{Pow}\,C = \{S \mid S \text{ is a subset of } C\},^{9}$$

and to every class function $f : C_1 \to C_2$ a class function

$$\mathsf{Pow}\,f : \mathsf{Pow}\,C_1 \to \mathsf{Pow}\,C_2$$

such that for every $S \in \mathsf{Pow}\,C_1$,

$$(\mathsf{Pow}\,f)(S) = \{f(s) \mid s \in S\}.$$

Notice that if the class C is actually a set, then

$$\mathsf{Pow}\,C = \mathscr{P}\,C.$$

Assume an endofunctor F on **Class**.

Definition 2.2.2. An F-*coalgebra* is an ordered pair $\langle C, \gamma\rangle$ such that the following are true:

(a) C is a class;

(b) γ is a class function from C to $F(C)$.

Assume an F-coalgebra $\langle C, \gamma\rangle$.

We call C the *carrier* of $\langle C, \gamma\rangle$, and γ the *cooperation* of $\langle C, \gamma\rangle$.

We say that $\langle C, \gamma\rangle$ *is small* if and only if C is a set.

We say that $\langle C, \gamma\rangle$ *is large* if and only if C is a proper class.

By this definition, a Pow-coalgebra is just an ordered pair $\langle C, \tau\rangle$ of a class C and a class function $\tau : C \to \mathsf{Pow}\,C$, which is precisely what we have recognized as another way to formalize the concept of transition system, with the caveat that C be a set.

Assume a Pow-coalgebra $\langle C, \tau\rangle$.

We call $\langle C, \tau\rangle$ a *transition coalgebra.*

We write $c \longrightarrow_\tau c'$ if and only if $c' \in \tau(c)$.

Assume a transition system $\langle S, T\rangle$.

The following is immediate:

[9] We use the verbal expression "S is a subset of C" instead of the mathematical expression "$S \subseteq C$" to emphasize the constraint that S be a set.

Proposition 2.2.3. *The following are true:*

(a) $s \longrightarrow_T s'$ *if and only if* $s \longrightarrow_{\mathsf{fun}\,T} s'$*;*

(b) if $\langle C, \tau \rangle$ *is small, then* $c \longrightarrow_\tau c'$ *if and only if* $c \longrightarrow_{\mathsf{rel}\,\tau} c'$*.*

This is all very nice, but why bother with this alternative formalization in the first place?

To begin with, we need to understand what the informal meaning of the concept of F-coalgebra is. For this, it will help to consider the dual concept of F-algebra first, where it is not hard to establish a connection with the more common concept of Σ-algebra, the principal object of study in the theory of universal algebra.

Definition 2.2.4. An *F-algebra* is an ordered pair $\langle C, \alpha \rangle$ such that the following are true:

(a) C is a class;

(b) α is a class function from $F(C)$ to C.

Assume an F-algebra $\langle C, \alpha \rangle$.

We call C the *carrier* of $\langle C, \alpha \rangle$, and α the *operation* of $\langle C, \alpha \rangle$.

We say that $\langle C, \alpha \rangle$ *is small* if and only if C is a set.

We say that $\langle C, \alpha \rangle$ *is large* if and only if C is a proper class.

Notice that the only difference between the definition of an F-algebra and that of an F-coalgebra is in the direction of the class function, which is reversed. Hence the duality.

Now, the way to think of the concept of F-algebra is as a generalization of the concept of Σ-algebra.

For example, consider a *single-sorted signature* Σ. We may think of Σ as a set of *operation symbols*, each annotated with a unique natural number, the *arity* of that symbol. A Σ-*algebra* is a semantic interpretation of Σ: a set S, the carrier set of the algebra, together with one n-ary operation f_σ on S for each operation symbol σ of arity n in Σ. These operations may be combined together into a single function

$$\alpha_\Sigma : \{\langle \sigma, \langle s_1, \ldots, s_{\text{arity of } \sigma} \rangle \rangle \mid \sigma \in \Sigma \text{ and } s_1, \ldots, s_{\text{arity of } \sigma} \in S\} \to S$$

such that for any pair $\langle \sigma, \langle s_1, \ldots, s_{\text{arity of } \sigma} \rangle \rangle$ in its domain,

$$\alpha_\Sigma(\langle \sigma, \langle s_1, \ldots, s_{\text{arity of } \sigma} \rangle \rangle) = f_\sigma(s_1, \ldots, s_{\text{arity of } \sigma}).$$

And if we view the domain of α_Σ as the image of an endofunctor F on **Class** that assigns to every class C the class

$$F(C) = \{\langle \sigma, \langle c_1, \ldots, c_{\text{arity of } \sigma} \rangle \rangle \mid \sigma \in \Sigma \text{ and } c_1, \ldots, c_{\text{arity of } \sigma} \in C\},$$

and to every class function $f : C_1 \to C_2$ a class function

$$F(f) : F(C_1) \to F(C_2)$$

such that for any $\langle \sigma, \langle c_1, \ldots, c_{\text{arity of } \sigma} \rangle \rangle \in F(C_1)$,

$$F(f)(\langle \sigma, \langle c_1, \ldots, c_{\text{arity of } \sigma} \rangle \rangle) = \langle \sigma, \langle f(c_1), \ldots, f(c_{\text{arity of } \sigma}) \rangle \rangle,$$

then a Σ-algebra is just a small F-algebra.

Algebra is about composition of things. In a Σ-algebra, this composition takes a very specific form: things are composed according to certain rules, the operations of the Σ-algebra, and each rule determines a unique thing for every ordered n-tuple of things, where n is arbitrary but fixed for that rule. There is no fundamental reason, though, why composition must be restricted to this form. One might, for example, think of a rule that determines a new composite thing for every possible set of things. And if C is the class of all things, then a rule of this kind can be represented simply as a class function from $\mathsf{Pow}\, C$ to C, which is precisely what a Pow-algebra is. In general, an F-algebra $\langle C, \alpha \rangle$ will represent one or several rules for composing particular combinations of things, as specified and encoded by F, into other things, as determined by α.

We may now appeal to the duality between the concepts of F-algebra and F-coalgebra to attach a plausible informal sense to the latter: if F-algebras are generalized rules of composition, then F-coalgebras are generalized rules of decomposition.

Under this interpretation, a transition coalgebra will represent a rule for decomposing each thing in the carrier of the coalgebra to a set of other things in it. This, of course, imposes a particular view on a transition system: each state of the system is a composite thing, decomposed by the cooperation of the corresponding coalgebra to the set of all immediate successors of it in the system. And this may not be the most intuitive view to impose on a transition system. But never mind. The merit of the coalgebraic approach is not in the interpretation per se, but in the machinery available for relating the corresponding decomposition structure of one system to that of another.

Assume F-coalgebras $\langle C_1, \gamma_1 \rangle$ and $\langle C_2, \gamma_2 \rangle$.

Definition 2.2.5. A *homomorphism from* $\langle C_1, \gamma_1 \rangle$ *to* $\langle C_2, \gamma_2 \rangle$ is a class function $h : C_1 \to C_2$ such that

$$F(h) \circ \gamma_1 = \gamma_2 \circ h.$$

Thus, h is a homomorphism from $\langle C_1, \gamma_1\rangle$ to $\langle C_2, \gamma_2\rangle$ just as long as it is a class function from C_1 to C_2, and the following diagram commutes:

$$\begin{tikzcd}
C_1 \arrow[r, "h"] \arrow[d, "\gamma_1"'] & C_2 \arrow[d, "\gamma_2"] \\
F(C_1) \arrow[r, "F(h)"'] & F(C_2)
\end{tikzcd}$$

We say that *h is a monomorphism from $\langle C_1, \gamma_1\rangle$ to $\langle C_2, \gamma_2\rangle$* if and only if h is an injective homomorphism from $\langle C_1, \gamma_1\rangle$ to $\langle C_2, \gamma_2\rangle$.

We say that *h is an epimorphism from $\langle C_1, \gamma_1\rangle$ to $\langle C_2, \gamma_2\rangle$* if and only if h is a surjective homomorphism from $\langle C_1, \gamma_1\rangle$ to $\langle C_2, \gamma_2\rangle$.

We say that *$\langle C_1, \gamma_1\rangle$ is a homomorphic image of $\langle C_2, \gamma_2\rangle$* if and only if there is an epimorphism from $\langle C_2, \gamma_2\rangle$ to $\langle C_1, \gamma_1\rangle$.

We say that *h is an isomorphism between $\langle C_1, \gamma_1\rangle$ and $\langle C_2, \gamma_2\rangle$* if and only if h is a bijective homomorphism from $\langle C_1, \gamma_1\rangle$ to $\langle C_2, \gamma_2\rangle$.

Proposition 2.2.6. *If h is an isomorphism between $\langle C_1, \gamma_1\rangle$ and $\langle C_2, \gamma_2\rangle$, then h^{-1} is an isomorphism between $\langle C_2, \gamma_2\rangle$ and $\langle C_1, \gamma_1\rangle$.*

Proof. See [56, prop. 2.3]. □

We say that *$\langle C_1, \gamma_1\rangle$ and $\langle C_2, \gamma_2\rangle$ are isomorphic* if and only if there is an isomorphism between $\langle C_1, \gamma_1\rangle$ to $\langle C_2, \gamma_2\rangle$.

We write $\langle C_1, \gamma_1\rangle \cong \langle C_2, \gamma_2\rangle$ if and only if $\langle C_1, \gamma_1\rangle$ and $\langle C_2, \gamma_2\rangle$ are isomorphic.

Assume an F-coalgebra $\langle C, \gamma\rangle$.

We say that *h is an endomorphism on $\langle C, \gamma\rangle$* if and only if h is a homomorphism from $\langle C, \gamma\rangle$ to $\langle C, \gamma\rangle$.

We say that *h is an automorphism on $\langle C, \gamma\rangle$* if and only if h is an isomorphism between $\langle C, \gamma\rangle$ to $\langle C, \gamma\rangle$.

The concept of homomorphism from one F-coalgebra to another is the coalgebraic counterpart of the concept of homomorphism from one F-algebra to another, which is a generalization, in the same sense as before, of the concept of homomorphism from one Σ-algebra to another. It is a structure-preserving map carrying the decomposition patterns of one coalgebra to those of another. In particular, it establishes a similarity of structure between its domain and range.

In the case of Pow, this similarity can take a very familiar form.

Example 2.2.7. Assume transition coalgebras $\langle C_1, \tau_1 \rangle$ and $\langle C_2, \tau_2 \rangle$.

Suppose that h is a homomorphism from $\langle C_1, \tau_1 \rangle$ to $\langle C_2, \tau_2 \rangle$.

Assume $c_1 \in C_1$.

Then

$$(\mathsf{Pow}\, h)(\tau_1(c_1)) = \tau_2(h(c_1)),$$

and hence, by definition of Pow,

$$\{h(c_1') \mid c_1' \in \tau_1(c_1)\} = \tau_2(h(c_1)).$$

By extensionality, this is equivalent to the following being true:

(i) if $c_1 \longrightarrow_{\tau_1} c_1'$, then $h(c_1) \longrightarrow_{\tau_2} h(c_1')$;

(ii) if $h(c_1) \longrightarrow_{\tau_2} c_2'$, then there is c_1' such that $c_1 \longrightarrow_{\tau_1} c_1'$ and $c_2' = h(c_1')$.

(i) and (ii) look very much like the defining clauses of the concept of bisimulation between transition systems. And indeed, if $\langle C_1, \tau_1 \rangle$ and $\langle C_2, \tau_2 \rangle$ are small, then, by Proposition 2.2.3(b), we may simply replace each instance of τ_1 with $\mathsf{rel}\, \tau_1$ and each instance of τ_2 with $\mathsf{rel}\, \tau_2$ in (i) and (ii), to conclude, by generalization, that $\mathsf{graph}\, h$ is a bisimulation between the transition systems $\langle C_1, \mathsf{rel}\, \tau_1 \rangle$ and $\langle C_2, \mathsf{rel}\, \tau_2 \rangle$.

From the perception of this connection between the concepts of homomorphism and bisimulation, it is a small step to a coalgebraic generalization of the latter.

Definition 2.2.8. A *bisimulation between* $\langle C_1, \gamma_1 \rangle$ *and* $\langle C_2, \gamma_2 \rangle$ is a binary class relation[10] $B : C_1 \leftrightarrow C_2$ such that there is an F-coalgebra $\langle \mathsf{graph}\, B, \beta \rangle$ such that $\mathsf{dpr}\, B$ is a homomorphism from $\langle \mathsf{graph}\, B, \beta \rangle$ to $\langle C_1, \gamma_1 \rangle$, and $\mathsf{cpr}\, B$ is a homomorphism from $\langle \mathsf{graph}\, B, \beta \rangle$ to $\langle C_2, \gamma_2 \rangle$.[11]

[10] A *binary class relation* R is an ordered triple $\langle D, C, G \rangle$ such that D is a class, C is a class, and $G \subseteq D \times C$. We write $\mathsf{dom}\, R$ for D, $\mathsf{cod}\, R$ for C, and $\mathsf{graph}\, R$ for G. We call $\mathsf{dom}\, R$ the *domain* of R, $\mathsf{cod}\, R$ the *codomain* of R, and $\mathsf{graph}\, R$ the *graph* of R.

[11] For every binary class relation R, we write $\mathsf{dpr}\, R$ for a function from $\mathsf{graph}\, R$ to $\mathsf{dom}\, R$ such that for any $\langle c_1, c_2 \rangle \in \mathsf{graph}\, R$, $(\mathsf{dpr}\, R)(\langle c_1, c_2 \rangle) = c_1$, and $\mathsf{cpr}\, R$ for a function from $\mathsf{graph}\, R$ to $\mathsf{cod}\, R$ such that for any $\langle c_1, c_2 \rangle \in \mathsf{graph}\, R$, $(\mathsf{cpr}\, R)(\langle c_1, c_2 \rangle) = c_2$. We call $\mathsf{dpr}\, R$ the *domain projection map* of R, and $\mathsf{cpr}\, R$ the *codomain projection map* of R.

Thus, B is a bisimulation between $\langle C_1, \gamma_1\rangle$ and $\langle C_2, \gamma_2\rangle$ just as long as it is a binary class relation between C_1 and C_2, and there is a class function $\beta : \mathsf{graph}\, B \to F(\mathsf{graph}\, B)$ such that the following diagram commutes:

$$\begin{tikzcd}
C_1 \arrow[d, "\gamma_1"'] & \mathsf{graph}\, B \arrow[l, "\mathsf{dpr}\, B"'] \arrow[r, "\mathsf{cpr}\, B"] \arrow[d, dashed, "\beta"] & C_2 \arrow[d, "\gamma_2"] \\
F(C_1) & F(\mathsf{graph}\, B) \arrow[l, "F(\mathsf{dpr}\, B)"] \arrow[r, "F(\mathsf{cpr}\, B)"'] & F(C_2)
\end{tikzcd}$$

We say that B *is a bisimulation on* $\langle C, \gamma\rangle$ if and only if B is a bisimulation between $\langle C, \gamma\rangle$ and $\langle C, \gamma\rangle$.

We say that B *is a bisimulation equivalence on* $\langle C, \gamma\rangle$ if and only if B is a bisimulation on $\langle C, \gamma\rangle$, and an equivalence class relation on C.

So, a binary class relation between two F-coalgebras is a bisimulation just as long as we can impart it with the structure of an F-coalgebra in a way that turns the projections from the graph of the class relation to the carriers of the two F-coalgebras into homomorphisms. In general, however, there might be more than one way to do this.

Example 2.2.9. Let $S = \{0, 1\}$, and τ be a function from S to $\mathsf{Pow}\, S$ defined by the following mapping:

$$0 \mapsto S;$$
$$1 \mapsto S.$$

We want to show that the full binary relation on S is a bisimulation on $\langle S, \tau\rangle$. In order to do so, we must find a function $\beta : S \times S \to \mathsf{Pow}(S \times S)$ such that both $\mathsf{proj}_1(S \times S)$ and $\mathsf{proj}_2(S \times S)$ are homomorphisms from the Pow-coalgebra $\langle S \times S, \beta\rangle$ to $\langle S, \tau\rangle$.[12]

One such function is defined by the following mapping:

$$\langle 0, 0\rangle \mapsto S \times S;$$
$$\langle 0, 1\rangle \mapsto S \times S;$$
$$\langle 1, 0\rangle \mapsto S \times S;$$
$$\langle 1, 1\rangle \mapsto S \times S.$$

[12] For every class C_1 and C_2, we write $\mathsf{proj}_1(C_1 \times C_2)$ for a function from $C_1 \times C_2$ to C_1 such that for any $\langle c_1, c_2\rangle \in C_1 \times C_2$, $(\mathsf{proj}_1(C_1 \times C_2))(\langle c_1, c_2\rangle) = c_1$, and $\mathsf{proj}_2(C_1 \times C_2)$ for a function from $C_1 \times C_2$ to C_2 such that for any $\langle c_1, c_2\rangle \in C_1 \times C_2$, $(\mathsf{proj}_1(C_1 \times C_2))(\langle c_1, c_2\rangle) = c_2$. We call $\mathsf{proj}_1(C_1 \times C_2)$ the *canonical projection map* from $C_1 \times C_2$ to C_1, and $\mathsf{proj}_2(C_1 \times C_2)$ the *canonical projection map* from $C_1 \times C_2$ to C_2.

Another is defined by the following mapping:

$$\begin{aligned}
&\langle 0,0\rangle \mapsto \{\langle 0,1\rangle, \langle 1,0\rangle\};\\
&\langle 0,1\rangle \mapsto \{\langle 0,0\rangle, \langle 1,1\rangle\};\\
&\langle 1,0\rangle \mapsto \{\langle 0,0\rangle, \langle 1,1\rangle\};\\
&\langle 1,1\rangle \mapsto \{\langle 0,1\rangle, \langle 1,0\rangle\}.
\end{aligned}$$

In fact, there is nothing special about the graph of the class relation and its projections either.

Theorem 2.2.10. *B is a bisimulation between $\langle C_1, \gamma_1\rangle$ and $\langle C_2, \gamma_2\rangle$ if and only if there is an F-coalgebra $\langle C, \gamma\rangle$, a homomorphism h_1 from $\langle C, \gamma\rangle$ to $\langle C_1, \gamma_1\rangle$, and a homomorphism h_2 from $\langle C, \gamma\rangle$ to $\langle C_2, \gamma_2\rangle$, such that*

$$B = h_1^{-1} \mathbin{;} h_2.$$[13]

Proof. See [56, lem. 5.3] and [23, thm. 5.11]. □

Assume F-coalgebras $\langle C_1', \gamma'\rangle$ and $\langle C_2', \gamma'\rangle$.

The following is immediate:

Corollary 2.2.11. *If B is a bisimulation between $\langle C_1, \gamma_1\rangle$ and $\langle C_2, \gamma_2\rangle$, h_1 a homomorphism from $\langle C_1, \gamma_1\rangle$ to $\langle C_1', \gamma_1'\rangle$, and h_2 a homomorphism from $\langle C_2, \gamma_2\rangle$ to $\langle C_2', \gamma_2'\rangle$, then*

$$h_1^{-1} \mathbin{;} B \mathbin{;} h_2$$

is a bisimulation between $\langle C_1', \gamma_1'\rangle$ and $\langle C_2', \gamma_2'\rangle$.

The following will come of use:

Theorem 2.2.12. *For every class-indexed family $\{B_i\}_{i\in I}$ of bisimulations between $\langle C_1, \gamma_1\rangle$ and $\langle C_2, \gamma_2\rangle$, there is a bisimulation B between $\langle C_1, \gamma_1\rangle$ and $\langle C_2, \gamma_2\rangle$ such that*

$$\mathsf{graph}\, B = \bigcup_{i\in I} \mathsf{graph}\, B_i.$$

Proof. See [23, thm. 5.6]. □

[13] For every binary class relation R_1 and R_2 such that $\mathsf{cod}\, R_1 = \mathsf{dom}\, R_2$, we write $R_1 \mathbin{;} R_2$ for a binary class relation between $\mathsf{dom}\, R_1$ and $\mathsf{cod}\, R_2$ such that for any $c_1 \in \mathsf{dom}\, R_1$ and any $c_2 \in \mathsf{cod}\, R_2$, $c_1\ (R_1 \mathbin{;} R_2)\ c_2$ if and only if there is c such that $c_1\ R_1\ c$ and $c\ R_2\ c_2$.

This coalgebraic definition of bisimulation was first introduced in [5], and does indeed generalize the concept of bisimulation between labelled transition systems. To see this, we must first go over the coalgebraic representation of such systems.

The functor that we are going to use is the endofunctor $\mathsf{Pow} \circ (L \times \mathsf{Id})$ on **Class**, namely the composite of Pow with the left product endofunctor $L \times \mathsf{Id}$ on **Class**, which assigns to every class C the class

$$\mathsf{Pow}(L \times C) = \{S \mid S \text{ is a subset of } L \times C\},$$

and to every class function $f : C_1 \to C_2$ a class function

$$\mathsf{Pow}(L \times f) : \mathsf{Pow}(L \times C_1) \to \mathsf{Pow}(L \times C_2)$$

such that for every $S \in \mathsf{Pow}(L \times C_1)$,

$$\mathsf{Pow}(L \times f)(S) = \{\langle l, f(s)\rangle \mid \langle l, s\rangle \in S\}.$$

By Proposition 2.2.1, an L-labelled transition system $\langle S, T\rangle$ can be represented as a $(\mathsf{Pow} \circ (L \times \mathsf{Id}))$-coalgebra, namely as $\langle S, \mathsf{fun}\, T\rangle$, and conversely, a $(\mathsf{Pow} \circ (L \times \mathsf{Id}))$-coalgebra $\langle C, \tau\rangle$ can be represented as an L-labelled transition system, namely as $\langle C, \mathsf{rel}\, \tau\rangle$, again with the caveat that C be a set.

Assume a $(\mathsf{Pow} \circ (L \times \mathsf{Id}))$-coalgebra $\langle C, \tau\rangle$.

We call $\langle C, \tau\rangle$ an L-labelled transition coalgebra.

We write $c \xrightarrow{l}_\tau c'$ if and only if $\langle l, c'\rangle \in \tau(c)$.

Assume an L-labelled transition system $\langle S, T\rangle$.

The following is immediate:

Proposition 2.2.13. *The following are true:*

(a) $s \xrightarrow{l}_T s'$ *if and only if* $s \xrightarrow{l}_{\mathsf{fun}\, T} s'$*;*

(b) if $\langle C, \tau\rangle$ *is small, then* $c \xrightarrow{l}_\tau c'$ *if and only if* $c \xrightarrow{l}_{\mathsf{rel}\, \tau} c'$*.*

Assume L-labelled transition coalgebras $\langle C_1, \tau_1\rangle$ and $\langle C_2, \tau_2\rangle$.

Proposition 2.2.14. *B is a bisimulation between $\langle C_1, \tau_1\rangle$ and $\langle C_2, \tau_2\rangle$ if and only if B is a binary class relation between C_1 and C_2, and for any c_1 and c_2 such that $c_1\ B\ c_2$, the following are true:*

(a) *if* $c_1 \xrightarrow{l}_{\tau_1} c'_1$*, then there is* c'_2 *such that* $c_2 \xrightarrow{l}_{\tau_2} c'_2$ *and* $c'_1\ B\ c'_2$*;*

(b) *if* $c_2 \xrightarrow{l}_{\tau_2} c'_2$*, then there is* c'_1 *such that* $c_1 \xrightarrow{l}_{\tau_1} c'_1$ *and* $c'_1\ B\ c'_2$*.*

Proof. Suppose that B is a bisimulation between $\langle C_1, \tau_1\rangle$ and $\langle C_2, \tau_2\rangle$.

Let $\langle \mathsf{graph}\, B, \beta\rangle$ be an L-labelled transition coalgebra such that $\mathsf{dpr}\, B$ is a homomorphism from $\langle \mathsf{graph}\, B, \beta\rangle$ to $\langle C_1, \tau_1\rangle$, and $\mathsf{cpr}\, B$ one from $\langle \mathsf{graph}\, B, \beta\rangle$ to $\langle C_2, \tau_2\rangle$.

Assume c_1 and c_2 such that $c_1\ B\ c_2$.

Then

$$\mathsf{Pow}(L \times \mathsf{dpr}\, B)(\beta(\langle c_1, c_2\rangle)) = \tau_1((\mathsf{dpr}\, B)(\langle c_1, c_2\rangle)),$$

and hence, by definition of $\mathsf{Pow} \circ (L \times \mathsf{Id})$ and $\mathsf{dpr}\, B$,

$$\{\langle l, c'_1\rangle \mid \langle l, \langle c'_1, c'_2\rangle\rangle \in \beta(\langle c_1, c_2\rangle)\} = \tau_1(c_1).$$

By extensionality, this is equivalent to the following being true:

(i) if $\langle c_1, c_2\rangle \xrightarrow{l}_{\beta} \langle c'_1, c'_2\rangle$, then $c_1 \xrightarrow{l}_{\tau_1} c'_1$;

(ii) if $c_1 \xrightarrow{l}_{\tau_1} c'_1$, then there is c'_2 such that $\langle c_1, c_2\rangle \xrightarrow{l}_{\beta} \langle c'_1, c'_2\rangle$.

And by symmetry, the following are true:

(iii) if $\langle c_1, c_2\rangle \xrightarrow{l}_{\beta} \langle c'_1, c'_2\rangle$, then $c_2 \xrightarrow{l}_{\tau_2} c'_2$;

(iv) if $c_2 \xrightarrow{l}_{\tau_2} c'_2$, then there is c'_1 such that $\langle c_1, c_2\rangle \xrightarrow{l}_{\beta} \langle c'_1, c'_2\rangle$.

By (ii) and (iii), (a) is true, and by (iv) and (i), (b) is true.

Thus, by generalization, for any c_1 and c_2 such that $c_1\ B\ c_2$, (a) and (b) are true.

Conversely, suppose that B is a binary class relation between C_1 and C_2, and for any c_1 and c_2 such that $c_1\ B\ c_2$, (a) and (b) are true.

Let β be a class function from $\mathsf{graph}\, B$ to $\mathsf{Pow}(L \times \mathsf{graph}\, B)$ such that for any $\langle c_1, c_2\rangle \in \mathsf{graph}\, B$,

$$\begin{aligned}\beta(\langle c_1, c_2\rangle) = \{\langle l, \langle c'_1, c'_2\rangle\rangle \mid{} & c_1 \xrightarrow{l}_{\tau_1} c'_1, \\ & c_2 \xrightarrow{l}_{\tau_2} c'_2, \\ & \text{and } \langle c'_1, c'_2\rangle \in \mathsf{graph}\, B\}.\end{aligned}$$

Assume $\langle c_1, c_2\rangle \in \mathsf{graph}\, B$.

Then the following is immediately true:

(v) if $\langle c_1, c_2\rangle \xrightarrow{l}_{\beta} \langle c_1', c_2'\rangle$, then $c_1 \xrightarrow{l}_{\tau_1} c_1'$.

Also, by (a) and (b), the following is true:

(vi) if $c_1 \xrightarrow{l}_{\tau_1} c_1'$, then there is c_2' such that $\langle c_1, c_2\rangle \xrightarrow{l}_{\beta} \langle c_1', c_2'\rangle$.

By (v), (vi), and extensionality,

$$\{\langle l, c_1'\rangle \mid \langle l, \langle c_1', c_2'\rangle\rangle \in \beta(\langle c_1, c_2\rangle)\} = \tau_1(c_1),$$

and hence, by definition of $\mathsf{Pow} \circ (L \times \mathsf{Id})$ and $\mathsf{dpr}\, B$,

$$\mathsf{Pow}(L \times \mathsf{dpr}\, B)(\beta(\langle c_1, c_2\rangle)) = \tau_1((\mathsf{dpr}\, B)(\langle c_1, c_2\rangle)).$$

And by symmetry,

$$\mathsf{Pow}(L \times \mathsf{cpr}\, B)(\beta(\langle c_1, c_2\rangle)) = \tau_2((\mathsf{cpr}\, B)(\langle c_1, c_2\rangle)).$$

Thus, by generalization, B is a bisimulation between $\langle C_1, \tau_1\rangle$ and $\langle C_2, \tau_2\rangle$. □

Assume L-labelled transition systems $\langle S_1, T_1\rangle$ and $\langle S_2, T_2\rangle$.

The following is immediate from Proposition 2.2.13(a), 2.2.14, and the definition of bisimulation between labelled transition systems:

Proposition 2.2.15. *B is a bisimulation between $\langle S_1, T_1\rangle$ and $\langle S_2, T_2\rangle$ if and only if B is a bisimulation between the L-labelled transition coalgebras $\langle S_1, \mathsf{fun}\, T_1\rangle$ and $\langle S_2, \mathsf{fun}\, T_2\rangle$.*

Proposition 2.2.14 and 2.2.15 can of course be adapted for transition coalgebras and transition systems: simply replace every instance of $\mathsf{Pow} \circ (L \times \mathsf{Id})$ with Pow, and erase every instance of l.

2.3 More on homomorphisms and bisimulations

We start from making the connection between the concepts of homomorphism and bisimulation precise.

The first thing to note is that every bisimulation equivalence is the equivalence kernel of a homomorphism. In a theory of sets, this is straightforward. But in a theory of classes, some care is needed.

Assume a class C and an equivalence class relation E on C.

We say that *q is a quotient of C with respect to E* if and only if q is a surjective class function such that $\mathsf{dom}\, q = C$ and $\mathsf{ker}\, q = E$.

In [5], existence of quotients is deduced from an assumed global form of the *Axiom of Choice*, and in [3], from a postulated quotient existence principle for classes. In a theory of sets, there is no such need: for every set S and every equivalence relation R on S, one can simply turn to the *quotient set* of S by R, namely the set

$$S\backslash R = \{\{s' \mid s \; R \; s'\} \mid s \in S\},$$

and the *canonical projection map* from S to $S\backslash R$, which assigns to any $s \in S$ the *equivalence class* of s, namely the set

$$[s]_R = \{s' \mid s \; R \; s'\}.$$

But in a theory of classes, these constructs are not, in general, well defined. For example, if R is the full binary class relation on a proper class C, then for every $c \in C$,

$$\{c' \mid c \; R \; c'\} = C,$$

which cannot be a member of any class.

Here, we will assume existence of quotients as a theorem of the underlying theory of classes, and not worry about its precise deduction. We only remark that in the standard *von Neumann-Bernays-Gödel* theory of classes, existence of quotients is readily deduced from the *Axiom of Limitation of Size*.

Assume an F-coalgebra $\langle C, \gamma\rangle$.

Proposition 2.3.1. *For every bisimulation equivalence B on $\langle C, \gamma\rangle$, and every quotient q of C with respect to B, there is exactly one F-coalgebra $\langle \mathsf{cod}\, q, \delta\rangle$ such that q is an epimorphism from $\langle C, \gamma\rangle$ to $\langle \mathsf{cod}\, q, \delta\rangle$.*

Proof. See [5, lem. 5.1 and prop. 6.1]. □

Thus, for every bisimulation equivalence B on $\langle C, \gamma\rangle$, and every quotient q of C with respect to B, there is exactly one class function $\delta : \mathsf{cod}\, q \to F(\mathsf{cod}\, q)$ such that the following diagram commutes:

$$\begin{tikzcd} C \arrow[r, "q"] \arrow[d, "\gamma"'] & \mathsf{cod}\, q \arrow[d, dashed, "\delta"] \\ F(C) \arrow[r, "F(q)"'] & F(\mathsf{cod}\, q) \end{tikzcd}$$

The converse is not true: the equivalence kernel of a homomorphism is not, in general, a bisimulation (see Example 2.3.3). But the graph of a homomorphism, or more accurately, considering our working definition of a binary class relation, the homomorphism itself, is.

Assume F-coalgebras $\langle C_1, \gamma_1 \rangle$ to $\langle C_2, \gamma_2 \rangle$.

Theorem 2.3.2. *h is a homomorphism from $\langle C_1, \gamma_1 \rangle$ to $\langle C_2, \gamma_2 \rangle$ if and only if h is a class function from C_1 to C_2, and a bisimulation between $\langle C_1, \gamma_1 \rangle$ and $\langle C_2, \gamma_2 \rangle$.*

Proof. See [56, thm. 2.5]. □

In other words, homomorphisms are functional bisimulations. This should not be surprising: homomorphisms are supposed to preserve structure, and bisimulations to capture equivalence of it. But do they really?

There is not much to be said about homomorphisms; they are too basic to doubt. Bisimulations, on the other hand, deserve investigation.

We begin with a formal statement of coalgebraic bisimilarity.

We say that c_1 *and* c_2 *are bisimilar among* $\langle C_1, \gamma_1 \rangle$ *and* $\langle C_2, \gamma_2 \rangle$ if and only if there is a bisimulation B between $\langle C_1, \gamma_1 \rangle$ and $\langle C_2, \gamma_2 \rangle$ such that $c_1 \; B \; c_2$.

We say that c_1 *and* c_2 *are bisimilar in* $\langle C, \gamma \rangle$ if and only if c_1 and c_2 are bisimilar among $\langle C, \gamma \rangle$ and $\langle C, \gamma \rangle$.

A disturbing fact about coalgebraic bisimilarity is that, unlike ordinary bisimilarity, it is not, in general, an equivalence concept.

Example 2.3.3. Let F be an endofunctor on **Class** that assigns to every class C the class

$$F(C) = \{\langle c_1, c_2, c_3 \rangle \mid \{c_1, c_2, c_3\} \subseteq C \text{ and } |\{c_1, c_2, c_3\}| < 3\},$$

and to every class function $f : C_1 \to C_2$ a class function

$$F(f) : F(C_1) \to F(C_2)$$

such that for every $\langle c_1, c_2, c_3\rangle \in F(C_1)$,

$$F(f)(\langle c_1, c_2, c_3\rangle) = \langle f(c_1), f(c_2), f(c_3)\rangle.$$

Let $S_1 = \{0, 1\}$, and γ_1 be a function from S_1 to $F(S_1)$ defined by the following mapping:

$$0 \mapsto \langle 0, 0, 1\rangle;$$
$$1 \mapsto \langle 0, 1, 1\rangle.$$

Let $S_2 = \{0\}$, and γ_2 be the unique function from S_2 to $F(S_2)$, namely a function from S_2 to $F(S_2)$ defined by the following mapping:

$$0 \mapsto \langle 0, 0, 0\rangle.$$

Let h be the unique function from S_1 to S_2, namely a function from S_1 to S_2 defined by the following mapping:

$$0 \mapsto 0;$$
$$1 \mapsto 0.$$

h is trivially a homomorphism from $\langle S_1, \gamma_1\rangle$ to $\langle S_2, \gamma_2\rangle$. But whereas $h(0)$ and $h(1)$ are equal, and thus, trivially bisimilar in $\langle S_2, \gamma_2\rangle$, 0 and 1 are not bisimilar in $\langle S_1, \gamma_1\rangle$, lest there be a binary relation B on S_1, and an F-coalgebra $\langle \mathsf{graph}\, B, \beta\rangle$ such that
$\langle 0, 1\rangle \in \mathsf{graph}\, B$ and

$$\beta(\langle 0, 1\rangle) = \langle\langle 0, 0\rangle, \langle 0, 1\rangle, \langle 1, 1\rangle\rangle.$$

Note 2.3.1. We cannot replace ordered triples with multisets of size 3 having at most two members of multiplicity greater than 0 to the same effect in Example 2.3.3.

Example 2.3.1.1. Let F be an endofunctor on **Class** that assigns to every class C the class

$$F(C) = \{m \mid m : C \to \omega,\ |\{c \mid m(c) \neq 0\}| < 3, \text{ and } \sum \mathsf{ran}\, m = 3\},$$

and to every class function $f : C_1 \to C_2$ a class function

$$F(f) : F(C_1) \to F(C_2)$$

such that for every $m \in F(C_1)$ and every $c_2 \in C_2$,

$$F(f)(m)(c_2) = \sum \{m(c_1) \mid f(c_1) = c_2\}.$$

For every class C, $F(C)$ is simply the class of all multisets of size 3 having just at most two members of multiplicity greater than 0 (see [62, chap. A]).

