# Labelled Execution Systems

*Eleftherios Matsikoudis*
*Edward A. Lee*

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2012-64
http://www.eecs.berkeley.edu/Pubs/TechRpts/2012/EECS-2012-64.html

May 7, 2012

# Report Documentation Page

*Form Approved*
*OMB No. 0704-0188*



| 1. REPORT DATE | 2. REPORT TYPE | 3. DATES COVERED |
|---|---|---|
| **07 MAY 2012** | | **00-00-2012 to 00-00-2012** |

| 4. TITLE AND SUBTITLE | 5a. CONTRACT NUMBER |
|---|---|
| **Labelled Execution Systems** | 5b. GRANT NUMBER |
| | 5c. PROGRAM ELEMENT NUMBER |
| 6. AUTHOR(S) | 5d. PROJECT NUMBER |
| | 5e. TASK NUMBER |
| | 5f. WORK UNIT NUMBER |
| 7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **University of California, Berkeley,Department of Electrical Engineering and Computer Sciences,Berkeley,CA,94720** | 8. PERFORMING ORGANIZATION REPORT NUMBER |
| 9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) | 10. SPONSOR/MONITOR'S ACRONYM(S) |
| | 11. SPONSOR/MONITOR'S REPORT NUMBER(S) |

12. DISTRIBUTION/AVAILABILITY STATEMENT
**Approved for public release; distribution unlimited**

13. SUPPLEMENTARY NOTES

14. ABSTRACT

**Interleaving theories have traditionally failed to integrate a satisfactory treatment of the so-called nite delay property". This is generally attributed to the expansion law of such theories, but actually the problem is rooted in the concept of labelled transition system. 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, in a canonical way, a suitable concept of bisimilarity among such systems, study the conditions under which that concept agrees with the intuitive understanding of equivalence of branching structure that one has for these systems, and examine their relationship with labelled transition systems, precisely characterizing the di erence in expressive power and branching complexity between the two kinds of systems.**

15. SUBJECT TERMS

| 16. SECURITY CLASSIFICATION OF: | | | 17. LIMITATION OF ABSTRACT | 18. NUMBER OF PAGES | 19a. NAME OF RESPONSIBLE PERSON |
|---|---|---|---|---|---|
| a. REPORT **unclassified** | b. ABSTRACT **unclassified** | c. THIS PAGE **unclassified** | **Same as Report (SAR)** | **58** | |

Acknowledgement

This work was supported in part by the Center for Hybrid and Embedded Software Systems (CHESS) at UC Berkeley, which receives support from the National Science Foundation (NSF awards \#0720882 (CSR-EHS: PRET), \#1035672 (CPS: PTIDES), and \#0931843 (ActionWebs)), the U. S. Army Research Lab (ARL \#W911NF-11-2-0038), the Air Force Research Lab (AFRL), the Multiscale Systems Center (MuSyC), one of six research centers funded under the Focus Center Research Program, a Semiconductor Research Corporation program, and the following companies: Bosch, National Instruments, Thales, and Toyota.

# Labelled Execution Systems*

Eleftherios Matsikoudis Edward A. Lee

**Abstract**

Interleaving theories have traditionally failed to integrate a satisfactory treatment of the so-called "finite delay property". This is generally attributed to the expansion law of such theories, but actually, the problem is rooted in the concept of labelled transition system. 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, in a canonical way, a suitable concept of bisimilarity among such systems, study the conditions under which that concept agrees with the intuitive understanding of equivalence of branching structure that one has for these systems, and examine their relationship with labelled transition systems, precisely characterizing the difference in expressive power and branching complexity between the two kinds of systems.

## 1 Introduction

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 the marriage of Plotkin's structural operational semantics (see [53]) and Keller's named transition systems (see [36]) (see [46, chap. 12], [13], [54], [11]). This marriage was the work of Robin Milner, and is most clearly expounded in [46], but was already present in [42], 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 [43], [46]), a typical equation asserted by the law is the following:

$$a.\mathbf{0} \mid b.\mathbf{0} = a.b.\mathbf{0} + b.a.\mathbf{0}. \tag{1}$$

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. And the intended meaning of (1) 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 [42, 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,

* This work was supported in part by the Center for Hybrid and Embedded Software Systems (CHESS) at UC Berkeley, which receives support from the National Science Foundation (NSF awards #0720882 (CSR-EHS: PRET), #1035672 (CPS: PTIDES), and #0931843 (ActionWebs)), the U. S. Army Research Lab (ARL #W911NF-11-2-0038), the Air Force Research Lab (AFRL), the Multiscale Systems Center (MuSyC), one of six research centers funded under the Focus Center Research Program, a Semiconductor Research Corporation program, and the following companies: Bosch, National Instruments, Thales, and Toyota.

> 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 admit 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:

$$\mathbf{fix}(X = a.X) \mid \mathbf{fix}(X = b.X) = \mathbf{fix}(X = a.X + b.X). \tag{2}$$

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 both 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. Interleaving became synonymous with *bounded indeterminacy* (see [24, chap. 9]), and the observational view was robbed of its power to express properties like *fairness* (see [55]) or the *finite delay property* (see [34]) (e.g., see [50]).

Of course, it is not the expansion law per se that is to blame for this confusion. In order to develop an observational theory of processes of some kind, one must decide 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 [38], [40], [9]).

To risk an analogy, 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]). And this rejection of the notion of actual infinity is detrimental to the expressiveness of the approach. A process is demoted to a graph, and every path through that graph is promoted to an execution of that process. This is what renders indeterminacy bounded, and the real reason behind the aforementioned confusion.

We prefer what we might say is a more *classical* approach, where we fix the unit of observation at the level of a complete execution of a process. And for that, we need a different kind of mathematical structure. The purpose of this work then is to introduce a new kind of system, in which, instead of labelled transitions, one has, essentially, sequences of labelled transitions. We call systems of this kind *labelled execution systems*.

By changing the unit of observation, we change the kind of experiment that we may use to investigate the behaviour of an agent (e.g., see [43, pp. 10–12]). In principle, we can adapt Milner's argument to this new

kind of experiment, and weather permitting, identify the behaviour of an agent with what we would call the *branching structure* of a labelled execution system modelling that agent. But although motivated by process algebra and its observational approach to concurrency theory, this work is not about the use of labelled execution systems as models of behaviour. Our goal here is to develop a thorough understanding of what this so-called "branching structure" of a labelled execution system is, more or less independently of what it might be used to model.

It is instructive to look back on how our understanding of the notion of branching structure evolved in the case of labelled transition systems. The notion originated in Milner's work, albeit in an indirect way: two systems were understood to have the same branching structure just as long as they were "observationally equivalent". Its use as a characterization of the behaviour of a communicating agent as experienced by an external observer experimenting on that agent generated some controversy (e.g., see [15]), but the notion itself proved to be an important one, both in theory and practice. Soon after its inception, it was honed by David Parks's concept of *bisimilarity* (see [51]), and before long, parlayed through Peter Aczel's work (see [3], [6]) into an overarching theory of *universal coalgebra* (e.g., see [58], [28]). This produced not only a direct formalization of the notion, in the shape of the concept of final coalgebra, but also a purely mathematical justification for it. It was now possible to speak of the notion as part of an extensional approach to systems, rather than an observational view of processes, and perhaps more importantly, expand its scope to any kind of mathematical structure amenable to coalgebraic treatment.

Here, we subject our newly introduced systems to such treatment. We find that the theory of coalgebra offers a convenient and powerful analytical framework for studying these systems and working out their relationship with the more traditional transition-based systems.