Let $S = \{0, 1\}$, and γ be a function from S to $F(S)$ defined by the following mapping:

$$\begin{aligned} &0 \mapsto [0, 0, 1];\\ &1 \mapsto [0, 1, 1]. \end{aligned}$$

We want to show that the full binary relation on S is a bisimulation on $\langle S, \gamma\rangle$. In order to do so, we must find a function $\beta : S \times S \to F(S \times S)$ such that both $\mathsf{proj}_1(S \times S)$ and $\mathsf{proj}_2(S \times S)$ are homomorphisms from the F-coalgebra $\langle S \times S, \beta\rangle$ to $\langle S, \gamma\rangle$.

One such function is defined by the following mapping:

$$\begin{aligned} &\langle 0, 0\rangle \mapsto [\langle 0, 0\rangle, \langle 0, 0\rangle, \langle 0, 0\rangle];\\ &\langle 0, 1\rangle \mapsto [\langle 0, 1\rangle, \langle 0, 1\rangle, \langle 1, 0\rangle];\\ &\langle 1, 0\rangle \mapsto [\langle 0, 1\rangle, \langle 1, 0\rangle, \langle 1, 0\rangle];\\ &\langle 1, 1\rangle \mapsto [\langle 1, 1\rangle, \langle 1, 1\rangle, \langle 1, 1\rangle]. \end{aligned}$$

Thus, 0 and 1 are bisimilar in $\langle S, \gamma\rangle$.

This should cast serious doubt on the coalgebraic notion of bisimulation: how can one hope to capture all of equivalence of structure using a notion, the induced similarity concept of which is not, in general, transitive?

This discrepancy was not lost on Aczel and Mendler, who, also in [5], generalized the coalgebraic concept of bisimulation further into that of what they called a *precongruence*, or in the case of an equivalence class relation, a *congruence*. This is a technically more complicated concept: to determine whether a binary class relation R on the carrier of an F-coalgebra $\langle C, \gamma\rangle$ is a precongruence on $\langle C, \gamma\rangle$, one has to invoke a quotient of C with respect to the equivalence class relation generated by R, compose its image under F with γ, and test whether R is contained in the equivalence kernel of the composite. It is also an intuitively more warranted concept, exactly formalizing the idea of a class relation that is compatible with the cooperation of a coalgebra. Every bisimulation on an F-coalgebra is a precongruence, but not every precongruence on an F-coalgebra is a bisimulation. In fact, the endofunctor that we used in Example 2.3.3 is one that was devised in [5] for the express purpose of demonstrating this separation between the two concepts.

Note 2.3.2. We can also use Example 2.3.3 to separate the two concepts. For by [31, prop. 4.2], the equivalence kernel of a homomorphism from an F-coalgebra $\langle C_1, \gamma_1\rangle$ to an F-coalgebra $\langle C_2, \gamma_2\rangle$ is a precongruence on $\langle C_1, \gamma_1\rangle$.

Note 2.3.3. In [5], the concept of precongruence was defined only for binary class relations on single F-coalgebras. However, one can use directed sums (see Definition 2.4.10), to extend this definition to binary class relations between pairs of different F-coalgebras.

Definition 2.3.3.1. A *precongruence on* $\langle C, \gamma\rangle$ is a binary class relation P on C such that for every quotient q with respect to the equivalence class relation generated by P,

$$\mathsf{graph}\, P \subseteq \mathsf{graph}\,\mathsf{ker}(F(q) \circ \gamma).$$

We say that P *is a congruence on* $\langle C, \gamma\rangle$ if and only if P is a precongruence on $\langle C, \gamma\rangle$, and an equivalence class relation on C.

The following is easy:

Proposition 2.3.3.2. *P is a precongruence on $\langle C, \gamma\rangle$ if and only if P is a class relation on C, and the equivalence class relation generated by P is a congruence on $\langle C, \gamma\rangle$.*

We say that c_1 *and* c_2 *are congruent in* $\langle C, \gamma\rangle$ if and only if there is a congruence P on $\langle C, \gamma\rangle$ such that $c_1\ P\ c_2$.

Definition 2.3.4. A *precongruence between* $\langle C_1, \gamma_1\rangle$ *and* $\langle C_2, \gamma_2\rangle$ is a binary class relation $P : C_1 \leftrightarrow C_2$ such that

$$(\mathsf{inj}_1(C_1 + C_2))^{-1} \,;\, P \,;\, \mathsf{inj}_2(C_1 + C_2)$$

is a precongruence on $\langle C_1, \gamma_1\rangle + \langle C_2, \gamma_2\rangle$.

We say that c_1 *and* c_2 *are congruent among* $\langle C_1, \gamma_1\rangle$ *and* $\langle C_2, \gamma_2\rangle$ if and only if there is a precongruence P between $\langle C_1, \gamma_1\rangle$ and $\langle C_2, \gamma_2\rangle$ such that $c_1\ P\ c_2$.

With this extended definition, we can appreciate Theorem 2.3.2 better.

First, we extend [5, prop. 6.1].

Proposition 2.3.3.3. *If B is a bisimulation between $\langle C_1, \gamma_1\rangle$ and $\langle C_2, \gamma_2\rangle$, then B is a precongruence between $\langle C_1, \gamma_1\rangle$ and $\langle C_2, \gamma_2\rangle$.*

Proof. Suppose that B is a bisimulation between $\langle C_1, \gamma_1\rangle$ and $\langle C_2, \gamma_2\rangle$.

Let $\langle \mathsf{graph}\, B, \beta\rangle$ be an F-coalgebra such that $\mathsf{dpr}\, B$ is a homomorphism from $\langle \mathsf{graph}\, B, \beta\rangle$ to $\langle C_1, \gamma_1\rangle$, and $\mathsf{cpr}\, B$ one from $\langle \mathsf{graph}\, B, \beta\rangle$ to $\langle C_2, \gamma_2\rangle$.

Let E be the equivalence class relation generated by

$$(\mathsf{inj}_1(C_1 + C_2))^{-1} \,;B\,; \mathsf{inj}_2(C_1 + C_2).$$

Let q be a quotient of $C_1 + C_2$ with respect to E.

Then

$$q \circ \mathsf{inj}_1(C_1 + C_2) \circ \mathsf{dpr}\, B = q \circ \mathsf{inj}_2(C_1 + C_2) \circ \mathsf{cpr}\, B,$$

and thus,

$$F(q) \circ F(\mathsf{inj}_1(C_1 + C_2)) \circ F(\mathsf{dpr}\, B) = F(q) \circ F(\mathsf{inj}_2(C_1 + C_2)) \circ F(\mathsf{cpr}\, B).$$

Let γ be the cooperation of $\langle C_1, \gamma_1\rangle + \langle C_2, \gamma_2\rangle$.

Assume c_1 and c_2 such that

$$c_1 \;((\mathsf{inj}_1(C_1 + C_2))^{-1} \,;B\,; \mathsf{inj}_2(C_1 + C_2))\; c_2.$$

Then there are c_1' and c_2' such that

$$(\mathsf{inj}_1(C_1 + C_2))(c_1') = c_1,$$
$$(\mathsf{inj}_2(C_1 + C_2))(c_2') = c_2,$$

and $c_1'\ B\ c_2'$. Thus,

$$\begin{aligned}
F(q)(\gamma(c_1)) &= F(q)(\gamma((\mathsf{inj}_1(C_1 + C_2))(c_1')))\\
&= F(q)(F(\mathsf{inj}_1(C_1 + C_2))(\gamma_1(c_1')))\\
&= F(q)(F(\mathsf{inj}_1(C_1 + C_2))(\gamma_1((\mathsf{dpr}\, B)(\langle c_1', c_2'\rangle))))\\
&= F(q)(F(\mathsf{inj}_1(C_1 + C_2))(F((\mathsf{cpr}\, B)(\beta(\langle c_1', c_2'\rangle)))))\\
&= F(q)(F(\mathsf{inj}_2(C_1 + C_2))(F((\mathsf{cpr}\, B)(\beta(\langle c_1', c_2'\rangle)))))\\
&= F(q)(F(\mathsf{inj}_2(C_1 + C_2))(\gamma_2((\mathsf{cpr}\, B)(\langle c_1', c_2'\rangle))))\\
&= F(q)(F(\mathsf{inj}_2(C_1 + C_2))(\gamma_2(c_2')))\\
&= F(q)(\gamma((\mathsf{inj}_2(C_1 + C_2))(c_2')))\\
&= F(q)(\gamma(c_2)).
\end{aligned}$$

Thus, by generalization, B is a precongruence between $\langle C_1, \gamma_1\rangle$ and $\langle C_2, \gamma_2\rangle$. □

Theorem 2.3.3.4. *h is a homomorphism from $\langle C_1, \gamma_1\rangle$ to $\langle C_2, \gamma_2\rangle$ if and only if h is a class function from C_1 to C_2, and a precongruence between $\langle C_1, \gamma_1\rangle$ and $\langle C_2, \gamma_2\rangle$.*

Proof. Suppose that h is a homomorphism from $\langle C_1, \gamma_1\rangle$ to $\langle C_2, \gamma_2\rangle$.

Then by Theorem 2.3.2, h is a bisimulation between $\langle C_1, \gamma_1\rangle$ and $\langle C_2, \gamma_2\rangle$, and thus, by Proposition 2.3.3.3, a precongruence between $\langle C_1, \gamma_1\rangle$ and $\langle C_2, \gamma_2\rangle$.

Conversely, suppose that h is a precongruence between $\langle C_1, \gamma_1\rangle$ and $\langle C_2, \gamma_2\rangle$.

Let E be the equivalence class relation generated by

$$(\mathsf{inj}_1(C_1 + C_2))^{-1} \,;h\,; \mathsf{inj}_2(C_1 + C_2).$$

Let q be a quotient of $C_1 + C_2$ with respect to E.

Then the following diagram commutes:

$$\begin{tikzcd}
\mathsf{graph}\,h \arrow[r, "\mathsf{dpr}\,h"] \arrow[d, "\mathsf{cpr}\,h"'] & C_1 \arrow[d, "q \circ \mathsf{inj}_1(C_1+C_2)"] \\
C_2 \arrow[r, "q \circ \mathsf{inj}_2(C_1+C_2)"'] & \mathsf{cod}\,q
\end{tikzcd}$$

Thus, the following diagram also commutes:

$$\begin{tikzcd}
F(\mathsf{graph}\,h) \arrow[r, "F(\mathsf{dpr}\,h)"] \arrow[d, "F(\mathsf{cpr}\,h)"'] & F(C_1) \arrow[d, "F(q) \circ F(\mathsf{inj}_1(C_1+C_2))"] \\
F(C_2) \arrow[r, "F(q) \circ F(\mathsf{inj}_2(C_1+C_2))"'] & F(\mathsf{cod}\,q)
\end{tikzcd}$$

E is the smallest equivalence class relation that contains

$$(\mathsf{inj}_1(C_1 + C_2))^{-1} \,;h\,; \mathsf{inj}_2(C_1 + C_2),$$

or equivalently, the smallest transitive relation that contains the reflexive-symmetric closure of it.

We use induction to prove that for any c_1 and c_2 such that $c_1 \; E \; c_2$, the following are true:

(i) if there are c_1' and c_2' such that

$$(\mathsf{inj}_1(C_1 + C_2))(c_1') = c_1$$

and

$$(\mathsf{inj}_2(C_1 + C_2))(c_2') = c_2,$$

then $h(c_1') = c_2'$;

(ii) if there are c_1' and c_2' such that

$$(\mathsf{inj}_2(C_1 + C_2))(c_1') = c_1$$

and

$$(\mathsf{inj}_1(C_1 + C_2))(c_2') = c_2,$$

then $h(c_2') = c_1'$;

(iii) if there are c_1' and c_2' such that

$$(\mathsf{inj}_1(C_1 + C_2))(c_1') = c_1$$

and

$$(\mathsf{inj}_1(C_1 + C_2))(c_2') = c_2,$$

then $h(c_1') = h(c_2')$;

(iv) if there are c_1' and c_2' such that

$$(\mathsf{inj}_2(C_1 + C_2))(c_1') = c_1$$

and

$$(\mathsf{inj}_2(C_1 + C_2))(c_2') = c_2,$$

then $c_1' = c_2'$.

If

$$c_1 \; ((\mathsf{inj}_1(C_1 + C_2))^{-1} \,;\, h \,;\, \mathsf{inj}_2(C_1 + C_2)) \; c_2,$$

then (i) is trivially true, and (ii), (iii), and (iv) are vacuously true.

If $c_1 = c_2$, then (i) and (ii) are vacuously true. Also, since $\mathsf{inj}_1(C_1 + C_2)$ is injective, (iii) is trivially true, and since $\mathsf{inj}_2(C_1 + C_2)$ is injective, (iv) is trivially true.

If

$$c_1 \; ((\mathsf{inj}_1(C_1 + C_2))^{-1} \,;\, h \,;\, \mathsf{inj}_2(C_1 + C_2))^{-1} \; c_2,$$

then (i) is vacuously true, (ii) is trivially true, and (iii) and (iv) are vacuously true.

Otherwise, there is c such that $c_1 \; E \; c$, and either

$$c \; ((\mathsf{inj}_1(C_1 + C_2))^{-1} \,;\, h \,;\, \mathsf{inj}_2(C_1 + C_2)) \; c_2$$

or

$$c \; ((\mathsf{inj}_1(C_1 + C_2))^{-1} \,;\, h \,;\, \mathsf{inj}_2(C_1 + C_2))^{-1} \; c_2.$$

If

$$c \; ((\mathsf{inj}_1(C_1 + C_2))^{-1} \,;\, h \,;\, \mathsf{inj}_2(C_1 + C_2)) \; c_2,$$

then there are c' and c_2' such that

$$(\mathsf{inj}_1(C_1 + C_2))(c') = c,$$

$$(\mathsf{inj}_2(C_1 + C_2))(c_2') = c_2,$$

and $h(c') = c_2'$. Thus, (ii) and (iii) are vacuously true. Also, by the induction hypothesis, if there is c_1' such that

$$(\mathsf{inj}_1(C_1 + C_2))(c_1') = c_1,$$

then $h(c_1') = h(c')$, and hence, $h(c_1') = c_2'$. Thus, (i) is true. Finally, by the induction hypothesis, if there is c_1' such that

$$(\mathsf{inj}_2(C_1 + C_2))(c_1') = c_1,$$

then $h(c') = c_1'$, and hence, $c_1' = c_2'$. Thus, (iv) is true.

Otherwise,

$$c \; ((\mathsf{inj}_1(C_1 + C_2))^{-1} \,;\, h \,;\, \mathsf{inj}_2(C_1 + C_2))^{-1} \; c_2.$$

Then there are c' and c_2' such that

$$(\mathsf{inj}_2(C_1 + C_2))(c') = c,$$

$$(\mathsf{inj}_1(C_1 + C_2))(c_2') = c_2,$$

and $c' = h(c_2')$. Thus, (i) and (iv) are vacuously true. Also, by the induction hypothesis, if there is c_1' such that

$$(\mathsf{inj}_2(C_1 + C_2))(c_1') = c_1,$$

then $c_1' = c'$, and hence, $h(c_2') = c_1'$. Thus, (ii) is true. Finally, by the induction hypothesis, if there is c_1' such that

$$(\mathsf{inj}_1(C_1 + C_2))(c_1') = c_1,$$

then $h(c_1') = c'$, and hence, $h(c_1') = h(c_2')$. Thus, (iii) is true.

Therefore, $q \circ \mathsf{inj}_2(C_1 + C_2)$ is injective, and thus, has a left inverse.

Another implication, which we are not using for this proof, is that

$$\{\langle c_1, c_2\rangle \mid q((\mathsf{inj}_1(C_1 + C_2))(c_1)) = q((\mathsf{inj}_2(C_1 + C_2))(c_2))\} = \mathsf{graph}\, h,$$

and thus, $\langle \mathsf{graph}\, h, \langle \mathsf{dpr}\, h, \mathsf{cpr}\, h\rangle\rangle$ is a pullback of $q \circ \mathsf{inj}_1(C_1 + C_2)$ and $q \circ \mathsf{inj}_2(C_1 + C_2)$ in F-**Coalg**.

Let g be a left inverse of $q \circ \mathsf{inj}_2(C_1 + C_2)$.

Since h is a class function, $\mathsf{dpr}\, h$ is bijective, and

$$h = (\mathsf{cpr}\, h) \circ (\mathsf{dpr}\, h)^{-1}.$$

Let γ be the cooperation of $\langle C_1, \gamma_1\rangle + \langle C_2, \gamma_2\rangle$.

Assume $c_1 \in C_1$.

Since h is a precongruence between $\langle C_1, \gamma_1\rangle$ and $\langle C_2, \gamma_2\rangle$,

$$F(q)(\gamma((\mathsf{inj}_1(C_1 + C_2))(c_1))) = F(q)(\gamma((\mathsf{inj}_2(C_1 + C_2))(h(c_1)))),$$

and hence,

$$F(q)(F(\mathsf{inj}_1(C_1 + C_2))(\gamma_1(c_1))) = F(q)(F(\mathsf{inj}_2(C_1 + C_2))(\gamma_2(h(c_1)))).$$

Thus,

$$\begin{aligned}
F(h)(\gamma_1(c_1)) &= F(\mathsf{cpr}\, h)(F(\mathsf{dpr}\, h)^{-1}(\gamma_1(c_1))) \\
&= F(g)(F(q)(F(\mathsf{inj}_2(C_1 + C_2))(F(\mathsf{cpr}\, h)(F(\mathsf{dpr}\, h)^{-1}(\gamma_1(c_1)))))) \\
&= F(g)(F(q)(F(\mathsf{inj}_1(C_1 + C_2))(F(\mathsf{dpr}\, h)(F(\mathsf{dpr}\, h)^{-1}(\gamma_1(c_1)))))) \\
&= F(g)(F(q)(F(\mathsf{inj}_1(C_1 + C_2))(\gamma_1(c_1)))) \\
&= F(g)(F(q)(F(\mathsf{inj}_2(C_1 + C_2))(\gamma_2(h(c_1))))) \\
&= \gamma_2(h(c_1)).
\end{aligned}$$

Thus, by generalization, h is a homomorphism from $\langle C_1, \gamma_1 \rangle$ to $\langle C_2, \gamma_2 \rangle$. □

An immediate corollary of Theorem 2.3.2 and 2.3.3.4 is that in the case of single-valued binary class relations, the concepts of bisimulation and precongruence coincide.

Here, mostly for the purpose of accessibility, we have decided to follow the approach of Rutten in [56], who advocates the coalgebraic concepts of bisimulation and bisimulation equivalence as formal duals to the algebraic ones of substitutive relation and congruence. His tacit preference over the more appropriate concepts of precongruence and congruence of [5] is partly justified by the the fact that more can be proved about bisimulations and bisimulation equivalences than precongruences and congruences. No matter: most of the theory in [56] is developed under the assumption that the endofunctor F *preserves weak pullbacks*, a technical condition under which the concepts of bisimulation and precongruence coincide. And although we will never need to make explicit mention of it, every particular endofunctor considered here will actually satisfy this condition, unless specifically intended not to.

2.4 Behaviour modelling and final coalgebras

Suppose that we wanted to use L-labelled transition systems to model the behaviour of processes of some kind. What we would wish for then is that there be a system diverse enough to model the behaviour of every process, but coarse enough not to distinguish between processes of equivalent behaviour. Is there such a system?

The way to use an L-labelled transition system to model the behaviour of a process is to map the process to a state of the system, and let the branching structure emanating from that state represent the behaviour of the process. Equivalence of behaviour then amounts to similarity of branching structure of some kind, and indeed determines the actual association between ‘behaviour’ and branching structure. If we let bisimilarity be that concept of similarity, and assume for simplicity that any state of every L-labelled transition system models the behaviour of some process, then our question becomes an enquiry over the existence of an L-labelled transition system $\langle S, T \rangle$ such that for every L-labelled transition system $\langle S', T' \rangle$ and any $s' \in S'$, there is $s \in S$ such that s' and s are bisimilar among $\langle S', T' \rangle$ and $\langle S, T \rangle$, and for any $s_1, s_2 \in S$, s_1 and s_2 are bisimilar in $\langle S, T \rangle$ if and only if $s_1 = s_2$.

We want to use Theorem 2.3.2 to turn this enquiry into an instance of a common universal construction problem, namely that of a terminal object of a category.

An object of a category is a *terminal object* of that category just as long as for every object of that category, there is exactly one arrow of that category from the latter to the former.

The category pertaining to our enquiry is that of all L-labelled transition coalgebras and all homomorphisms between them. But once again, we work generally.

First, notice that for every F-coalgebra $\langle C, \gamma\rangle$, $\mathsf{id}\, C$ is an endomorphism on $\langle C, \gamma\rangle$, and for every homomorphism h_1 from an F-coalgebra $\langle C_1, \gamma_1\rangle$ to an F-coalgebra $\langle C_2, \gamma_2\rangle$, and every homomorphism h_2 from $\langle C_2, \gamma_2\rangle$ to an F-coalgebra $\langle C_2, \gamma_2\rangle$, $h_2 \circ h_1$ is a homomorphism from $\langle C_1, \gamma_1\rangle$ to $\langle C_2, \gamma_2\rangle$. Thus, F-coalgebras and their homomorphisms form a category.

We write F-**Coalg** for the category whose objects are all the F-coalgebras, and arrows all the homomorphisms from one F-coalgebra to another.

Note that for any homomorphism h from an F-coalgebra $\langle C_1, \gamma_1\rangle$ to an F-coalgebra $\langle C_2, \gamma_2\rangle$, the domain and codomain of h as an arrow of F-**Coalg** are $\langle C_1, \gamma_1\rangle$ and $\langle C_2, \gamma_2\rangle$ respectively, and not to be confused with the domain and codomain of h as an arrow of **Class**, which are C_1 and C_2 respectively.

We say that $\langle C, \gamma\rangle$ *is final in* F-**Coalg** if and only if for every F-coalgebra $\langle C', \gamma'\rangle$, there is exactly one homomorphism from $\langle C', \gamma'\rangle$ to $\langle C, \gamma\rangle$.

We use "final" here rather than "terminal" only to conform with common practice in the germane literature: an F-coalgebra is final in F-**Coalg** if and only if it is a terminal object of F-**Coalg**.

Notice that if $\langle C_1, \gamma_1\rangle$ and $\langle C_2, \gamma_2\rangle$ are both final in F-**Coalg**, then

$$\langle C_1, \gamma_1\rangle \cong \langle C_2, \gamma_2\rangle,$$

lest there be another endomorphism, apart from the identity map, on either of them. In plain words, all final F-coalgebras are isomorphic to one another.

Theorem 2.3.2 suggests that there might be a connection between our enquiry and the notion of finality. We set out to make this connection, if any, precise.

We say that $\langle C, \gamma\rangle$ *is weakly complete in* F-**Coalg** if and only if for every F-coalgebra $\langle C', \gamma'\rangle$ and any $c' \in C'$, there is $c \in C$ such that c' and c are bisimilar among $\langle C', \gamma'\rangle$ and $\langle C, \gamma\rangle$.

Note that our use of the term "weakly complete" is different from, and in general, strictly more inclusive than that in [2], [5], and [3].

Note 2.4.1. In [2], [5], and [3], an F-coalgebra $\langle C, \gamma\rangle$ was called "weakly complete" if and only if for every small F-coalgebra $\langle C', \gamma'\rangle$, there is a homomorphism from $\langle C', \gamma'\rangle$ to $\langle C, \gamma\rangle$. Such an F-coalgebra is always weakly complete in our sense.

Proposition 2.4.1.1. *If for every small F-coalgebra $\langle C', \gamma' \rangle$, there is a homomorphism from $\langle C', \gamma' \rangle$ to $\langle C, \gamma \rangle$, then $\langle C, \gamma \rangle$ is weakly complete in F-**Coalg**.*

Proof. Suppose that for every small F-coalgebra $\langle C', \gamma' \rangle$, there is a homomorphism from $\langle C', \gamma' \rangle$ to $\langle C, \gamma \rangle$.

Assume an F-coalgebra $\langle C'', \gamma'' \rangle$.

Assume $c'' \in C''$.

By Lemma 2.4.8, there is a small F-coalgebra $\langle C', \gamma' \rangle$ such that $c'' \in C'$ and $\langle C', \gamma' \rangle \leq \langle C'', \gamma'' \rangle$. And by hypothesis, there is a homomorphism h from $\langle C', \gamma' \rangle$ to $\langle C, \gamma \rangle$.

Let $R = (C' \hookrightarrow C'')^{-1} \, ; h$.

Then, by Theorem 2.2.10, R is a bisimulation between $\langle C'', \gamma'' \rangle$ and $\langle C, \gamma \rangle$. And clearly, $c'' \; R \; h(c'')$. Thus, there is $c \in C$, namely $h(c'')$, such that c'' and c are bisimilar among $\langle C', \gamma' \rangle$ and $\langle C, \gamma \rangle$.

Thus, by generalization, $\langle C, \gamma \rangle$ is weakly complete in F-**Coalg**. □

However, an F-coalgebra that is weakly complete in our sense need not be weakly complete in the sense of [2], [5], and [3].

Example 2.4.1.2. Consider the left product endofunctor $\omega \times \mathsf{Id}$, which assigns to every class C the class

$$\omega \times C = \{\langle n, c \rangle \mid n \in \omega \text{ and } c \in C\},$$

and to every class function $f : C_1 \to C_2$, a class function

$$\omega \times f : \omega \times C_1 \to \omega \times C_2$$

such that for any $\langle n, c \rangle \in C_1$,

$$(\omega \times f)(\langle n, c \rangle) = \langle n, f(c) \rangle.$$

Let $C = \{\langle s, \mathsf{tail}^n \, s \rangle \mid s \in \mathscr{S}_{\mathrm{inf}} \, \omega \text{ and } n \in \omega\}$, and γ be a function from C to $\omega \times C$ such that for any $\langle s, \mathsf{tail}^n \, s \rangle \in C$,

$$\gamma(\langle s, \mathsf{tail}^n \, s \rangle) = \langle \mathsf{head} \, \mathsf{tail}^n \, s, \langle s, \mathsf{tail}^{n+1} \, s \rangle \rangle.$$

For every $(\omega \times \mathsf{Id})$-coalgebra $\langle C', \gamma' \rangle$ and any $c' \in C'$, if s is an infinite sequence such that for every $n \in \omega$,

$$\mathsf{head} \, \mathsf{tail}^n \, s = (\mathsf{proj}_1(\omega \times C'))((\mathsf{proj}_2(\omega \times C') \circ \gamma')^n(c')),$$

then, by an easy induction, it is the only such infinite sequence, and c' and $\langle s, s\rangle$ are bisimilar in $\langle C', \gamma'\rangle$ and $\langle C, \gamma\rangle$. Thus, $\langle C, \gamma\rangle$ is weakly complete in $(\omega \times \mathsf{Id})$-**Coalg**.

Now let C' be the set of all integers, and γ' a function from C' to $\omega \times C'$ such that for any integer i,

$$\gamma'(i) = \begin{cases} \langle 2i, i-1\rangle & \text{if } 0 \leq i; \\ \langle -2i-1, i-1\rangle & \text{otherwise.} \end{cases}$$

$\langle C', \gamma'\rangle$ is a small $(\omega \times \mathsf{Id})$-coalgebra. However, there is no homomorphism from $\langle C', \gamma'\rangle$ to $\langle C, \gamma\rangle$, lest there be an order-embedding from the standardly ordered set of all integers to the standardly ordered set of all natural numbers.

Also, an F-coalgebra that is weakly complete in the sense of [2], [5], and [3] need not be weakly final in F-**Coalg**.

Example 2.4.1.3. Let $\langle C, \gamma\rangle$ be a direct sum of all small Pow-coalgebras.

Then for every small Pow-coalgebra $\langle C', \gamma'\rangle$, there is a homomorphism from $\langle C', \gamma'\rangle$ to $\langle C, \gamma\rangle$, namely the canonical injection map from C' to C.

Let Ord be the class of all ordinal numbers, and ϵ a class function from Ord to $\mathsf{Pow}\,\mathrm{Ord}$ such that for every ordinal number α,

$$\epsilon(\alpha) = \{\beta \mid \beta < \alpha\}.$$

Then, by an easy transfinite induction argument, for every Pow-coalgebra $\langle C'', \gamma''\rangle$, any homomorphism from $\langle \mathrm{Ord}, \epsilon\rangle$ to $\langle C'', \gamma''\rangle$ is injective. Also, again by an easy transfinite induction argument, if h is homomorphism from $\langle \mathrm{Ord}, \epsilon\rangle$ to $\langle C, \gamma\rangle$, there is a small Pow-coalgebra $\langle C''', \gamma'''\rangle$ such that if ι is the canonical injection map from C''' to C, then $\mathsf{ran}\, h \subseteq \mathsf{ran}\, \iota$. Thus, there is no homomorphism from $\langle \mathrm{Ord}, \epsilon\rangle$ to $\langle C, \gamma\rangle$, lest there be an injective class function from the class of all ordinal numbers to a set, and hence, $\langle C, \gamma\rangle$ is not weakly final in Pow-**Coalg**.

The following is immediate from Theorem 2.3.2:

Proposition 2.4.1. *If $\langle C, \gamma\rangle$ is final in F-**Coalg**, then $\langle C, \gamma\rangle$ is weakly complete in F-**Coalg**.*

The notion of weak completeness is meant as a coalgebraic generalization the first of the two conditions of our enquiry. The one introduced next is meant as a coalgebraic generalization of the second.

We say that $\langle C, \gamma\rangle$ *is strongly extensional* if and only if for every $c_1, c_2 \in C$, $c_1 = c_2$ if and only if c_1 and c_2 are bisimilar in $\langle C, \gamma\rangle$.

We use the term "strongly extensional" here in the same way that Rutten and Turi did in [57]. This choice of term was suggested by the special case of Pow, where it was used quite literally, in reference to a stronger form of the Axiom of Extensionality, one better suited to a theory of sets that need not be well founded (see [2]). All other uses of it in [2], [5], and [3] are ultimately equivalent to that in [57].

Note 2.4.2. In [2] and [5], an F-coalgebra $\langle C, \gamma\rangle$ was called "strongly extensional" if and only if for every small F-coalgebra $\langle C', \gamma'\rangle$, there is at most one homomorphism from $\langle C', \gamma'\rangle$ to $\langle C, \gamma\rangle$, whereas in [3], if and only if for every F-coalgebra $\langle C', \gamma'\rangle$, there is at most one homomorphism from $\langle C', \gamma'\rangle$ to $\langle C, \gamma\rangle$. All three notions coincide.

Proposition 2.4.2.1. *The following are equivalent:*

(a) $\langle C, \gamma\rangle$ *is strongly extensional;*

(b) for every F*-coalgebra* $\langle C', \gamma'\rangle$*, there is at most one homomorphism from* $\langle C', \gamma'\rangle$ *to* $\langle C, \gamma\rangle$*;*

(c) for every small F*-coalgebra* $\langle C', \gamma'\rangle$*, there is at most one homomorphism from* $\langle C', \gamma'\rangle$ *to* $\langle C, \gamma\rangle$*.*

Proof. By [23, thm. 6.13], (a) and (b) are equivalent. And since (b) trivially implies (c), it suffices to prove that (c) implies (b).

Suppose that for every small F-coalgebra $\langle C', \gamma'\rangle$, there is at most one homomorphism from $\langle C', \gamma'\rangle$ to $\langle C, \gamma\rangle$.

Suppose, toward contradiction, that there is an F-coalgebra $\langle C'', \gamma''\rangle$, and homomorphisms h_1 and h_2 from $\langle C'', \gamma''\rangle$ to $\langle C, \gamma\rangle$ such that $h_1 \neq h_2$. Then there is $c'' \in C''$ such that $h_1(c'') \neq h_2(c'')$. By Lemma 2.4.8, there is a small F-coalgebra $\langle C', \gamma'\rangle$ such that $c'' \in C'$ and $\langle C', \gamma'\rangle \leq \langle C'', \gamma''\rangle$. Thus, both $h_1 \circ (C' \hookrightarrow C'')$ and $h_2 \circ (C' \hookrightarrow C'')$ are homomorphisms from $\langle C', \gamma'\rangle$ to $\langle C, \gamma\rangle$. However,

$$(h_1 \circ (C' \hookrightarrow C''))(c'') = h_1(c'') \neq h_2(c'') = (h_2 \circ (C' \hookrightarrow C''))(c''),$$

and thus,

$$h_1 \circ (C' \hookrightarrow C'') \neq h_2 \circ (C' \hookrightarrow C''),$$

contrary to our hypothesis.

Therefore, for every F-coalgebra $\langle C'', \gamma'' \rangle$, there is at most one homomorphism from $\langle C', \gamma' \rangle$ to $\langle C, \gamma \rangle$. □

Theorem 2.4.2. *If $\langle C, \gamma \rangle$ is final in F-**Coalg**, then $\langle C, \gamma \rangle$ is strongly extensional.*

Proof. See [57, thm. 2.4]. □

Theorem 2.4.2 is equivalent to the statement that any final F-coalgebra $\langle C, \gamma \rangle$ satisfies what is now known as the *coinduction proof principle*, whereby for every $c_1, c_2 \in C$, in order to prove that $c_1 = c_2$, one need only find a bisimulation B on $\langle C, \gamma \rangle$ such that $c_1 \; B \; c_2$ (see [56, thm. 9.2]).

Note 2.4.3. The coinduction proof principle applies to every *simple* F-coalgebra, namely every F-coalgebra that has no proper quotient, or equivalently, every F-coalgebra $\langle C, \gamma \rangle$ such that for every F-coalgebra $\langle C', \gamma' \rangle$, if h is an epimorphism from $\langle C, \gamma \rangle$ to $\langle C', \gamma' \rangle$, then h is an isomorphism between $\langle C, \gamma \rangle$ to $\langle C', \gamma' \rangle$.

Dually, the induction proof principle applies to every *minimal* F-algebra, namely every F-algebra that has no proper subalgebra, or equivalently, every F-algebra $\langle C, \alpha \rangle$ such that for every F-algebra $\langle C', \alpha' \rangle$, if h is a monomorphism from $\langle C', \alpha' \rangle$ to $\langle C, \alpha \rangle$, then h is an isomorphism between $\langle C', \alpha' \rangle$ to $\langle C, \alpha \rangle$.

These statements appear explicitly in [57, sec. 6.3], which is, apparently, where the term "coinduction" was used for the first time in this coalgebraic setting.

By [23, lem. 4.12], an F-coalgebra $\langle C, \gamma \rangle$ is simple, in the above sense, if and only if for any $c_1, c_2 \in C$, $c_1 = c_2$ if and only if c_1 and c_2 are congruent in $\langle C, \gamma \rangle$.

Note that in [23], an F-coalgebra $\langle C, \gamma \rangle$ is called simple if and only if it satisfies the coinduction proof principle (see [23, def. 6.12]).

Now, by Proposition 2.2.15 and 2.4.1, and Theorem 2.4.2, any system having a coalgebraic representation that is final in $(\mathsf{Pow} \circ (L \times \mathsf{Id}))$-**Coalg** will satisfy both conditions of our enquiry. So if we can prove that the converse is also true, then we need look no further.

Before attempting such a proof, there is a small issue that we need to resolve regarding the first of these conditions and our coalgebraic generalization of it. By Proposition 2.2.1(b), 2.2.13, and 2.2.15, the former is equivalent to the statement, "for every small L-labelled transition coalgebra $\langle C, \tau\rangle$ and any $c \in C$, there is $s \in S$ such that c and s are bisimilar among $\langle C, \tau\rangle$ and $\langle S, \mathsf{fun}\, T\rangle$". Because of the term "small" in the first quantifier of this statement, saying that $\langle S, \mathsf{fun}\, T\rangle$ is weakly complete in $(\mathsf{Pow} \circ (L \times \mathsf{Id}))$-**Coalg** seems to impose a much stronger constraint on the system $\langle S, T\rangle$. Whether this is in fact a stronger constraint is really a matter of whether an increase in the size of the carrier beyond that of a set can bring about new kinds of behaviour in a labelled transition coalgebra.

In [5], Aczel and Mendler introduced a general condition meant precisely to guard against this type of possibility. They called F *set-based* if and only if for every class C and any $c \in F(C)$, there is a subset S of C, and $s \in F(S)$, such that $c = F(S \hookrightarrow C)(s)$.[14]

Now, it is quite obvious that $\mathsf{Pow} \circ (L \times \mathsf{Id})$ is set-based. What is not so obvious is that, actually, every endofunctor on **Class** is set-based. Adamek, Milius, and Velebil proved this surprising fact in [7] for the standard set-theoretic model of **Class**, using a classical result from combinatorial set theory, but the same ideas extend to other models of **Class** (for example, see [15]).

Here, we will be using this fact under the guise of *The Small Subcoalgebra Lemma* of [5].

Definition 2.4.3. A *subcoalgebra of* $\langle C, \gamma\rangle$ is an F-coalgebra $\langle C', \gamma'\rangle$ such that $C' \subseteq C$, and $C' \hookrightarrow C$ is a homomorphism from $\langle C', \gamma'\rangle$ to $\langle C, \gamma\rangle$.

Thus, $\langle C', \gamma'\rangle$ is a subcoalgebra of $\langle C, \gamma\rangle$ just as long as it is an F-coalgebra, C' is a subclass of C, and the following diagram commutes:

$$\begin{tikzcd}
C' \arrow[r, "C' \hookrightarrow C"] \arrow[d, "\gamma'"'] & C \arrow[d, "\gamma"] \\
F(C') \arrow[r, "F(C' \hookrightarrow C)"'] & F(C)
\end{tikzcd}$$

We write $\langle C', \gamma'\rangle \leq \langle C, \gamma\rangle$ if and only if $\langle C', \gamma'\rangle$ is a subcoalgebra of $\langle C, \gamma\rangle$.

The concept of subcoalgebra is the coalgebraic counterpart of the concept of subalgebra, which is a generalization, again in the same sense as before, of the concept of Σ-subalgebra. It is a part of the original coalgebra, that is closed, in a suitably generalized sense, under the decomposition rules of the latter.

[14] For every class C_1 and C_2 such that $C_1 \subseteq C_2$, we write $C_1 \hookrightarrow C_2$ for a function from C_1 fo C_2 such that for any $c_1 \in C_1$, $(C_1 \hookrightarrow C_2)(c_1) = c_1$. We call $C_1 \hookrightarrow C_2$ the *inclusion map* from C_1 to C_2.

For example, if $\langle S, T\rangle$ is a transition system, and $\langle S', \gamma'\rangle$ a subcoalgebra of $\langle S, \mathsf{fun}\, T\rangle$, then S' is a set of states of $\langle S, T\rangle$ that is closed under the transition relation of $\langle S, T\rangle$, and $\mathsf{rel}\,\gamma'$ is the restriction of that transition relation onto that set of states.

As one might expect, the cooperation of a subcoalgebra is uniquely determined by its carrier.

Proposition 2.4.4. *If $\langle C_1, \gamma_1\rangle \leq \langle C, \gamma\rangle$, $\langle C_2, \gamma_2\rangle \leq \langle C, \gamma\rangle$, and $C_1 = C_2$, then $\gamma_1 = \gamma_2$.*

Proof. See [56, prop. 6.1]. □

The following can be used as criteria for choosing an eligible carrier:

Theorem 2.4.5. *If h is a homomorphism from $\langle C_1, \gamma_1\rangle$ to $\langle C_2, \gamma_2\rangle$, then there is a class function $\rho : \mathsf{ran}\, h \to F(\mathsf{ran}\, h)$ such that*

$$\langle \mathsf{ran}\, h, \rho\rangle \leq \langle C_2, \gamma_2\rangle.$$[15]

Proof. See [56, thm. 6.3]. □

Theorem 2.4.6. *For every class-indexed family $\{\langle C_i, \gamma_i\rangle\}_{i\in I}$ of subcoalgebras of $\langle C, \gamma\rangle$, there is a class function $\upsilon : \bigcup_{i\in I} C_i \to F(\bigcup_{i\in I} C_i)$ such that*

$$\langle \bigcup_{i\in I} C_i, \upsilon\rangle \leq \langle C, \gamma\rangle.$$

Proof. See [23, thm. 4.7]. □

Finally, every homomorphism factorizes, in a unique fashion, through every subcoalgebra of its codomain F-coalgebra that contains its range.

Proposition 2.4.7. *If h is a homomorphism from $\langle C_1, \gamma_1\rangle$ to $\langle C_2, \gamma_2\rangle$, and $\langle C, \gamma\rangle$ is a subcoalgebra of $\langle C_2, \gamma_2\rangle$ such that $\mathsf{ran}\, h \subseteq C$, then there is exactly one homomorphism h' from $\langle C_1, \gamma_1\rangle$ to $\langle C, \gamma\rangle$ such that*

$$h = (C \hookrightarrow C_2) \circ h'.$$

Proof. See [56, prop. 6.5]. □

[15] For every class function f, we write $\mathsf{ran}\, f$ for the class $\{y \mid \text{there is } x \in \mathsf{dom}\, f \text{ such that } y = f(x)\}$. We call $\mathsf{ran}\, f$ the *range* of f.

Thus, if h, $\langle C_1, \gamma_1\rangle$, $\langle C_2, \gamma_2\rangle$, and $\langle C, \gamma\rangle$ are as in Proposition 2.4.7, then there is exactly one homomorphism h' from $\langle C_1, \gamma_1\rangle$ to $\langle C, \gamma\rangle$ such that the following diagram commutes:

$$\begin{tikzcd}
\langle C_1, \gamma_1\rangle \arrow[rr, "h"] \arrow[dr, dashed, "h'"'] & & \langle C_2, \gamma_2\rangle \\
& \langle C, \gamma\rangle \arrow[ur, "C \hookrightarrow C_2"'] &
\end{tikzcd}$$

Theorem 2.4.6 and Proposition 2.4.7 can be used to arrange the subcoalgebras of an F-coalgebra into a complete lattice (see [23, cor. 4.9]).

The Small Subcoalgebra Lemma was a key lemma in [5], and will be a key lemma here as well.

Lemma 2.4.8. *For every subset S of C, there is a small F-coalgebra $\langle C', \gamma'\rangle$ such that $S \subseteq C'$ and $\langle C', \gamma'\rangle \leq \langle C, \gamma\rangle$.*

Proof. See [5, lem. 2.2] and [7, thm. 2.2]. □

Our fist use of it is in resolving our last issue.

Theorem 2.4.9. *$\langle C, \gamma\rangle$ is weakly complete in F-**Coalg** if and only if for every small F-coalgebra $\langle C', \gamma'\rangle$ and any $c' \in C'$, there is $c \in C$ such that c' and c are bisimilar among $\langle C', \gamma'\rangle$ and $\langle C, \gamma\rangle$.*

Proof. If $\langle C, \gamma\rangle$ is weakly complete in F-**Coalg**, then, trivially, for every small F-coalgebra $\langle C', \gamma'\rangle$ and any $c' \in C'$, there is $c \in C$ such that c' and c are bisimilar among $\langle C', \gamma'\rangle$ and $\langle C, \gamma\rangle$.

Conversely, suppose that for every small F-coalgebra $\langle C', \gamma'\rangle$ and any $c' \in C'$, there is $c \in C$ such that c' and c are bisimilar among $\langle C', \gamma'\rangle$ and $\langle C, \gamma\rangle$.

Assume an F-coalgebra $\langle C'', \gamma''\rangle$.

Assume $c'' \in C''$.

By Lemma 2.4.8, there is a small F-coalgebra $\langle C', \gamma'\rangle$ such that $c'' \in C'$ and $\langle C', \gamma'\rangle \leq \langle C'', \gamma''\rangle$. By hypothesis, there is $c \in C$ such that c'' and c are bisimilar among $\langle C', \gamma'\rangle$ and $\langle C, \gamma\rangle$. Thus, there is a bisimulation B between $\langle C', \gamma'\rangle$ and $\langle C, \gamma\rangle$ such that $c''\ B\ c$.

Let $R = (C' \hookrightarrow C'')^{-1} \,;B$.

Then, by Corollary 2.2.11, R is a bisimulation between $\langle C'', \gamma'' \rangle$ and $\langle C, \gamma \rangle$. And clearly, $c'' \mathrel{R} c$. Thus, c'' and c are bisimilar among $\langle C', \gamma' \rangle$ and $\langle C, \gamma \rangle$.

Thus, by generalization, $\langle C, \gamma \rangle$ is weakly complete in F-**Coalg**. □

By Proposition 2.2.1(b), 2.2.13, and 2.2.14, and Theorem 2.4.9, an L-labelled transition system $\langle S, T \rangle$ will satisfy the first condition of our enquiry if and only if $\langle S, \mathsf{fun}\, T \rangle$ is weakly complete in $(\mathsf{Pow} \circ (L \times \mathsf{Id}))$-**Coalg**, and by Proposition 2.2.14, it will satisfy the second if and only if $\langle S, \mathsf{fun}\, T \rangle$ is strongly extensional. We are thus left with the problem of proving that if $\langle S, \mathsf{fun}\, T \rangle$ is both weakly complete in $(\mathsf{Pow} \circ (L \times \mathsf{Id}))$-**Coalg**, and strongly extensional, then it is final in $(\mathsf{Pow} \circ (L \times \mathsf{Id}))$-**Coalg**.

Although it is indeed possible to prove this, such proof does not generalize as one might expect: weak completeness and strong extensionality do not, in general, imply finality. But before we see an example attesting to this claim, we need to introduce a simple coalgebra construction that generalizes the notion of disjoint union, allowing us to merge different coalgebras into a single whole.

Assume a class-indexed family $\{\langle C_i, \gamma_i \rangle\}_{i \in I}$ of F-coalgebras.

Definition 2.4.10. The *direct sum of* $\{\langle C_i, \gamma_i \rangle\}_{i \in I}$ is an F-coalgebra $\langle C, \gamma \rangle$ such that the following are true:

(a) C is the disjoint union[16] of $\{C_i\}_{i \in I}$;

(b) γ is a class function from C to $F(C)$ such that for any $j \in I$ and any $c \in C_j$,

$$\gamma(\langle j, c \rangle) = F(\mathsf{inj}_j \sum_{i \in I} C_i)(\gamma_j(c)).$$ [17]

We write $\sum_{i \in I} \langle C_i, \gamma_i \rangle$ for the direct sum of $\{\langle C_i, \gamma_i \rangle\}_{i \in I}$.

We write $\langle C_1, \gamma_1 \rangle + \langle C_2, \gamma_2 \rangle$ for $\sum_{i \in \{1,2\}} \langle C_i, \gamma_i \rangle$.

Notice that for any $j \in I$, the canonical injection map $\mathsf{inj}_j \sum_{i \in I} C_i$ is trivially a homomorphism from $\langle C_j, \gamma_j \rangle$ to $\sum_{i \in I} \langle C_i, \gamma_i \rangle$.

The most important, and practically defining property of the direct sum is the following:

[16] For every class-indexed family $\{C_i\}_{i \in I}$ of classes, the disjoint union of $\{C_i\}_{i \in I}$ is the class $\{\langle i, c \rangle \mid i \in I \text{ and } c \in C_i\}$. We write $\sum_{i \in I} C_i$ for the disjoint union of $\{C_i\}_{i \in I}$.

[17] For every class-indexed family $\{C_i\}_{i \in I}$ of classes and any $j \in I$, we write $\mathsf{inj}_j \sum_{i \in I} C_i$ for a function from C_j to $\sum_{i \in I} C_i$ such that for any $c \in C_j$, $(\mathsf{inj}_j \sum_{i \in I} C_i)(c) = \langle j, c \rangle$. We call $\mathsf{inj}_j \sum_{i \in I} C_i$ the *canonical injection map* from C_j to $\sum_{i \in I} C_i$.

Proposition 2.4.11. *For every class-indexed family $\{h_i\}_{i\in I}$ such that for any $i \in I$, h_i is a homomorphism from $\langle C_i, \gamma_i\rangle$ to $\langle C, \gamma\rangle$, there is exactly one homomorphism h from $\sum_{i\in I} \langle C_i, \gamma_i\rangle$ to $\langle C, \gamma\rangle$ such that for any $j \in I$,*

$$h_j = h \circ \mathsf{inj}_j \sum_{i\in I} C_i.$$

Proof. See [23, lem. 4.1]. □

Thus, for every F-coalgebra $\langle C, \gamma\rangle$ and class-indexed family $\{h_i\}_{i\in I}$ such that for any $i \in I$, h_i is a homomorphism from $\langle C_i, \gamma_i\rangle$ to $\langle C, \gamma\rangle$, there is exactly one *mediating* homomorphism h from $\sum_{i\in I} \langle C_i, \gamma_i\rangle$ to $\langle C, \gamma\rangle$ such that the following diagram commutes:

$$\begin{tikzcd}
\langle C_j, \gamma_j\rangle \arrow[r, "\mathsf{inj}_j \sum_{i\in I} C_i"] \arrow[dr, "h_j"'] & \sum_{i\in I} \langle C_i, \gamma_i\rangle \arrow[d, dashed, "h"] \\
& \langle C, \gamma\rangle
\end{tikzcd}$$

The disjoint sum construction is one particular instance of the more abstract category-theoretic concept of coproduct, which is defined by use of the property of Proposition 2.4.11, only generalized to a category of arbitrary objects and arrows. Another is the disjoint union construction.

Note 2.4.4. What about the concept of product?

In [55, p. 5], Rutten writes the following:

> The *direct sum* (or *coproduct*) of any collection of transition systems consists of the disjoint union of their carriers together with (the transition structure determined by) the disjoint union of their transition relations. In general, the *product* (in the category of transition systems) of two transition systems need not exist. For instance, let $S = \{0, 1, 2\}$ with $\alpha_S(0) = \{0, 1\}$, and $\alpha_S(1) = \alpha_S(2) = \emptyset$. There does not exist a product of $\langle S, \alpha_S\rangle$ with itself.

This is false: there does exist a product of $\langle S, \alpha_S\rangle$ with itself in Pow-**Coalg**.

Let $P = \{0, \langle 1,1\rangle, \langle 1,2\rangle, \langle 2,1\rangle, \langle 2,2\rangle\}$, and τ be a class function from P to $\mathsf{Pow}\,P$ defined by the following mapping:

$$\begin{aligned} 0 &\mapsto \{0, \langle 1,1\rangle\};\\ \langle 1,1\rangle &\mapsto \emptyset;\\ \langle 1,2\rangle &\mapsto \emptyset;\\ \langle 2,1\rangle &\mapsto \emptyset;\\ \langle 2,2\rangle &\mapsto \emptyset. \end{aligned}$$

Let π_1 be a function from P to S defined by the following mapping:

$$\begin{aligned} 0 &\mapsto 0;\\ \langle 1,1\rangle &\mapsto 1;\\ \langle 1,2\rangle &\mapsto 1;\\ \langle 2,1\rangle &\mapsto 2;\\ \langle 2,2\rangle &\mapsto 2. \end{aligned}$$

Let π_2 be a function from P to S defined by the following mapping:

$$\begin{aligned} 0 &\mapsto 0;\\ \langle 1,1\rangle &\mapsto 1;\\ \langle 1,2\rangle &\mapsto 2;\\ \langle 2,1\rangle &\mapsto 1;\\ \langle 2,2\rangle &\mapsto 2. \end{aligned}$$

Then $\langle P, \tau\rangle$, together with π_1 and π_2, is a product of $\langle S, \alpha_S\rangle$ with itself in Pow-**Coalg**.

More generally, for every endofunctor F on class, all small products exist in F-**Coalg** (see [7, rem. 3.20]).

We are now ready to proceed with our promised example.

Example 2.4.12. Let F be as in Example 2.3.3.

Call an F-coalgebra $\langle C, \gamma\rangle$ *nameless* if and only if for any $c \in C$,

$$|\mathsf{ran}\,\gamma(c)| = 2.$$

The first thing to notice about a nameless F-coalgebra $\langle C, \gamma \rangle$ is that if c_1 and c_2 are bisimilar in $\langle C, \gamma \rangle$, and c_2 and c_3 are bisimilar in $\langle C, \gamma \rangle$, then c_1 and c_3 are bisimilar in $\langle C, \gamma \rangle$; that is, bisimilarity in $\langle C, \gamma \rangle$ is a transitive concept, and thus, by Theorem 2.2.12, the largest bisimulation on $\langle C, \gamma \rangle$ is a bisimulation equivalence.

Let $\langle C, \gamma \rangle$ be a direct sum of all nameless F-coalgebras.

Clearly, $\langle C, \gamma \rangle$ is itself nameless. It is also weakly complete in F-**Coalg**. To see this, notice that every F-coalgebra $\langle C', \gamma' \rangle$ is a homomorphic image of a nameless F-coalgebra $\langle C'', \gamma'' \rangle$ such that

$$C'' = C' + \{c'' \mid \text{there is } c' \in C' \text{ such that } \gamma'(c') = \langle c'', c'', c'' \rangle\}.$$

and for any $\langle i, c' \rangle \in C''$, if

$$\gamma'(c') = \langle c'_1, c'_2, c'_3 \rangle,$$

then

$$\gamma''(\langle c', i \rangle) = \begin{cases} \langle \langle 0, c'_1 \rangle, \langle 0, c'_2 \rangle, \langle 0, c'_3 \rangle \rangle & \text{if } |\{c'_1, c'_2, c'_3\}| = 2; \\ \langle \langle 0, c'_1 \rangle, \langle 0, c'_2 \rangle, \langle 1, c'_3 \rangle \rangle & \text{otherwise.} \end{cases}$$

Weak completeness then follows from Theorem 2.2.10 and 2.4.9.

Let q be a quotient of C with respect to the largest bisimulation equivalence on $\langle C, \gamma \rangle$.

By Proposition 2.3.1, there is exactly one F-coalgebra $\langle \mathsf{cod}\, q, \delta \rangle$ such that q is an epimorphism from $\langle C, \gamma \rangle$ to $\langle \mathsf{cod}\, q, \delta \rangle$. And by an argument similar to that in Example 2.3.3, one can see that $\langle \mathsf{cod}\, q, \delta \rangle$ is also nameless. More importantly, $\langle \mathsf{cod}\, q, \delta \rangle$ is both weakly complete in F-**Coalg**, and strongly extensional. Weak completeness follows from weak completeness of $\langle C, f \rangle$ and Corollary 2.2.11, and strong extensionality by the straightforward observation that the inverse image of a bisimulation under a homomorphism from one nameless F-coalgebra to another is itself a bisimulation, which, by Theorem 2.3.2, is just a consequence of the fact that bisimilarity in a nameless F-coalgebra is transitive. However, $\langle \mathsf{cod}\, q, \delta \rangle$ is not final in F-**Coalg**, lest it be isomorphic to $\langle S_2, \gamma_2 \rangle$, and thus, 0 and 1 bisimilar in $\langle S_1, \gamma_1 \rangle$ of Example 2.3.3.

This issue is just another manifestation of the aforementioned separation between the coalgebraic concepts of bisimulation and precongruence; had we used the latter to define the notion of strong extensionality, there would be no issue.

Fortunately, there is a simple way around this.

We say that $\langle C, \gamma \rangle$ *is complete in* F-**Coalg** if and only if for every F-coalgebra $\langle C', \gamma' \rangle$ and any $c' \in C'$, there is exactly one $c \in C$ such that c' and c are bisimilar among $\langle C', \gamma' \rangle$ and $\langle C, \gamma \rangle$.

By an easy corollary of Proposition 2.2.14, bisimilarity among L-labelled transition coalgebras is a transitive concept, and thus, an L-labelled transition coalgebra is both weakly complete in $(\mathsf{Pow} \circ (L \times \mathsf{Id}))$-**Coalg**, and strongly extensional, if and only if it is complete in $(\mathsf{Pow} \circ (L \times \mathsf{Id}))$-**Coalg**. Therefore, we may just as well forget about weak completeness and strong extensionality, and work with completeness instead, whose equivalence to finality generalizes nicely.

Apart from establishing this generalized equivalence, the following shows that when dealing with the two notions, one need only worry about coalgebras that are small:

Theorem 2.4.13. *The following are equivalent:*

*(a) $\langle C, \gamma\rangle$ is final in F-**Coalg**;*

(b) for every small F-coalgebra $\langle C', \gamma'\rangle$, there is exactly one homomorphism from $\langle C', \gamma'\rangle$ to $\langle C, \gamma\rangle$;

*(c) $\langle C, \gamma\rangle$ is complete in F-**Coalg**;*

(d) for every small F-coalgebra $\langle C', \gamma'\rangle$, and any $c' \in C'$, there is exactly one $c \in C$ such that c' and c are bisimilar among $\langle C', \gamma'\rangle$ and $\langle C, \gamma\rangle$.

Proof. Trivially, (a) implies (b), and (c) implies (d). Therefore, it suffices to prove that (b) implies (c), and (d) implies (a).

Suppose that for every small F-coalgebra $\langle C', \gamma'\rangle$, there is exactly one homomorphism from $\langle C', \gamma'\rangle$ to $\langle C, \gamma\rangle$.

Assume an F-coalgebra $\langle C', \gamma'\rangle$.

Assume $c' \in C'$.

By Lemma 2.4.8, there is a small F-coalgebra $\langle C'', \gamma''\rangle$ such that $c' \in C''$ and $\langle C'', \gamma''\rangle \leq \langle C', \gamma'\rangle$. By hypothesis, there is exactly one homomorphism h from $\langle C'', \gamma''\rangle$ to $\langle C, \gamma\rangle$.

Let $R = (C'' \hookrightarrow C')^{-1} \,;h$.

Then, by Theorem 2.2.10, R is a bisimulation between $\langle C', \gamma'\rangle$ and $\langle C, \gamma\rangle$. And clearly, $c' \; R \; h(c')$. Thus, there is $c \in C$, namely $h(c')$, such that c' and c are bisimilar among $\langle C', \gamma'\rangle$ and $\langle C, \gamma\rangle$.

Suppose, toward contradiction, that there are $c_1, c_2 \in C$ such that c' and c_1 are bisimilar among $\langle C', \gamma'\rangle$ and $\langle C, \gamma\rangle$, c' and c_2 are bisimilar among $\langle C', \gamma'\rangle$ and $\langle C, \gamma\rangle$, and $c_1 \neq c_2$. Then there are bisimulations B_1 and B_2 between $\langle C', \gamma'\rangle$ and $\langle C, \gamma\rangle$ such that $c' \; B_1 \; c_1$ and

$c' \; B_2 \; c_2$. By Theorem 2.2.12, there is a bisimulation B between $\langle C', \gamma' \rangle$ and $\langle C, \gamma \rangle$ such that

$$\mathsf{graph}\, B = \mathsf{graph}\, B_1 \cup \mathsf{graph}\, B_2.$$

Let $\langle \mathsf{graph}\, B, \beta \rangle$ be an F-coalgebra such that $\mathsf{dpr}\, B$ is a homomorphism from $\langle \mathsf{graph}\, B, \beta \rangle$ to $\langle C', \gamma' \rangle$, and $\mathsf{cpr}\, B$ one from $\langle \mathsf{graph}\, B, \beta \rangle$ to $\langle C, \gamma \rangle$.

By Lemma 2.4.8, there is a small F-coalgebra $\langle G, \beta' \rangle$ such that $\{\langle c', c_1 \rangle, \langle c', c_2 \rangle\} \subseteq G$ and $\langle G, \beta' \rangle \leq \langle \mathsf{graph}\, B, \beta \rangle$.

Let B' be a binary class relation between C' and C such that

$$\mathsf{graph}\, B' = G.$$

Clearly, B' is a bisimulation between $\langle C', \gamma' \rangle$ and $\langle C, \gamma \rangle$.

By Theorem 2.4.5, there is a class function $\rho : \mathsf{ran}\, \mathsf{dpr}\, B' \to F(\mathsf{ran}\, \mathsf{dpr}\, B')$ such that

$$\langle \mathsf{ran}\, \mathsf{dpr}\, B', \rho \rangle \leq \langle C', \gamma' \rangle.$$

And by Proposition 2.4.7, there is exactly one homomorphism π from $\langle \mathsf{graph}\, B', \beta \rangle$ to $\langle \mathsf{ran}\, \mathsf{dpr}\, B', \rho \rangle$ such that

$$\mathsf{dpr}\, B' = (\mathsf{ran}\, \mathsf{dpr}\, B' \hookrightarrow C') \circ \pi.$$

Since $\langle \mathsf{graph}\, B', \beta' \rangle$ is small, $\langle \mathsf{ran}\, \mathsf{dpr}\, B', \rho \rangle$ is small. Thus, by hypothesis, there is exactly one homomorphism h' from $\langle \mathsf{ran}\, \mathsf{dpr}\, B', \rho \rangle$ to $\langle C, \gamma \rangle$. Then both $h' \circ \pi$ and $\mathsf{cpr}\, B'$ are homomorphisms from $\langle \mathsf{graph}\, B', \beta' \rangle$ to $\langle C, \gamma \rangle$. However,

$$(h' \circ \pi)(\langle c', c_1 \rangle) = h'(c') = (h' \circ \pi)(\langle c', c_2 \rangle)$$

and

$$(\mathsf{cpr}\, B')(\langle c', c_1 \rangle) = c_1 \neq c_2 = (\mathsf{cpr}\, B')(\langle c', c_2 \rangle),$$

and thus,

$$h' \circ \pi \neq \mathsf{cpr}\, B',$$

contrary to our hypothesis.

Therefore, there is at most one $c \in C$ such that c' and c are bisimilar among $\langle C', \gamma' \rangle$ and $\langle C, \gamma \rangle$.

Thus, there is exactly one $c \in C$, namely $h(c')$, such that c' and c are bisimilar among $\langle C', \gamma' \rangle$ and $\langle C, \gamma \rangle$.

Thus, by generalization, $\langle C, \gamma\rangle$ is complete in F-**Coalg**.

We have thereby proved that (b) implies (c). It remains to prove that (d) implies (a).

Suppose that for every for every small F-coalgebra $\langle C', \gamma'\rangle$, and any $c' \in C'$, there is exactly one $c \in C$ such that c' and c are bisimilar among $\langle C', \gamma'\rangle$ and $\langle C, \gamma\rangle$.

Assume an F-coalgebra $\langle C'', \gamma''\rangle$.

For every small subcoalgebra $\langle C', \gamma'\rangle$ of $\langle C'', \gamma''\rangle$, let $h_{\langle C', \gamma'\rangle}$ be a class function from $\langle C', \gamma'\rangle$ to $\langle C, \gamma\rangle$ such that for any $c' \in C'$, $h_{\langle C', \gamma'\rangle}(c')$ is the unique $c \in C$ such that c' and c are bisimilar among $\langle C', \gamma'\rangle$ and $\langle C, \gamma\rangle$.

Let h be a binary class relation between $\langle C'', \gamma''\rangle$ and $\langle C, \gamma\rangle$ such that

$$\mathsf{graph}\, h = \bigcup \{\mathsf{graph}\, h_{\langle C', \gamma'\rangle} \mid \langle C', \gamma'\rangle \text{ is a small subcoalgebra of } \langle C'', \gamma''\rangle\}.$$

We claim that h is a homomorphism from $\langle C'', \gamma''\rangle$ to $\langle C, \gamma\rangle$.

We first need to prove that h is a class function.

Assume $c'' \in C''$.

By Lemma 2.4.8, there is a small F-coalgebra $\langle C', \gamma'\rangle$ such that $c'' \in C''$ and $\langle C', \gamma'\rangle \leq \langle C'', \gamma''\rangle$. Thus, $\langle c'', h_{\langle C', \gamma'\rangle}(c'')\rangle \in \mathsf{graph}\, h$.

Thus, by generalization, $\mathsf{dom}\, h = C''$.

Suppose, toward contradiction, that there are $\langle c'', c_1\rangle, \langle c'', c_2\rangle \in \mathsf{graph}\, h$ such that $c_1 \neq c_2$. Then there is a small subcoalgebra $\langle C'_1, \gamma'_1\rangle$ of $\langle C'', \gamma''\rangle$ such that

$$h_{\langle C'_1, \gamma'_1\rangle}(c'') = c_1,$$

and a small subcoalgebra $\langle C'_2, \gamma'_2\rangle$ of $\langle C'', \gamma''\rangle$ such that

$$h_{\langle C'_2, \gamma'_2\rangle}(c'') = c_2.$$

Thus, there is a bisimulation B_1 between $\langle C'_1, \gamma'_1\rangle$ and $\langle C, \gamma\rangle$ such that $c''\ B_1\ c_1$, and a bisimulation B_2 between $\langle C'_2, \gamma'_2\rangle$ and $\langle C, \gamma\rangle$ such that $c''\ B_2\ c_2$. Now, by Theorem 2.4.6, there is a function $\upsilon : C'_1 \cup C'_2 \to F(C'_1 \cup C'_2)$ such that

$$\langle C'_1 \cup C'_2, \upsilon\rangle \leq \langle C'', \gamma''\rangle.$$

And by Proposition 2.4.7, $C'_1 \hookrightarrow (C'_1 \cup C'_2)$ is a homomorphism from $\langle C'_1, \gamma'_1\rangle$ to $\langle C'_1 \cup C'_2, \upsilon\rangle$, and $C'_2 \hookrightarrow (C'_1 \cup C'_2)$ one from $\langle C'_2, \gamma'_2\rangle$ to $\langle C'_1 \cup C'_2, \upsilon\rangle$.

Let $R_1 = (C'_1 \hookrightarrow (C'_1 \cup C'_2))^{-1} \mathbin{;} B_1$.

Let $R_2 = (C'_2 \hookrightarrow (C'_1 \cup C'_2))^{-1} \mathbin{;} B_2$.

Then, by Corollary 2.2.11, both R_1 and R_2 are bisimulations between $\langle C'_1 \cup C'_2, \upsilon\rangle$ and $\langle C, \gamma\rangle$. And clearly, $c''\ R_1\ c_1$ and $c''\ R_2\ c_2$, contrary to our hypothesis.

Therefore, for every $\langle c'', c_1\rangle, \langle c'', c_2\rangle \in \mathsf{graph}\, h$, $c_1 = c_2$.

Thus, h is a class function from C'' to C.

We move on to prove that h is a homomorphism from $\langle C'', \gamma''\rangle$ to $\langle C, \gamma\rangle$.

Assume $c'' \in C''$.

By Lemma 2.4.8, there is a small F-coalgebra $\langle C', \gamma'\rangle$ such that $c'' \in C'$ and $\langle C', \gamma'\rangle \leq \langle C, \gamma\rangle$.

Let B be a bisimulation between $\langle C', \gamma'\rangle$ and $\langle C, \gamma\rangle$ such that $c''\ B\ h_{\langle C', \gamma'\rangle}(c'')$.

Let $\langle \mathsf{graph}\, B, \beta\rangle$ be an F-coalgebra such that $\mathsf{dpr}\, B$ is a homomorphism from $\langle \mathsf{graph}\, B, \beta\rangle$ to $\langle C', \gamma'\rangle$, and $\mathsf{cpr}\, B$ one from $\langle \mathsf{graph}\, B, \beta\rangle$ to $\langle C, \gamma\rangle$.

Suppose, toward contradiction, that

$$h \circ (C' \hookrightarrow C'') \circ \mathsf{dpr}\, B \neq \mathsf{cpr}\, B.$$

Then there is $\langle c'', c\rangle \in \mathsf{graph}\, B$ such that

$$(h \circ (C' \hookrightarrow C'') \circ \mathsf{dpr}\, B)(\langle c'', c\rangle) \neq (\mathsf{cpr}\, B)(\langle c'', c\rangle).$$

Thus, $h(c'') \neq c$. However, $h(c'') = h_{\langle C', \gamma'\rangle}(c'')$, and thus, $h_{\langle C', \gamma'\rangle}(c'') \neq c$, contrary to our hypothesis.

Therefore,

$$h \circ (C' \hookrightarrow C'') \circ \mathsf{dpr}\, B = \mathsf{cpr}\, B.$$

Then

$$\begin{aligned}
F(h)(\gamma''(c'')) &= F(h)(F(C' \hookrightarrow C'')(\gamma'(c''))) \\
&= F(h)(F(C' \hookrightarrow C'')(\gamma'((\mathsf{dpr}\, B)(\langle c'', h_{\langle C', \gamma'\rangle}(c'')\rangle)))) \\
&= F(h)(F(C' \hookrightarrow C'')(F(\mathsf{dpr}\, B)(\beta(\langle c'', h_{\langle C', \gamma'\rangle}(c'')\rangle)))) \\
&= F(h \circ (C' \hookrightarrow C'') \circ \mathsf{dpr}\, B)(\beta(\langle c'', h_{\langle C', \gamma'\rangle}(c'')\rangle)) \\
&= F(\mathsf{cpr}\, B)(\beta(\langle c'', h_{\langle C', \gamma'\rangle}(c'')\rangle)) \\
&= \gamma((\mathsf{cpr}\, B)(\langle c'', h_{\langle C', \gamma'\rangle}(c'')\rangle)) \\
&= \gamma(h_{\langle C', \gamma'\rangle}(c'')) \\
&= \gamma(h(c'')).
\end{aligned}$$

Thus, by generalization, h is a homomorphism from $\langle C'', \gamma'' \rangle$ to $\langle C, f \rangle$.

Suppose, toward contradiction, that there are homomorphisms h_1 and h_2 from $\langle C'', \gamma'' \rangle$ to $\langle C, \gamma \rangle$ such that $h_1 \neq h_2$. Then there is $c'' \in C''$ such that

$$h_1(c'') \neq h_2(c'').$$

And by Lemma 2.4.8, there is a small F-coalgebra $\langle C', \gamma' \rangle$ such that $c'' \in C'$ and $\langle C', \gamma' \rangle \leq \langle C, \gamma \rangle$. By Theorem 2.3.2, both $h_1 \circ (C' \hookrightarrow C'')$ and $h_2 \circ (C' \hookrightarrow C'')$ are bisimulations between $\langle C', \gamma' \rangle$ to $\langle C, \gamma \rangle$. Thus, c'' and $h_1(c'')$ are bisimilar among $\langle C', \gamma' \rangle$ and $\langle C, \gamma \rangle$, and c'' and $h_2(c'')$ are bisimilar among $\langle C', \gamma' \rangle$ and $\langle C, \gamma \rangle$, contrary to our hypothesis.

Therefore, there is at most one homomorphism from $\langle C'', \gamma'' \rangle$ to $\langle C, \gamma \rangle$.

Thus, there is exactly one homomorphism from $\langle C'', \gamma'' \rangle$ to $\langle C, \gamma \rangle$.

Thus, by generalization, $\langle C, \gamma \rangle$ is final in F-**Coalg**. □

The equivalence of (a) and (b) was already sketched by Aczel in [2], whereas the implication from (a) to (c) can be found in [23, thm. 6.4]. But the one in the reverse direction is, to our knowledge, a new result. Altogether, Theorem 2.4.13 is a powerful characterization of final coalgebras, justifying their prominent role as semantic models of behaviour.

2.5 Existence of final coalgebras

At this point, there are two questions left for us to answer. First, is there a final L-labelled transition coalgebra? And second, if there is one, is it a coalgebraic representation of an L-labelled transition system?

The answer to the first question was already contained in the *Final Coalgebra Theorem* of [2], which has since been generalized to assert the existence of a final coalgebra for every endofunctor on **Class**.

Theorem 2.5.1. *There is an F-coalgebra that is final in F-***Coalg***.*

Proof. See [5, thm. 2.1] and [7, thm. 2.2]. □

By now, there have been several different proofs of Theorem 2.5.1 (see [7] and references therein). Assuming [7, thm. 2.2], or equivalently, Lemma 2.4.8, the proof in [5] is perhaps the most elementary, and surely the most natural from the non-category-theorist point of view. It amounts to forming a direct sum of all small F-coalgebras, and constructing a quotient of it with respect to the largest congruence on it, or equivalently in this case, the equivalence class relation generated by the largest bisimulation on it.

Note 2.5.1. In [5], Aczel and Mendler constructed the quotient of a coproduct of all small coalgebras with respect to the largest congruence on it, which, by Proposition 2.3.3.2, is also the largest precongruence on it. That we can use the equivalence class relation generated by the largest bisimulation on it instead is not entirely obvious, but nevertheless, true.