The main contributions of this work are the following:

- we define labelled execution systems (see Definition 4.1);
- we represent labelled execution systems coalgebraically (see Definition 4.2 and Proposition 4.3), and use that representation to obtain, in a canonical way, a suitable concept of bisimilarity among such systems (see Proposition 4.4, Definition 4.5, and Proposition 4.6);
- we show that for the obtained concept of bisimilarity to agree with the intuitive understanding of equivalence of branching structure that one has for labelled execution systems, two properties are necessary: suffix closure and fusion closure (see Section 4.3);
- we truncate the executions of a labelled execution system to get a uniquely determined underlying labelled transition system, and prove that bisimilarity among labelled execution systems implies bisimilarity among their respective underlying labelled transition systems (see Theorem 4.13);
- we show that the converse fails if any of the following four properties fails: suffix closure, fusion closure, limit closure, and what we call *impossibility of indeterminate termination* (see Section 4.5);
- we prove that essentially these four properties constitute a complete characterization of what we call *generable* labelled execution systems, namely labelled executions systems that are generated from their respective underlying labelled transition systems (see Theorem 4.24);
- we prove that bisimilarity among generable labelled execution systems is equivalent to bisimilarity among their respective underlying labelled transition systems (see Theorem 4.32), rendering the characterization of generable labelled execution systems a characterization of the difference in complexity of branching structure between arbitrary labelled executions systems and labelled transition systems (see also Theorem 4.25, Corollary 4.26 and Theorem 4.33).

The rest of this document is organized into five sections. In Section 2, we provide a brief, but rather comprehensive, overview of the theory of universal coalgebra. In Section 3, we review the concept of labelled transition system and Park's concept of bisimilarity, and go over their coalgebraic representation. The main body of this work is in Section 4, where we define labelled execution systems, work out their

coalgebraic representation, and use that representation to study their branching structure, and characterize their relationship with labelled transition systems. In Section 5, we discuss related work. We conclude in Section 6 with a few comments on the implications of our results, and some directions for future work.

We would like to finish this introduction with a few remarks.

First, throughout this work, we maintain a formal distinction between systems and their coalgebraic representations. This is done for the benefit of the non-coalgebraist, who we think could use a more careful exposition of the correspondence between the two, and the more application-oriented reader, who might only be interested in results on systems.

Second, working with coalgebras, we will need to make reference to a few common concepts from category theory, e.g., the concepts of category, functor, natural transformation, etc., and we will assume some minimal level of familiarity with these concepts on the reader's part. For a gentle introduction to such concepts, and category theory in general, we refer to [52].

And third, we choose the category of all classes and all class functions between them as the underlying category of our coalgebraic framework. This is a "surprisingly coalgebra-friendly category" (see [8, p. 3]), which is one of the main reasons for choosing to work with it. And while it is possible to avoid such a super-large category, by only considering endofunctors that are bounded in some suitable sense, we think that classes provide for a cleaner, unobscured presentation of our ideas (see also discussion at end of Section 3). Any concerns of foundational nature regarding the excessive size of this category can be addressed in one way or another (e.g., see [41, chap. I] or [7, chap. 2]), but a thorough treatment would be out of place here, and in any case, we will avoid impredicative constructions and comprehension principles that test the consistency of the theory.

# 2 Background

In this section, we briefly overview the theory of universal coalgebra. We deliberately include here more of the theory than is strictly needed for the purposes of this work, what we think is justified on the following grounds: (i) we think that the exposure of the broader community to the theory is, regrettably, at best limited, and (ii) we believe that the reader will be able to better appreciate our methods and results once equipped with a more rounded view of the theory. For a more detailed introduction, we refer the interested reader to [58] or [28].

## 2.1 Coalgebras

Assume an endofunctor $F$ on **Class**.[1]

**Definition 2.1.** An *$F$-coalgebra* is an ordered pair $\langle C, \gamma \rangle$ such that the following are true:

(a) $C$ is a class;

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

In general, an $F$-coalgebra $\langle C, \gamma \rangle$ will represent one or several rules for decomposing things from $C$, as determined by $\gamma$, into particular combinations of things from $C$, as encoded by $F$.

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

[1] We write **Class** for the category whose objects are all the classes, and arrows all the class functions[2].

[2] 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 $d \in D$, there is exactly one $c$ such that $\langle d, c \rangle \in G$. 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 $f$. We write $f : C_1 \to C_2$ if and only if $\mathsf{dom}\, f = C_1$ and $\mathsf{cod}\, f = C_2$.

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

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

## 2.2 Homomorphisms

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

**Definition 2.2.** 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

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

or equivalently, the following diagram commutes:

$$\begin{array}{ccc} C_1 & \xrightarrow{\quad h \quad} & C_2 \\ {\scriptstyle \gamma_1}\downarrow & & \downarrow{\scriptstyle \gamma_2} \\ F(C_1) & \xrightarrow[F(h)]{} & F(C_2) \end{array}$$

A homomorphism from one $F$-coalgebra to another is a structure-preserving map that carries the decomposition patterns of the first coalgebra to those of the second. In particular, it establishes a similarity of structure between its domain and range (see Theorem 2.10).

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 $\langle C_1, \gamma_1 \rangle$ is a *homomorphic image* of $\langle C_2, \gamma_2 \rangle$ if and only if there is a surjective homomorphism 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 2.3.** *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 [58, prop. 2.3].[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 $\langle C_1, \gamma_1 \rangle$ and $\langle C_2, \gamma_2 \rangle$.

Isomorphic $F$-coalgebras are structurally identical, and for all practical purposes, may be thought of as the same object.

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

The following is easy:

**Proposition 2.4.** *The following are true:*

*(a)* $\mathsf{id}\, C$ *is an automorphism on* $\langle C, \gamma \rangle$*;*

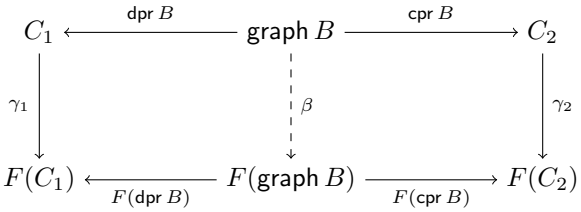
[3] Quite often, we shall cite results from bibliographic references in our proofs. And in many cases, these results will have been stated and proved in a different setting. For example, in [58], Rutten works with the category **Set** of all sets and all functions between them, not **Class**. But in all cases, their proofs will remain valid in our setting here, unchanged or only trivially modified.

*(b) if $h_1$ is a homomorphism from $\langle C_1, \gamma_1\rangle$ to $\langle C, \gamma\rangle$, and $h_2$ is a homomorphism from $\langle C, \gamma\rangle$ to $\langle C_2, \gamma_2\rangle$, then $h_2 \circ h_1$ is a homomorphism from $\langle C_1, \gamma_1\rangle$ to $\langle C_2, \gamma_2\rangle$.*

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.

## 2.3 Bisimulations

**Definition 2.5.** A *bisimulation* between $\langle C_1, \gamma_1\rangle$ and $\langle C_2, \gamma_2\rangle$ is a binary class relation[4] $B : C_1 \leftrightarrow C_2$ 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 $\langle C_2, \gamma_2\rangle$,[5] or equivalently, the following diagram commutes:




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 $c_1\ B\ c_2$.

By Definition 2.5, a binary class relation between two $F$-coalgebras is a bisimulation just as long as we can impart its graph with the structure of an $F$-coalgebra that turns the projection maps from the graph to the carriers of the two $F$-coalgebras into homomorphisms. But really, there is nothing particularly special about such an $F$-coalgebra.

**Theorem 2.6.** *$B$ 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*

$$B = h_1^{-1}\,;h_2.$$[6]

*Proof.* See [58, lem. 5.3] and [28, thm. 5.11]. □

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

The following is immediate:

**Corollary 2.7.** *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 $h_1^{-1}\,;B\,;h_2$ is a bisimulation between $\langle C_1', \gamma_1'\rangle$ and $\langle C_2', \gamma_2'\rangle$.*

[4] 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 $R$. We write $R : C_1 \leftrightarrow C_2$ if and only if $\mathsf{dom}\, R = C_1$ and $\mathsf{cod}\, R = C_2$.

[5] For every binary class relation $R$, we write $\mathsf{dpr}\, R$ for a class 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 class 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 $R$.