Proposition 2.5.1.1. *If for every small F-coalgebra $\langle C', \gamma' \rangle$, there is a homomorphism from $\langle C', \gamma' \rangle$ to $\langle C, \gamma \rangle$, then for every $c_1, c_2 \in C$, the following are equivalent:*

(a) c_1 and c_2 are congruent in $\langle C, \gamma \rangle$;

(b) there is $c \in C$ such that c_1 and c are bisimilar in $\langle C, \gamma \rangle$, and c and c_2 are bisimilar in $\langle C, \gamma \rangle$.

Proof. Suppose that for every small F-coalgebra $\langle C', \gamma' \rangle$, there is a homomorphism from $\langle C', \gamma' \rangle$ to $\langle C, \gamma \rangle$.

Assume $c_1, c_2 \in C$.

Suppose that c_1 and c_2 are congruent in $\langle C, \gamma \rangle$.

Then there is a congruence P on $\langle C, \gamma \rangle$ such that $c_1 \; P \; c_2$.

By Lemma 2.4.8, there is a small F-coalgebra $\langle C', \gamma' \rangle$ such that $\{c_1, c_2\} \subseteq C'$ and $\langle C', \gamma' \rangle \leq \langle C, \gamma \rangle$.

Let q be a quotient of C with respect to P.

By [5, lem. 5.1], there is a class function $\delta : \mathsf{cod}\, q \to F(\mathsf{cod}\, q)$ such that q is an epimorphism from $\langle C, \gamma \rangle$ to $\langle \mathsf{cod}\, q, \delta \rangle$. By Theorem 2.4.5, there is a class function $\rho : \mathsf{ran}(q \circ (C' \hookrightarrow C)) \to F(\mathsf{ran}(q \circ (C' \hookrightarrow C)))$ such that

$$\langle \mathsf{ran}(q \circ (C' \hookrightarrow C)), \rho \rangle \leq \langle \mathsf{cod}\, q, \delta \rangle.$$

By Proposition 2.4.7, there is exactly one homomorphism h from $\langle C', \gamma' \rangle$ to $\langle \mathsf{ran}(q \circ (C' \hookrightarrow C)), \rho \rangle$ such that

$$q \circ (C' \hookrightarrow C) = (\mathsf{ran}(q \circ (C' \hookrightarrow C)) \hookrightarrow \mathsf{cod}\, q) \circ h.$$

And clearly, $h(c_1) = h(c_2)$.

Assume $c' \in C'$.

Since $\langle C', \gamma' \rangle$ is small, $\langle \mathsf{ran}(q \circ (C' \hookrightarrow C)), \rho \rangle$ is small. Thus, by hypothesis, there is a homomorphism h' from $\langle \mathsf{ran}(q \circ (C' \hookrightarrow C)), \rho \rangle$ to $\langle C, \gamma \rangle$.

Let $B = (C' \hookrightarrow C)^{-1} ; (h' \circ h)$.

Then, by Theorem 2.2.10, both B and B^{-1} are bisimulations on $\langle C, \gamma \rangle$. And clearly, $c_1 \; B \; h'(h(c_1))$ and $h'(h(c_2)) \; B^{-1} \; c_2$. However, $h(c_1) = h(c_2)$, and hence, $h'(h(c_1)) = h'(h(c_2))$. Thus, there is $c \in C$, namely $h'(h(c_1))$, such that c_1 and c are bisimilar in $\langle C, \gamma \rangle$, and c and c_2 are bisimilar in $\langle C, \gamma \rangle$.

Conversely, suppose that there is $c \in C$ such that c_1 and c are bisimilar in $\langle C, \gamma \rangle$, and c and c_2 are bisimilar in $\langle C, \gamma \rangle$.

Then, by [5, prop. 6.1], c_1 and c are congruent in $\langle C, \gamma \rangle$, and c and c_2 are congruent in $\langle C, \gamma \rangle$. Thus, by [23, lem. 4.16] there is a congruence P on $\langle C, \gamma \rangle$ such that $c_1 \; P \; c$ and $c \; P \; c_2$, and hence, $c_1 \; P \; c_2$. Thus, c_1 and c_2 are congruent in $\langle C, \gamma \rangle$.

Thus, by generalization, (a) and (b) are equivalent. □

By [5, lem. 4.3], there is a largest congruence, and by [56, cor. 5.6], a largest bisimulation on $\langle C, \gamma \rangle$. What Proposition 2.5.1.1 then implies is that if for every small F-coalgebra $\langle C', \gamma' \rangle$, there is a homomorphism from $\langle C', \gamma' \rangle$ to $\langle C, \gamma \rangle$, as is the case with any coproduct of all small F-coalgebras, then the largest congruence on $\langle C, \gamma \rangle$ coincides with the equivalence class relation generated by the largest bisimulation on $\langle C, \gamma \rangle$.

Note that according to Rutten, for every F-coalgebra, the largest bisimulation on it is an equivalence class relation (see [56, cor. 5.6]). This is false. For an easy counterexample, consider the direct sum of the F-coalgebras $\langle S_1, \gamma_1 \rangle$ and $\langle S_2, \gamma_2 \rangle$ of Example 2.3.3.

We could also forgo some of the generality of Theorem 2.5.1, in favour of an even simpler final coalgebra construction, one that would be, for instance, more amenable to formal reasoning.

For example, if we assume that F is an *ω^{op}-continuous* endofunctor on **Class**, namely one that preserves limits of ω^{op}-indexed diagrams in **Class**, and let 1 be a terminal object of **Class**, or equivalently, a set with just one member, and ! the unique class function from $F(1)$ to 1, then we can fashion a final F-coalgebra out of a limit of the following diagram (see [60, lem. 2]):

$$\begin{tikzcd} 1 & F(1) \arrow[l, "!"'] & F(F(1)) \arrow[l, "F(!)"'] & \cdots \arrow[l, "F(F(!))"'] \end{tikzcd}$$

Another, fascinating example is *The Special Final Coalgebra Theorem* of [2], which, under the hypothesis of a universe of sets that need not be well founded, asserts that if F *preserves inclusion maps*, and is *uniform on maps*, a condition capturing the informal idea of an endofunctor whose action on class functions is completely determined by its action on classes, then the largest fixed point of the object part of F, which is readily obtained, in this case, as

$$\bigcup\{S \mid S \text{ is a subset of } F(S)\},$$

together with the identity map on it, is a final F-coalgebra.

All the same, it is only the existence of a final F-coalgebra that will be of any interest to us here.

The answer to the second question is suggested by *Lambek's Lemma*.

Proposition 2.5.2. *If $\langle C, \gamma\rangle$ is final in F-**Coalg**, then γ is bijective.*

Proof. See [33, lem. 2.2]. □

If $\langle C, \tau\rangle$ is final in $(\mathsf{Pow} \circ (L \times \mathsf{Id}))$-**Coalg**, then τ is a bijective class function from C to $\mathsf{Pow}(L \times C)$, which, for obvious cardinality reasons, is possible only if C is a proper class. Thus, even though there is a final L-labelled transition coalgebra, it, being large, cannot be a coalgebraic representation of an L-labelled transition system, and consequently, there is no L-labelled transition system satisfying both conditions of our enquiry.

Although perhaps disconcerting, this is not something that should come as a complete surprise. Having left the cardinality of the branching degree of a state in a system unchecked, it is only reasonable to expect that the different types of branching structure be too many to collect inside a set. One could, for example, bound, in a suitable sense, the endofunctor F, to ensure that a final coalgebra be small (see [31, cor. 3.3]). In the case of $\mathsf{Pow} \circ (L \times \mathsf{Id})$, this would correspond to bounding that branching degree cardinality, and for a model of process behaviour, this may or may not be a natural thing to do. But in the case of Pow, and for a model of a theory of sets, for example, it is definitely not.

Here, we will feel comfortable working with large coalgebras, and refrain from imposing additional constraints, just for the sake of size. Indeed, the very reason that we have decided to work within a theory of classes in the first place was not having to worry about size at all. If we want to understand 'behaviour' in terms of branching structure, then it seems inappropriate to constrain that structure by means that only reflect our own preconceptions about how that ‘behaviour’ may come about. And a bound on the branching degree of that structure seems to do just that: reflect our own bias toward a kind of process that “computes” its own evolution using a fixed set of predefined resources.

Before we leave this chapter, we note that the notion of an F-coalgebra can be defined for any endofunctor F on any arbitrary category, concrete or not. See, for example, [57], for an adaptation of the theory to the categories of metric spaces and ordered sets, or [51], for an entirely axiomatic approach.

Chapter 3

Execution Systems

3.1 From transitions to executions

Perhaps the first thing to decide when setting out to develop an observational theory of processes of some kind is what the unit of observation should be. For a theory based on the concept of labelled transition system, this unit is effectively fixed to what can be represented by a single transition: a single action or event. But at that scale of observation, it is only the local properties of the behaviour of a process that carry over to the model. Non-local properties, specifically those concerning infinite executions of the process, do not. For examples of the first kind, one may look at *safety* properties, such as mutual exclusion or deadlock freedom, whereas for examples of the second kind, one may look at *liveness* properties, such as termination or guaranteed service (see [8]).

We might think of a labelled transition system as an *intuitionistic* approach to a model of a process. For an intuitionist, an infinite execution cannot possibly claim existence as a completed totality of actions or events. It is only “a manifold of possibilities open towards infinity; it remains forever in the status of creation, but is not a closed realm of things existing in themselves” (see [63, p. 9]). This rejection of the notion of *actual infinity* is detrimental to the expressiveness of a theory. All limits of partial executions become complete executions in the model, which is the source of *bounded indeterminacy* (see [19, chap. 9]), and the reason behind the well known problems with properties like *fairness* ([50]) or *finite delay* ([30]) (for example, see [45]).

Here, instead, we take what we might say is a more *classical* approach, and fix our unit of observation at the level of a complete execution of a process. And for that, we need a different type of mathematical structure.

Definition 3.1.1. An *execution system* is an ordered pair $\langle S, E\rangle$ such that the following are true:

(a) S is a set;

(b) E is a binary relation between S and $\mathscr{S}\, S$.[1]

Assume an execution system $\langle S, E\rangle$.

We write $s \rhd_E e$ if and only if $s\ E\ e$.

We call any $s \in S$ a *state* of $\langle S, E\rangle$, and any $\langle s, e\rangle \in \mathsf{graph}\, E$ an *execution* of $\langle S, E\rangle$.

Notice that an execution is an ordered pair of a state and a sequence of states, rather than a single sequence of states, what might have seemed a more natural option. And while we do think that there is a certain clarity in distinguishing the starting state of an execution from any subsequent step, this was mainly a choice of mathematical convenience. Its merit will soon become apparent.

The idea of an execution system has been around at least since the early days of temporal logic in computer science, in the form of a type of semantic structure called a *path structure* in [53] (for example, see [52], [1], and [34]). The states of a system would represent the various memory configurations and control locations traversed in the course of a computation of a possibly concurrent program, and the executions those computations permitted by the assumed implementation, and over which the modalities of the logic were to be interpreted. Here, we will need one more thing: labels.

Definition 3.1.2. An *L-labelled execution system* is an ordered pair $\langle S, E\rangle$ such that the following are true:

(a) S is a set;

(b) E is a binary relation between S and $\mathscr{S}(L \times S)$.

Assume an L-labelled execution system $\langle S, E\rangle$.

We write $s \rhd_E e$ if and only if $s\ E\ e$.

We call any $s \in S$ a *state* of $\langle S, E\rangle$, and any $\langle s, e\rangle \in \mathsf{graph}\, T$ an *execution* of $\langle S, E\rangle$.

Despite the rich cross-fertilization of ideas between temporal logic and process algebra, and the obvious parallel between semantic structures and labelled systems in the two fields, path structures were never really assimilated for use as models of process theories. In fact, the concept of labelled execution system is almost absent from the literature. Looking back

[1] For every set S, we write $\mathscr{S}\, S$ for the set of all finite and infinite sequences over S.

to it, we could only find a handful of sporadic instances of the general notion. We discuss them at the end of this chapter, where we also offer our own opinion on this surprising omission.

3.2 From systems to coalgebras and back

As our choice of formalization should have made obvious, the concept of labelled execution system is a direct generalization of that of labelled transition system. The idea of a single step from one state to another is replaced by that of an "admissible" path through the system over which a sequence of steps can be taken. The result is a more elaborate notion of branching structure. And if we are to associate this notion with 'behaviour' of some kind, we need to understand what constitutes similarity and dissimilarity of it. In other words, we need a concept of branching equivalence suited to labelled execution systems. What should that concept be?

In [57], Rutten and Turi propose a simple approach to this type of problem: all we have to do is find a suitable endofunctor to represent our systems coalgebraically. We can then use that endofunctor to instantiate the "parametric" concept of bisimulation of Definition 2.2.8, and obtain not only the equivalence concept that we seek, but a model too that is *fully abstract* with respect to that concept (see Theorem 2.4.13 and 2.5.1).[2] This is straightforward here.

We write Seq for an endofunctor on **Class** that assigns to every class C the class

$$\mathsf{Seq}\, C = \{s \mid \text{there is a subset } S \text{ of } C \text{ such that } s \in \mathscr{S}\, S\},$$

and to every class function $f : C_1 \to C_2$ a class function

$$\mathsf{Seq}\, f : \mathsf{Seq}\, C_1 \to \mathsf{Seq}\, C_2$$

such that for every $s \in \mathsf{Seq}\, C_1$,

$$(\mathsf{Seq}\, f)(s) = \begin{cases} \langle\ \rangle & \text{if } s = \langle\ \rangle; \\ \langle f(\mathsf{head}\, s)\rangle \cdot (\mathsf{Seq}\, f)(\mathsf{tail}\, s) & \text{otherwise.} \end{cases}$$

Notice that if the class C is a actually a set, then

$$\mathsf{Seq}\, C = \mathscr{S}\, C.$$

[2] The tacit assumption here is that the instantiated concept of bisimulation does indeed induce an equivalence concept. See Example 2.3.3 for a case where it does not.

At this point, the reader may protest against the seeming circularity in the way we have specified the action of Seq on class functions; what may look like a harmless definition by recursion is really a descending argument over a possibly infinite deduction sequence: there is no *base case*. A little thought, however, will suffice to convince oneself that there is nothing ambiguous about it. $\mathsf{Seq}\, f$ is a simple *lift* of f to sequences over $\mathsf{dom}\, f$. Informally, if $\langle c_0, c_1, \ldots \rangle$ is a sequence over $\mathsf{dom}\, f$, then $(\mathsf{Seq}\, f)(\langle c_0, c_1, \ldots \rangle)$ is the result of replacing each c_i in that sequence with its own image under f, namely the sequence $\langle f(c_0), f(c_1), \ldots \rangle$. We are thus entitled to use this contentious form of specification as a definition. The question is how do we justify it formally.

In principle, we could use induction on the index of a sequence to prove that each point in the sequence is uniquely determined. But we can do better.

First, notice that $\mathsf{Seq}\, C_2$ can be given the structure of a $(\{\langle\ \rangle\} + (C_2 \times \mathsf{Id}))$-coalgebra, where $\{\langle\ \rangle\} + (C_2 \times \mathsf{Id})$ is the composite of the left sum endofunctor $\{\langle\ \rangle\} + \mathsf{Id}$ on **Class** with the left product endofunctor $C_2 \times \mathsf{Id}$ on **Class**: simply let σ_2 be a class function from $\mathsf{Seq}\, C_2$ to $\{\langle\ \rangle\} + (C_2 \times \mathsf{Seq}\, C_2)$ such that for every $s \in \mathsf{Seq}\, C_2$,

$$\sigma_2(s) = \begin{cases} (\mathsf{inj}_1(\{\langle\ \rangle\} + (C_2 \times \mathsf{Seq}\, C_2)))(\langle\ \rangle) & \text{if } s = \langle\ \rangle; \\ (\mathsf{inj}_2(\{\langle\ \rangle\} + (C_2 \times \mathsf{Seq}\, C_2)))(\langle \mathsf{head}\, s, \mathsf{tail}\, s \rangle) & \text{otherwise.} \end{cases}$$

Now, we can use f to give $\mathsf{Seq}\, C_1$ the structure of a $(\{\langle\ \rangle\} + (C_2 \times \mathsf{Id}))$-coalgebra as well: just let σ_1 be a class function from $\mathsf{Seq}\, C_1$ to $\{\langle\ \rangle\} + (C_2 \times \mathsf{Seq}\, C_1)$ such that for every $s \in \mathsf{Seq}\, C_1$,

$$\sigma_1(s) = \begin{cases} (\mathsf{inj}_1(\{\langle\ \rangle\} + (C_2 \times \mathsf{Seq}\, C_1)))(\langle\ \rangle) & \text{if } s = \langle\ \rangle; \\ (\mathsf{inj}_2(\{\langle\ \rangle\} + (C_2 \times \mathsf{Seq}\, C_1)))(\langle f(\mathsf{head}\, s), \mathsf{tail}\, s \rangle) & \text{otherwise.} \end{cases}$$

All our definition says then is that $\mathsf{Seq}\, f$ is a homomorphism from $\langle \mathsf{Seq}\, C_1, \sigma_1 \rangle$ to $\langle \mathsf{Seq}\, C_2, \sigma_2 \rangle$. And the existence and uniqueness of this homomorphism follows from the fact that $\langle \mathsf{Seq}\, C_2, \sigma_2 \rangle$ is actually final in $(\{\langle\ \rangle\} + (C_2 \times \mathsf{Id}))$-**Coalg**, as the reader may wish to prove.

With experience, all this can be inferred immediately by simple inspection of the form of the defining equations. More importantly, the same type of argument can be applied to the case of any endofunctor, even if there is no readily available representation of a final coalgebra amenable to inductive reasoning (see [44]). A definition that relies in this way on the finality of the implicit target coalgebra is what we call a definition by *corecursion*. This particular use of the term appears to have originated with [11], and is justified by the duality to the more familiar notion of definition by *recursion*, which, in similar fashion, relies on the initiality of the implicit source algebra (see [28]).

We can now compose Seq with the left product endofunctor $L \times \mathsf{Id}$ on **Class** to obtain the endofunctor $\mathsf{Seq} \circ (L \times \mathsf{Id})$ on **Class**, which assigns to every class C the class

$$\mathsf{Seq}(L \times C) = \{s \mid \text{there is a subset } S \text{ of } L \times C \text{ such that } s \in \mathscr{S}\, S\},$$

and to every class function $f : C_1 \to C_2$ a class function

$$\mathsf{Seq}(L \times f) : \mathsf{Seq}(L \times C_1) \to \mathsf{Seq}(L \times C_2)$$

such that for every $s \in \mathsf{Seq}(L \times C_1)$,

$$(\mathsf{Seq}(L \times f))(s) = \begin{cases} \langle\ \rangle & \text{if } s = \langle\ \rangle; \\ \langle\langle \mathsf{first\,head}\, s, f(\mathsf{sec\,head}\, s)\rangle\rangle \cdot (\mathsf{Seq}(L \times f))(\mathsf{tail}\, s) & \text{otherwise.}^{3} \end{cases}$$

This is another instance of a definition by corecursion. Informally, if $\langle\langle l_0, c_0\rangle, \langle l_1, c_1\rangle, \ldots\rangle$ is a sequence over $\mathsf{dom}\, f$, then

$$\mathsf{Seq}(L \times f)(\langle\langle l_0, c_0\rangle, \langle l_1, c_1\rangle, \ldots\rangle) = \langle\langle l_0, f(c_0)\rangle, \langle l_1, f(c_1)\rangle, \ldots\rangle.$$

Finally, we can compose Pow with $\mathsf{Seq} \circ (L \times \mathsf{Id})$ to obtain the endofunctor $\mathsf{Pow} \circ \mathsf{Seq} \circ (L \times \mathsf{Id})$ on **Class**, which assigns to every class C the class

$$\mathsf{Pow\,Seq}(L \times C) = \{S \mid S \text{ is a subset of } \mathsf{Seq}(L \times C)\},$$

and to every class function $f : C_1 \to C_2$ a class function

$$\mathsf{Pow\,Seq}(L \times f) : \mathsf{Pow\,Seq}(L \times C_1) \to \mathsf{Pow\,Seq}(L \times C_2)$$

such that for every $S \in \mathsf{Pow\,Seq}(L \times C_1)$,

$$(\mathsf{Pow\,Seq}(L \times f))(S) = \{(\mathsf{Seq}(L \times f))(s) \mid s \in S\}.$$

Just as we did with transition systems, we can take now advantage of the formal analogy between the concepts of binary relation and set-valued function, and use Proposition 2.2.1 to obtain our coalgebraic representation. This unity of treatment is the reward of our aforementioned formalization choice.

By Proposition 2.2.1 then, an L-labelled execution system $\langle S, E\rangle$ can be represented as a $(\mathsf{Pow} \circ \mathsf{Seq} \circ (L \times \mathsf{Id}))$-coalgebra, namely as $\langle S, \mathsf{fun}\, E\rangle$, and conversely, a $(\mathsf{Pow} \circ \mathsf{Seq} \circ (L \times \mathsf{Id}))$-coalgebra $\langle C, \varepsilon\rangle$ can be represented as an L-labelled execution system, namely as $\langle C, \mathsf{rel}\, \varepsilon\rangle$, again with the caveat that C be a set.

[3] For every ordered pair $\langle x, y\rangle$, we write $\mathsf{first}\, \langle x, y\rangle$ for x, and $\mathsf{sec}\, \langle x, y\rangle$ for y.

Assume a $(\mathsf{Pow} \circ \mathsf{Seq} \circ (L \times \mathsf{Id}))$-coalgebra $\langle C, \varepsilon \rangle$.

We call $\langle C, \varepsilon \rangle$ an *L-labelled execution coalgebra.*

We write $c \rhd_\varepsilon e$ if and only if $e \in \varepsilon(c)$.

Assume an L-labelled execution system $\langle S, E \rangle$.

The following is immediate:

Proposition 3.2.1. *The following are true:*

(a) $s \rhd_T e$ if and only if $s \rhd_{\mathsf{fun}\, T} e$;

(b) if $\langle C, \varepsilon \rangle$ is small, then $c \rhd_\varepsilon e$ if and only if $c \rhd_{\mathsf{rel}\, \varepsilon} e$.

At this stage, we could already use Proposition 3.2.4 below as our definition of bisimulation between labelled execution systems. But we prefer a different, more operational one that will help us develop some insight into the concept. And for this, we need some more preparation.

Assume a binary class relation $R : C_1 \leftrightarrow C_2$.

We write $\mathsf{Seq}(L \times R)$ for a binary class relation between $\mathsf{Seq}(L \times C_1)$ and $\mathsf{Seq}(L \times C_2)$ such that for every $e_1 \in \mathsf{Seq}(L \times C_1)$ and every $e_2 \in \mathsf{Seq}(L \times C_2)$,

$$\begin{array}{lcl} e_1 \; \mathsf{Seq}(L \times R) \; e_2 & \Longleftrightarrow & \text{there is } e \in \mathsf{Seq}(L \times \mathsf{graph}\, R) \\ & & \text{such that } e_1 = \mathsf{Seq}(L \times \mathsf{dpr}\, B)(e) \\ & & \text{and } e_2 = \mathsf{Seq}(L \times \mathsf{cpr}\, B)(e). \end{array}$$

$\mathsf{Seq}(L \times R)$ is a simple lift of R to pairs of sequences over $\mathsf{dom}\, R$ and $\mathsf{cod}\, R$. Informally, if $\langle c_0, c_1, \ldots \rangle$ is a sequence over $\mathsf{dom}\, R$, and $\langle c'_0, c'_1, \ldots \rangle$ a sequence over $\mathsf{cod}\, R$, then

$$\langle c_0, c_1, \ldots \rangle \; \mathsf{Seq}(L \times R) \; \langle c'_0, c'_1, \ldots \rangle$$

if and only if $c_0 \; R \; c'_0$, $c_1 \; R \; c'_1$, etc. This is, of course, an abuse of notation. But it is a very mild form of abuse. For if R is actually a class function, then $\mathsf{Seq}(L \times R)$ is the same as the image of R under $\mathsf{Seq} \circ (L \times \mathsf{Id})$.

Assume L-labelled execution coalgebras $\langle C_1, \varepsilon_1 \rangle$ and $\langle C_2, \varepsilon_2 \rangle$.

Proposition 3.2.2. *B is a bisimulation between $\langle C_1, \varepsilon_1 \rangle$ and $\langle C_2, \varepsilon_2 \rangle$ if and only if B is a binary class relation between $\langle C_1, \varepsilon_1 \rangle$ and $\langle C_2, \varepsilon_2 \rangle$, and for any c_1 and c_2 such that $c_1 \; B \; c_2$, the following are true:*

(a) if $c_1 \rhd_{\varepsilon_1} e_1$, then there is e_2 such that $c_2 \rhd_{\varepsilon_2} e_2$ and $e_1 \; \mathsf{Seq}(L \times B) \; e_2$;

(b) *if* $c_2 \rhd_{\varepsilon_2} e_2$*, then there is* e_1 *such that* $c_1 \rhd_{\varepsilon_1} e_1$ *and* $e_1\ \mathsf{Seq}(L \times B)\ e_2$*.*

Proof. Suppose that B is a bisimulation between $\langle C_1, \varepsilon_1\rangle$ and $\langle C_2, \varepsilon_2\rangle$.

Let $\langle \mathsf{graph}\, B, \beta\rangle$ be an L-labelled execution coalgebra such that $\mathsf{dpr}\, B$ is a homomorphism from $\langle \mathsf{graph}\, B, \beta\rangle$ to $\langle C_1, \varepsilon_1\rangle$, and $\mathsf{cpr}\, B$ one from $\langle \mathsf{graph}\, B, \beta\rangle$ to $\langle C_2, \varepsilon_2\rangle$.

Assume c_1 and c_2 such that $c_1\ B\ c_2$.

Then

$$\mathsf{Pow}\,\mathsf{Seq}(L \times \mathsf{dpr}\, B)(\beta(\langle c_1, c_2\rangle)) = \varepsilon_1((\mathsf{dpr}\, B)(\langle c_1, c_2\rangle)),$$

and hence, by definition of $\mathsf{Pow} \circ \mathsf{Seq} \circ (L \times \mathsf{Id})$ and $\mathsf{dpr}\, B$,

$$\{\mathsf{Seq}(L \times \mathsf{dpr}\, B)(e) \mid e \in \beta(\langle c_1, c_2\rangle)\} = \varepsilon_1(c_1).$$

By extensionality, this is equivalent to the following being true:

(i) if $\langle c_1, c_2\rangle \rhd_\beta e$, then $c_1 \rhd_{\varepsilon_1} \mathsf{Seq}(L \times \mathsf{dpr}\, B)(e)$;

(ii) if $c_1 \rhd_{\varepsilon_1} e_1$, then there is e such that $\mathsf{Seq}(L \times \mathsf{dpr}\, B)(e) = e_1$ and $\langle c_1, c_2\rangle \rhd_\beta e$.

And by symmetry, the following are true:

(iii) if $\langle c_1, c_2\rangle \rhd_\beta e$, then $c_2 \rhd_{\varepsilon_2} \mathsf{Seq}(L \times \mathsf{cpr}\, B)(e)$;

(iv) if $c_2 \rhd_{\varepsilon_2} e_2$, then there is e such that $\mathsf{Seq}(L \times \mathsf{cpr}\, B)(e) = e_2$ and $\langle c_1, c_2\rangle \rhd_\beta e$.

By (ii), (iii), and definition of $\mathsf{Seq}(L \times B)$, (a) is true, and by (i), (iv), and definition of $\mathsf{Seq}(L \times B)$, (b) is true.

Thus, by generalization, for any c_1 and c_2 such that $c_1\ B\ c_2$, (a) and (b) are true.

Conversely, suppose that B is a binary class relation between C_1 and C_2, and for any c_1 and c_2 such that $c_1\ B\ c_2$, (a) and (b) are true.

Let β be a class function from $\mathsf{graph}\, B$ to $\mathsf{Pow}\,\mathsf{Seq}(L \times \mathsf{graph}\, B)$ such that for any $\langle c_1, c_2\rangle \in \mathsf{graph}\, B$,

$$\begin{aligned}\beta(\langle c_1, c_2\rangle) = \{e \mid\ & c_1 \rhd_E \mathsf{Seq}(L \times \mathsf{dpr}\, B)(e),\\ & c_2 \rhd_E \mathsf{Seq}(L \times \mathsf{cpr}\, B)(e),\\ & \text{and } e \in \mathsf{Seq}(L \times \mathsf{graph}\, B)\}.\end{aligned}$$

Assume $\langle c_1, c_2\rangle \in \mathsf{graph}\, B$.

Then the following is immediately true:

(v) if $\langle c_1, c_2\rangle \rhd_\beta e$, then $c_1 \rhd_{\varepsilon_1} \mathsf{Seq}(L \times \mathsf{dpr}\, B)(e)$.

Also, by (a), (b), and the definition of $\mathsf{Seq}(L \times B)$, the following is true:

(vi) if $c_1 \rhd_{\varepsilon_1} e_1$, then there is e such that $\mathsf{Seq}(L \times \mathsf{dpr}\, B)(e) = e_1$ and $\langle c_1, c_2\rangle \rhd_\beta e$.

By (v), (vi), and extensionality,

$$\{\mathsf{Seq}(L \times \mathsf{dpr}\, B)(e) \mid e \in \beta(\langle c_1, c_2\rangle)\} = \varepsilon_1(c_1).$$

and hence, by definition of $\mathsf{Pow} \circ \mathsf{Seq} \circ (L \times \mathsf{Id})$ and $\mathsf{dpr}\, B$,

$$\mathsf{Pow}\,\mathsf{Seq}(L \times \mathsf{dpr}\, B)(\beta(\langle c_1, c_2\rangle)) = \varepsilon_1((\mathsf{dpr}\, B)(\langle c_1, c_2\rangle)).$$

And by symmetry,

$$\mathsf{Pow}\,\mathsf{Seq}(L \times \mathsf{cpr}\, B)(\beta(\langle c_1, c_2\rangle)) = \varepsilon_2((\mathsf{cpr}\, B)(\langle c_1, c_2\rangle)).$$

Thus, by generalization, B is a bisimulation between $\langle C_1, \varepsilon_1\rangle$ and $\langle C_2, \varepsilon_2\rangle$. □

Proposition 3.2.2 has the same feel as Proposition 2.2.14. The difference is that the local check of correspondence of transitions has been replaced by a non-local test of agreement along entire executions. This is conceptually in tune with our intended change in scale of observation from one type of system to the other.

Assume L-labelled execution systems $\langle S_1, E_1\rangle$ and $\langle S_2, E_2\rangle$.

Definition 3.2.3. A *bisimulation between* $\langle S_1, E_1\rangle$ *and* $\langle S_2, E_2\rangle$ is a binary relation $B : S_1 \leftrightarrow S_2$ such that for any s_1 and s_2 such that $s_1\ B\ s_2$, the following are true:

(a) if $s_1 \rhd_{E_1} e_1$, then there is e_2 such that $s_2 \rhd_{E_2} e_2$ and $e_1\ \mathsf{Seq}(L \times B)\ e_2$;

(b) if $s_2 \rhd_{E_2} e_2$, then there is e_1 such that $s_1 \rhd_{E_1} e_1$ and $e_1\ \mathsf{Seq}(L \times B)\ e_2$.

The following is of course immediate:

Proposition 3.2.4. *B is a bisimulation between $\langle S_1, E_1\rangle$ and $\langle S_2, E_2\rangle$ if and only if B is a bisimulation between the L-labelled execution coalgebras $\langle S_1, \mathsf{fun}\, E_1\rangle$ and $\langle S_2, \mathsf{fun}\, E_2\rangle$.*

We say that B *is a bisimulation on* $\langle S, E\rangle$ if and only if B is a bisimulation between $\langle S, E\rangle$ and $\langle S, E\rangle$.

We say that B *is a bisimulation equivalence on* $\langle S, E\rangle$ if and only if B is a bisimulation on $\langle S, E\rangle$, and an equivalence relation on S.

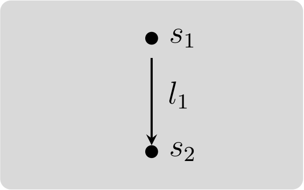
We say that s_1 *and* s_2 *are bisimilar among* $\langle S_1, E_1\rangle$ *and* $\langle S_2, E_2\rangle$ if and only if there is a bisimulation B between $\langle S_1, E_1\rangle$ and $\langle S_2, E_2\rangle$ such that $s_1\ B\ s_2$.

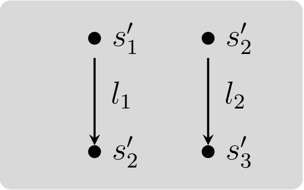
We say that s_1 *and* s_2 *are bisimilar in* $\langle S, E\rangle$ if and only if s_1 and s_2 are bisimilar among $\langle S, E\rangle$ and $\langle S, E\rangle$.

3.3 Abrahamson systems

Informally, we can explain bisimilarity of states of labelled execution systems in the same way as we did in the case of labelled transition systems. Only now, paths are not implicitly inferred from a transition relation, but explicitly stipulated as part of the system structure. And this can have some peculiar side effects.

For example, consider two $\{l_1, l_2\}$-labelled execution systems, whose executions are as depicted in the following left and right frames respectively:

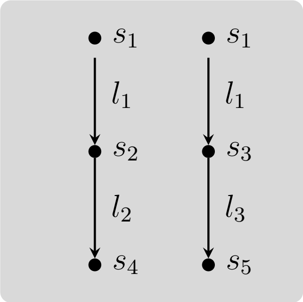


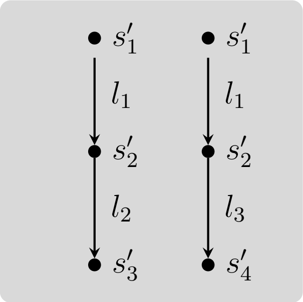


Then s_1 and s'_1 are not bisimilar among the two systems, simply because the execution starting from s'_2 carries a different label than the one starting from s_1. But why should we care if it does? The only execution starting from s'_1 has only one step, labelled l_1, and is in perfect agreement with the only execution starting from s_1, which also has only one step, also labelled l_1. So, intuitively, there is no difference in branching potential between the two states. We must therefore conclude that bisimilarity is, in this case, inconsistent with our informal sense of equivalence of branching structure.

A plausible remedy for this would, informally, be the following: for any path beginning at a given state, discount any branch off that path that is not a suffix of another path beginning at that same state. And indeed, this would work for this particular case. But there are more problems.

Consider two $\{l_1, l_2\}$-labelled execution systems, whose executions are as depicted in the following left and right frames respectively:





Then s_1 and s'_1 are bisimilar among the two systems. But intuitively, there is difference in branching potential between the two states: the two executions starting from s_1 diverge right away at s_1, with steps that carry identical labels, whereas those starting from s'_1 diverge after the first step, at s'_2, with steps that carry different labels. Of course, the explanation here is that there is no execution starting from s'_2, and so, conceptually, the choice between the diverging steps is already made at s'_1. But then, what is the point of having the two executions share the state s'_2?

By now, the reader should begin to suspect what the source of our problems is: what we have called "state" in our systems does not really behave as such. In a type of system that is supposed to serve as a modelling device for processes of some kind, it is essential that "the future behavior depends only upon the current state, and not upon how that state was reached". And this is not always the case here. What we need to do is constrain the structure of our systems so that it is.

Of course, this idea is not new. The quote above is from [34, p. 176], where Lamport required that the set of paths in a path structure be *suffix closed*, in the sense that for any path in the set, any suffix of that path is again a path in the set. It was later observed in [20] that this is not enough: one must also require that the set of paths be *fusion closed*, in the sense that for any prefix of a path in the set, and any suffix of another path in the set, if the former ends at the state at which the latter begins, then their fusion at that state is again a path in the set (see [52]). And apparently, it was Abrahamson, in [1], that first considered path structures that satisfied both requirements (see [16]).

We now adapt these requirements to our own setting.

We say that $\langle S, E\rangle$ *is Abrahamson* if and only if the following are true:

(i) for every s, l, s', and e', if $s \rhd_E \langle\langle l, s'\rangle\rangle \cdot e'$, then $s' \rhd_E e'$;

(ii) for every s, l, s', e'_1, and e'_2, if $s \rhd_E \langle\langle l, s'\rangle\rangle \cdot e'_1$ and $s' \rhd_E e'_2$, then $s \rhd_E \langle\langle l, s'\rangle\rangle \cdot e'_2$.

Here, (i) corresponds to suffix closure, and (ii), conditioned on (i), to fusion closure.

In an Abrahamson system, there is a clear notion of a "possible next step" relation, which induces the construction of an associated, or better, underlying labelled transition system. From a mathematical standpoint, this construction makes sense for a non-Abrahamson system as well, and is most conveniently carried out on the coalgebra side of the theory.

Assume a class C.

We write $\eta(C)$ for a class function from $\mathsf{Pow}\,\mathsf{Seq}(L \times C)$ to $\mathsf{Pow}(L \times C)$ such that for every $S \in \mathsf{Pow}\,\mathsf{Seq}(L \times C)$,

$$\eta(C)(S) = \{\mathsf{head}\, s \mid s \in S \text{ and } s \neq \langle\ \rangle\}.$$

Our choice of notation here is not arbitrary. We think of η as an operator that assigns to every class C a class function from its image under $\mathsf{Pow} \circ \mathsf{Seq} \circ (L \times \mathsf{Id})$ to its image under $\mathsf{Pow} \circ (L \times \mathsf{Id})$. And what is interesting about this operator is that for every class function $f : C_1 \to C_2$,

$$\eta(C_2) \circ \mathsf{Pow}\,\mathsf{Seq}(L \times f) = \mathsf{Pow}(L \times f) \circ \eta(C_1),$$

or equivalently, the following diagram commutes:

$$\begin{tikzcd}
\mathsf{Pow}\,\mathsf{Seq}(L \times C_1) \arrow[r, "\eta(C_1)"] \arrow[d, "\mathsf{Pow}\,\mathsf{Seq}(L \times f)"'] & \mathsf{Pow}(L \times C_1) \arrow[d, "\mathsf{Pow}(L \times f)"] \\
\mathsf{Pow}\,\mathsf{Seq}(L \times C_2) \arrow[r, "\eta(C_2)"'] & \mathsf{Pow}(L \times C_2)
\end{tikzcd}$$

In the language of category theory, this makes η a *natural transformation* from $\mathsf{Pow} \circ \mathsf{Seq} \circ (L \times \mathsf{Id})$ to $\mathsf{Pow} \circ (L \times \mathsf{Id})$.

The concept of natural transformation is of great importance in the study and application of category theory. It is defined using the same property as above, only generalized to a pair of arbitrary functors with a common domain category and a common codomain one. Pierce helps us visualize the essence of the concept by inviting us to "imagine 'sliding' the picture defined by [the first functor] onto the picture defined by [the second one]" (see [47, p. 41]).

The reason why it is of interest to us here that η is a natural transformation is a theorem by Rutten, according to which, every natural transformation ν from an endofunctor F_1 on **Class** to an endofunctor F_2 on **Class** induces a functor from F_1-**Coalg** to F_2-**Coalg** that assigns to every F_1-coalgebra $\langle C, \gamma\rangle$ the F_2-coalgebra $\langle C, \eta(C) \circ \gamma\rangle$, and to every homomorphism h from an F_1-coalgebra $\langle C_1, \gamma_1\rangle$ to an F_1-coalgebra $\langle C_2, \gamma_2\rangle$ that same class function h, which is now a homomorphism from the F_2-coalgebra $\langle C_1, \eta(C_1) \circ \gamma_1 \rangle$ to the F_2-coalgebra $\langle C_2, \eta(C_2) \circ \gamma_2 \rangle$ (see [56, thm. 15.1]). In other words, the induced functor preserves homomorphisms, and thus, by Theorem 2.2.10, bisimulations too.

In our case, the functor induced by η is a *forgetful* functor, which, informally, keeps only the first step, if any, from any execution starting from any state, and discards the rest.

The following is immediate from Rutten's theorem, but a more direct proof would require only little extra work:

Proposition 3.3.1. *If h is a homomorphism from $\langle C_1, \varepsilon_1 \rangle$ to $\langle C_2, \varepsilon_2 \rangle$, then h is a homomorphism from the L-labelled transition coalgebra $\langle C_1, \eta(C_1) \circ \varepsilon_1 \rangle$ to the L-labelled transition coalgebra $\langle C_2, \eta(C_2) \circ \varepsilon_2 \rangle$.*

The following is now immediate from Theorem 2.2.10 and Proposition 3.3.1.

Proposition 3.3.2. *If B is a bisimulation between $\langle C_1, \varepsilon_1 \rangle$ and $\langle C_2, \varepsilon_2 \rangle$, then B is a bisimulation between the L-labelled transition coalgebras $\langle C_1, \eta(C_1) \circ \varepsilon_1 \rangle$ and $\langle C_2, \eta(C_2) \circ \varepsilon_2 \rangle$.*

Of course, we can translate all this back to the system side of the theory.

Assume a binary relation $E : S \leftrightarrow \mathscr{P}(L \times S)$.

We write $\mathsf{trans}\, E$ for a binary relation between S and $L \times S$ such that for any $s \in S$ and any $\langle l, s' \rangle \in L \times S$,

$$s\ (\mathsf{trans}\, E)\ \langle l, s' \rangle \iff \text{there is } e \text{ such that } s\ E\ e,\ e \neq \langle\ \rangle, \text{ and } \mathsf{head}\, e = \langle l, s' \rangle.$$

The following is trivial:

Proposition 3.3.3. $\mathsf{trans}\, E = \mathsf{rel}(\eta(S) \circ \mathsf{fun}\, E)$.

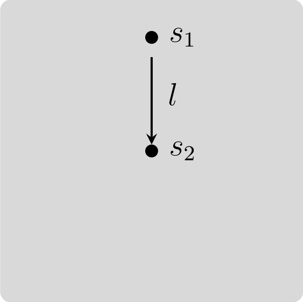
The following is now immediate from Proposition 2.2.1(b), 2.2.15, 3.2.4, 3.3.2, and 3.3.3:

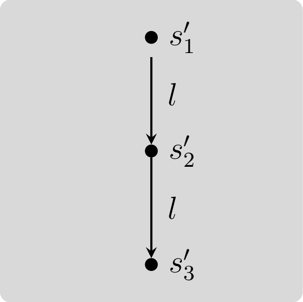
Proposition 3.3.4. *If B is a bisimulation between $\langle S_1, E_1 \rangle$ and $\langle S_2, E_2 \rangle$, then B is a bisimulation between the L-labelled transition systems $\langle S_1, \mathsf{trans}\, E_1 \rangle$ and $\langle S_2, \mathsf{trans}\, E_2 \rangle$.*

3.4 Generable systems

The converse of Proposition 3.3.4 is of course false, as is that of Proposition 3.3.1 and 3.3.2. But it is instructive to see exactly where it fails. We go over it through a series of simple examples.

First, suppose that $\langle S, E\rangle$ and $\langle S', E'\rangle$ are two $\{l\}$-labelled execution systems, whose executions are as depicted in the following left and right frames respectively:

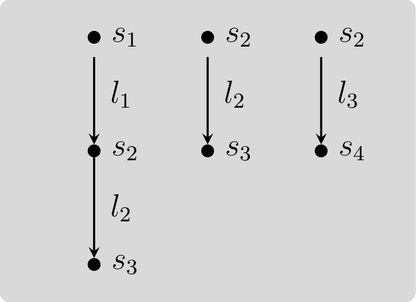


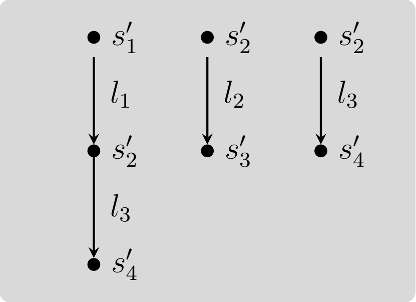


Then s_1 and s_1' are bisimilar among the $\{l\}$-labelled transition systems $\langle S, \mathsf{trans}\, E\rangle$ and $\langle S', \mathsf{trans}\, E'\rangle$, but not among $\langle S, E\rangle$ and $\langle S', E'\rangle$.

The problem is easy to spot here. The two systems have one execution each. But whereas the execution of the first system has only one step, the execution of the second has two. And that second step, which is the cause for s_1 and s_1' not being bisimilar among the two systems, is dropped during the labelled transition system construction.

Now suppose that $\langle S, E\rangle$ and $\langle S', E'\rangle$ are two $\{l_1, l_2, l_3\}$-labelled execution systems, whose executions are as depicted in the following left and right frames respectively:





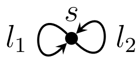
Then s_1 and s_1' are bisimilar among the now $\{l_1, l_2, l_2\}$-labelled transition systems $\langle S, \mathsf{trans}\, E\rangle$ and $\langle S', \mathsf{trans}\, E'\rangle$, but not among $\langle S, E\rangle$ and $\langle S', E'\rangle$.

Here the problem is of a different nature. Every step of every execution is accounted for in the constructed labelled transition systems. However, the two longer executions, starting from s_1 and s_1' respectively, disagree on their second step, and that disagreement is masked by the agreement of executions starting from s_2 and s_2' respectively.

These two examples were specially chosen to target the two defining clauses of the Abrahamson property. Specifically, and informally, the systems in the first example are not suffix closed, thus violating clause (i) of the property, whereas those in the second are not fusion closed, thus violating clause (ii). Overall, none of them is Abrahamson. And since our construction was based on the idea of a "possible next step" relation, which, in the case of a non-Abrahamson system, is a conceptually ambiguous notion, it is no surprise that non-bisimilar states turn bisimilar in the constructed systems.

With Abrahamson systems, things get much more interesting. In the rest of our examples, we shall focus on such systems. And as afforded with such systems, we shall communicate their structure more casually, simply drawing a diagram of the underlying labelled transition system, and describing the set of paths in that diagram that correspond to their executions.

Consider then the $\{l_1, l_2\}$-labelled transition system, with l_1 and l_2 different, portrayed in the following diagram:

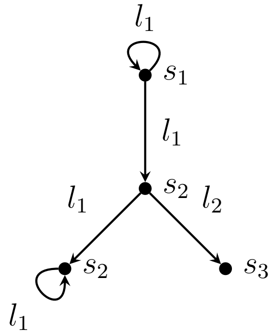


One $\{l_1, l_2\}$-labelled execution system lying over this labelled transition system is the one whose executions correspond to all infinite paths in the diagram. Another is the one whose executions correspond to those infinite paths that go through each of the two loops infinitely often. And of course, s is not bisimilar with itself among the two.

We may think of each of the two $\{l_1, l_2\}$-labelled execution systems in this example as a specification of a scheduling policy between two processes, forever iterating over l_1 and l_2 respectively, on a single processing unit. Under the first policy, the scheduler is only required to guarantee progress of execution, simply picking at random one process at a time. Under the second, it is further required to be *fair*, taking care that there is no point in time after which a process is forever neglected. But whereas its behaviour in the first case is completely specified by the underlying $\{l_1, l_2\}$-labelled transition system, in the second case, it cannot be specified by any $\{l_1, l_2\}$-labelled transition system alone.

Besides demonstrating the failure of the converse of Proposition 3.3.4 for Abrahamson systems, this example attempts to display the increase in expressive power and branching complexity that moving from a labelled transition to a labelled execution system can bring. But it does so inadequately. For one need not really move to a labelled execution system to specify the behaviour of the scheduler under that second policy. One can just augment the given labelled transition system with the set of all infinite sequences over $\{l_1, l_2\}$ corresponding to a fair interleaving of the two processes. And in fact, the concept of bisimulation between labelled transition systems can be generalized to account for this kind of augmentation by simply adding a third clause to Definition 2.1.3 that tests for inclusion between the sets of "admissible" sequences of labels associated with each state. This gives rise to the less known concept of *fortification equivalence*, one of the alternative approaches to the semantics of finite delay considered by Milner in [38], and a perfectly adequate approach to the specification of the two scheduling policies in our example. What we want is another example that will expose the shortcomings of this type of approach, and vindicate our present venture.

Consider then the $\{l_1, l_2\}$-labelled transition system, with l_1 and l_2 different, portrayed in the following diagram:



The first $\{l_1, l_2\}$-labelled execution system that we wish to consider here is the unique Abrahamson system whose executions starting from s_1 correspond to all maximal paths in this diagram. The second is the one whose executions are all the executions of the first, except the single infinite execution stuttering around s_1. And because of this exception, s_1 is not bisimilar with itself among the two systems.

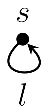
This beautiful example is from [4], where it was used to attack precisely the type of approach discussed above. Here, it is perhaps convenient to think of the two systems as modelling the behaviour of two distinct processes, both initialized at s_1. The first process will either loop around s_1 forever, or iterate through it for a finite, indeterminate number of times before progressing to s_2. From there on, a single indeterminate choice will decide its fate. The second process, on the other hand, is not allowed to loop around s_1 forever. It must eventually advance to s_2, from where on it behaves just like the first one. What sets the behaviour of the two processes apart is, of course, the infinite stuttering around s_1, permitted for the first process, but not the second. However, this is something that cannot be determined by the sequences of actions that the two processes perform in the course of their executions, for the trace of that infinite stuttering is matched by that of every infinite execution that eventually loops around s_2. And yet the two processes ought to be distinguished. For during that infinite stuttering, the first process may always choose to branch off to a state from which it can perform l_2, whereas, in every execution having that trace, the second must eventually reach a state from which it cannot ever do so.

With respect to the failure of the converse of Proposition 3.3.4, both this and the previous example point at the same problem: the existence of an infinite path in the diagram that does not correspond to any execution of a system, but whose every finite prefix is a prefix of another path that does.

This too is something that has already come up in the investigation of path structures in temporal logic. In [20], Emerson called a set of paths *limit closed* provided that for every infinite, strictly increasing chain of finite prefixes of paths in the set, the limit of that chain, in the standard topology of sequences, is again a path in the set. This property was apparently also first considered in [1]. But it was Emerson in [20] who proved the independence of all three closure properties, and the equivalence of their conjunction to the existence of a transition relation generating the given set of paths. Apart from the absence of labels, which has no bearing in this particular discussion, Emerson's setup was different in that paths were always infinite. But this too is of no importance in our examples, which, in light of Emerson's result, appear to implicate violation of limit closure in the failure of the underlying labelled transition system to subsume all the branching information relevant to a given Abrahamson system.

Our next example is perhaps the most curious one.

Consider the simple $\{l\}$-labelled transition system portrayed in the following diagram:



There are exactly three Abrahamson $\{l\}$-labelled execution systems that one can lay over this labelled transition system. The first is the one whose only execution corresponds to the only infinite path in the diagram. The second is the one whose executions correspond to all finite paths in the diagram. And of course, the third is the one whose executions are all executions of the first and second system. But s is not bisimilar with itself among any two of the three.

Informally, the second system is not limit closed, and this is one part of the problem. But the first and third are, and so there must be something more going on here. The answer is in the difference between Emerson's setup and ours mentioned earlier. Here, executions are not always infinite. In a system that is, informally, suffix closed, if there is a finite execution, then there is an empty execution. And an empty execution creates a type of branching that is impossible to mimic in a labelled transition system.

In an Abrahamson system that is used to model the behaviour of a process, an empty execution can be used to model termination. But if there is another, non-empty execution starting from the same state, then termination becomes a branching choice, one that does not show up in the "possible next step" relation of the system. This feature of *indeterminate termination*, as we might call it, can seem a little odd at first, but is really a highly versatile mechanism, particularly useful in modelling idling in absence of input stimuli, as we shall see in the next chapter.

Finally, consider the labelled transition system of the following, trivial diagram:

There are exactly two labelled execution systems that one can lay over this labelled transition system: one that has one execution, the empty execution, and one that has no execution. And of course, s is not bisimilar with itself among the two.

This degenerate case deserves little comment. We only remark that in a suffix closed system, if a state has no execution starting from it, then it has no execution going through it.

At this point, we have found five possible causes of failure for the converse of Proposition 3.3.4. We have chosen our examples carefully, to examine each of the five separately and independently from one another. And we have observed how each of the first three connects to violation of one of the three closure properties that have been shown to collectively characterize sets of infinite paths generable by a transition relation. But finite paths add another dimension to the problem, rendering Emerson's characterization result obsolete. What we will show next is that impossibility of indeterminate termination, along with a non-triviality condition guarding against the occurrence of an isolated state, can be added to the conditions of suffix, fusion, and limit closure, to produce a complete characterization of system generability, insensitive to the length of the executions.

First, we need to make the notion of generability precise. For generality, we transfer ourselves again to the coalgebra side of the theory.

Assume a class function $\tau : C \to \mathsf{Pow}(L \times C)$.

Assume $c \in C$.

Assume $e \in \mathsf{Seq}(L \times C)$.

We say that e *is a τ-orbit of* c if and only if the following are true:

(i) one of the following is true:

  (1) $\tau(c) = \emptyset$ and $e = \langle\ \rangle$;

  (2) there is l, c', and e' such that $\langle l, c' \rangle \in \tau(c)$ and $e = \langle\langle l, c' \rangle\rangle \cdot e'$;

(ii) for every $n \in \omega$, if $\mathsf{tail}^n\, e \neq \langle\ \rangle$, then one of the following is true:

  (1) there is l and c' such that $\tau(c') = \emptyset$ and $\mathsf{tail}^n\, e = \langle\langle l, c' \rangle\rangle$;

  (2) there is l, c', l', c'', and e'' such that $\langle l', c'' \rangle \in \tau(c')$ and $\mathsf{tail}^n\, e = \langle\langle l, c' \rangle\rangle \cdot \langle\langle l', c'' \rangle\rangle \cdot e''$.

Here again, it is the computational interpretation that is most helpful. If we think of τ as a representation of the control flow graph of a possibly indeterminate sequential program, then a τ-orbit of c corresponds to a total execution of that program starting from the node represented by c.

Now, we would like to say that a class function from C to $\mathsf{Pow}\,\mathsf{Seq}(L \times C))$ is generated by τ just as long as it assigns to any $c \in C$ the set of all τ-orbits of c. But first, we need to make sure that this really is a set, and not a proper class.

We write $\mathrm{W}_\tau(c)$ for a class function from ω to $\mathsf{Pow}(L \times C)$ such that

$$\mathrm{W}_\tau(c)(0) = \tau(c)$$

and for every $n \in \omega$,

$$\mathrm{W}_\tau(c)(n+1) = \bigcup \{\tau(c') \mid \text{there is } l \text{ such that } \langle l, c' \rangle \in \mathrm{W}_\tau(c)(n)\}.$$

We think of $\mathrm{W}_\tau(c)$ as a wave emitted by c, and propagating through $L \times C$ according to τ, and $\mathrm{W}_\tau(c)(n)$ as the wavefront at the nth time instance.

Proposition 3.4.1. *If e is a τ-orbit of c, then for every $n \in \omega$, if $\mathsf{tail}^n\, e \neq \langle\ \rangle$, then $\mathsf{head}\,\mathsf{tail}^n\, e \in \mathrm{W}_\tau(c)(n)$.*

Proof. We use induction.

If $n = 0$, then $\mathsf{tail}^n\, e = e$. Thus, if $\mathsf{tail}^n\, e \neq \langle\ \rangle$, then there is l, c', and e' such that $\langle l, c' \rangle \in \tau(c)$ and

$$\mathsf{tail}^n\, e = \langle \langle l, c' \rangle \rangle \cdot e'.$$

Hence, $\mathsf{head}\,\mathsf{tail}^n\, e \in \mathrm{W}_\tau(c)(n)$.

Otherwise, there is $m \in \omega$ such that $n = m + 1$. Then, if $\mathsf{tail}^n\, e \neq \langle\ \rangle$, then $\mathsf{tail}^m\, e \neq \langle\ \rangle$. Thus, there is l, c', l', c'', and e'' such that $\langle l', c'' \rangle \in \tau(c')$

$$\mathsf{tail}^m\, e = \langle \langle l, c' \rangle \rangle \cdot \langle \langle l', c'' \rangle \rangle \cdot e''.$$

By the induction hypothesis, $\mathsf{head}\,\mathsf{tail}^m\, e \in \mathrm{W}_\tau(c)(m)$, and so, $\langle l, c' \rangle \in \mathrm{W}_\tau(c)(m)$. Thus, $\langle l', c'' \rangle \in \mathrm{W}_\tau(c)(m+1)$, and since

$$\mathsf{head}\,\mathsf{tail}^n\, e = \mathsf{head}\,\mathsf{tail}^{m+1}\, e = \langle l', c'' \rangle,$$

$\mathsf{head}\,\mathsf{tail}^n\, e \in \mathrm{W}_\tau(c)(n)$.

Therefore, for every $n \in \omega$, if $\mathsf{tail}^n\, e \neq \langle\ \rangle$, then $\mathsf{head}\,\mathsf{tail}^n\, e \in \mathrm{W}_\tau(c)(n)$. □

Proposition 3.4.2. *For every $n \in \omega$, $\mathrm{W}_\tau(c)(n)$ is a set.*

Proof. We use induction.

If $n = 0$, then $\mathrm{W}_\tau(c)(n) = \tau(c)$, which, by definition of Pow, is a set.

Otherwise, there is $m \in \omega$ such that $n = m + 1$. By the induction hypothesis, $\mathrm{W}_\tau(c)(m)$ is a set. Then clearly, $\mathrm{W}_\tau(c)(m + 1)$ is a set, and so $\mathrm{W}_\tau(c)(n)$ is a set.

Therefore, for every $n \in \omega$, $\mathrm{W}_\tau(c)(n)$ is a set. □

Proposition 3.4.3. $\{e \mid e \in \mathsf{Seq}(L \times C) \text{ and } e \text{ is a } \tau\text{-orbit of } c\}$ *is a set.*

Proof. By Proposition 3.4.1, for every $e \in \mathsf{Seq}(L \times C)$, if e is a τ-orbit of c, then

$$\mathsf{graph}\, e \subseteq C \times \bigcup\{\mathrm{W}_\tau(c)(n) \mid n \in \omega\}.$$

Thus,

$$\{\mathsf{graph}\, e \mid e \in \mathsf{Seq}(L \times C) \text{ and } e \text{ is a } \tau\text{-orbit of } c\}$$

is a set, and by replacement,

$$\{e \mid e \in \mathsf{Seq}(L \times C) \text{ and } e \text{ is a } \tau\text{-orbit of } c\}$$

is a set. □

Proposition 3.4.2 and 3.4.3 will be taken for granted in the sequel.

We write $\mathsf{gen}\,\tau$ for a class function from C to $\mathsf{Pow}\,\mathsf{Seq}(L \times C)$ such that for any $c \in C$,

$$(\mathsf{gen}\,\tau)(c) = \{e \mid e \in \mathsf{Seq}(L \times C) \text{ and } e \text{ is a } \tau\text{-orbit of } c\}.$$

Assume a class function $\varepsilon : C \to \mathsf{Pow}\,\mathsf{Seq}(L \times C)$.

We say that τ *generates* ε if and only if $\mathsf{gen}\,\tau = \varepsilon$.

We say that ε *is generable* if and only if there is a class function from C to $\mathsf{Pow}(L \times C)$ that generates ε.

Now, suppose that ε is indeed generable. Can there be more than one class function from C to $\mathsf{Pow}(L \times C)$ that generates ε?

We could perhaps use the following tentative argument to convince ourselves that this cannot be the case: if τ_1 and τ_2 are two different class functions from C to $\mathsf{Pow}(L \times C)$, then there must be $c \in C$ and $\langle l, c' \rangle \in L \times C$ such that either $\langle l, c' \rangle \in \tau_1(c)$ and $\langle l, c' \rangle \notin \tau_2(c)$, or

$\langle l, c' \rangle \notin \tau_1(c)$ and $\langle l, c' \rangle \in \tau_2(c)$; and assuming, without any loss of generality, the former, we can prefix any τ_1-orbit of c' with $\langle l, c' \rangle$ to get a τ_1-orbit of c that cannot be a τ_2-orbit of c. But how do we know if there is a τ_1-orbit of c' to prefix with $\langle l, c' \rangle$?

If $\tau_1(c') = \emptyset$, then $\langle \ \rangle$ is a τ_1-orbit of c'. If $\tau_1(c') \neq \emptyset$, then we would again expect that there is at least one τ_1-orbit of c'. For we could imagine constructing one by first choosing a pair $\langle l', c'' \rangle$ in $\tau_1(c')$, then a pair $\langle l'', c''' \rangle$ in $\tau_1(c'')$, then a pair $\langle l''', c'''' \rangle$ in $\tau_1(c''')$, and so on forever, or until we reach a point where there is no pair to choose. If we never reach such a point, then this construction will involve an infinite number of choices. This suggests that the Axiom of Choice, or some other, weaker form of it, might be necessary to prove the statement that for every suitable τ and c, there is a τ-orbit of c. And indeed, this statement is equivalent to the *Axiom of Dependent Choice*.

We will need the following lemma:

Lemma 3.4.4. *For every $n \in \omega$, if there is $\langle l, c' \rangle \in \mathrm{W}_\tau(c)(n)$ and $e' \in \mathsf{Seq}(L \times C)$ such that e' is a τ-orbit of c', then there is $e \in \mathsf{Seq}(L \times C)$ such that e is a τ-orbit of c.*

Proof. We use induction.

If $n = 0$, then $\langle l, c' \rangle \in \tau(c)$, and $\langle \langle l, c' \rangle \rangle \cdot e'$ is a τ-orbit of c.

Otherwise, there is $m \in \omega$ such that $n = m + 1$. By definition of $\mathrm{W}_\tau(c)$, there is $\langle l', c'' \rangle \in \mathrm{W}_\tau(c)(m)$ such that $\langle l, c' \rangle \in \tau(c'')$, and $\langle \langle l, c' \rangle \rangle \cdot e'$ is a τ-orbit of c''. Thus, by the induction hypothesis, there is $e \in \mathsf{Seq}(L \times C)$ such that e is a τ-orbit of c.

Therefore, for every $n \in \omega$, if there is $\langle l, c' \rangle \in \mathrm{W}_\tau(c)(n)$ and $e' \in \mathsf{Seq}(L \times C)$ such that e' is a τ-orbit of c', then there is $e \in \mathsf{Seq}(L \times C)$ such that e is a τ-orbit of c. □

Theorem 3.4.5. *The following are equivalent:*

(a) for every class C, every class function $\tau : C \to \mathsf{Pow}(L \times C)$, and any $c \in C$, there is $e \in \mathsf{Seq}(L \times C)$ such that e is a τ-orbit of c;

(b) for every non-empty set S and every binary relation R on S, if for every $s \in S$, there is s' such that $s \mathrel{R} s'$, then there is an infinite sequence d over S such that for every $n \in \omega$, $\mathsf{head}\,\mathsf{tail}^n\, d \mathrel{R} \mathsf{head}\,\mathsf{tail}^{n+1}\, d$.

Proof. Suppose that (a) is true.

Assume a non-empty set S.

Let l be a label in L.

Let τ be a class function from S to $\mathsf{Pow}(L \times S)$ such that for every $s \in S$,

$$\tau(s) = \{\langle l, s' \rangle \mid s \mathrel{R} s'\}.$$

Let s be a member of S.

Since (a) is true, there is $e \in \mathsf{Seq}(L \times S)$ such that e is a τ-orbit of s. And by an easy induction, for every $n \in \omega$, $\mathsf{tail}^n\, e \neq \langle\ \rangle$.

Let d be an infinite sequence over S such that for every $n \in \omega$,

$$\mathsf{head}\,\mathsf{tail}^n\, d = \mathsf{sec}\,\mathsf{head}\,\mathsf{tail}^n\, e.$$

Then, by an easy induction, for every $n \in \omega$, $\mathsf{head}\,\mathsf{tail}^n\, d\ R\ \mathsf{head}\,\mathsf{tail}^{n+1}\, d$.

Thus, by generalization, (b) is true.

Conversely, suppose that (b) is true.

Assume a class C, a class function $\tau : C \to \mathsf{Pow}(L \times C)$, and $c \in C$.

If $\tau(c) = \emptyset$, then $\langle\ \rangle$ is a τ-orbit of c.

Otherwise, $\mathrm{W}_\tau(c)(0) \neq \emptyset$.

If there is $n \in \omega$ and $c' \in \mathrm{W}_\tau(c)(n)$ such that $\tau(c') = \emptyset$, then $\langle\ \rangle$ is a τ-orbit of c'. Thus, by Lemma 3.4.4, there is $e \in \mathsf{Seq}(L \times C)$ such that e is a τ-orbit of c.

Otherwise, for every $n \in \omega$ and every $c' \in \mathrm{W}_\tau(c)(n)$, $\tau(c') \neq \emptyset$.

Let $S = \bigcup\{\mathrm{W}_\tau(c)(n) \mid n \in \omega\}$.

Then, since $\mathrm{W}_\tau(c)(0) \neq \emptyset$, $S \neq \emptyset$.

Let R be a binary relation on S such that for every $\langle l, c'\rangle, \langle l', c''\rangle \in S$,

$$\langle l, c'\rangle\ R\ \langle l', c''\rangle \iff \langle l', c''\rangle \in \tau(c').$$

Then for every $\langle l, c'\rangle \in S$, there is $\langle l', c''\rangle$ such that $\langle l, c'\rangle\ R\ \langle l', c''\rangle$, and thus, since (b) is true, there is an infinite sequence d over S such that for every $n \in \omega$, $\mathsf{head}\,\mathsf{tail}^n\, d\ R\ \mathsf{head}\,\mathsf{tail}^{n+1}\, d$. And clearly, there is $n \in \omega$ and $\langle l, c'\rangle \in \mathrm{W}_\tau(c)(n)$ such that

$$\mathsf{head}\, d = \langle l, c'\rangle$$

and $\mathsf{tail}\, d$ is a τ-orbit of c'. Thus, by Lemma 3.4.4, there is $e \in \mathsf{Seq}(L \times C)$ such that e is a τ-orbit of c.

Thus, by generalization, (a) is true. □

Here, we accept the Axiom of Dependent Choice, and so we will take Theorem 3.4.5(a) for granted.

We can now make our tentative argument formal.

Assume class functions $\tau_1, \tau_2 : C \to \mathsf{Pow}(L \times C)$.

Proposition 3.4.6. *If $\tau_1 \neq \tau_2$, then $\mathsf{gen}\,\tau_1 \neq \mathsf{gen}\,\tau_2$.*

Proof. Suppose that $\tau_1 \neq \tau_2$.

Then there is c, l, and c' such that either $\langle l, c' \rangle \in \tau_1(c)$ and $\langle l, c' \rangle \notin \tau_2(c)$, or $\langle l, c' \rangle \notin \tau_1(c)$ and $\langle l, c' \rangle \in \tau_2(c)$.

Without any loss of generality, assume the former.

Let e' be a sequence in $\mathsf{Seq}(L \times C)$ that is a τ_1-orbit of c'.

Let $e = \langle \langle l, c' \rangle \rangle \cdot e'$.

Then $e \in (\mathsf{gen}\,\tau_1)(c)$, but $e \notin (\mathsf{gen}\,\tau_2)(c)$. Thus, $\mathsf{gen}\,\tau_1 \neq \mathsf{gen}\,\tau_2$. □

If we think of gen as an operator from cooperations of L-labelled transition coalgebras to cooperations of L-labelled execution coalgebras, then we can read Proposition 3.4.6 as saying that that operator is injective. So it must have a left inverse. The following shows that that left inverse is the composition on the left with the image of the carrier of the corresponding L-labelled execution coalgebra under η:

Proposition 3.4.7. *The following are true:*

(a) $\eta(C) \circ \mathsf{gen}\,\tau = \tau$;

(b) if ε is generable, then $\varepsilon = \mathsf{gen}(\eta(C) \circ \varepsilon)$.

Proof. Assume $c \in C$.

Assume $\langle l, c' \rangle \in L \times C$.

Suppose that $\langle l, c' \rangle \in (\eta(C) \circ \mathsf{gen}\,\tau)(c)$.

Then there is $e \in (\mathsf{gen}\,\tau)(c)$ such that $\mathsf{head}\,e = \langle l, c' \rangle$, and thus, $\langle l, c' \rangle \in \tau(c)$.

Conversely, suppose that $\langle l, c' \rangle \in \tau(c)$.

Let e' be a sequence in $\mathsf{Seq}(L \times C)$ that is a τ-orbit of c'.

Then $\langle \langle l, c' \rangle \rangle \cdot e' \in (\mathsf{gen}\,\tau)(c)$, and thus, $\langle l, c' \rangle \in (\eta(C) \circ \mathsf{gen}\,\tau)(c)$.

Thus, $\langle l, c' \rangle \in (\eta(C) \circ \mathsf{gen}\,\tau)(c)$ if and only if $\langle l, c' \rangle \in \tau(c)$.

Thus, by generalization, (a) is true.

We will now use (a) to prove (b).

Suppose that ε is generable.

Then there is a class function $\tau' : C \to \mathsf{Pow}(L \times C)$ such that

$$\mathsf{gen}\,\tau' = \varepsilon.$$

Thus,

$$\eta(C) \circ \mathsf{gen}\,\tau' = \eta(C) \circ \varepsilon,$$

and hence, by (a),

$$\tau' = \eta(C) \circ \varepsilon.$$

Thus, (b) is true. □

Before we move on to our characterization theorem, we have one last stop to make. We have built our notion of generability around the idea of a τ-orbit. And we have tried to formalize the latter in the most conceptually direct way. But as effective as that formalization has been, there is still reason to consider another one. First, it is ugly. And second, there is a very simple but powerful proof rule that it is entirely oblivious to.

We say that ε *is consistent with* τ if and only if for any $c \in C$ and any $e \in \varepsilon(c)$, one of the following is true:

(i) $\tau(c) = \emptyset$ and $e = \langle\ \rangle$;

(ii) there is l, c', and e' such that $\langle l, c' \rangle \in \tau(c)$, $e' \in \varepsilon(c')$, and $e = \langle\langle l, c' \rangle\rangle \cdot e'$.

Theorem 3.4.8. *The following are equivalent:*

(a) e is a τ-orbit of c;

(b) there is a class function $\varepsilon : C \to \mathsf{Pow\,Seq}(L \times C)$ such that ε is consistent with τ, and $e \in \varepsilon(c)$.

Proof. Suppose that (a) is true.

Let ε be a class function from C to $\mathsf{Pow\,Seq}(L \times C)$ such that for every c' and e', $e' \in \varepsilon(c')$ if and only if one of the following is true:

(i) $c' = c$ and $e' = e$;

(ii) there is $n \in \omega$ and l such that $\mathsf{tail}^n\, e = \langle\langle l, c'\rangle\rangle \cdot e'$.

Assume $c' \in C$ and $e' \in \varepsilon(c')$.

Suppose that $c' = c$ and $e' = e$.

If $\tau(c') = \emptyset$ and $e' = \langle\ \rangle$, then clause (i) of the consistency property is true.

Otherwise, there is l', c'', and e'' such that $\langle l', c''\rangle \in \tau(c')$ and

$$e' = \langle\langle l', c''\rangle\rangle \cdot e''.$$

Then, by (ii), $e'' \in \varepsilon(c'')$. Thus, clause (ii) of the consistency property is true.

Otherwise, there is $n \in \omega$ and l such that $\mathsf{tail}^n\, e = \langle\langle l, c'\rangle\rangle \cdot e'$.

If $e' = \langle\ \rangle$, then $\tau(c') = \emptyset$, and thus, clause (i) of the consistency property is true.

Otherwise, there is l', c'', and e'' such that $\langle l', c''\rangle \in \tau(c')$ and

$$e' = \langle\langle l', c''\rangle\rangle \cdot e''.$$

Then, by (ii), $e'' \in \varepsilon(c'')$. Thus, clause (ii) of the consistency property is true.

Therefore, ε is consistent with τ.

Thus, (b) is true.

Conversely, suppose (b) is true,

Then clause (i) of the property of being a τ-orbit of c is true.