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

**Theorem 2.8.** *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 [28, thm. 5.6]. □

Definition 2.5 was introduced in [6] as a generalization of the concept of bisimulation between labelled transition systems (see Proposition 3.6 and 3.7). It is supposed to capture equivalence of structure. But to what extent does it?

A disturbing fact is that, unlike bisimilarity among labelled transition systems (see Section 3), the coalgebraic concept of bisimilarity is not, in general, an equivalence concept.

*Example* 2.9. 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 $\gamma_1$ 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 $\gamma_2$ 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 $\langle S_2, \gamma_2\rangle$. Thus, by Theorem 2.6, 0 and $h(0)$, and similarly, 1 and $h(1)$, are bisimilar among $\langle S_1, \gamma_1\rangle$ and $\langle S_2, \gamma_2\rangle$ (see also Theorem 2.10). 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

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

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

This discrepancy was not lost on Aczel and Mendler, who, also in [6], 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 $\gamma$, and test whether $R$ is contained in the equivalence kernel of the composite. But 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 2.9 is one that was devised in [6] for the express purpose of demonstrating this separation between the two concepts.

Here, mostly for the purpose of accessibility, we have decided to follow the approach of Rutten in [58], 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 [6] 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 [58] 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, with the sole exception of that in Example 2.9.

That said, the inadequacy of the concept of bisimulation is not felt in any but the most contrived cases. For all intents and purposes, bisimilarity is a sufficient measure of equivalence of structure. The following is a case in point:

**Theorem 2.10.** *$h$ 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 [58, thm. 2.5]. □

In a word, homomorphisms are functional bisimulations.

## 2.4 Subcoalgebras

**Definition 2.11.** 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 $\langle C, \gamma\rangle$,[7] or equivalently, the following diagram commutes:

$$\begin{array}{ccc} C' & \xrightarrow{C' \hookrightarrow C} & C \\ {\scriptstyle \gamma'}\downarrow & & \downarrow{\scriptstyle \gamma} \\ F(C') & \xrightarrow[F(C' \hookrightarrow C)]{} & F(C) \end{array}$$

[7] 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 $C_2$.

A subcoalgebra of an $F$-coalgebra is a part of that $F$-coalgebra, that is closed, in a suitably generalized sense, under the decomposition rules of the latter.

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

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

**Proposition 2.12.** *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 [58, prop. 6.1]. □

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

**Theorem 2.13.** *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 $\langle \mathsf{ran}\, h, \rho \rangle \leq \langle C_2, \gamma_2 \rangle$.*[8]

*Proof.* See [58, thm. 6.3]. □

**Theorem 2.14.** *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 $\langle \bigcup_{i \in I} C_i, \upsilon \rangle \leq \langle C, \gamma \rangle$.*

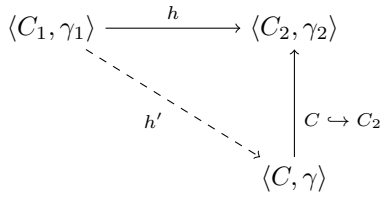
*Proof.* See [28, thm. 4.7]. □

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

**Proposition 2.15.** *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*

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

*or equivalently, the following diagram commutes:*




*Proof.* See [58, prop. 6.5]. □

Theorem 2.14 and Proposition 2.15 can be used to arrange the subcoalgebras of an $F$-coalgebra into a complete lattice (see [28, cor. 4.9]).

We conclude this brief account on subcoalgebras with The Small Subcoalgebra Lemma of [6], which is in fact equivalent to [8, thm. 2.2], namely the surprising fact that every endofunctor on **Class** is set-based.[9]

**Lemma 2.16.** *For every subset $S$ of $C$, there is a small $F$-coalgebra $\langle C', \gamma' \rangle$ such that $S \subseteq C'$ and $\langle C', \gamma' \rangle \leq \langle C, \gamma \rangle$.*

[8] 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 $f$.

[9] An endofunctor $F$ on **Class** is *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 $c = F(S \hookrightarrow C)(s)$.

*Proof.* See [6, lem. 2.2] and [8, thm. 2.2]. □

## 2.5 Direct sums

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

**Definition 2.17.** 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[10] of $\{C_i\}_{i\in I}$;

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

$$\gamma((\mathsf{inj}_j \sum_{i\in I} C_i)(c)) = F(\mathsf{inj}_j \sum_{i\in I} C_i)(\gamma_j(c)).^{11}$$

The concept of direct sum lifts the concept of disjoint union from classes to $F$-coalgebras, allowing us to merge different $F$-coalgebras into a single whole.

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

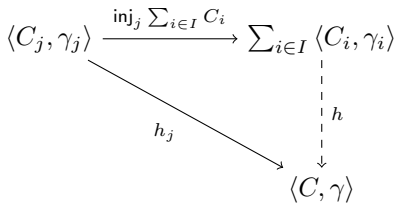
Notice that for any $j \in I$, the canonical injection map $\mathsf{inj}_j \sum_{i\in I} C_i$ is by definition a homomorphism from $\langle C_j, \gamma_j\rangle$ to $\sum_{i\in I} \langle C_i, \gamma_i\rangle$.

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

**Proposition 2.18.** *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 $j \in I$,*

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

*or equivalently, the following diagram commutes:*




*Proof.* See [28, lem. 4.1]. □

## 2.6 Final coalgebras

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 $\langle C, \gamma\rangle$.

[10] 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 $\{C_i\}_{i\in I}$.

[11] 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 $\sum_{i\in I} C_i$.

We use the word "final" here instead of the word "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 all final $F$-coalgebras are isomorphic to one another, lest there be another endomorphism, apart from the identity map, on any of them.

Final $F$-coalgebras are interesting because they are complex enough to subsume the structure of every other $F$-coalgebra, but coarse enough to do so only modulo equivalence of structure.

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 $\langle C, \gamma \rangle$.

**Theorem 2.19.** *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 2.16, 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 2.6, $R$ 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 2.8, 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 2.16, 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 2.13, 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 2.15, there is exactly one homomorphism $\pi$ 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 2.16, 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 2.14, 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 2.15, $C_1' \hookrightarrow (C_1' \cup C_2')$ 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 $R_1 = (C_1' \hookrightarrow (C_1' \cup C_2'))^{-1} \,;\, B_1$.

Let $R_2 = (C_2' \hookrightarrow (C_1' \cup C_2'))^{-1} \,;\, B_2$.

Then, by Corollary 2.7, 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 2.16, 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 2.16, 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 2.10, 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 [3], whereas the implication from (a) to (c) can be found in [28, thm. 6.4]. But the one in the reverse direction is, to our knowledge, a new result. Altogether, Theorem 2.19 is a powerful characterization of final coalgebras, justifying their prominent role as semantic models of behaviour.

By Theorem 2.19, any final $F$-coalgebra $\langle C, \gamma \rangle$ is *strongly extensional*, or equivalently, 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 [58, thm. 9.2]).

**Theorem 2.20.** *There is an $F$-coalgebra that is final in $F$-**Coalg**.*

*Proof.* See [6, thm. 2.1] and [8, thm. 2.2]. □

By now, there have been several different proofs of Theorem 2.20 (see [8] and references therein). Assuming [8, thm. 2.2], or equivalently, Lemma 2.16, the proof in [6] 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.

**Proposition 2.21.** *If $\langle C, \gamma \rangle$ is final in $F$-**Coalg**, then $\gamma$ is bijective.*

*Proof.* See [37, lem. 2.2]. □

Proposition 2.21 is known as *Lambek's Lemma*, and for many interesting endofunctors, including those considered in Section 3 and 4, it implies that the carrier of the final coalgebra is a proper class.

## 2.7 Covarieties

Assume a full[12] subcategory **S** of $F$-**Coalg**.

We say that **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 **S**, and $\langle C', \gamma' \rangle$ is a homomorphic image of $\langle C, \gamma \rangle$, then $\langle C', \gamma' \rangle$ is in **S**.