By an easy induction, for every $n \in \omega$, if $\mathsf{tail}^n\, e \neq \langle\ \rangle$, then there is l and c' such that

$$\mathsf{head}\,\mathsf{tail}^n\, e = \langle l, c'\rangle$$

and $\mathsf{tail}^{n+1}\, e \in \varepsilon(c')$.

Assume $n \in \omega$.

Suppose that $\mathsf{tail}^n\, e \neq \langle\ \rangle$.

Then there is l and c' such that

$$\mathsf{head}\,\mathsf{tail}^n\, e = \langle l, c'\rangle$$

and $\mathsf{tail}^{n+1}\, e \in \varepsilon(c')$.

If $\mathsf{tail}^{n+1}\, e = \langle\ \rangle$, then $\tau(c') = \emptyset$.

Otherwise, there is l', c'', and e'' such that $\langle l', c'' \rangle \in \tau(c')$, $e'' \in \varepsilon(c'')$, and

$$\mathsf{tail}^{n+1}\, e = \langle \langle l', c'' \rangle \rangle \cdot e''.$$

Thus, by generalization, clause (ii) of the property of being a τ-orbit of c is true.

Therefore, (a) is true. □

The following is immediate:

Corollary 3.4.9. *If ε is consistent with τ, then for any $c \in C$,*

$$\varepsilon(c) \subseteq (\mathsf{gen}\, \tau)(c).$$

The following is now straightforward:

Corollary 3.4.10. $\mathsf{gen}\, \tau$ *is consistent with* τ*.*

Proof. Assume $c \in C$ and $e \in (\mathsf{gen}\, \tau)(c)$.

By Theorem 3.4.8, there is a class function $\varepsilon : C \to \mathsf{Pow}\,\mathsf{Seq}(L \times C)$ such that ε is consistent with τ, and $e \in \varepsilon(c)$.

If $\tau(c) = \emptyset$ and $e = \langle\ \rangle$, then there is nothing to prove.

Otherwise, there is l, c', and e' such that $\langle l, c' \rangle \in \tau(c)$, $e' \in \varepsilon(c')$, and

$$e = \langle \langle l, c' \rangle \rangle \cdot e'.$$

And by Corollary 3.4.9, $e' \in (\mathsf{gen}\, \tau)(c')$.

Thus, by generalization, $\mathsf{gen}\, \tau$ is consistent with τ. □

Corollary 3.4.9 is the proof rule that we referred to earlier, which is basically an instance of the *coinduction proof technique* described in [42]. This deserves a brief digression.

We write $\mathcal{G}_\tau(\varepsilon)$ for a class function from C to $\mathsf{Pow}\,\mathsf{Seq}(L \times C)$ such that for any $c \in C$,

$$\mathcal{G}_\tau(\varepsilon)(c) = \begin{cases} \{\langle\ \rangle\} & \text{if } \tau(c) = \emptyset; \\ \{\langle \langle l, c' \rangle \rangle \cdot e' \mid \langle l, c' \rangle \in \tau(c) \text{ and } e' \in \varepsilon(c')\} & \text{otherwise.} \end{cases}$$

Here again, our notation is not arbitrary. We think of $\mathcal{G}_\tau$ as an operator on class functions from C to $\mathsf{Pow}\,\mathsf{Seq}(L \times C)$. And the interesting thing about this operator is that it preserves the pointwise ordering of class functions from C to $\mathsf{Pow}\,\mathsf{Seq}(L \times C)$ induced by the inclusion class relation on $\mathsf{Pow\,Seq}(L \times C)$: for every class function $\varepsilon_1, \varepsilon_2 : C \to \mathsf{Pow\,Seq}(L \times C)$, if for any $c \in C$,

$$\varepsilon_1(c) \subseteq \varepsilon_2(c),$$

then for any $c \in C$,

$$\mathcal{G}_\tau(\varepsilon_1)(c) \subseteq \mathcal{G}_\tau(\varepsilon_2)(c).$$

What is more, ε is consistent with τ if and only if ε is a *post-fixed point* of $\mathcal{G}_\tau$, or equivalently, for any $c \in C$,

$$\varepsilon(c) \subseteq \mathcal{G}_\tau(\varepsilon)(c).$$

And it is not hard to see that $\mathsf{gen}\,\tau$ is the *greatest fixed point* of $\mathcal{G}_\tau$, with respect to the aforementioned pointwise ordering. Therefore, we can read Corollary 3.4.9 as saying that every post-fixed point of $\mathcal{G}_\tau$ is below the greatest fixed point of $\mathcal{G}_\tau$ in that ordering, which is precisely what the coinduction proof technique of [42] mandates. Unlike the latter, we could not have used Tarski's *Lattice-theoretical Fixpoint Theorem* (see [61, thm. 1]) to deduce our proof rule here. For if C is a proper class, then $\mathsf{Pow\,Seq}(L \times C)$ is not a complete lattice under inclusion, and so neither is the induced ordering of class functions from C to $\mathsf{Pow\,Seq}(L \times C)$. Nevertheless, the principle is the same.

Note that an ordered set can be viewed as a category, an order-preserving function on that set as a functor on that category, and a post-fixed point of that function as a coalgebra for that functor. And if that ordered set is a complete lattice, then, by Tarski's fixed-point theorem, there is a final coalgebra for that functor. And so the coinduction proof technique of [42] is just another variation of the general finality theme that lies behind the coinduction proof principle of Theorem 2.4.2. The same, of course, is true for the more *ad hoc* proof rule of Corollary 3.4.9.

For a historical account on the emergence of coinduction in computer science, we refer to [58].

We have now finally reached our generability characterization theorem.

Theorem 3.4.11. *ε is generable if and only if the following are true:*

(a) for every c, l, c', and e', if $\langle\langle l, c'\rangle\rangle \cdot e' \in \varepsilon(c)$, then $e' \in \varepsilon(c')$;

(b) for every c, l, c', e'_1, and e'_2, if $\langle\langle l, c'\rangle\rangle \cdot e'_1 \in \varepsilon(c)$ and $e'_2 \in \varepsilon(c')$, then $\langle\langle l, c'\rangle\rangle \cdot e'_2 \in \varepsilon(c)$;

(c) for any $c \in C$ and every infinite sequence s, if for every $n \in \omega$, there is $e \in \varepsilon(c)$ such that for every $k < n + 1$, $\mathsf{tail}^k\, e \neq \langle\ \rangle$ and

$$\mathsf{head\,tail}^k\, s = \mathsf{head\,tail}^k\, e,$$

then $s \in \varepsilon(c)$;

(d) for every c and e, if $e \in \varepsilon(c)$ and $\langle\ \rangle \in \varepsilon(c)$, then $e = \langle\ \rangle$;

(e) for any $c \in C$, $\varepsilon(c) \neq \emptyset$.

Proof. Suppose that ε is generable.

Then there is a class function $\tau : C \to \mathsf{Pow}(L \times C)$ such that $\varepsilon = \mathsf{gen}\,\tau$.

For every c, l, c', and e', if $\langle\langle l, c'\rangle\rangle \cdot e' \in (\mathsf{gen}\,\tau)(c)$, then, by Corollary 3.4.10, $e' \in (\mathsf{gen}\,\tau)(c)$, and thus, (a) is true.

For every c, l, c', e'_1, and e'_2, if $\langle\langle l, c'\rangle\rangle \cdot e'_1 \in (\mathsf{gen}\,\tau)(c)$ and $e'_2 \in (\mathsf{gen}\,\tau)(c')$, then, by Corollary 3.4.10, $\langle\langle l, c'\rangle\rangle \cdot e'_2 \in (\mathsf{gen}\,\tau)(c)$, and thus, (b) is true.

Assume $c \in C$ and an infinite sequence s.

Suppose that for every $n \in \omega$, there is $e \in (\mathsf{gen}\,\tau)(c)$ such that for every $k < n + 1$, $\mathsf{tail}^k\, e \neq \langle\ \rangle$ and

$$\mathsf{head}\,\mathsf{tail}^k\, s = \mathsf{head}\,\mathsf{tail}^k\, e.$$

Assume $n \in \omega$.

If $n = 0$, then there is $e \in (\mathsf{gen}\,\tau)(c)$ such that $e \neq \langle\ \rangle$ and

$$\mathsf{head}\, s = \mathsf{head}\, e.$$

Thus, there is l, c', and e' such that $\langle l, c'\rangle \in \tau(c)$ and

$$s = \langle\langle l, c'\rangle\rangle \cdot e'.$$

Otherwise, there is $m \in \omega$ such that $n = m + 1$. Then there is $e \in (\mathsf{gen}\,\tau)(c)$ such that $\mathsf{tail}^m\, e \neq \langle\ \rangle$ and

$$\mathsf{head}\,\mathsf{tail}^m\, s = \mathsf{head}\,\mathsf{tail}^m\, e,$$

and $\mathsf{tail}^{m+1}\, e \neq \langle\ \rangle$ and

$$\mathsf{head}\,\mathsf{tail}^{m+1}\, s = \mathsf{head}\,\mathsf{tail}^{m+1}\, e.$$

Thus, there is l, c', l', c'', and e'' such that $\langle l', c''\rangle \in \tau(c')$ and

$$\mathsf{tail}^m\, s = \langle\langle l, c'\rangle\rangle \cdot \langle\langle l', c''\rangle\rangle \cdot e''.$$

Thus, by generalization, s is a τ-orbit of c, and hence, $s \in (\mathsf{gen}\,\tau)(c)$.

Thus, by generalization, (c) is true.

For every c and e, if $e \in (\mathsf{gen}\,\tau)(c)$ and $\langle\ \rangle \in (\mathsf{gen}\,\tau)(c)$, then, by Corollary 3.4.10, $\tau(c) = \emptyset$, and hence, $e = \langle\ \rangle$. Thus, (d) is true.

By Theorem 3.4.5 and the Axiom of Dependent Choice, (e) is true.

Conversely, suppose that (a), (b), (c), (d), and (e) are true.

We prove that $\varepsilon = \mathsf{gen}(\boldsymbol{\eta}(C) \circ \varepsilon)$.

Assume $c \in C$ and $e \in \varepsilon(c)$.

If $e = \langle\ \rangle$, then, by (d), $\varepsilon(c) = \{\langle\ \rangle\}$, and thus, $\boldsymbol{\eta}(C)(\varepsilon(c)) = \emptyset$.

Otherwise, there is l, c', and e' such that

$$e = \langle\langle l, c'\rangle\rangle \cdot e'.$$

Thus, by definition of $\boldsymbol{\eta}$, $\langle l, c'\rangle \in \boldsymbol{\eta}(C)(\varepsilon(c))$, and by (a), $e' \in \varepsilon(c')$.

Thus, by generalization, ε is consistent with $\boldsymbol{\eta}(C) \circ \varepsilon$, and by Corollary 3.4.9, for any $c \in C$,

$$\varepsilon(c) \subseteq (\mathsf{gen}(\boldsymbol{\eta}(C) \circ \varepsilon))(c).$$

Assume $c \in C$ and $e \in (\mathsf{gen}(\boldsymbol{\eta}(C) \circ \varepsilon))(c)$.

If $e = \langle\ \rangle$, then $\boldsymbol{\eta}(C)(\varepsilon(c)) = \emptyset$. Thus, by (e), $e \in \varepsilon(c)$.

Otherwise, there is l, c', and e'_2 such that $\langle l, c'\rangle \in \boldsymbol{\eta}(C)(\varepsilon(c))$ and

$$e = \langle\langle l, c'\rangle\rangle \cdot e'_2.$$

Suppose that there is $n \in \omega$ such that $\mathsf{tail}^{n+1}\, e = \langle\ \rangle$.

Let n be the least member of ω such that $\mathsf{tail}^{n+1}\, e = \langle\ \rangle$.

Then there is l' and c'' such that $\boldsymbol{\eta}(C)(\varepsilon(c'')) = \emptyset$ and

$$\mathsf{tail}^{n}\, e = \langle\langle l', c''\rangle\rangle.$$

We use induction to prove that for every $j < n+1$, there is l'' and c''' such that

$$\mathsf{head}\,\mathsf{tail}^{j}\, e = \langle l'', c'''\rangle$$

and $\mathsf{tail}^{j+1}\, e \in \varepsilon(c''')$.

If $j = n$, then

$$\mathsf{head}\,\mathsf{tail}^{j}\, e = \langle l', c''\rangle$$

and

$$\mathsf{tail}^{j+1}\, e = \langle\ \rangle.$$

And since $\eta(C)(\varepsilon(c'')) = \emptyset$, by (e), $\mathsf{tail}^{j+1}\, e \in \varepsilon(c'')$.

Otherwise, there is $k < n+1$ such that $j+1 = k$. Then there is l'', c''', l''', c'''', and e_2'''' such that $\langle l''', c''''\rangle \in \eta(C)(\varepsilon(c'''))$ and

$$\mathsf{head}\,\mathsf{tail}^{j}\, e = \langle l'', c'''\rangle.$$

and

$$\mathsf{tail}^{j+1}\, e = \mathsf{tail}^{k}\, e = \langle\langle l''', c''''\rangle\rangle \cdot e_2''''.$$

By the induction hypothesis, $e_2'''' \in \varepsilon(c'''')$. Since $\langle l''', c''''\rangle \in \eta(C)(\varepsilon(c'''))$, there is e_1'''' such that $\langle\langle l''', c''''\rangle\rangle \cdot e_1'''' \in \varepsilon(c''')$. Thus, by (b), $\mathsf{tail}^{j+1}\, e \in \varepsilon(c''')$.

Therefore, $e_2' \in \varepsilon(c')$. And since $\langle l, c'\rangle \in \eta(C)(\varepsilon(c))$, there is e_1' such that $\langle\langle l, c'\rangle\rangle \cdot e_1' \in \varepsilon(c)$. Thus, by (b), $e \in \varepsilon(c)$.

Otherwise, for every $n \in \omega$, $\mathsf{tail}^{n+1}\, e \neq \langle\ \rangle$.

We use induction to prove that for every $n, k \in \omega$, there is l', c'', and e'' such that

$$\mathsf{head}\,\mathsf{tail}^{n}\, e = \langle l', c''\rangle,$$

$e'' \in \varepsilon(c'')$, and for every $i < k+1$, $\mathsf{tail}^{i}\, e'' \neq \langle\ \rangle$ and

$$\mathsf{head}\,\mathsf{tail}^{n+1+i}\, e = \mathsf{head}\,\mathsf{tail}^{i}\, e''.$$

Suppose that $k = 0$.

Then there is l', c'', l'', c''', and e''' such that $\langle l'', c'''\rangle \in \eta(C)(\varepsilon(c''))$ and

$$\mathsf{tail}^{n}\, e = \langle\langle l', c''\rangle\rangle \cdot \langle\langle l'', c'''\rangle\rangle \cdot e'''.$$

And since $\langle l'', c'''\rangle \in \eta(C)(\varepsilon(c''))$, there is e'' such that $e'' \in \varepsilon(c'')$ and

$$\mathsf{head}\, e'' = \langle l'', c'''\rangle.$$

Otherwise, there is $j \in \omega$ such that $k = j+1$.

Then there is l', c'', l'', c''', and e''' such that $\langle l'', c'''\rangle \in \eta(C)(\varepsilon(c''))$ and

$$\mathsf{tail}^{n}\, e = \langle\langle l', c''\rangle\rangle \cdot \langle\langle l'', c'''\rangle\rangle \cdot e'''.$$

By the induction hypothesis, there is e_2''' such that $e_2''' \in \varepsilon(c''')$ and for every $i < j+1$, $\mathsf{tail}^i\, e_2''' \neq \langle\ \rangle$ and

$$\mathsf{head}\,\mathsf{tail}^{n+2+i}\, e = \mathsf{head}\,\mathsf{tail}^i\, e_2'''.$$

Since $\langle l'', c'''\rangle \in \eta(C)(\varepsilon(c''))$, there is e_1''' such that $\langle\langle l'', c'''\rangle\rangle \cdot e_1''' \in \varepsilon(c'')$. Thus, by (b), $\langle\langle l'', c'''\rangle\rangle \cdot e_2''' \in \varepsilon(c'')$. And clearly, for every $i < k+1$, $\mathsf{tail}^i(\langle\langle l'', c'''\rangle\rangle \cdot e_2''') \neq \langle\ \rangle$ and

$$\mathsf{head}\,\mathsf{tail}^{n+1+i}\, e = \mathsf{head}\,\mathsf{tail}^i(\langle\langle l'', c'''\rangle\rangle \cdot e_2''').$$

Therefore, for every $n \in \omega$, there is $e' \in \varepsilon(c')$ such that for every $k < n+1$, $\mathsf{tail}^k\, e' \neq \langle\ \rangle$ and

$$\mathsf{head}\,\mathsf{tail}^k\, e_2' = \mathsf{head}\,\mathsf{tail}^m\, e'.$$

Thus, by (c), $e_2' \in \varepsilon(c')$. And since $\langle l, c'\rangle \in \eta(C)(\varepsilon(c))$, there is e_1' such that $\langle\langle l, c'\rangle\rangle \cdot e_1' \in \varepsilon(c)$. Thus, by (b), $\langle\langle l, c'\rangle\rangle \cdot e_2' \in \varepsilon(c)$, and hence, $e \in \varepsilon(c)$.

Thus, by generalization, for any $c \in C$,

$$\varepsilon(c) \supseteq (\mathsf{gen}(h(C) \circ \varepsilon))(c).$$

Thus, $\varepsilon = \mathsf{gen}(h(C) \circ \varepsilon)$, and hence, ε is generable. □

Clause (a) of Theorem 3.4.11 corresponds to suffix closure, clause (b), conditioned on (a), to fusion closure, and clause (c) to limit closure. Clause (d) asserts the impossibility of indeterminate termination. Finally, clause (e) is the non-triviality condition discussed earlier, and essentially replaces Emerson's *left totality* condition on the generating transition relation (see [20]).

Each of these five properties has come about in connection with a different cause of failure of the converse of Proposition 3.3.4, and hence of Proposition 3.3.1 and 3.3.2. And if we have been thorough enough, we should expect that the conjunction of all five properties be sufficient a condition for eliminating that failure altogether. This turns out to be the case.

We say that $\langle C, \varepsilon\rangle$ *is generable* if and only if ε is generable.

Theorem 3.4.12. *If $\langle C_1, \varepsilon_1\rangle$ and $\langle C_2, \varepsilon_2\rangle$ are generable, then h is a homomorphism from $\langle C_1, \varepsilon_1\rangle$ to $\langle C_2, \varepsilon_2\rangle$ if and only if h is a homomorphism from the L-labelled transition coalgebra $\langle C_1, \eta(C_1) \circ \varepsilon_1\rangle$ to the L-labelled transition coalgebra $\langle C_2, \eta(C_2) \circ \varepsilon_2\rangle$.*

Proof. Suppose that $\langle C_1, \varepsilon_1\rangle$ and $\langle C_2, \varepsilon_2\rangle$ are generable.

Suppose that h is a homomorphism from $\langle C_1, \varepsilon_1\rangle$ to $\langle C_2, \varepsilon_2\rangle$.

Then, by Proposition 3.3.1, h is a homomorphism from the L-labelled transition coalgebra $\langle C_1, \eta(C_1) \circ \varepsilon_1 \rangle$ to the L-labelled transition coalgebra $\langle C_2, \eta(C_2) \circ \varepsilon_2 \rangle$.

Conversely, suppose that h is a homomorphism from $\langle C_1, \eta(C_1) \circ \varepsilon_1 \rangle$ to $\langle C_2, \eta(C_2) \circ \varepsilon_2 \rangle$.

Let ε_2' be a class function from C_2 to $\mathsf{Pow}\,\mathsf{Seq}(L \times C_2)$ such that for any $c_2 \in C_2$,

$$\begin{aligned}\varepsilon_2'(c_2) = \{e_2 \mid {} & \text{there is } c_1 \in C_1 \text{ and } e_1 \in \varepsilon_1(c_1) \\ & \text{such that } h(c_1) = c_2 \text{ and } (\mathsf{Seq}(L \times h))(e_1) = e_2\}.\end{aligned}$$

Assume $c_2 \in C_2$ and $e_2 \in \varepsilon_2'(c_2)$.

Then there is $c_1 \in C_1$ and $e_1 \in \varepsilon_1(c_1)$ such that

$$h(c_1) = c_2$$

and

$$(\mathsf{Seq}(L \times h))(e_1) = e_2.$$

Suppose that $e_1 = \langle\ \rangle$.

Then $e_2 = \langle\ \rangle$. Also, $(\eta(C_1) \circ \varepsilon_1)(c_1) = \emptyset$, and since h is a homomorphism from $\langle C_1, \eta(C_1) \circ \varepsilon_1 \rangle$ to $\langle C_2, \eta(C_2) \circ \varepsilon_2 \rangle$, $(\eta(C_2) \circ \varepsilon_2)(c_2) = \emptyset$.

Otherwise, by Corollary 3.4.10, there is l, c_1', and e_1' such that $\langle l, c_1' \rangle \in (\eta(C_1) \circ \varepsilon_1)(c_1)$, $e_1' \in \varepsilon_1(c_1')$, and

$$e_1 = \langle \langle l, c_1' \rangle \rangle \cdot e_1'.$$

Then $(\mathsf{Seq}(L \times h))(e_1') \in \varepsilon_2'(h(c_1'))$ and

$$e_2 = \langle \langle l, h(c_1') \rangle \rangle \cdot (\mathsf{Seq}(L \times h))(e_1').$$

And since h is a homomorphism from $\langle C_1, \eta(C_1) \circ \varepsilon_1 \rangle$ to $\langle C_2, \eta(C_2) \circ \varepsilon_2 \rangle$, $\langle l, h(c_1') \rangle \in (\eta(C_2) \circ \varepsilon_2)(c_2)$.

Thus, by generalization, ε_2' is consistent with $\eta(C_2) \circ \varepsilon_2$. Then, by Corollary 3.4.9, for any $c_2 \in C_2$,

$$\varepsilon_2'(c_2) \subseteq \varepsilon_2(c_2).$$

And clearly, for any $c_1 \in C_1$,

$$(\mathsf{Pow}\,\mathsf{Seq}(L \times h))(\varepsilon_1(c_1)) \subseteq \varepsilon_2'(h(c_1)).$$

Therefore, for any $c_1 \in C_1$,

$$(\mathsf{Pow}\,\mathsf{Seq}(L \times h))(\varepsilon_1(c_1)) \subseteq \varepsilon_2(h(c_1)).$$

We use induction to prove that for any $c_1 \in C_1$ and any finite $e_2 \in \varepsilon_2(h(c_1))$, there is $e_1 \in \varepsilon_1(c_1)$ such that

$$(\mathsf{Seq}(L \times h))(e_1) = e_2.$$

If $e_2 = \langle\ \rangle$, then $(\eta(C_2) \circ \varepsilon_2)(h(c_1)) = \emptyset$, and since h is a homomorphism from $\langle C_1, \eta(C_1) \circ \varepsilon_1 \rangle$ to $\langle C_2, \eta(C_2) \circ \varepsilon_2 \rangle$, $(\eta(C_1) \circ \varepsilon_1)(c_1) = \emptyset$. Thus, $\langle\ \rangle \in \varepsilon_1(c_1)$.

Otherwise, by Corollary 3.4.10, there is l, c_2', and e_2' such that $\langle l, c_2' \rangle \in (\eta(C_2) \circ \varepsilon_2)(h(c_1))$, $e_2' \in \varepsilon_2(h(c_1'))$, and

$$e_2 = \langle \langle l, c_2' \rangle \rangle \cdot e_2'.$$

And since h is a homomorphism from $\langle C_1, \eta(C_1) \circ \varepsilon_1 \rangle$ to $\langle C_2, \eta(C_2) \circ \varepsilon_2 \rangle$, there is c_1' such that $\langle l, c_1' \rangle \in (\eta(C_1) \circ \varepsilon_1)(c_1)$ and

$$h(c_1') = c_2'.$$

Thus, by the induction hypothesis, there is $e_1' \in \varepsilon_1(c_1')$ such that

$$(\mathsf{Seq}(L \times h))(e_1') = e_2',$$

and hence, $\langle \langle l, c_1' \rangle \rangle \cdot e_1' \in \varepsilon_1(c_1)$. And clearly,

$$(\mathsf{Seq}(L \times h))(\langle \langle l, c_1' \rangle \rangle \cdot e_1') = e_2.$$

We now prove that for any $c_1 \in C_1$ and any infinite $e_2 \in \varepsilon_2(h(c_1))$, there is $e_1 \in \varepsilon_1(c_1)$ such that

$$(\mathsf{Seq}(L \times h))(e_1) = e_2.$$

Assume $c_1 \in C_1$ and an infinite $e_2 \in \varepsilon_2(h(c_1))$.

By Corollary 3.4.10 and an easy induction, for every $n \in \omega$, there is l and c_2' such that

$$\mathsf{head}\,\mathsf{tail}^n\, e_2 = \langle l, c_2' \rangle$$

and $\mathsf{tail}^{n+1}\, e_2 \in \varepsilon_2(c_2')$.

Let W be a function from ω to $(L \times C_1) \times \omega$ such that

$$W(0) = \{\langle\langle l, c_1'\rangle, 0\rangle \mid \langle l, c_1'\rangle \in (\eta(C_1) \circ \varepsilon_1)(c_1) \text{ and } \mathsf{head}\, e_2 = \langle l, h(c_1')\rangle\}$$

and for every $n \in \omega$,

$$\begin{aligned} W(n+1) = \{\langle\langle l', c_1''\rangle, n+1\rangle \mid {} & \text{there is } \langle\langle l, c_1'\rangle, n\rangle \in W(n) \\ & \text{such that } \langle l', c_1''\rangle \in (\eta(C_1) \circ \varepsilon_1)(c_1') \\ & \text{and } \mathsf{head}\, \mathsf{tail}^{n+1}\, e_2 = \langle l', h(c_1'')\rangle\}. \end{aligned}$$

Since e_2 is non-empty, there is l and c_2' such that

$$\mathsf{head}\, e_2 = \langle l, c_2'\rangle.$$

And since h is a homomorphism from $\langle C_1, \eta(C_1) \circ \varepsilon_1\rangle$ to $\langle C_2, \eta(C_2) \circ \varepsilon_2\rangle$, there is c_1' such that $\langle l, c_1'\rangle \in (\eta(C_1) \circ \varepsilon_1)(c_1)$ and

$$h(c_1') = c_2'.$$

Thus, $W(0) \neq \emptyset$.

Let $S = \bigcup\{W(n) \mid n \in \omega\}$.

Then, since $W(0) \neq \emptyset$, $S \neq \emptyset$.

Let R be a binary relation on S such that for every $\langle\langle l, c_1'\rangle, m\rangle, \langle\langle l', c_1''\rangle, n\rangle \in S$,

$$\langle\langle l, c_1'\rangle, m\rangle \; R \; \langle\langle l', c_1''\rangle, n\rangle \iff \langle l', c_1''\rangle \in (\eta(C_1) \circ \varepsilon_1)(c_1') \text{ and } n = m + 1.$$

Assume $\langle\langle l, c_1'\rangle, n\rangle \in S$.

Then

$$\mathsf{head}\, \mathsf{tail}^{n}\, e_2 = \langle l, h(c_1')\rangle$$

and $\mathsf{tail}^{n+1}\, e_2 \in \varepsilon_2(h(c_1'))$. Thus, by Corollary 3.4.10, there is l', c_2'', and e_2'' such that $\langle l', c_2''\rangle \in (\eta(C_2) \circ \varepsilon_2)(h(c_1'))$, $e_2'' \in \varepsilon_2(c_2'')$, and

$$\mathsf{tail}^{n+1}\, e_2 = \langle\langle l', c_2''\rangle\rangle \cdot e_2''.$$

And since h is a homomorphism from $\langle C_1, \eta(C_1) \circ \varepsilon_1\rangle$ to $\langle C_2, \eta(C_2) \circ \varepsilon_2\rangle$, there is c_1'' such that $\langle l', c_1''\rangle \in (\eta(C_1) \circ \varepsilon_1)(c_1')$ and

$$h(c_1'') = c_2''.$$

Thus, $\langle\langle l', c_1''\rangle, n+1\rangle \in S$ and $\langle\langle l, c_1'\rangle, n\rangle \; R \; \langle\langle l', c_1''\rangle, n+1\rangle$.

Thus, by generalization, for every $s \in S$, there is s' such that $s \mathrel{R} s'$. Then, by the Axiom of Dependent Choice, there is an infinite sequence d over S such that for every $n \in \omega$, $\mathsf{head}\,\mathsf{tail}^n\, d \mathrel{R} \mathsf{head}\,\mathsf{tail}^{n+1}\, d$.

Let $n = \mathsf{sec}\,\mathsf{head}\, d$.

We use induction to prove that for every $j < n + 1$, there is l, c_1', and e_1' such that $\langle\langle l, c_1'\rangle, j\rangle \in W(j)$, $e_1' \in \varepsilon_1(c_1')$, and

$$(\mathsf{Seq}(L \times h))(\langle\langle l, c_1'\rangle\rangle \cdot e_1') = \mathsf{tail}^j\, e_2.$$

If $j = n$, then there is l and c_1' such that

$$\mathsf{head}\, d = \langle\langle l, c_1'\rangle, j\rangle.$$

And clearly, $\langle\langle l, c_1'\rangle, j\rangle \in W(j)$, $(\mathsf{Seq}(\mathsf{proj}_1((L \times C_1) \times \omega)))(\mathsf{tail}\, d) \in \varepsilon_1(c_1')$ and

$$(\mathsf{Seq}(L \times h))(\langle l, c_1'\rangle \cdot (\mathsf{Seq}(\mathsf{proj}_1((L \times C_1) \times \omega)))(\mathsf{tail}\, d)) = \mathsf{tail}^j\, e_2.$$

Otherwise, there is $k < n + 1$ such that $j + 1 = k$. By the induction hypothesis, there is l', c_1'', and e_1'' such that $\langle\langle l', c_1''\rangle, k\rangle \in W(k)$, $e_1'' \in \varepsilon_1(c_1'')$, and

$$(\mathsf{Seq}(L \times h))(\langle\langle l', c_1''\rangle\rangle \cdot e_1'') = \mathsf{tail}^k\, e_2.$$

Since $\langle\langle l', c_1''\rangle, k\rangle \in W(k)$, there is $\langle\langle l, c_1'\rangle, j\rangle \in W(j)$ such that $\langle l', c_1''\rangle \in (\boldsymbol{\eta}(C_1) \circ \varepsilon_1)(c_1')$ and $\mathsf{head}\,\mathsf{tail}^k\, e_2 = \langle l', h(c_1'')\rangle$. And clearly, $\langle\langle l', c_1''\rangle\rangle \cdot e_1'' \in \varepsilon_1(c_1')$ and

$$(\mathsf{Seq}(L \times h))(\langle\langle l, c_1'\rangle\rangle \cdot e_1') = \mathsf{tail}^j\, e_2.$$

Therefore, there is l, c_1', and e_1' such that $\langle\langle l, c_1'\rangle, e_1'\rangle \in W(0)$, $e_1' \in \varepsilon_1(c_1')$, and

$$(\mathsf{Seq}(L \times h))(\langle\langle l, c_1'\rangle\rangle \cdot e_1') = e_2.$$

And by definition of W, $\langle l, c_1'\rangle \in (\boldsymbol{\eta}(C_1) \circ \varepsilon_1)(c_1)$. Thus, $\langle l, c_1'\rangle \cdot e_1' \in \varepsilon_1(c_1)$.

Thus, by generalization, for any $c_1 \in C_1$,

$$(\mathsf{Pow}\,\mathsf{Seq}(L \times h))(\varepsilon_1(c_1)) \supseteq \varepsilon_2(h(c_1)).$$

Thus,

$$(\mathsf{Pow}\,\mathsf{Seq}(L \times h)) \circ \varepsilon_1 = \varepsilon_2 \circ h,$$

and hence, h is a homomorphism from $\langle C_1, \varepsilon_1\rangle$ to $\langle C_2, \varepsilon_2\rangle$. □

The following is immediate from Theorem 2.2.10 and 3.4.12:

Theorem 3.4.13. *If $\langle C_1, \varepsilon_1\rangle$ and $\langle C_2, \varepsilon_2\rangle$ are generable, then B is a bisimulation between $\langle C_1, \varepsilon_1\rangle$ and $\langle C_2, \varepsilon_2\rangle$ if and only if B is a bisimulation between the L-labelled transition coalgebra $\langle C_1, \eta(C_1) \circ \varepsilon_1\rangle$ and the L-labelled transition coalgebra $\langle C_2, \eta(C_2) \circ \varepsilon_2\rangle$.*

Once more, we can translate all this back to the system side of the theory.

Assume a binary relation $T : S \leftrightarrow L \times S$.

Assume a binary relation $E : S \leftrightarrow \mathscr{S}(L \times S)$.

We write $\mathcal{E}_T(E)$ for a binary relation between S and $\mathscr{S}(L \times S)$ such that for any $s \in S$ and every $e \in \mathscr{S}(L \times S)$,

$$s\ \mathcal{E}_T(E)\ e \iff \begin{array}{l} \text{either there is no } \langle l, s'\rangle \text{ such that } s\ T\ \langle l, s'\rangle\text{, and } e = \langle\ \rangle, \\ \text{or there is } \langle l, s'\rangle \text{ such that } s\ T\ \langle l, s'\rangle,\ \mathsf{head}\, e = \langle l, s'\rangle\text{, and } s'\ E\ \mathsf{tail}\, e. \end{array}$$

The following is trivial:

Proposition 3.4.14. $\mathcal{E}_T(E) = \mathsf{rel}\, \mathcal{G}_{\mathsf{fun}\, T}(\mathsf{fun}\, E)$.

We think of $\mathcal{E}_T$ as a function on binary relations between S and $\mathscr{S}(L \times S)$. And the reason that we are interested in this function is that it preserves the ordering of binary relations between S and $\mathscr{S}(L \times S)$ induced by the inclusion relation on their graphs: for every binary relation $E_1, E_2 : S \leftrightarrow \mathscr{S}(L \times S)$, if

$$\mathsf{graph}\, E_1 \subseteq \mathsf{graph}\, E_2,$$

then

$$\mathsf{graph}\, \mathcal{E}_T(E_1) \subseteq \mathsf{graph}\, \mathcal{E}_T(E_2).$$

This ordering is of course a complete lattice, and hence, by Tarski's Lattice-theoretical Fixpoint Theorem, so is the set of all fixed points of $\mathcal{E}_T$.

We write $\mathsf{exec}\, T$ for the greatest fixed point of $\mathcal{E}_T$.

Notice that here, the coinduction proof technique of [42] is directly applicable.

The following follows from Proposition 3.4.14 and the fact that $\mathsf{gen}\,\mathsf{fun}\, T$ is the greatest fixed point of $\mathcal{G}_{\mathsf{fun}\, T}$:

Proposition 3.4.15. $\mathsf{exec}\, T = \mathsf{rel}\,\mathsf{gen}\,\mathsf{fun}\, T$.

We say that T *generates* E if and only if $\mathsf{exec}\, T = E$.

We say that E is generable if and only if there is a binary relation between S and $L \times S$ that generates E.

The following is immediate from Proposition 2.2.1 and 3.4.15:

Proposition 3.4.16. *E is generable if and only if $\mathsf{fun}\, E$ is generable.*

The following is immediate from Proposition 2.2.1, 3.3.3, 3.4.15, and 3.4.16:

Proposition 3.4.17. *The following are true:*

(a) $\mathsf{trans}\, \mathsf{exec}\, T = T$*;*

(b) if E is generable, then $\mathsf{exec}\, \mathsf{trans}\, E = E$*.*

We say that $\langle S, E\rangle$ *is generable* if and only if E is generable.

The following is immediate from Proposition 3.4.16:

Proposition 3.4.18. *$\langle S, E\rangle$ is generable if and only if the L-labelled execution coalgebra $\langle S, \mathsf{fun}\, E\rangle$ is generable.*

The following is immediate from Proposition 2.2.1(b), 2.2.15, 3.2.4, and 3.3.3, and Theorem 3.4.13:

Theorem 3.4.19. *If $\langle S_1, E_1\rangle$ and $\langle S_2, E_2\rangle$ are generable, then B is a bisimulation between $\langle S_1, E_1\rangle$ and $\langle S_2, E_2\rangle$ if and only if B is a bisimulation between the L-labelled transition systems $\langle S_1, \mathsf{trans}\, E_1\rangle$ and $\langle S_2, \mathsf{trans}\, E_2\rangle$.*

We would like to finish this section with a few remarks.