[12] A subcategory **S** of a category **C** is *full* if and only if the arrows between any two objects in **S** are all the arrows between the two objects in **C**.

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 2.22.** 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 a generalization of the concept of $\Sigma$-variety, what Birkhoff called a *family of algebras* when he introduced the concept in [14]. Here, it is of interest because it allows us to generalize the results of Section 2.6 to cases where certain kinds of structure have been systematically ruled out (e.g., see Section 4.3).

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 $\langle C, \gamma \rangle$.

Just as in $F$-**Coalg**, all final $F$-coalgebras are isomorphic to one another.

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 2.19:

**Theorem 2.23.** *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 2.19. □

The reason that we are able to reuse the proof of Theorem 2.19 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.

By Theorem 2.23, every $F$-coalgebra that is final in an $F$-covariety satisfies the coinduction proof principle.

The following is a generalization of Theorem 2.20:

**Theorem 2.24.** *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 2.13, 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 2.15, 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 $\iota$ 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 2.23, $\langle \mathsf{ran}\, h, \rho\rangle$ is final in $\mathbf{C}$. □

Theorem 2.24 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 [4, thm 2.2]).

The following is a generalization of Proposition 2.21:

**Proposition 2.25.** *For every $F$-covariety $\mathbf{C}$, if $\langle C, \gamma\rangle$ is final in $\mathbf{C}$, then $\gamma$ is bijective.*

*Proof.* See proof of Proposition 2.21. □

As with Theorem 2.23, the reason that we are able to reuse the proof of Proposition 2.21 here is that, because of the closure properties of an $F$-covariety, all relevant constructions in that proof can be carried out inside $\mathbf{C}$.

# 3 Labelled transition systems and coalgebras

The concept of labelled transition system is the paradigmatic example of a coalgebra. Indeed, the theory of coalgebra was largely inspired by that concept (e.g., see [6]). Here, we formalize it in a somewhat nonstandard way, namely using a binary relation rather than a ternary one, and go over its coalgebraic treatment anew, with the intent of drawing the reader's attention to the unity of formal treatment between this section and the next.

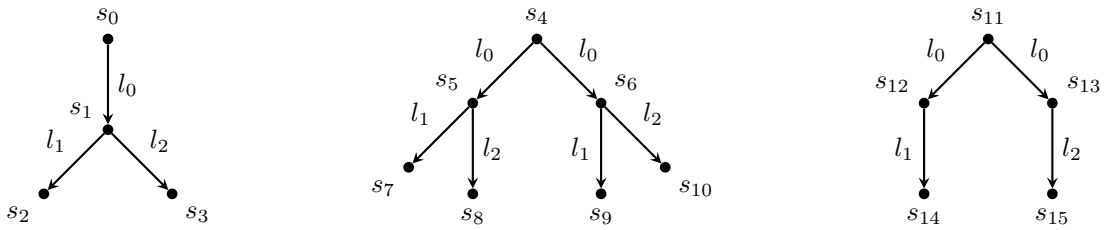



**Figure 1.** $s_0$ and $s_4$ are bisimilar; neither of them is bisimilar to $s_{11}$.

## 3.1 Labelled transition systems

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

**Definition 3.1.** An *$L$-labelled 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 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 [48], where they appeared in tabular as well as pictorial form. In their present form, they seem to have been introduced by Keller in [36], 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 [43] and the advent of process algebra.

In fact, considering how easy it is to "cook up yet another variant process calculus or algebra" (see [2, p. 39]), it is not unreasonable to suggest that a process algebra is no more than an algebra of labelled transition systems. For in the absence of a "hard and obdurate" kind of external reality, it is the model that gives meaning to form, and that model is hardly ever anything other than a labelled transition system. And really, the only, seemingly tacit constraint on the latter is that equivalence of branching structure be a congruence relation. This is the notion of equivalence corresponding to the branching end of the spectrum of [62], and "the real fruit" of process algebra as a whole.

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

**Definition 3.2.** 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 $s_1$ 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 $s_1\ B\ s_2$.

For example, consider the three diagrams of Figure 1, which are of course pictures of labelled transition systems: $s_0$ and $s_4$ are bisimilar; neither of them is bisimilar to $s_{11}$.

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.

The concept of bisimulation is due to David Park (see [51]), and is without doubt the most significant contribution of the theory of concurrency to the broader arena of computer science and mathematics at large.[13] After learning about Milner's work in [45], 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 [3]).[14] 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 [6] to bear a final coalgebra theorem and a categorical definition of bisimulation, eventually leading to the general theory of universal coalgebra glimpsed in Section 2.

## 3.2 Labelled transition coalgebras

Consider once more the concept of labelled transition system. We have formalized this as an ordered pair of a set and a binary relation. But there is another way: to look at the binary relation as a set-valued function.

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

We write $\mathsf{fun}\, R$ for a function from $S_1$ to $\mathscr{P}\, S_2$ such that for any $s_1 \in S_1$,[15]

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

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

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 3.3.** *The following are true:*

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

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

Proposition 3.3 suggests an alternative, coalgebraic formalization of the concept of $L$-labelled transition system.

[13] To be fair, the concept of bisimulation has been independently discovered in the fields of modal logic and set theory as well. See [60] for a comprehensive historical account.

[14] Forti and Honsell had already discovered and used the concept of bisimulation to that effect in [27].

[15] For every set $S$, we write $\mathscr{P}\, S$ for the *power set* of $S$, namely the set of all subsets of $S$.

We write $\mathsf{Pow}$ for the *power-set endofunctor* on **Class**, namely an endofunctor on **Class** that assigns to every class $C$ the class

$$\mathsf{Pow}\, C = \{S \mid S \text{ is a set, and } S \subseteq C\},$$

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

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

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

$$\mathsf{Pow}(L \times C) = \{S \mid S \text{ is a set, and } S \subseteq L \times C\},$$

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 $S \in \mathsf{Pow}(L \times C_1)$,

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

An $L$-labelled transition system $\langle S, T\rangle$ can then 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 $\langle C, \tau\rangle$ as an $L$-labelled transition system, namely as $\langle C, \mathsf{rel}\, \tau\rangle$, with the caveat, of course, that $C$ be a set. But this is only an arbitrary constraint on the size of a system. There is no fundamental reason why we should disqualify a system for being too large.

**Definition 3.4.** An *$L$-labelled transition coalgebra* is a $(\mathsf{Pow} \circ (L \times \mathsf{Id}))$-coalgebra.

We write $L$-**LTC** for the category whose objects are all the $L$-labelled transition coalgebras, and arrows all the homomorphisms from one $L$-labelled transition coalgebra to another.

Formally, we will treat $L$-labelled transition systems and $L$-labelled transition coalgebras as distinct concepts. But informally, we shall think of an $L$-labelled transition coalgebra as an $L$-labelled transition system, no matter how large the carrier of the coalgebra is.

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

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

The following is immediate:

**Proposition 3.5.** *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 3.6.** *$B$ 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 $\beta$ be a class function from $\mathsf{graph}\, B$ to $\mathsf{Pow}(L \times \mathsf{graph}\, B)$ such that for any $\langle c_1, c_2 \rangle \in \mathsf{graph}\, B$,

$$\beta(\langle c_1, c_2 \rangle) = \{\langle l, \langle c_1', c_2' \rangle \rangle \mid c_1 \xrightarrow{l}_{\tau_1} c_1',\ c_2 \xrightarrow{l}_{\tau_2} c_2', \text{ and } \langle c_1', c_2' \rangle \in \mathsf{graph}\, B\}.$$

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 $\langle C_2, \tau_2 \rangle$. □

The following is immediate from Proposition 3.5(a), 3.6, and the definition of bisimulation between labelled transition systems:

**Proposition 3.7.** *$B$ 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 $\langle S_2, \mathsf{fun}\, T_2 \rangle$.*

Now, 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 complex 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 the following are true:

(i) 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 $\langle S, T \rangle$;

(ii) for any $s_1, s_2 \in S$, $s_1$ and $s_2$ are bisimilar in $\langle S, T \rangle$ if and only if $s_1 = s_2$.