Proposition 3.4.17 and Theorem 3.4.19 confirm what has been implicit throughout this section: generable labelled execution systems are just another representation of labelled transition systems. This is even more evident in the coalgebra side of the theory, where, by Proposition 3.4.7 and Theorem 3.4.12, the category of all generable labelled execution coalgebras and all homomorphisms between them is isomorphic to $(\mathsf{Pow} \circ (L \times \mathsf{Id}))$-$\mathbf{Coalg}$. Thus, for all practical purposes, generable labelled execution coalgebras are equivalent to labelled transition coalgebras.

In light of this equivalence, Theorem 3.4.11 does not just characterize generable labelled execution coalgebras. It marks the boundary between the expressive power of labelled transition coalgebras and labelled execution coalgebras. And what it implies is that there is no sense in choosing the latter over the former, unless we are willing to give up one or more of the five properties listed in the respective clauses of Theorem 3.4.11.

In the next chapter, we will give up two of these properties: limit closure and impossibility of indeterminate termination. Giving up limit closure will enable us to faithfully model the finite delay property, so intrinsically bound to the notion of asynchrony. And indeterminate termination will provide us with the means to simulate the behaviour of a capricious environment that may at any time cease to produce input stimuli.

3.5 Behaviour modelling and covarieties

In Section 2.4, in order to motivate the study of final coalgebras, we considered a hypothetical scenario, in which we used L-labelled transition systems to model the behaviour of processes of some kind. And for simplicity, we assumed that any state of every L-labelled transition system modelled the behaviour of some process of that kind. Here, we wish to consider a different scenario, in which we use L-labelled execution systems instead. And this time, we make no such simplifying assumption.

We have already seen examples of labelled execution systems that are not suitable for modelling the behaviour of processes. In Section 3.3, we argued that non-Abrahamson systems should be excluded from consideration. And in the next chapter, we will exclude even more systems, leaving only those that conform with our intuitive notion of behaviour of an asynchronous process.

In cases like these, a final L-labelled execution coalgebra is no longer the right choice of model. Sure, every behaviour modelled within the class of systems under consideration is accounted for exactly once in such a coalgebra. But there are more behaviours in there, which are neither needed nor wanted. What is required then is a suitable generalization of the final coalgebra approach of the previous chapter.

We will need a couple of last concepts from category theory.

A *subcategory* of a category is a collection of objects and arrows of that category that is closed under the identity, domain, codomain, and composition operation of that category.

A *full subcategory* of a category is a subcategory of that category, whose arrows between any two objects are all the arrows between the two objects in that category.

Notice that a full subcategory is completely determined by its objects.

In Section 2.4, we were interested in the terminal object of F-**Coalg**. Here, by a similar reduction, we are interested in the terminal object of a full subcategory of F-**Coalg**.

Assume a full subcategory $\mathbf{S}$ of F-**Coalg**.

We say that $\langle C, \gamma \rangle$ *is final in* $\mathbf{S}$ if and only if $\langle C, \gamma \rangle$ is in $\mathbf{S}$, and for every F-coalgebra $\langle C', \gamma' \rangle$ in $\mathbf{S}$, there is exactly one homomorphism from $\langle C', \gamma' \rangle$ to $\langle C, \gamma \rangle$.

Clearly, not every full subcategory of F-**Coalg** has a final F-coalgebra. But as we will see, every "reasonably specified" one does.

We say that $\mathbf{S}$ *is closed under the formation of homomorphic images* if and only if for every F-coalgebra $\langle C, \gamma \rangle$, if $\langle C, \gamma \rangle$ is in $\mathbf{S}$, and $\langle C', \gamma' \rangle$ is a homomorphic image of $\langle C, \gamma \rangle$, then $\langle C', \gamma' \rangle$ is in $\mathbf{S}$.

We say that $\mathbf{S}$ *is closed under the formation of subcoalgebras* if and only if for every F-coalgebra $\langle C, \gamma \rangle$, if $\langle C, \gamma \rangle$ is in $\mathbf{S}$, and $\langle C', \gamma' \rangle$ is a subcoalgebra of $\langle C, \gamma \rangle$, then $\langle C', \gamma' \rangle$ is in $\mathbf{S}$.

We say that $\mathbf{S}$ *is closed under the formation of direct sums* if and only if for every class-indexed family $\{\langle C_i, \gamma_i \rangle\}_{i \in I}$ of F-coalgebras, if for every $i \in I$, $\langle C_i, \gamma_i \rangle$ is in $\mathbf{S}$, then $\sum_{i \in I} \langle C_i, \gamma_i \rangle$ is in $\mathbf{S}$.

Definition 3.5.1. An F*-covariety* is a full subcategory $\mathbf{C}$ of F-**Coalg** such that the following are true:

(a) $\mathbf{C}$ is closed under the formation of homomorphic images;

(b) $\mathbf{C}$ is closed under the formation of subcoalgebras;

(c) $\mathbf{C}$ is closed under the formation of direct sums.

The concept of F-covariety is the coalgebraic counterpart of the concept of F-variety, which is a generalization, in the same sense as before, of the concept of Σ-variety, what Birkhoff called a *family of algebras* when he introduced the concept in [13]. It is our idea of a "reasonably specified" full subcategory of F-**Coalg**.

For example, the category of all Abrahamson labelled execution coalgebras, defined in the obvious way, and all homomorphisms between them is a $(\mathsf{Pow} \circ \mathsf{Seq} \circ (L \times \mathsf{Id}))$-covariety, as is the category of all generable labelled execution coalgebras and all homomorphisms between them. But we will not need to prove this here.

Just as we did with the notion of finality, we generalize the notion of completeness to full subcategories.

We say that $\langle C, \gamma \rangle$ *is complete in* $\mathbf{S}$ if and only if $\langle C, \gamma \rangle$ is in $\mathbf{S}$, and for every F-coalgebra $\langle C', \gamma' \rangle$ in $\mathbf{S}$ and any $c' \in C'$, there is exactly one $c \in C$ such that c' and c are bisimilar among $\langle C', \gamma' \rangle$ and $\langle C, \gamma \rangle$.

The following is a generalization of Theorem 2.4.13:

Theorem 3.5.2. *For every F-covariety $\mathbf{C}$, the following are true:*

(a) $\langle C, \gamma \rangle$ *is final in* $\mathbf{C}$*;*

(b) for every small F-coalgebra $\langle C', \gamma' \rangle$ in $\mathbf{C}$, there is exactly one homomorphism from $\langle C', \gamma' \rangle$ to $\langle C, \gamma \rangle$;

(c) $\langle C, \gamma \rangle$ *is complete in* $\mathbf{C}$*;*

(d) for every small F-coalgebra $\langle C', \gamma' \rangle$ in $\mathbf{C}$, and any $c' \in C'$, there is exactly one $c \in C$ such that c' and c are bisimilar among $\langle C', \gamma' \rangle$ and $\langle C, \gamma \rangle$.

Proof. See proof of Theorem 2.4.13. □

The reason that we were able to reuse the proof of Theorem 2.4.13 here without any modification or adjustment is that, because of the closure properties of an F-covariety, all relevant constructions in that proof can be carried out inside $\mathbf{C}$. The only structures in that proof that are not necessarily in $\mathbf{C}$ are the bisimulations, which are not supposed to either.

The following is a generalization of Theorem 2.5.1:

Theorem 3.5.3. *For every F-covariety $\mathbf{C}$, there is an F-coalgebra that is final in $\mathbf{C}$.*

Proof. Assume an F-covariety $\mathbf{C}$.

Let $\langle C, \gamma \rangle$ be an F-coalgebra that is final in F-**Coalg**.

Let $\langle C', \gamma' \rangle$ be a direct sum of all small F-coalgebras in $\mathbf{C}$.

Let h be the unique homomorphism from $\langle C', \gamma' \rangle$ to $\langle C, \gamma \rangle$.

Then, by Theorem 2.4.5, there is a class function $\rho : \mathsf{ran}\, h \to F(\mathsf{ran}\, h)$ such that

$$\langle \mathsf{ran}\, h, \rho \rangle \leq \langle C, \gamma \rangle.$$

And by Proposition 2.4.7, there is exactly one homomorphism h' from $\langle C', f' \rangle$ to $\langle \mathsf{ran}\, h, \rho \rangle$ such that

$$h = (\mathsf{ran}\, h \hookrightarrow C) \circ h'.$$

Clearly, h' is surjective, and thus, $\langle \mathsf{ran}\, h, \rho \rangle$ is a homomorphic image of $\langle C', f' \rangle$. And since $\mathbf{C}$ is an F-covariety, $\langle \mathsf{ran}\, h, \rho \rangle$ is an F-coalgebra in $\mathbf{C}$.

We claim that $\langle \mathsf{ran}\, h, \rho \rangle$ is final in $\mathbf{C}$.

Assume a small F-coalgebra $\langle C'', \gamma'' \rangle$ in $\mathbf{C}$.

Let ι be the canonical injection map from C'' to C'.

Then $h' \circ \iota$ is a homomorphism from $\langle C'', \gamma'' \rangle$ to $\langle \mathsf{ran}\, h, \rho \rangle$.

Suppose, toward contradiction, that there are homomorphisms h_1 and h_2 from $\langle C'', \gamma'' \rangle$ to $\langle \mathsf{ran}\, h, \rho \rangle$ such that

$$h_1 \neq h_2.$$

Then both $(\mathsf{ran}\, h \hookrightarrow C) \circ h_1$ and $(\mathsf{ran}\, h \hookrightarrow C) \circ h_2$ are homomorphisms from $\langle C'', \gamma'' \rangle$ to $\langle C, \gamma \rangle$. And since $(\mathsf{ran}\, h \hookrightarrow C)$ is injective,

$$(\mathsf{ran}\, h \hookrightarrow C) \circ h_1 \neq (\mathsf{ran}\, h \hookrightarrow C) \circ h_2,$$

contrary to $\langle C, \gamma \rangle$ being final in F-**Coalg**.

Therefore, there is at most one homomorphism from $\langle C'', \gamma'' \rangle$ to $\langle \mathsf{ran}\, h, \rho \rangle$.

Thus, there is exactly one homomorphism, namely $h' \circ \iota$, from $\langle C'', \gamma'' \rangle$ to $\langle \mathsf{ran}\, h, \rho \rangle$.

Thus, by generalization and Theorem 3.5.2, $\langle \mathsf{ran}\, h, \rho \rangle$ is final in **C**. □

Theorem 3.5.3 is easy enough to be already known. But being unable to trace it in the literature, we have taken care to prove it here (but see [3, thm 2.2]).

3.6 Execution systems in the literature

The process algebra literature is dominated by the concept of labelled transition system. And to some extent, this is understandable. For process algebra emerged from a marriage of Plotkin's structural operational semantics (see [48]), and Keller's named transition systems (see [32]) (see [41, chap. 12], [12], [49], and [10]). This marriage was the work of Robin Milner, and is most clearly expounded in [41], but was already present in [36], where the so-called *expansion law* was stated for the first time.

The expansion law has been a constant source of controversy in the theory of concurrency. In the language of Milner's CCS (see [40], [41]), a typical equation asserted by the law is the following:

$$a.\mathbf{0} \mid b.\mathbf{0} = a.b.\mathbf{0} + b.a.\mathbf{0}.$$

Here, 'a' and 'b' stand for arbitrary actions, '$\mathbf{0}$' for the inactive agent, which is incapable of performing any action, '.' for sequential composition, '|' for parallel composition, and '+' for alternative composition. Thus, the intended meaning of the equation is that the parallel execution of a and b is “equivalent”, in some sense, to the indeterminate serialization of the two.

In order to justify the expansion law, and the blurring between causal dependence and temporal precedence resulting from it, Milner wrote the following in [36, p. 81]:

> We do not yet know how to frame a sufficiently general law without, in a sense, explicating parallelism in terms of non-determinism. More precisely, this means that we explicate a (parallel) composition by presenting all serializations - or interleavings - of its possible atomic actions. This has the disadvantage that we lose distinction between causally necessary sequence, and sequence which is fictitiously imposed upon causally independent actions; [. . .]. However, it may be justified to ignore it if we can accept the view that, in observing (communicating with) a composite system, we make our observations in a definite time sequence, thereby causing a sequencing of actions which, for the system itself, are causally independent.

Effectively, what he argued for was a dichotomy between causation and observation in the theory of concurrency. And what he proposed as an observational view to the theory was the interleaving of the atomic actions of the various agents inside a system as would be perceived by a single, sequential observer outside the system. But what he failed to recognize was that the expansion law is in fact inconsistent with that view.

To understand the mismatch, consider the following equation derived from the expansion law, again in the language of CCS:

$$\mathbf{fix}(X = a.X) \mid \mathbf{fix}(X = b.X) = \mathbf{fix}(X = a.X + b.X).$$

Here, we use recursion expressions to define agents with infinite behaviour. Thus, $\mathbf{fix}(X = a.X)$ is an agent that forever iterates a, $\mathbf{fix}(X = b.X)$ one that forever iterates b, and $\mathbf{fix}(X = a.X + b.X)$ one that at each iteration, does either a or b, indeterminately choosing between the two. But whereas every infinite sequence over $\{a, b\}$ is a trace of a possible execution of $\mathbf{fix}(X = a.X + b.X)$, not every such sequence is consistent with what could be perceived by a sequential observer of $\mathbf{fix}(X = a.X) \mid \mathbf{fix}(X = b.X)$. Indeed, only those sequences that contain an infinite number of a’s and an infinite number of b’s are. For if $\mathbf{fix}(X = a.X)$ and $\mathbf{fix}(X = b.X)$ execute in parallel, each of them must eventually perform an infinite number of actions, and each of these actions must eventually be perceived by any sequential observer of $\mathbf{fix}(X = a.X) \mid \mathbf{fix}(X = b.X)$.

All this goes unnoticed in the finite case, because interleaving the executions of two finite agents is ultimately equivalent to indeterminately alternating between the two. But the expansion law blindly carried that equivalence over to the infinite case. And this created confusion. The word “interleaving” became synonymous to the word “indeterminate”, and the observational view was robbed of its power to express liveness properties such as finite delay or termination.

Of course, it is not the expansion law per se that is to blame for this confusion. Interleaving is an operation on executions, not transitions, and the use of labelled transition systems was always going to cause problems with it. But instead of replacing transitions with executions in their systems, people started augmenting them with all kinds of different pieces of information that would allow them to distinguish the “admissible” sequences of transitions from the “inadmissible” ones. And more often than not, the result was a type of modelling structure that could no longer claim adherence to the observational view. The few attempts that did use executions directly, at least those that we are aware of (see [18], [17]), were not concerned with organizing them into structures and looking at their branching properties, and anyway, seem to have received only scant attention, possibly due to their poor presentation.

The first place where we do see executions organized into structures is not process algebra, but temporal logic. These so-called path structures (see Section 3.1) are quite popular in the beginning. We do not see a formal concept of bisimulation for them, but there is definitely interest in their branching properties. The notions of suffix, fusion, and limit closure are all defined in connection with path structures. Eventually, they give way to Kripke structures, inherited from modal logic, and claimed to provide “a setting more appropriate to concurrency” (see [21, p. 152]). They do not. But despite the voices of reason asking for a separation between implementation and correctness issues in reasoning about concurrent programs (for example, see [16]), transitions remain in the lead role.

In [26], Hennessy and Stirling introduce what appears to be the first type of labelled execution system in the literature. They call systems of that type *general transition systems*, and in their definition, demand not only suffix and fusion closure, but prefix closure as well, with the justification that it “also appears to be natural” (see [26, p. 27]). They also define a concept of *extended bisimulation* for such systems, which is basically the same as our formally derived concept of bisimulation between labelled executions systems (see Definition 3.2.3). The focus in [26] is in logic, and specifically, in a generalization of Hennessy-Milner Logic (see [25]) to general transition systems. But what is surprising is that no attempt is later made to apply the ideas of general transition systems and extended bisimulations to the semantics of processes.

More than ten years later, these ideas pop up in a “very rough and incomplete draft” of Aczel (see [4]), who is aware of Hennessy’s work in [24], a precursor of [26], but apparently, unaware of the work in [26] (see footnote in [4, p. 3]). Aczel’s intention is to apply the final universe approach of [3] to the semantics of Milner’s SCCS with finite delay (see [38]). But his execution is indeed “very rough and incomplete”. The proposed type of structure is a generalized type of labelled transition system, where each state is equipped with the set of all infinite sequences of transitions “admissible” from that state. An added condition of “stability” makes structures of that type ultimately equivalent to the general transition systems of [26], but only because the latter are prefix closed. Eventually, these structures are represented as coalgebras over **Class**, and [3, thm. 2.2] is used to prove the existence of a final coalgebra in the full subcategory of all such coalgebras that are “stable”.

The only other place where we find these ideas applied to the semantics of processes is [27]. The starting point is again Milner’s SCCS with finite delay, and the structures used are practically the same as in [4]. But the approach is purely categorical. Indeed, the main goal in [27] is showing how much can be done within category theory alone. Apparently, a lot, but not without cost.

Comparing [26], [4], and [27] with our work here, there are two things that we think stand out and would like to mention.

First, as regards the general idea underlying the concept of labelled execution system, we find that in all three of [26], [4], and [27], the notion of indeterminate termination, and its use in modelling the behaviour of reactive systems, has been completely overlooked. This is easy to put right in [26], where prefix closure is an added feature, but not so in [4] and [27], where the property is practically built into the structure of a system.

Second, as regards the formalization of the idea, we believe that the present approach represents a great simplification, both conceptually and notationally, over what was done in all three of [26], [4], and [27], and hope that the reader will appreciate the power and elegance of our framework, which, we think, are felt in every part of the theory.

It should be emphasized that the precedence of [26], [4], and [27] over our work here is not causal, only temporal. Our ideas were developed, and for the most part, worked out before any acquaintance with these studies. The above review was mainly driven by our curiosity to understand why ideas that in retrospect seem so natural have not found their way into the household of the average concurrency theorist.

In the end, one can only speculate. But one thing is certain: if matters of pedagogy have played any role in this, transition semantics have definitely profited from it; for people like pictures, and execution systems are impossible to draw.

Chapter 4

Sequential Asynchronous Processes

4.1 Our intuitive notion of asynchronous process

What is a process? This is a question that has troubled concurrency theorists ever since the birth of the field almost fifty year ago. And yet, there is still very little agreement as to what the answer is. With the advent of process algebra in the eighties came the well known controversy between so-called *true concurrency* and interleaving semantics, adding another dimension to the problem.

The central figure in this controversy was of course Robin Milner, who, as we have seen, was the one to argue for the dichotomy between causation and observation, which can be tersely described as the difference between events partially ordered by causal dependence, and actions totally ordered by temporal precedence (see [9]). Milner's idea was nothing less than brilliant. But it was marred by the inadequacies of the used model, the labelled transition system.

Our work thus far has been to fix just that. And we have done so by introducing a model that we believe can do justice to the interleaving approach. We now want to use this model to define an abstract notion of asynchronous process.

The word *asynchronous* is derived from the Greek word *ασύγχρονος*. The latter is a compound of the privative prefix *a*-, the prefix *συν*-, in the sense of *of identical*, and *χρόνος*, the Greek word for *time*. It is used in a strict sense to assert that one does not coincide in time with another, or, more often, in a loose sense to assert that one need not coincide in time with another. And in the context of concurrency theory, it is the only in a loose sense that it can be used without any contradiction. For if two events must be non-synchronized, then there is a synchronization between the presence of the one and absence of the other.

The most important implication from this intuitive notion of asynchrony is that communication must be unidirectional. In other words, in an asynchronous setting, if a process wants to communicate with another process, then the most it can do is pass a message for it to read at its own leisure. And this means that there must be some place to leave that message. Therefore, a process is determined by a number of ports, or mailboxes, as we like to call them, which is the interface of the process to the outside world.

Based on this, we shall now give an informal, anthropomorphic description of how we think an asynchronous process works. The precise concept will only be understood after the axioms are presented. But this informal description will be very useful in understanding the intuition behind the mathematics, and we shall return frequently to it. We only advise the reader not to take it too literally.

We shall then think of an asynchronous process as an organization of people. And this organization will communicate with the outside world through the exchange of messages. Specifically, it will have a number of mailboxes, and a number of workers and messengers. We will not constrain the internal structure of the mailboxes at all. These may act as bounded or unbounded buffers, first-in-first-out queues or push-and-pop stacks, single-token places or multisets. The workers inside the organization will be responsible for the operation of the organization, and their access to these mailboxes will too be left unconstrained. For all we know, a worker might choose to ignore all messages, or might be sitting behind a mailbox waiting for the next message. Finally, the messengers will be responsible for delivering messages to other organizations, and thus, represent the communication capabilities of the organization. And while the number of mailboxes will be considered fixed for an organization, the number of workers and messengers will, in principle, be allowed to vary.

Here, we will focus on a specific type of asynchronous process, a sequential asynchronous process. What this corresponds to in our anthropomorphic description is the constraint that there be just one worker. Messengers may by many and variable, but there is always going to be a single worker.

4.2 Asynchronous process types

Our first step is to formalize this anthropomorphic notion of interface of a process.

Definition 4.2.1. An *asynchronous process type* is an ordered pair $\langle I, O\rangle$ such that the following are true:

(a) I is a function such that the following are true:

  (i) $\mathsf{dom}\, I$ is countable;

   (ii) for any $i \in \mathsf{dom}\, I$, $I(i)$ is a non-empty set;

(b) O is a function such that the following are true:

   (i) $\mathsf{dom}\, O$ is countable;

   (ii) for any $o \in \mathsf{dom}\, O$, $O(o)$ is a non-empty set;

(c) $\mathsf{dom}\, I$ and $\mathsf{dom}\, O$ are disjoint.

In an asynchronous process type $\langle I, O\rangle$, I represents that mailboxes of a process, and the types of messages that can be deposited to them. O, on the other hand, represents all the mailboxes that a process might deposit a message to, and all the kinds of messages it might deposit. This creates some redundancy, but a typed approach greatly simplifies things.

Note that the constraint that $\mathsf{dom}\, I$ and $\mathsf{dom}\, O$ be disjoint makes sense from an observational point of view: if a process deposits a message to itself, only it can observe that deposition.

One interesting question here is why we have decided to make the number of input and output channels countable when we have intentionally left everything else unbounded. The reason is simple: our executions are finite or infinite sequences, as they must be if we want our processes to maintain a connection with reality. And thus, a process with an uncountable number of channels could never communicate over all of them through its lifetime.

4.3 Systems and coalgebras

The type of an asynchronous process determines the different kinds of actions that it can perform. Basically, these are input actions, which are to be understood as the event of deposition of a message to a mailbox of the process, output actions, which correspond to the deposition of a message by a messenger of the process to some mailbox, and internal actions, which model the work done internally by the workers of the process.

Assume an asynchronous process type $\langle I, O\rangle$.

We write $\mathsf{act}_{\text{in}}\,\langle I, O\rangle$ for $\{\langle i, m\rangle \mid i \in \mathsf{dom}\, I \text{ and } m \in I(i)\}$.

We write $\mathsf{act}_{\text{out}}\,\langle I, O\rangle$ for $\{\langle o, m\rangle \mid o \in \mathsf{dom}\, O \text{ and } m \in O(o)\}$.

We write τ for $\emptyset$.

We write $\mathsf{act}\,\langle I, O\rangle$ for $\mathsf{act}_{\text{in}}\,\langle I, O\rangle \cup \mathsf{act}_{\text{out}}\,\langle I, O\rangle \cup \{\tau\}$.

We call any $\langle i, m\rangle \in \mathsf{act}_{\mathrm{in}}\,\langle I, O\rangle$ an *input action*, any $\langle o, m\rangle \in \mathsf{act}_{\mathrm{out}}\,\langle I, O\rangle$ an *output action*, and τ an *internal action*.

Notice that the choice of $\emptyset$ for τ is arbitrary. Any object different from an ordered pair would do. The symbol itself is used for conformance with the literature.

It is also worth noting that there is no predefined set of actions. Any pair of functions satisfying the definition of an asynchronous process type will do.

We now begin to formalize our notion of process. The first step is to define a notion of path. This is understood as any finite or infinite sequence of steps through the underlying transition system of a labelled execution system.

Assume an $(\mathsf{act}\,\langle I, O\rangle)$-labelled execution system $\langle S, E\rangle$.

We write $\mathsf{paths}\,\langle S, E\rangle$ for the largest subset X of $S \times \mathscr{S}(\mathsf{act}\,\langle I, O\rangle \times S)$ such that for every $\langle s, e\rangle \in X$, one of the following is true:

(i) $e = \langle\ \rangle$;

(ii) there is α, s', and e' such that $s \xrightarrow{\alpha}_{\mathsf{trans}\,E} s'$, $\langle s', e'\rangle \in X$, and

$$e = \langle\langle \alpha, s'\rangle\rangle \cdot e'.$$

The existence of a largest such set is easily verified by straightforward order-theoretic arguments, which we will henceforth omit.

We next begin to define a notion of fairness for such paths. Intuitively, we want every different party involved in the execution of a process to be given the chance to proceed. This include the worker of the process, the messengers, and of course the environment. However, we want to place as little constraints as possible, making sure that in the end, every possible interleaving of them will be part of the behaviour of the process.

Assume $\langle s, e\rangle \in \mathsf{paths}\,\langle S, E\rangle$.

We say that τ *is eventually in* $\langle s, e\rangle$ if and only if $\langle s, e\rangle$ is a member of the smallest set X such that the following are true:

(i) for every s', s'', and e'', if $s' \xrightarrow{\tau}_{\mathsf{trans}\,E} s''$ and $\langle s'', e''\rangle \in \mathsf{paths}\,\langle S, E\rangle$, then $\langle s', \langle\langle \tau, s''\rangle\rangle \cdot e''\rangle \in X$;

(ii) for every s', α, s'', and e'', if $s' \xrightarrow{\alpha}_{\mathsf{trans}\,E} s''$ and $\langle s'', e''\rangle \in X$, then $\langle s', \langle\langle \alpha, s''\rangle\rangle \cdot e''\rangle \in X$.

We say that τ *eventually need not be in* $\langle s, e\rangle$ if and only if $\langle s, e\rangle$ is a member of the smallest set X such that the following are true:

(i) for every s', if for every s'', $s' \not\xrightarrow{\tau}_{\mathsf{trans}\,E} s''$, then for every e', if $\langle s', e'\rangle \in \mathsf{paths}\,\langle S, E\rangle$, then $\langle s', e'\rangle \in X$;

(ii) for every s', α, s'', and e'', if $s' \xrightarrow{\alpha}_{\mathsf{trans}\,E} s''$ and $\langle s'', e''\rangle \in X$, then $\langle s', \langle\langle \alpha, s''\rangle\rangle \cdot e''\rangle \in X$.

The meaning of these two predicates should be clear. The first asserts that in an execution, the worker does at some point make progress, and the second that at some point in the execution, the worker cannot make progress.

The following is meant to take care the progress of a messenger.

Assume $\langle o, m\rangle \in \mathsf{act}_{\mathsf{out}}\,\langle I, O\rangle$ and s' such that $s \xrightarrow{\langle o,m\rangle}_{\mathsf{trans}\,E} s'$.

We say that $\langle\langle o, m\rangle, s'\rangle$ *is eventually in* $\langle s, e\rangle$ if and only if $\langle s, e\rangle$ is a member of the smallest set X such that the following are true:

(i) for every e', if $\langle s', e'\rangle \in \mathsf{paths}\,\langle S, E\rangle$, then $\langle s, \langle\langle\langle o, m\rangle, s'\rangle\rangle \cdot e'\rangle \in X$;

(ii) for every α, s'_α, and e', if $s \xrightarrow{\alpha}_{\mathsf{trans}\,E} s'_\alpha$, $\langle s'_\alpha, e'\rangle \in \mathsf{paths}\,\langle S, E\rangle$, and there is s'' such that $s' \xrightarrow{\alpha}_{\mathsf{trans}\,E} s''$, $s'_\alpha \xrightarrow{\langle o,m\rangle}_{\mathsf{trans}\,E} s''$, and $\langle\langle o, m\rangle, s''\rangle$ is eventually in $\langle s'_\alpha, e'\rangle$, then $\langle s, \langle\langle \alpha, s'_\alpha\rangle\rangle \cdot e'\rangle \in X$.

Here, the situation is more complex. The problem is that at any point, there may be more than one messenger with identical messages and destinations. If we used the same type of predicate as in the case of an internal action, there would be cases where a messenger was put on hold forever, just because another messenger with the same message and destination was making progress. And for all we know, the behaviour of the worker may depend on the identity of the available messengers. In Definition 4.3.1, the necessary confluence constraints are imposed so that the above method does indeed guarantee that every single messenger will eventually make progress.

With these inductively defined predicates, we can now define a notion of fairness coinductively.

We say that $\langle s, e\rangle$ *is fair in* $\langle S, E\rangle$ if and only if $\langle s, e\rangle$ is a member of the largest subset X of $\mathsf{paths}\,\langle S, E\rangle$ such that for every $\langle s', e'\rangle \in X$, one of the following is true:

(i) $e' = \langle\ \rangle$, and for every s'', the following are true:

(1) $s \not\xrightarrow{\tau}_{\mathsf{trans}\,E} s''$;

(2) for every $\langle o, m\rangle \in \mathsf{act}_{\mathsf{out}}\,\langle I, O\rangle$, $s \not\xrightarrow{\langle o,m\rangle}_{\mathsf{trans}\,E} s''$;

(ii) there is $\langle i, m\rangle \in \mathsf{act}_{\text{in}}\,\langle I, O\rangle$, s'', and e'' such that $s' \xrightarrow{\langle i,m\rangle}_{\mathsf{trans}\,E} s''$, $\langle s'', e''\rangle \in X$,

$$e' = \langle\langle\langle i, m\rangle, s''\rangle\rangle \cdot e'',$$

and the following are true:

(1) if there is $\langle o, m_o\rangle \in \mathsf{act}_{\text{out}}\,\langle I, O\rangle$ and s''_o such that $s' \xrightarrow{\langle o,m_o\rangle}_{\mathsf{trans}\,E} s''_o$, then there is s''' such that $s'' \xrightarrow{\langle o,m_o\rangle}_{\mathsf{trans}\,E} s'''$, $s''_o \xrightarrow{\langle i,m\rangle}_{\mathsf{trans}\,E} s'''$, and $\langle\langle o, m_o\rangle, s'''\rangle$ is eventually in $\langle s'', e''\rangle$;

(2) if there is s''_τ such that $s' \xrightarrow{\tau}_{\mathsf{trans}\,E} s''_\tau$, then either τ is eventually in $\langle s'', e''\rangle$, or τ eventually need not be in $\langle s'', e''\rangle$;

(iii) there is $\langle o, m\rangle \in \mathsf{act}_{\text{out}}\,\langle I, O\rangle$, s'', and e'' such that $s' \xrightarrow{\langle o,m\rangle}_{\mathsf{trans}\,E} s''$, $\langle s'', e''\rangle \in X$,

$$e' = \langle\langle\langle o, m\rangle, s''\rangle\rangle \cdot e'',$$

and if there is s''_τ such that $s' \xrightarrow{\tau}_{\mathsf{trans}\,E} s''_\tau$, then either τ is eventually in $\langle s'', e''\rangle$, or τ eventually need not be in $\langle s'', e''\rangle$;

(iv) there is s'' and e'' such that $s' \xrightarrow{\tau}_{\mathsf{trans}\,E} s''$, $\langle s'', e''\rangle \in X$,

$$e' = \langle\langle \tau, s''\rangle\rangle \cdot e'',$$

and if there is $\langle o, m\rangle \in \mathsf{act}_{\text{out}}\,\langle I, O\rangle$ and s''_o such that $s' \xrightarrow{\langle o,m\rangle}_{\mathsf{trans}\,E} s''_o$, then there is s''' such that $s'' \xrightarrow{\langle o,m\rangle}_{\mathsf{trans}\,E} s'''$, $s''_o \xrightarrow{\tau}_{\mathsf{trans}\,E} s'''$, and $\langle\langle o, m\rangle, s'''\rangle$ is eventually in $\langle s'', e''\rangle$.

We note that similar approaches of iterated induction and coinduction to fairness have been used before (for example, see [45]).

By requiring the executions of a process to coincide with all fair paths in the underlying labelled transition system, we guarantee that all different interleavings of the worker, the messengers, and the environment are accounted for.

Definition 4.3.1. A *sequential asynchronous process system* of type $\langle I, O\rangle$ is an $(\mathsf{act}\,\langle I, O\rangle)$-labelled execution system $\langle S, E\rangle$ such that the following are true:

(a) for any $s \in S$ and any $\langle i, m\rangle \in \mathsf{act}_{\text{in}}\,\langle I, O\rangle$, there is s' such that $s \xrightarrow{\langle i,m\rangle}_{\mathsf{trans}\,E} s'$;

(b) for every s, s'_1, and s'_2, and any $\langle i, m\rangle \in \mathsf{act}_{\text{in}}\,\langle I, O\rangle$, if $s \xrightarrow{\langle i,m\rangle}_{\mathsf{trans}\,E} s'_1$ and $s \xrightarrow{\langle i,m\rangle}_{\mathsf{trans}\,E} s'_2$, then $s'_1 = s'_2$;

(c) for every s, s'_1, and s'_2, and any $\langle i_1, m_1\rangle, \langle i_1, m_2\rangle \in \mathsf{act}_{\text{in}}\,\langle I, O\rangle$, if $s \xrightarrow{\langle i_1,m_1\rangle}_{\mathsf{trans}\,E} s'_1$ $s \xrightarrow{\langle i_2,m_2\rangle}_{\mathsf{trans}\,E} s'_2$, and $i_1 \neq i_2$, then there is s'' such that $s'_1 \xrightarrow{\langle i_2,m_2\rangle}_{\mathsf{trans}\,E} s''$ and $s'_2 \xrightarrow{\langle i_1,m_1\rangle}_{\mathsf{trans}\,E} s''$;

(d) for every s, s'_i, and s'_o, any $\langle i, m_i\rangle \in \mathsf{act}_{\text{in}}\,\langle I, O\rangle$, and any $\langle o, m_o\rangle \in \mathsf{act}_{\text{out}}\,\langle I, O\rangle$, if $s \xrightarrow{\langle i,m_i\rangle}_{\mathsf{trans}\,E} s'_i$ and $s \xrightarrow{\langle o,m_o\rangle}_{\mathsf{trans}\,E} s'_o$, then there is s'' such that $s'_i \xrightarrow{\langle o,m_o\rangle}_{\mathsf{trans}\,E} s''$ and $s'_o \xrightarrow{\langle i,m_i\rangle}_{\mathsf{trans}\,E} s''$;

(e) for every s, s'_1, and s'_2, and any $\langle o_1, m_1\rangle, \langle o_1, m_2\rangle \in \mathsf{act}_{\text{out}}\,\langle I, O\rangle$ if $s \xrightarrow{\langle o_1,m_1\rangle}_{\mathsf{trans}\,E} s'_1$ and $s \xrightarrow{\langle o_2,m_2\rangle}_{\mathsf{trans}\,E} s'_2$, then there is s'' such that $s'_1 \xrightarrow{\langle o_2,m_2\rangle}_{\mathsf{trans}\,E} s''$ and $s'_2 \xrightarrow{\langle o_1,m_1\rangle}_{\mathsf{trans}\,E} s''$;

(f) for every s, s'_o, and s'_τ, and any $\langle o, m\rangle \in \mathsf{act}_{\text{out}}\,\langle I, O\rangle$, if $s \xrightarrow{\langle o,m\rangle}_{\mathsf{trans}\,E} s'_o$ and $s \xrightarrow{\tau}_{\mathsf{trans}\,E} s'_\tau$, then there is s'' such that $s'_o \xrightarrow{\tau}_{\mathsf{trans}\,E} s''$ and $s'_\tau \xrightarrow{\langle o,m\rangle}_{\mathsf{trans}\,E} s''$;

(g) for every s, s'_o, and s'', and any $\langle o, m\rangle \in \mathsf{act}_{\text{out}}\,\langle I, O\rangle$, if $s \xrightarrow{\langle o,m\rangle}_{\mathsf{trans}\,E} s'_o$ and $s \xrightarrow{\tau}_{\mathsf{trans}\,E} s''$, then there is s'_τ such that $s \xrightarrow{\tau}_{\mathsf{trans}\,E} s'_\tau$ and $s'_\tau \xrightarrow{\langle o,m\rangle}_{\mathsf{trans}\,E} s''$;

(h) for every s and e, $s \rhd_E e$ if and only if $\langle s, e\rangle$ is fair in $\langle S, E\rangle$.