By an easy corollary to Proposition 3.6, bisimilarity among $L$-labelled transition coalgebras is a transitive concept, and thus, by Theorem 2.19 and Proposition 3.7, a system will satisfy conditions (i) and (ii) of our enquiry if and only if it has a coalgebraic representation that is final in $L$-**LTC**. And by Theorem 2.20, there is indeed a final coalgebra in $L$-**LTC**. But by Proposition 2.21, the carrier of every such coalgebra is, for obvious cardinality reasons, a proper class.

This should not come as a 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 bound, in a suitable sense, the endofunctor to ensure that a final coalgebra be small (e.g., see [35, 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, one could even argue that this is a natural thing to do. But 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, in one way or another, "computes" its own evolution using a fixed set of predefined resources.

# 4 Labelled execution systems and coalgebras

This section contains the main body of this work, where we introduce and study our newly proposed systems. For the most part, we treat these systems in the guise of coalgebras. But for the benefit of the non-coalgebraist, as well as the more application-oriented reader, we make sure to translate all our findings back into the language of systems.

## 4.1 Labelled execution systems

**Definition 4.1.** 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 $\mathscr{S}(L \times S)$.[16]

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 $\langle s, e\rangle \in \mathsf{graph}\, E$ an *execution* of $\langle S, E\rangle$.

Notice that an execution is an ordered pair of a state and a sequence of ordered pairs of labels and states, rather than a single odd-length alternating sequence of states and labels, 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 general idea of an execution system, if only unlabelled, has been around at least since the early days of temporal logic in computer science, in the form of a semantic structure called a *path structure* in [57] (e.g., see [56], [1], and [39]). 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. But 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 to it, we could only find a handful of sporadic instances of the general notion. We discuss them in Section 5.

## 4.2 Labelled execution coalgebras

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 [59], 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 2.5, and obtain not only the equivalence

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

concept that we seek, but a model too that is *fully abstract* with respect to that concept (see Theorem 2.19 and 2.20).[17] This is straightforward here.

We write $\mathsf{Seq}$ for the *sequence-set endofunctor* on **Class**, namely an endofunctor on **Class** that assigns to every class $C$ the class

$$\mathsf{Seq}\, C = \{s \mid \text{there is } S \text{ such that } S \subseteq C \text{ and } s \in \mathscr{S}\, S\},$$

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 actually a set, then

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

At this point, the reader may protest against the seeming circularity in the way we have specified the action of $\mathsf{Seq}$ 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 that 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 $\sigma_2$ 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 $\sigma_1$ 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.

[17] The tacit assumption here is that the instantiated concept of bisimulation does indeed induce an equivalence concept. See Example 2.9 for a case where it does not.

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 [49]). 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 [12], 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 [33]).

We now compose $\mathsf{Seq}$ 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

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

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 $s \in \mathsf{Seq}(L \times C_1)$,

$$(\mathsf{Seq}(L \times f))(s) = \begin{cases} \langle\ \rangle & \text{if } s = \langle\ \rangle; \\ \langle\langle \mathsf{first\ head}\, s, f(\mathsf{sec\ head}\, s)\rangle\rangle \cdot (\mathsf{Seq}(L \times f))(\mathsf{tail}\, s) & \text{otherwise.}[18] \end{cases}$$

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 compose $\mathsf{Pow}$ 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

$$\mathsf{Pow\,Seq}(L \times C) = \{S \mid S \text{ is a set, and } S \subseteq \mathsf{Seq}(L \times C)\},$$

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 labelled transition systems, we can now take advantage of the formal analogy between the concepts of binary relation and set-valued function, and use Proposition 3.3 to obtain our coalgebraic representation. This unity of treatment is the reward of our aforementioned formalization choice.

An $L$-labelled execution system $\langle S, E\rangle$ can then 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 $\langle C, \varepsilon\rangle$ as an $L$-labelled execution system, namely as $\langle C, \mathsf{rel}\, \varepsilon\rangle$, again with the caveat that $C$ be a set.

**Definition 4.2.** An *$L$-labelled execution coalgebra* is a $(\mathsf{Pow} \circ \mathsf{Seq} \circ (L \times \mathsf{Id}))$-coalgebra.

[18] 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 $y$.

We write $L$-**LEC** for the category whose objects are all the $L$-labelled execution coalgebras, and arrows all the homomorphisms from one $L$-labelled execution coalgebra to another.

Once more, we will formally treat $L$-labelled execution systems and $L$-labelled execution coalgebras as distinct concepts, but we shall informally think of an $L$-labelled execution coalgebra as an $L$-labelled execution system, no matter how large the carrier of the coalgebra is.

Assume an $L$-labelled execution coalgebra $\langle C, \varepsilon \rangle$.

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

The following is immediate:

**Proposition 4.3.** *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 4.6 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 $e_2 \in \mathsf{Seq}(L \times C_2)$,

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

$\mathsf{Seq}(L \times R)$ is a simple lift of $R$ to pairs of sequences over $L \times \mathsf{dom}\, R$ and $L \times \mathsf{cod}\, R$. Informally, if $\langle \langle l_0, c_0 \rangle, \langle l_1, c_1 \rangle, \ldots \rangle$ is a sequence over $\mathsf{dom}\, R$, and $\langle \langle l'_0, c'_0 \rangle, \langle l'_1, c'_1 \rangle, \ldots \rangle$ a sequence over $\mathsf{cod}\, R$, then

$$\langle \langle l_0, c_0 \rangle, \langle l_1, c_1 \rangle, \ldots \rangle \; \mathsf{Seq}(L \times R) \; \langle \langle l'_0, c'_0 \rangle, \langle l'_1, c'_1 \rangle, \ldots \rangle$$

if and only if $l_0 = l'_0$ and $c_0 \; R \; c'_0$, $l_1 = l'_1$ and $c_1 \; R \; c'_1$, etc. Our choice of notation may be justified by the fact that

$$\begin{aligned} \mathsf{Seq}(L \times R) &= \mathsf{Seq}(L \times ((\mathsf{dpr}\, R)^{-1} ; \mathsf{cpr}\, R)) \\ &= (\mathsf{Seq}(L \times \mathsf{dpr}\, R))^{-1} ; \mathsf{Seq}(L \times \mathsf{cpr}\, R). \end{aligned}$$

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

**Proposition 4.4.** *$B$ 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 $\beta$ be a class function from $\mathsf{graph}\,B$ to $\mathsf{Pow}\,\mathsf{Seq}(L \times \mathsf{graph}\,B)$ such that for any $\langle c_1, c_2\rangle \in \mathsf{graph}\,B$,

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

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 4.4 is similar to Proposition 3.6. 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 $\langle S_2, E_2\rangle$.


**Figure 2.** $s_0$ and $s_2$ are not bisimilar, even though the only execution starting from $s_0$ is in perfect agreement with the only execution starting from $s_2$.

**Definition 4.5.** 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 $e_1\ \mathsf{Seq}(L \times B)\ e_2$.

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 $s_1\ B\ s_2$.

The following is of course immediate:

**Proposition 4.6.** *$B$ 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 $\langle S_2, \mathsf{fun}\, E_2\rangle$.*

## 4.3 Abrahamson systems and coalgebras

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 $\{l_0, l_1\}$-labelled execution systems, whose executions are as depicted in the left and right frames respectively of Figure 2. $s_0$ and $s_2$ are not bisimilar, simply because there is an execution starting from $s_3$, and no execution starting from $s_1$. But why should we care if there is? The only execution starting from $s_2$ has only one step, labelled $l_0$, and is in perfect agreement with the only execution starting from $s_0$, which also has only one step, also labelled $l_0$. 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 $\{l_0, l_1, l_2\}$-labelled execution systems, whose executions are as depicted in the left and right frames respectively of Figure 3. $s_0$ and $s_5$ are bisimilar, but intuitively, there is difference in branching potential between the two states: the two executions starting from $s_0$ diverge right away at $s_0$, with steps that carry identical labels, whereas those starting from $s_5$ diverge after the first step, at $s_6$, with steps that carry different labels. Of course, the explanation here is that there is no execution starting from $s_6$, and so, conceptually, the choice between the diverging steps is already made at $s_5$. But then, what is the point of having the two executions share the state $s_6$?