We briefly go over the clauses of this definition.

Clause (a) corresponds to the most basic assumption of asynchrony, that we should be able to deposit any kind of message in any mailbox at any time. Diagrammatically, we have the following representation:

$$\begin{tikzcd} s \arrow[d, dashed, "i\ ?\ m"] \\ s' \end{tikzcd} \qquad (\text{all } \langle i, m\rangle \in \mathsf{act}_{\text{in}}\,\langle I, O\rangle)$$

Here, the dashed line is used to signify existence of the corresponding transition, which is understood as a transition of the underlying labelled transition system of the sequential asynchronous process system.

Clause (b) formalizes the idea that the effect of deposition of a message in a mailbox is uniquely determined. In other words, there are no different ways of depositing a message in a mailbox. Diagrammatically, we have the following:

$$\begin{tikzcd}
s \arrow[r, "i \mathbin{?} m"] \arrow[d, "i \mathbin{?} m"'] & s'_1 & \Longrightarrow \quad s'_1 = s'_2 \\
s'_2 & &
\end{tikzcd}$$

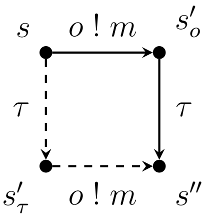
Clause (c) is the last clause concerned with input, and what it states is essentially that different mailboxes are physically distinct. Thus, depositing messages to two different mailboxes in any order will have the same effect in the content of each mailbox. This is portrayed in the following confluent diagram:

$$\begin{tikzcd}
p \arrow[r, "i_1 \mathbin{?} m_1"] \arrow[d, "i_2 \mathbin{?} m_2"'] & s'_1 \arrow[d, dashed, "i_2 \mathbin{?} m_2"] & (i_1 \neq i_2) \\
s'_2 \arrow[r, dashed, "i_1 \mathbin{?} m_1"'] & s'' &
\end{tikzcd}$$

Clauses (d), (e), and (f) are concerned with output, and essentially, assert that a messenger can never be recalled. Once a decision is made, and a messenger is dispatched, that messenger will insist until his job is done. This is expressed by the confluence of the following three diagrams:

$$\begin{tikzcd}
s \arrow[r, "o \mathbin{!} m_o"] \arrow[d, "i \mathbin{?} m_i"'] & s'_o \arrow[d, dashed, "i \mathbin{?} m_i"] \\
s'_i \arrow[r, dashed, "o \mathbin{!} m_o"'] & s''
\end{tikzcd}
\qquad
\begin{tikzcd}
s \arrow[r, "o_1 \mathbin{!} m_1"] \arrow[d, "o_2 \mathbin{!} m_2"'] & s'_1 \arrow[d, dashed, "o_2 \mathbin{!} m_2"] \\
s'_2 \arrow[r, dashed, "o_1 \mathbin{!} m_1"'] & s''
\end{tikzcd}
\qquad
\begin{tikzcd}
s \arrow[r, "o \mathbin{!} m"] \arrow[d, "\tau"'] & s'_o \arrow[d, dashed, "\tau"] \\
s'_\tau \arrow[r, dashed, "o \mathbin{!} m"'] & s''
\end{tikzcd}$$

Clause (g) is the last of the clauses that constrain the underlying labelled transition system, and is, as we shall see, somewhat controversial. Essentially, what it asserts is that it cannot be the case that a worker waits for a messenger to continue with his job. This clearly precludes the situation where there is only one person in the organization, switching between a worker and a messenger. This is the intuitive idea behind the following commutative diagram:



Finally, clause (h) is the place where our work begins to pay off. All previous clauses concern transitions, and while more general than the typical approach to asynchrony (for example, see [59]), are after all expressed in terms of labelled transition systems. It is in (h), where we begin to express properties about our systems that are impossible to capture with labelled transition systems. And this will become all the more important, when we use our coalgebraic framework to obtain an extensional model for these properties.

To be formal, we repeat all the above, this time for labelled execution coalgebras.

Assume an $(\mathsf{act}\,\langle I, O\rangle)$-labelled execution coalgebra $\langle C, \varepsilon\rangle$.

We write $\mathsf{paths}\,\langle C, \varepsilon\rangle$ for the largest subclass X of $C \times \mathsf{Seq}(\mathsf{act}\,\langle I, O\rangle \times C)$ such that for every $\langle c, e\rangle \in X$, one of the following is true:

(i) $e = \langle\ \rangle$;

(ii) there is α, c', and e' such that $c \xrightarrow{\alpha}_{\eta(C)\circ\varepsilon} c'$, $\langle c', e'\rangle \in X$, and

$$e = \langle\langle \alpha, c'\rangle\rangle \cdot e'.$$

Assume $\langle c, e\rangle \in \mathsf{paths}\,\langle C, \varepsilon\rangle$.

We say that τ *is eventually in* $\langle c, e\rangle$ if and only if $\langle c, e\rangle$ is a member of the smallest set X such that the following are true:

(i) for every c', c'', and e'', if $c' \xrightarrow{\tau}_{\eta(C)\circ\varepsilon} c''$ and $\langle c'', e''\rangle \in \mathsf{paths}\,\langle C, \varepsilon\rangle$, then $\langle c', \langle\langle \tau, c''\rangle\rangle \cdot e''\rangle \in X$;

(ii) for every c', α, c'', and e'', if $c' \xrightarrow{\alpha}_{\eta(C)\circ\varepsilon} c''$ and $\langle c'', e''\rangle \in X$, then $\langle c', \langle\langle \alpha, c''\rangle\rangle \cdot e''\rangle \in X$.

We say that τ *eventually need not be in* $\langle c, e\rangle$ if and only if $\langle c, e\rangle$ is a member of the smallest set X such that the following are true:

(i) for every c', if for every c'', $c' \not\xrightarrow{\tau}_{\eta(C)\circ\varepsilon} c''$, then for every e', if $\langle c', e'\rangle \in \mathsf{paths}\,\langle C, \varepsilon\rangle$, then $\langle c', e'\rangle \in X$;

(ii) for every c', α, c'', and e'', if $c' \xrightarrow{\alpha}_{\eta(C)\circ\varepsilon} c''$ and $\langle c'', e'' \rangle \in X$, then $\langle c', \langle\langle \alpha, c'' \rangle\rangle \cdot e'' \rangle \in X$.

Assume $\langle o, m \rangle \in \mathsf{act}_{\text{out}} \langle I, O \rangle$ and c' such that $c \xrightarrow{\langle o,m \rangle}_{\eta(C)\circ\varepsilon} c'$.

We say that $\langle \langle o, m \rangle, c' \rangle$ *is eventually in* $\langle c, e \rangle$ if and only if $\langle c, e \rangle$ is a member of the smallest set X such that the following are true:

(i) for every e', if $\langle c', e' \rangle \in \mathsf{paths} \langle C, \varepsilon \rangle$, then $\langle c, \langle\langle\langle o, m \rangle, c' \rangle\rangle \cdot e' \rangle \in X$;

(ii) for every α, c'_α, and e', if $c \xrightarrow{\alpha}_{\eta(C)\circ\varepsilon} c'_\alpha$, $\langle c'_\alpha, e' \rangle \in \mathsf{paths} \langle C, \varepsilon \rangle$, and there is c'' such that $c' \xrightarrow{\alpha}_{\eta(C)\circ\varepsilon} c''$, $c'_\alpha \xrightarrow{\langle o,m \rangle}_{\eta(C)\circ\varepsilon} c''$, and $\langle \langle o, m \rangle, c'' \rangle$ is eventually in $\langle c'_\alpha, e' \rangle$, then $\langle c, \langle\langle \alpha, c'_\alpha \rangle\rangle \cdot e' \rangle \in X$.

We say that $\langle c, e \rangle$ *is fair in* $\langle C, \varepsilon \rangle$ if and only if $\langle c, e \rangle$ is a member of the largest subset X of $\mathsf{paths} \langle C, \varepsilon \rangle$ such that for every $\langle c', e' \rangle \in X$, one of the following is true:

(i) $e' = \langle\ \rangle$, and for every c'', the following are true:

  (1) $c \not\xrightarrow{\tau}_{\eta(C)\circ\varepsilon} c''$;

  (2) for every $\langle o, m \rangle \in \mathsf{act}_{\text{out}} \langle I, O \rangle$, $c \not\xrightarrow{\langle o,m \rangle}_{\eta(C)\circ\varepsilon} c''$;

(ii) there is $\langle i, m \rangle \in \mathsf{act}_{\text{in}} \langle I, O \rangle$, c'', and e'' such that $c' \xrightarrow{\langle i,m \rangle}_{\eta(C)\circ\varepsilon} c''$, $\langle c'', e'' \rangle \in X$,

$$e' = \langle\langle\langle i, m \rangle, c'' \rangle\rangle \cdot e'',$$

and the following are true:

  (1) if there is $\langle o, m_o \rangle \in \mathsf{act}_{\text{out}} \langle I, O \rangle$ and c''_o such that $c' \xrightarrow{\langle o,m_o \rangle}_{\eta(C)\circ\varepsilon} c''_o$, then there is c''' such that $c'' \xrightarrow{\langle o,m_o \rangle}_{\eta(C)\circ\varepsilon} c'''$, $c''_o \xrightarrow{\langle i,m \rangle}_{\eta(C)\circ\varepsilon} c'''$, and $\langle \langle o, m_o \rangle, c''' \rangle$ is eventually in $\langle c'', e'' \rangle$;

  (2) if there is c''_τ such that $c' \xrightarrow{\tau}_{\eta(C)\circ\varepsilon} c''_\tau$, then either τ is eventually in $\langle c'', e'' \rangle$, or τ eventually need not be in $\langle c'', e'' \rangle$;

(iii) there is $\langle o, m \rangle \in \mathsf{act}_{\text{out}} \langle I, O \rangle$, c'', and e'' such that $c' \xrightarrow{\langle o,m \rangle}_{\eta(C)\circ\varepsilon} c''$, $\langle c'', e'' \rangle \in X$,

$$e' = \langle\langle\langle o, m \rangle, c'' \rangle\rangle \cdot e'',$$

and if there is c''_τ such that $c' \xrightarrow{\tau}_{\eta(C)\circ\varepsilon} c''_\tau$, then either τ is eventually in $\langle c'', e'' \rangle$, or τ eventually need not be in $\langle c'', e'' \rangle$;

(iv) there is c'' and e'' such that $c' \xrightarrow{\tau}_{\eta(C)\circ\varepsilon} c''$, $\langle c'', e'' \rangle \in X$,

$$e' = \langle\langle \tau, c'' \rangle\rangle \cdot e'',$$

and if there is $\langle o, m \rangle \in \mathsf{act}_{\text{out}} \langle I, O \rangle$ and c''_o such that $c' \xrightarrow{\langle o,m \rangle}_{\eta(C)\circ\varepsilon} c''_o$, then there is c''' such that $c'' \xrightarrow{\langle o,m \rangle}_{\eta(C)\circ\varepsilon} c'''$, $c''_o \xrightarrow{\tau}_{\eta(C)\circ\varepsilon} c'''$, and $\langle\langle o, m \rangle, c''' \rangle$ is eventually in $\langle c'', e'' \rangle$.

Definition 4.3.2. A *sequential asynchronous process coalgebra* of type $\langle I, O \rangle$ is an $(\mathsf{act}\,\langle I, O \rangle)$-labelled execution coalgebra $\langle C, \varepsilon \rangle$ such that the following are true:

(a) for any $c \in C$ and any $\langle i, m \rangle \in \mathsf{act}_{\text{in}} \langle I, O \rangle$, there is c' such that $c \xrightarrow{\langle i,m \rangle}_{\eta(C)\circ\varepsilon} c'$;

(b) for every c, c'_1, and c'_2, and any $\langle i, m \rangle \in \mathsf{act}_{\text{in}} \langle I, O \rangle$, if $c \xrightarrow{\langle i,m \rangle}_{\eta(C)\circ\varepsilon} c'_1$ and $c \xrightarrow{\langle i,m \rangle}_{\eta(C)\circ\varepsilon} c'_2$, then $c'_1 = c'_2$;

(c) for every c, c'_1, and c'_2, and any $\langle i_1, m_1 \rangle, \langle i_1, m_2 \rangle \in \mathsf{act}_{\text{in}} \langle I, O \rangle$, if $c \xrightarrow{\langle i_1,m_1 \rangle}_{\eta(C)\circ\varepsilon} c'_1$ $c \xrightarrow{\langle i_2,m_2 \rangle}_{\eta(C)\circ\varepsilon} c'_2$, and $i_1 \neq i_2$, then there is c'' such that $c'_1 \xrightarrow{\langle i_2,m_2 \rangle}_{\eta(C)\circ\varepsilon} c''$ and $c'_2 \xrightarrow{\langle i_1,m_1 \rangle}_{\eta(C)\circ\varepsilon} c''$;

(d) for every c, c'_i, and c'_o, any $\langle i, m_i \rangle \in \mathsf{act}_{\text{in}} \langle I, O \rangle$, and any $\langle o, m_o \rangle \in \mathsf{act}_{\text{out}} \langle I, O \rangle$, if $c \xrightarrow{\langle i,m_i \rangle}_{\eta(C)\circ\varepsilon} c'_i$ and $c \xrightarrow{\langle o,m_o \rangle}_{\eta(C)\circ\varepsilon} c'_o$, then there is c'' such that $c'_i \xrightarrow{\langle o,m_o \rangle}_{\eta(C)\circ\varepsilon} c''$ and $c'_o \xrightarrow{\langle i,m_i \rangle}_{\eta(C)\circ\varepsilon} c''$;

(e) for every c, c'_1, and c'_2, and any $\langle o_1, m_1 \rangle, \langle o_1, m_2 \rangle \in \mathsf{act}_{\text{out}} \langle I, O \rangle$ if $c \xrightarrow{\langle o_1,m_1 \rangle}_{\eta(C)\circ\varepsilon} c'_1$ and $c \xrightarrow{\langle o_2,m_2 \rangle}_{\eta(C)\circ\varepsilon} c'_2$, then there is c'' such that $c'_1 \xrightarrow{\langle o_2,m_2 \rangle}_{\eta(C)\circ\varepsilon} c''$ and $c'_2 \xrightarrow{\langle o_1,m_1 \rangle}_{\eta(C)\circ\varepsilon} c''$;

(f) for every c, c'_o, and c'_τ, and any $\langle o, m \rangle \in \mathsf{act}_{\text{out}} \langle I, O \rangle$, if $c \xrightarrow{\langle o,m \rangle}_{\eta(C)\circ\varepsilon} c'_o$ and $c \xrightarrow{\tau}_{\eta(C)\circ\varepsilon} c'_\tau$, then there is c'' such that $c'_o \xrightarrow{\tau}_{\eta(C)\circ\varepsilon} c''$ and $c'_\tau \xrightarrow{\langle o,m \rangle}_{\eta(C)\circ\varepsilon} c''$;

(g) for every c, c'_o, and c'', and any $\langle o, m \rangle \in \mathsf{act}_{\text{out}} \langle I, O \rangle$, if $c \xrightarrow{\langle o,m \rangle}_{\eta(C)\circ\varepsilon} c'_o$ and $c \xrightarrow{\tau}_{\eta(C)\circ\varepsilon} c''$, then there is c'_τ such that $c \xrightarrow{\tau}_{\eta(C)\circ\varepsilon} c'_\tau$ and $c'_\tau \xrightarrow{\langle o,m \rangle}_{\eta(C)\circ\varepsilon} c''$;

(h) for every c and e, $c \rhd_E e$ if and only if $\langle c, e \rangle$ is fair in $\langle C, \varepsilon \rangle$.

Clearly, sequential asynchronous process systems and small sequential asynchronous process coalgebras are the same objects in different guises.

Assume a sequential asynchronous process system $\langle S, E \rangle$ of type $\langle I, O \rangle$.

Assume a sequential asynchronous process coalgebra $\langle C, \varepsilon\rangle$ of type $\langle I, O\rangle$.

The following is immediate from :

Proposition 4.3.3. *The following are true:*

(a) the $(\mathsf{act}\,\langle I, O\rangle)$*-labelled execution coalgebra* $\langle S, \mathsf{fun}\, E\rangle$ *is a sequential asynchronous process coalgebra of type* $\langle I, O\rangle$*;*

(b) if $\langle C, \varepsilon\rangle$ *is small, then the* $(\mathsf{act}\,\langle I, O\rangle)$*-labelled execution system* $\langle C, \mathsf{rel}\,\varepsilon\rangle$ *is a sequential asynchronous process system of type* $\langle I, O\rangle$*.*

We write $\mathbf{SAPC}_{\langle I,O\rangle}$ for the category whose objects are all the asynchronous process coalgebras of type $\langle I, O\rangle$, and arrows all the homomorphisms from one asynchronous process coalgebra of type $\langle I, O\rangle$ to another.

Clearly, $\mathbf{SAPC}_{\langle I,O\rangle}$ is a full subcategory of $(\mathsf{Pow} \circ \mathsf{Seq} \circ (\mathsf{act}\,\langle I, O\rangle \times \mathsf{Id}))$-$\mathbf{Coalg}$.

The following is straightforward from Proposition 3.3.1 and a routine proof by cases:

Theorem 4.3.4. $\mathbf{SAPC}_{\langle I,O\rangle}$ *is a* $(\mathsf{Pow} \circ \mathsf{Seq} \circ (\mathsf{act}\,\langle I, O\rangle \times \mathsf{Id}))$*-covariety.*

The following is immediate from Theorem 3.5.3 and 4.3.4:

Corollary 4.3.5. *There is a sequential asynchronous process coalgebra of type* $\langle I, O\rangle$ *that is final in* $\mathbf{SAPC}_{\langle I,O\rangle}$*.*

4.4 Axiomatization

Corollary 4.3.5 is the basis for our axiomatization. Essentially, what are axioms state are that processes are members of a sequential asynchronous process coalgebra of type $\langle I, O\rangle$ that is final in $\mathbf{SAPC}_{\langle I,O\rangle}$. This approach to the definition of a process is inspired by [2] and [57].

We will not be too obsessed with formalistic issues, as we want these axioms expressed in the same manner they are to be used.

In the following, we will used accented and subscripted variants of p for processes. We will write $\langle \mathrm{P}, \rhd\rangle$ for the class of all process and the cooperation of the corresponding coalgebra. Essentially, this is a metalinguistic device to refer to the model of the theory from inside the theory. This will enable us to express an extremal axiom that guarantees that are coalgebra is final in $\mathbf{SAPC}_{\langle I,O\rangle}$. And the consistency of that axiom will be a consequence of Corollary 4.3.5.

Axiom of Unimpeded Input. *For every p and any $\langle i, m\rangle \in \mathsf{act}_{\text{in}} \langle I, O\rangle$, there is p' such that $p \xrightarrow{\langle i,m\rangle} p'$.*

Axiom of Determinate Input. *For every p, p'_1, and p'_2, and any $\langle i, m\rangle \in \mathsf{act}_{\text{in}} \langle I, O\rangle$, if $p \xrightarrow{\langle i,m\rangle} p'_1$ and $p \xrightarrow{\langle i,m\rangle} p'_2$, then $p'_1 = p'_2$.*

Axiom of Independent Input. *For every p, p'_1, and p'_2, and any $\langle i_1, m_1\rangle, \langle i_1, m_2\rangle \in \mathsf{act}_{\text{in}} \langle I, O\rangle$, if $p \xrightarrow{\langle i_1,m_1\rangle} p'_1$ $p \xrightarrow{\langle i_2,m_2\rangle} p'_2$, and $i_1 \neq i_2$, then there is p'' such that $p'_1 \xrightarrow{\langle i_2,m_2\rangle} p''$ and $p'_2 \xrightarrow{\langle i_1,m_1\rangle} p''$.*

Axiom of Irrevocable Output. *The following are true:*

(a) for every p, p'_i, and p'_o, any $\langle i, m_i\rangle \in \mathsf{act}_{\text{in}} \langle I, O\rangle$, and any $\langle o, m_o\rangle \in \mathsf{act}_{\text{out}} \langle I, O\rangle$, if $p \xrightarrow{\langle i,m_i\rangle} p'_i$ and $p \xrightarrow{\langle o,m_o\rangle} p'_o$, then there is p'' such that $p'_i \xrightarrow{\langle o,m_o\rangle} p''$ and $p'_o \xrightarrow{\langle i,m_i\rangle} p''$;

(b) for every p, p'_1, and p'_2, and any $\langle o_1, m_1\rangle, \langle o_1, m_2\rangle \in \mathsf{act}_{\text{out}} \langle I, O\rangle$ if $p \xrightarrow{\langle o_1,m_1\rangle} p'_1$ and $p \xrightarrow{\langle o_2,m_2\rangle} p'_2$, then there is p'' such that $p'_1 \xrightarrow{\langle o_2,m_2\rangle} p''$ and $p'_2 \xrightarrow{\langle o_1,m_1\rangle} p''$;

(c) for every p, p'_o, and p'_τ, and any $\langle o, m\rangle \in \mathsf{act}_{\text{out}} \langle I, O\rangle$, if $p \xrightarrow{\langle o,m\rangle} p'_o$ and $p \xrightarrow{\tau} p'_\tau$, then there is p'' such that $p'_o \xrightarrow{\tau} p''$ and $p'_\tau \xrightarrow{\langle o,m\rangle} p''$.

Postulate of Delegated Output. *For every p, p'_o, and p'', and any $\langle o, m\rangle \in \mathsf{act}_{\text{out}} \langle I, O\rangle$, if $p \xrightarrow{\langle o,m\rangle} p'_o$ and $p \xrightarrow{\tau} p''$, then there is p'_τ such that $p \xrightarrow{\tau} p'_\tau$ and $p'_\tau \xrightarrow{\langle o,m\rangle} p''$.*

Axiom of Finite Delay. *For every p and e, $p \rhd e$ if and only if $\langle p, e\rangle$ is fair in $\langle \mathrm{P}, \rhd\rangle$.*

The Extremal Axiom. *$\langle \mathrm{P}, \rhd\rangle$ is final in $\mathbf{SAPC}_{\langle I,O\rangle}$.*

The Extremal Axiom can further be reduced in the obvious way within the language of the theory using Theorem 3.5.3.

4.5 The Postulate of Delegated Output

The Postulate of Delegated Output stands out, as it is the only one we have titled a "postulate" instead of an "axiom". This is done with the purpose of initiating a discussion about its use and connection to the notion of asynchrony. Intuitively, even though it is practically assumed in every formalization of asynchrony that we know of, we would like to exclude it. Because we would like to consider computational processes running asynchronously on a non-distributed system, and exchanging messages over say queues or buffer as asynchronous. But this creates problems.

In [14], Brock and Ackerman came up with their famous anomaly that proved that relational semantics was never going to be sufficient for the denotational characterization of interactive systems. The following example is from [54], and demonstrates two programs that have the same history relation, yet different behaviour when in a feedback configuration:

```
either
    write $o : 0$
    block read $i$
    write $o : 0$
or
    block read $a$
    write $o : 0$
    write $o : 1$
end either
```

```
either
    write $o : 0$
    block read $i$
    write $o : 0$
or
    block read $i$
    write $o : 0$
    write $o : 1$
or
    write $o : 0$
    block read $i$
    write $o : 1$
end either
```

In [29], Jonsson proved that in order to obtain a compositional semantics, one must at least add enough information to describe the behaviour of a process as observed by a linear, sequential observer. And as it turns out, his result rests on the Postulate of Delegated Output.

The next example shows two different components that display the same behaviour with respect to such an observer, but still behave differently in a feedback configuration. The pseudocode below is understood as the program of a sequential process that reads from a buffer i and writes to a buffer o. Notice that reads are internal actions, while writes are external. And this is what violates the Postulate of Delegated Output: each of these two processes, when in feedback, after writing something to its own buffer, that written value is there to be read the next time a read is attempted. The reader is invited to verify that the traces of the two processes agree as they are, but disagree when in feedback.

```
either
    write $o : 0$
    write $o : 0$
or
    write $o : 0$
    if read $i = $ void then
        write $o : 0$
    end if
end either
```

```
if read $i \neq$ void
    write $o : 0$
    write $o : 0$
else
    write $o : 0$
    if read $i = $ void then
        write $o : 0$
    end if
end if
```

It is not hard to see that the problem is created only in direct feedback, and so perhaps it is instead that mechanism that is ill conceived. But more work is required to understand this new kind of anomaly.

Bibliography

[1] Karl Abrahamson. *Decidability and Expressiveness of Logics of Programs.* PhD thesis, University of Washington at Seattle, 1980.

[2] Peter Aczel. *Non-well-founded Sets.* Number 14 in Lecture Notes. CLSI, 1988.

[3] Peter Aczel. Final universes of processes. In *Proceedings of the 9th International Conference on Mathematical Foundations of Programming Semantics*, pages 1–28, London, UK, 1994. Springer-Verlag.

[4] Peter Aczel. A semantic universe for the study of fairness. VERY ROUGH AND INCOMPLETE DRAFT, October 1996.

[5] Peter Aczel and Nax Paul Mendler. A final coalgebra theorem. In *Category Theory and Computer Science*, pages 357–365, London, UK, 1989. Springer-Verlag.

[6] Jiří Adámek, Horst Herrlich, and George Strecker. *Abstract and Concrete Categories.* New York, NY, USA, 2004.

[7] Jiří Adámek, Stefan Milius, and Jiří Velebil. On coalgebra based on classes. *Theoretical Computer Science*, 316(1-3):3–23, 2004. Recent Developments in Domain Theory: A collection of papers in honour of Dana S. Scott.

[8] Bowen Alpern and Fred B. Schneider. Defining liveness. *Information Processing Letters*, 21(4):181–185, 1985.

[9] Jos C. M. Baeten. The total order assumption. In *Proceedings of the First North American Process Algebra Workshop*, NAPAW '92, pages 231–240, London, UK, 1993. Springer-Verlag.

[10] Jos C. M. Baeten. A brief history of process algebra. *Theoretical Computer Science*, 335(2-3):131–146, 2005.

[11] Jon Barwise and Lawrence Moss. *Vicious Circles.* Number 60 in Lecture Notes. CLSI, 1996.

[12] Martin Berger. An interview with Robin Milner. http://www.informatics.sussex.ac.uk/users/mfb21/interviews/milner/, September 2003.

[13] Garrett Birkhoff. On the structure of abstract algebras. *Mathematical Proceedings of the Cambridge Philosophical Society*, 31(04):433–454, 1935.

[14] J. Dean Brock and William B. Ackerman. Scenarios: A model of non-determinate computation. In J. Díaz and I. Ramos, editors, *Formalization of Programming Concepts*, volume 107 of *Lecture Notes in Computer Science*, pages 252–259. Springer Berlin / Heidelberg, 1981.

[15] Daniela Cancila, Furio Honsell, and Marina Lenisa. Some properties and some problems on set functors. *Electronic Notes in Theoretical Computer Science*, 164(1):67–84, 2006. Proceedings of the Eighth Workshop on Coalgebraic Methods in Computer Science (CMCS 2006), Eighth Workshop on Coalgebraic Methods in Computer Science.

[16] Constantin Courcoubetis, Moshe Y. Vardi, and Pierre Wolper. Reasoning about fair concurrent programs. In *Proceedings of the eighteenth annual ACM symposium on Theory of computing*, STOC '86, pages 283–294, New York, NY, USA, 1986. ACM.

[17] Philippe Darondeau. About fair asynchrony. *Theoretical Computer Science*, 37:305–336, 1985.

[18] Philippe Darondeau and Laurent Kott. On the observational semantics of fair parallelism. In Josep Diaz, editor, *Automata, Languages and Programming*, volume 154 of *Lecture Notes in Computer Science*, pages 147–159. Springer Berlin / Heidelberg, 1983.

[19] Edsger W. Dijkstra. *A Discipline of Programming.* Prentice-Hall, 1976.

[20] E. Allen Emerson. Alternative semantics for temporal logics. *Theoretical Computer Science*, 26(1-2):121–130, 1983.

[21] E. Allen Emerson and Joseph Y. Halpern. "Sometimes" and "not never" revisited: On branching versus linear time temporal logic. *J. ACM*, 33:151–178, January 1986.

[22] Marco Forti and Furio Honsell. Set theory with free construction principles. *Annali della Scuola Normale Superiore di Pisa, Classe di Scienze, 4^e série*, 10(3):493–522, 1983.

[23] H. Peter Gumm. Elements of the general theory of coalgebras. Lecture Notes for LUATCS'99 at Rand Africaans University, Johannesburg, South Africa, 1999.

[24] Matthew Hennessy. Axiomatising finite delay operators. *Acta Informatica*, 21:61–88, 1984.

[25] Matthew Hennessy and Robin Milner. Algebraic laws for nondeterminism and concurrency. *J. ACM*, 32:137–161, January 1985.

[26] Matthew Hennessy and Colin Stirling. The power of the future perfect in program logics. *Information and Control*, 67(1-3):23–52, 1985.

[27] Thomas T. Hildebrandt. A fully abstract presheaf semantics of SCCS with finite delay. *Electronic Notes in Theoretical Computer Science*, 29:102–126, 1999. CTCS '99, Conference on Category Theory and Computer Science.

[28] Bart Jacobs and Jan J. M. M. Rutten. A Tutorial on (Co)Algebras and (Co)Induction. *EATCS Bulletin*, 62:62–222, 1997.

[29] Bengt Jonsson. A fully abstract trace model for dataflow networks. In *Proceedings of the 16th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, POPL '89, pages 155–165, New York, NY, USA, 1989. ACM.

[30] Richard M. Karp and Raymond E. Miller. Parallel program schemata. *Journal of Computer and System Sciences*, 3(2):147–195, 1969.

[31] Yasuo Kawahara and Masao Mori. A small final coalgebra theorem. *Theoretical Computer Science*, 233(1-2):129–145, 2000.

[32] Robert M. Keller. Formal verification of parallel programs. *Commun. ACM*, 19(7):371–384, 1976.

[33] Joachim Lambek. A fixpoint theorem for complete categories. *Mathematische Zeitschrift*, 103:151–161, 1968.

[34] Leslie Lamport. "Sometime" is sometimes "not never": On the temporal logic of programs. In *Proceedings of the 7th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, POPL '80, pages 174–185, New York, NY, USA, 1980. ACM.

[35] Saunders Mac Lane. *Categories for the Working Mathematician.* Number 5 in Graduate Texts in Mathematics. Springer, 2nd edition, 1998.

[36] Robin Milner. Synthesis of communicating behaviour. In *7th MFCS: Mathematical Foundations of Computer Science*, volume 64 of *Lecture Notes in Computer Science*, pages 71–83. Springer, 1978.

[37] Robin Milner. *A Calculus of Communicating Systems*, volume 92 of *Lecture Notes in Computer Science*. Springer, 1980.

[38] Robin Milner. A finite delay operator in synchronous CCS. Technical Report CSR-116-82, University of Edinburgh, 1982.

[39] Robin Milner. Calculi for synchrony and asynchrony. *Theoretical Computer Science*, 25(3):267–310, 1983.

[40] Robin Milner. A calculus of communicating systems. Report ECS-LFCS-86-7, Laboratory for Foundations of Computer Science, Department of Computer Science, University of Edinburgh, August 1986. (First published by Springer-Verlag as Vol.92 of Lecture Notes in Computer Science).

[41] Robin Milner. *Communication and Concurrency.* Prentice-Hall, Upper Saddle River, NJ, USA, 1989.

[42] Robin Milner and Mads Tofte. Co-induction in relational semantics. *Theoretical Computer Science*, 87(1):209–220, 1991.

[43] Edward F. Moore. Gedanken-experiments on sequential machines. *Automata Studies*, 34:129–153, 1956.

[44] Lawrence S. Moss and Norman Danner. On the foundations of corecursion. *Logic Journal of IGPL*, 5(2):231–257, 1997.

[45] David Park. On the semantics of fair parallelism. In Dines Bjoorner, editor, *Abstract Software Specifications*, volume 86 of *Lecture Notes in Computer Science*, pages 504–526. Springer Berlin / Heidelberg, 1980.

[46] David Park. Concurrency and automata on infinite sequences. In Peter Deussen, editor, *Theoretical Computer Science*, volume 104 of *Lecture Notes in Computer Science*, pages 167–183. Springer Berlin / Heidelberg, 1981.

[47] Benjamin C. Pierce. *Basic Category Theory for Computer Scientists.* Foundations of Computing. The MIT Press, 1991.

[48] Gordon D. Plotkin. A structural approach to operational semantics (Aarhus notes). Technical Report DAIMI FN–19, Computer Science Department, Aarhus University, September 1981.

[49] Gordon D. Plotkin. The origins of structural operational semantics. *Journal of Logic and Algebraic Programming*, 60-61:3–15, 2004.

[50] Amir Pnueli. The temporal semantics of concurrent programs. In Gilles Kahn, editor, *Semantics of Concurrent Computation*, volume 70 of *Lecture Notes in Computer Science*, pages 1–20. Springer Berlin / Heidelberg, 1979.

[51] John Power and Hiroshi Watanabe. An axiomatics for categories of coalgebras. *Electronic Notes in Theoretical Computer Science*, 11:158–175, 1998. CMCS '98, First Workshop on Coalgebraic Methods in Computer Science.

[52] Vaughan R. Pratt. Process logic: Preliminary report. In *Proceedings of the 6th ACM SIGACT-SIGPLAN symposium on Principles of programming languages*, POPL '79, pages 93–100, New York, NY, USA, 1979. ACM.

[53] M. Reynolds. An axiomatization of full computation tree logic. *The Journal of Symbolic Logic*, 66(3):1011–1057, 2001.

[54] James R. Russell. Full abstraction for nondeterministic dataflow networks. In *Foundations of Computer Science, 1989., 30th Annual Symposium on*, pages 170–175, 1989.

[55] Jan J. M. M. Rutten. A calculus of transition systems (towards universal coalgebra). Technical report, CWI (Centre for Mathematics and Computer Science), Amsterdam, The Netherlands, The Netherlands, 1995.

[56] Jan J. M. M. Rutten. Universal coalgebra: a theory of systems. *Theor. Comput. Sci.*, 249(1):3–80, 2000.

[57] Jan J. M. M. Rutten and Daniele Turi. On the foundations of final semantics: Non-standard sets, metric spaces, partial orders. In *Proceedings of the REX Workshop on Sematics: Foundations and Applications*, pages 477–530, London, UK, 1993. Springer-Verlag.

[58] Davide Sangiorgi. On the origins of bisimulation and coinduction. *ACM Trans. Program. Lang. Syst.*, 31(4):1–41, 2009.

[59] Peter Selinger. First-order axioms for asynchrony. In Antoni Mazurkiewicz and Józef Winkowski, editors, *CONCUR '97: Concurrency Theory*, volume 1243 of *Lecture Notes in Computer Science*, pages 376–390. Springer Berlin / Heidelberg, 1997.

[60] Michael B. Smyth and Gordon D. Plotkin. The category-theoretic solution of recursive domain equations. *SIAM Journal on Computing*, 11(4):761–783, 1982.

[61] Alfred Tarski. A lattice-theoretical fixpoint theorem and its applications. *Pacific J. of Mathematics*, 5:285–309, 1955.

[62] Terese. *Term Rewriting Systems*. Number 55 in Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 2003.

[63] Hermann Weyl. Mathematics and logic. *The American Mathematical Monthly*, 53(1):2–13, 1946.