**Figure 3.** $s_0$ and $s_5$ are bisimilar, even though the two executions starting from $s_0$ diverge right away at $s_0$, whereas those starting from $s_5$ diverge after the first step at $s_6$.

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 [39, 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 [25] 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 [56]). And apparently, it was Abrahamson, in [1], that first considered path structures that satisfied both requirements (see [19]).

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 $s \rhd_E \langle\langle l, s'\rangle\rangle \cdot e'_2$.

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

(iii) for every $c$, $l$, $c'$, and $e'$, if $c \rhd_\varepsilon \langle\langle l, c'\rangle\rangle \cdot e'$, then $c' \rhd_\varepsilon e'$;

(iv) for every $c$, $l$, $c'$, $e'_1$, and $e'_2$, if $c \rhd_\varepsilon \langle\langle l, c'\rangle\rangle \cdot e'_1$ and $c' \rhd_\varepsilon e'_2$, then $c \rhd_\varepsilon \langle\langle l, c'\rangle\rangle \cdot e'_2$.

Here, (i) and (iii) correspond to suffix closure, and (ii) and (iv), assuming (i) and (iii) respectively, to fusion closure.

The following is immediate from 4.3(a):

**Proposition 4.7.** *$\langle S, E\rangle$ is Abrahamson if and only if the $L$-labelled execution coalgebra $\langle S, \mathsf{fun}\, E\rangle$ is Abrahamson.*

We write $L$-$\mathbf{LEC}_{\mathrm{Abr}}$ for the category whose objects are all the Abrahamson $L$-labelled execution coalgebras, and arrows all the homomorphisms from one Abrahamson $L$-labelled execution coalgebra to another.

$L$-$\mathbf{LEC}_{\mathrm{Abr}}$ is clearly a full subcategory of $L$-$\mathbf{LEC}$, and a good example of a case where certain kinds of structure have been systematically ruled out. If we are to use an $L$-labelled execution system to model the behaviour of processes of some kind, then we must rule non-Abrahamson systems out. And in that case, a final $L$-labelled execution coalgebra is no longer the right choice of model; it is simply to big, containing

nonsensical pieces of structure that are neither needed nor wanted. The following allows us to cut any such coalgebra down to size:

**Theorem 4.8.** *$L$-**LEC**$_{\mathrm{Abr}}$ is a $(\mathsf{Pow} \circ \mathsf{Seq} \circ (L \times \mathsf{Id}))$-covariety.*

*Proof.* Assume an Abrahamson $L$-labelled execution coalgebra $\langle C, \varepsilon \rangle$.

Suppose that $\langle C_{\mathrm{hi}}, \varepsilon_{\mathrm{hi}} \rangle$ is a homomorphic image of $\langle C, \varepsilon \rangle$.

Then there is a surjective homomorphism $h$ from $\langle C, \varepsilon \rangle$ to $\langle C_{\mathrm{hi}}, \varepsilon_{\mathrm{hi}} \rangle$.

Assume $c_{\mathrm{hi}}$, $l$, $c'_{\mathrm{hi}}$, and $e'_{\mathrm{hi}}$ such that $c_{\mathrm{hi}} \rhd_{\varepsilon_{\mathrm{hi}}} \langle \langle l, c'_{\mathrm{hi}} \rangle \rangle \cdot e'_{\mathrm{hi}}$.

Since $h$ is a surjective function from $C$ to $C_{\mathrm{hi}}$, there is $c \in C$ such that $c_{\mathrm{hi}} = h(c)$. And since $h$ is a homomorphism from $\langle C, \varepsilon \rangle$ to $\langle C_{\mathrm{hi}}, \varepsilon_{\mathrm{hi}} \rangle$, there is $c'$ and $e'$ such that $c \rhd_{\varepsilon} \langle \langle l, c' \rangle \rangle \cdot e'$ and

$$\langle \langle l, c'_{\mathrm{hi}} \rangle \rangle \cdot e'_{\mathrm{hi}} = \langle \langle l, h(c') \rangle \rangle \cdot \mathsf{Seq}(L \times h)(e').$$

Since $\langle C, \varepsilon \rangle$ is Abrahamson, $c' \rhd_{\varepsilon} e'$, and thus, since $h$ is a homomorphism from $\langle C, \varepsilon \rangle$ to $\langle C_{\mathrm{hi}}, \varepsilon_{\mathrm{hi}} \rangle$, $c'_{\mathrm{hi}} \rhd_{\varepsilon_{\mathrm{hi}}} e'_{\mathrm{hi}}$.

Assume $c_{\mathrm{hi}}$, $l$, $c'_{\mathrm{hi}}$, $e'_{\mathrm{hi}1}$ and $e'_{\mathrm{hi}2}$ such that $c_{\mathrm{hi}} \rhd_{\varepsilon_{\mathrm{hi}}} \langle \langle l, c'_{\mathrm{hi}} \rangle \rangle \cdot e'_{\mathrm{hi}1}$ and $c'_{\mathrm{hi}} \rhd_{\varepsilon_{\mathrm{hi}}} e'_{\mathrm{hi}2}$.

Since $h$ is a surjective function from $C$ to $C_{\mathrm{hi}}$, there is $c \in C$ such that $c_{\mathrm{hi}} = h(c)$. And since $h$ is a homomorphism from $\langle C, \varepsilon \rangle$ to $\langle C_{\mathrm{hi}}, \varepsilon_{\mathrm{hi}} \rangle$, there is $c'$, $e'_1$, and $e'_2$ such that $c \rhd_{\varepsilon} \langle \langle l, c' \rangle \rangle \cdot e'_1$,

$$\langle \langle l, c'_{\mathrm{hi}} \rangle \rangle \cdot e'_{\mathrm{hi}1} = \langle \langle l, h(c') \rangle \rangle \cdot \mathsf{Seq}(L \times h)(e'_1),$$

$c' \rhd_{\varepsilon} e'_2$, and

$$e'_{\mathrm{hi}2} = \mathsf{Seq}(L \times h)(e'_2).$$

Since $\langle C, \varepsilon \rangle$ is Abrahamson, $c \rhd_{\varepsilon} \langle \langle l, c' \rangle \rangle \cdot e'_2$, and thus, since $h$ is a homomorphism from $\langle C, \varepsilon \rangle$ to $\langle C_{\mathrm{hi}}, \varepsilon_{\mathrm{hi}} \rangle$, $c_{\mathrm{hi}} \rhd_{\varepsilon_{\mathrm{hi}}} \langle \langle l, c'_{\mathrm{hi}} \rangle \rangle \cdot e'_{\mathrm{hi}2}$.

Thus, by generalization, $\langle C_{\mathrm{hi}}, \varepsilon_{\mathrm{hi}} \rangle$ is Abrahamson.

Thus, by generalization, $L$-**LEC**$_{\mathrm{Abr}}$ is closed under the formation of homomorphic images.

Assume an Abrahamson $L$-labelled execution coalgebra $\langle C, \varepsilon \rangle$.

Suppose that $\langle C_{\mathrm{sub}}, \varepsilon_{\mathrm{sub}} \rangle$ is a subcoalgebra of $\langle C, \varepsilon \rangle$.

Then $C_{\mathrm{sub}} \subseteq C$ and $C_{\mathrm{sub}} \hookrightarrow C$ is a homomorphism from $\langle C_{\mathrm{sub}}, \varepsilon_{\mathrm{sub}} \rangle$ to $\langle C, \varepsilon \rangle$.

Assume $c$, $l$, $c'$, and $e'$ such that $c \rhd_{\varepsilon_{\mathrm{sub}}} \langle \langle l, c' \rangle \rangle \cdot e'$.

Since $C_{\mathrm{sub}} \subseteq C$, $c, c' \in C$. And since $C_{\mathrm{sub}} \hookrightarrow C$ is a homomorphism from $\langle C_{\mathrm{sub}}, \varepsilon_{\mathrm{sub}} \rangle$ to $\langle C, \varepsilon \rangle$, $c \rhd_{\varepsilon} \langle \langle l, c' \rangle \rangle \cdot e'$. Since $\langle C, \varepsilon \rangle$ is Abrahamson, $c' \rhd_{\varepsilon} e'$, and thus, since $C_{\mathrm{sub}} \hookrightarrow C$ is a homomorphism from $\langle C_{\mathrm{sub}}, \varepsilon_{\mathrm{sub}} \rangle$ to $\langle C, \varepsilon \rangle$, $c' \rhd_{\varepsilon_{\mathrm{sub}}} e'$.

Assume $c$, $l$, $c'$, $e'_1$ and $e'_2$ such that $c \rhd_{\varepsilon_{\mathrm{sub}}} \langle \langle l, c' \rangle \rangle \cdot e'_1$ and $c' \rhd_{\varepsilon_{\mathrm{sub}}} e'_2$.

Since $C_{\mathrm{sub}} \subseteq C$, $c, c' \in C$. And since $C_{\mathrm{sub}} \hookrightarrow C$ is a homomorphism from $\langle C_{\mathrm{sub}}, \varepsilon_{\mathrm{sub}} \rangle$ to $\langle C, \varepsilon \rangle$, $c \rhd_{\varepsilon} \langle \langle l, c' \rangle \rangle \cdot e'_1$ and $c' \rhd_{\varepsilon} e'_2$. Since $\langle C, \varepsilon \rangle$ is Abrahamson, $c \rhd_{\varepsilon} \langle \langle l, c' \rangle \rangle \cdot e'_2$ and thus, since $C_{\mathrm{sub}} \hookrightarrow C$ is a homomorphism from $\langle C_{\mathrm{sub}}, \varepsilon_{\mathrm{sub}} \rangle$ to $\langle C, \varepsilon \rangle$, $c \rhd_{\varepsilon_{\mathrm{sub}}} \langle \langle l, c' \rangle \rangle \cdot e'_2$.

Thus, by generalization, $\langle C_{\mathrm{sub}}, \varepsilon_{\mathrm{sub}} \rangle$ is Abrahamson.

Thus, by generalization, $L$-**LEC**$_{\mathrm{Abr}}$ is closed under the formation of subcoalgebras.

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

Let $\langle C, \varepsilon\rangle = \sum_{i\in I} \langle C_i, \gamma_i\rangle$.

Assume $c$, $l$, $c'$, and $e'$ such that $c \rhd_\varepsilon \langle\langle l, c'\rangle\rangle \cdot e'$.

Since $\langle C, \varepsilon\rangle = \sum_{i\in I} \langle C_i, \gamma_i\rangle$, there is $j \in I$ and $c_j \in C_j$ such that $c = (\mathsf{inj}_j \sum_{i\in I} C_i)(c_j)$. And since $\mathsf{inj}_j \sum_{i\in I} C_i$ is a homomorphism from $\langle C_j, \varepsilon_j\rangle$ to $\langle C, \varepsilon\rangle$, there is $c'_j$ and $e'_j$ such that $c_j \rhd_{\varepsilon_j} \langle\langle l, c'_j\rangle\rangle \cdot e'_j$ and

$$\langle\langle l, c'\rangle\rangle \cdot e' = \langle\langle l, (\mathsf{inj}_j \sum_{i\in I} C_i)(c'_j)\rangle\rangle \cdot \mathsf{Seq}(L \times \mathsf{inj}_j \sum_{i\in I} C_i)(e'_j).$$

Since $\langle C_j, \varepsilon_j\rangle$ is Abrahamson, $c'_j \rhd_{\varepsilon_j} e'_j$, and thus, since $\mathsf{inj}_j \sum_{i\in I} C_i$ is a homomorphism from $\langle C_j, \varepsilon_j\rangle$ to $\langle C, \varepsilon\rangle$, $c' \rhd_\varepsilon e'$.

Assume $c$, $l$, $c'$, $e'_1$ and $e'_2$ such that $c \rhd_\varepsilon \langle\langle l, c'\rangle\rangle \cdot e'_1$ and $c' \rhd_\varepsilon e'_2$.

Since $\langle C, \varepsilon\rangle = \sum_{i\in I} \langle C_i, \gamma_i\rangle$, there is $j \in I$ and $c_j \in C_j$ such that $c = (\mathsf{inj}_j \sum_{i\in I} C_i)(c_j)$. And since $\mathsf{inj}_j \sum_{i\in I} C_i$ is a homomorphism from $\langle C_j, \varepsilon_j\rangle$ to $\langle C, \varepsilon\rangle$, there is $c'_j$, $e'_{j1}$ and $e'_{j2}$ such that $c_j \rhd_{\varepsilon_j} \langle\langle l, c'_j\rangle\rangle \cdot e'_{j1}$,

$$\langle\langle l, c'\rangle\rangle \cdot e'_1 = \langle\langle l, (\mathsf{inj}_j \sum_{i\in I} C_i)(c'_j)\rangle\rangle \cdot \mathsf{Seq}(L \times \mathsf{inj}_j \sum_{i\in I} C_i)(e'_{j1}),$$

$c'_j \rhd_{\varepsilon_j} e'_{j2}$, and

$$e'_2 = \mathsf{Seq}(L \times \mathsf{inj}_j \sum_{i\in I} C_i)(e'_{j2}).$$

Since $\langle C_j, \varepsilon_j\rangle$ is Abrahamson, $c_j \rhd_{\varepsilon_j} \langle\langle l, c'_j\rangle\rangle \cdot e'_{2j}$ and thus, since $\mathsf{inj}_j \sum_{i\in I} C_i$ is a homomorphism from $\langle C_j, \varepsilon_j\rangle$ to $\langle C, \varepsilon\rangle$, $c \rhd_\varepsilon \langle\langle l, c'\rangle\rangle \cdot e'_2$.

Thus, by generalization, $\langle C, \varepsilon\rangle$, or equivalently, $\sum_{i\in I} \langle C_i, \gamma_i\rangle$, is Abrahamson.

Thus, by generalization, $L$-$\mathbf{LEC}_{\mathrm{Abr}}$ is closed under the formation of direct sums.

Thus, $L$-$\mathbf{LEC}_{\mathrm{Abr}}$ is a $(\mathsf{Pow} \circ \mathsf{Seq} \circ (L \times \mathsf{Id}))$-covariety. □

The following is immediate from Theorem 2.24:

**Corollary 4.9.** *There is an $L$-labelled execution coalgebra that is final in $L$-$\mathbf{LEC}_{\mathrm{Abr}}$.*

Thus, by Theorem 2.23, there is a strongly extensional Abrahamson $L$-labelled execution coalgebra that subsumes the structure of every other Abrahamson $L$-labelled execution coalgebra.

## 4.4 Underlying labelled transition systems and coalgebras

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 $\eta$ 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:

$$\begin{array}{ccc} \mathsf{Pow}\,\mathsf{Seq}(L \times C_1) & \xrightarrow{\eta(C_1)} & \mathsf{Pow}(L \times C_1) \\ \Big\downarrow{\scriptstyle \mathsf{Pow}\,\mathsf{Seq}(L \times f)} & & \Big\downarrow{\scriptstyle \mathsf{Pow}(L \times f)} \\ \mathsf{Pow}\,\mathsf{Seq}(L \times C_2) & \xrightarrow[\eta(C_2)]{} & \mathsf{Pow}(L \times C_2) \end{array}$$

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

The reason why it is of interest to us here that $\eta$ is a natural transformation is a theorem by Rutten, according to which, every natural transformation $\nu$ 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 [58, thm. 15.1]). In other words, the induced functor preserves homomorphisms, and thus, by Theorem 2.6, bisimulations too.

In our case, the functor induced by $\eta$ 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:

**Theorem 4.10.** *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 2.6 and 4.10:

**Corollary 4.11.** *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 $E : S \leftrightarrow \mathscr{S}(L \times S)$.

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

The following is now immediate from Proposition 3.3(b), 3.7, and 4.6, Corollary 4.11, and Proposition 4.12:

**Theorem 4.13.** *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 $\langle S_2, \mathsf{trans}\,E_2 \rangle$.*


**Figure 4.** $s_0$ and $s_2$ are bisimilar among the two underlying $\{l\}$-labelled transition systems, which are identical, but not among the two $\{l\}$-labelled execution systems.


**Figure 5.** $s_0$ and $s_4$ are bisimilar among the two underlying $\{l_0, l_1, l_2\}$-labelled transition systems, which are identical, but not among the two $\{l_0, l_1, l_2\}$-labelled execution systems.

## 4.5 Generable systems and coalgebras

The converse of Theorem 4.13 is of course false, as is that of Theorem 4.10 and Corollary 4.11. But it is instructive to see exactly where it fails. We go over it through a series of simple examples.

First, suppose that $\langle S_1, E_1\rangle$ and $\langle S_2, E_2\rangle$ are two $\{l\}$-labelled execution systems, whose executions are as depicted in the left and right frames respectively of Figure 4. Then $s_0$ and $s_2$ are bisimilar among the $\{l\}$-labelled transition systems $\langle S_1, \mathsf{trans}\, E_1\rangle$ and $\langle S_2, \mathsf{trans}\, E_2\rangle$, but not among $\langle S_1, E_1\rangle$ and $\langle S_2, E_2\rangle$.

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 $s_0$ and $s_2$ not being bisimilar among the two systems, is dropped during the underlying labelled transition system construction.

Now suppose that $\langle S_1, E_1\rangle$ and $\langle S_2, E_2\rangle$ are two $\{l_0, l_1, l_2\}$-labelled execution systems, whose executions are as depicted in the left and right frames respectively of Figure 5. Then $s_0$ and $s_4$ are bisimilar among the $\{l_0, l_1, l_2\}$-labelled transition systems $\langle S_1, \mathsf{trans}\, E_1\rangle$ and $\langle S_2, \mathsf{trans}\, E_2\rangle$, but not among $\langle S_1, E_1\rangle$ and $\langle S_2, E_2\rangle$.

Here the problem is of a different nature. Every step of every execution is accounted for in the underlying labelled transition systems. However, the two longer executions, starting from $s_0$ and $s_4$ respectively, disagree on their second step, and that disagreement is masked by the agreement of executions starting from $s_1$ and $s_5$ 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 can turn bisimilar in the underlying systems.


**Figure 6.** $s$ is not bisimilar with itself among the overlying $\{l_0, l_1\}$-labelled execution system whose executions correspond to all infinite paths in the diagram, and that whose executions correspond to the infinite paths that go through each of the two loops infinitely often.

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 $\{l_0, l_1\}$-labelled transition system, with $l_0 \neq l_1$, portrayed in Figure 6. One $\{l_0, l_1\}$-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.

Notice that if we let $l_0 = a$ and $l_1 = b$, then the two $\{l_0, l_1\}$-labelled execution systems in this example become models of the right and left side agents respectively of (2). Accordingly, we may think of each of these two systems as a specification of a scheduling policy between two processes, forever iterating over $l_0$ and $l_1$ 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 $\{l_0, l_1\}$-labelled transition system, in the second case, it cannot be specified by any $\{l_0, l_1\}$-labelled transition system alone.

Besides demonstrating the failure of the converse of Theorem 4.13 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 $\{l_0, l_1\}$ 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 3.2 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 [44], 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 $\{l_0, l_1\}$-labelled transition system, with $l_0 \neq l_1$, portrayed in Figure 7. The first $\{l_0, l_1\}$-labelled execution system that we wish to consider here is the unique Abrahamson system whose executions starting from $s_0$ 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 $s_0$. And because of this exception, $s_0$ is not bisimilar with itself among the two systems.

This beautiful example is from [5], 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 $s_0$. The first process will either loop around $s_0$ forever, or iterate through it for a finite, indeterminate number of times before progressing to $s_1$. From there on, a single indeterminate choice will decide its fate. The second process, on the other hand, is not allowed to loop

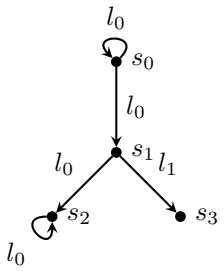



**Figure 7.** $s_0$ is not bisimilar with itself among the overlying Abrahamson $\{l_0, l_1\}$-labelled execution system whose executions starting from $s_0$ correspond to all maximal paths in the diagram, and that whose executions are all the executions of the first system except the infinite execution stuttering around $s_0$.


**Figure 8.** $s$ is not bisimilar with itself among the overlying $\{l\}$-labelled execution system whose single execution corresponds to the only infinite path in the diagram, and that whose executions correspond to all finite paths and the only infinite path.

around $s_0$ forever. It must eventually advance to $s_1$, 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 $s_0$, 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 $l_1$, 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 Theorem 4.13, 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 [25], Emerson called a set of paths *limit closed* provided that for every infinite, strictly increasing chain of finite prefixes of paths in that set, the limit of that chain, in the standard topology of sequences, is again a path in the set. This is essentially a continuity property implying that a set of paths be determined by the finite prefixes of paths in that set, and was apparently also first considered in [1]. But it was Emerson in [25] 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 Figure 8. 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

**Figure 9.** $s$ is not bisimilar with itself among the overlying $\{l\}$-labelled execution system that has no execution, and that whose only execution is the empty execution.

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.

Finally, consider the trivial labelled transition system portrayed in Figure 9. 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 Theorem 4.13. 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 $\tau$-*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 $\tau$ as a representation of the control flow graph of a possibly indeterminate sequential program, then a $\tau$-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 $\tau$ just as long as it assigns to any $c \in C$ the set of all $\tau$-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 $\omega$ 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 $\tau$, and $\mathrm{W}_\tau(c)(n)$ as the wavefront at the $n$th time instance.

**Proposition 4.14.** *If $e$ is a $\tau$-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 4.15.** *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 $\mathsf{Pow}$, 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 4.16.** *$\{e \mid e \in \mathsf{Seq}(L \times C)$ and $e$ is a $\tau$-orbit of $c\}$ is a set.*

*Proof.* By Proposition 4.14, for every $e \in \mathsf{Seq}(L \times C)$, if $e$ is a $\tau$-orbit of $c$, then

$$\mathsf{graph}\, e \subseteq \bigcup \{\{n\} \times \mathrm{W}_\tau(c)(n) \mid n \in \omega\}.$$

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 4.15 and 4.16 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 $\tau$ *generates* $\varepsilon$ if and only if $\mathsf{gen}\, \tau = \varepsilon$.

We say that $\varepsilon$ is *generable* if and only if there is a class function from $C$ to $\mathsf{Pow}(L \times C)$ that generates $\varepsilon$.

Now, suppose that $\varepsilon$ is indeed generable. Can there be more than one class function from $C$ to $\mathsf{Pow}(L \times C)$ that generates $\varepsilon$?

We could perhaps use the following tentative argument to convince ourselves that this cannot be the case: if $\tau_1$ and $\tau_2$ 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 $\tau_1$-orbit of $c'$ with $\langle l, c' \rangle$ to get a $\tau_1$-orbit of $c$ that cannot be a $\tau_2$-orbit of $c$. But how do we know if there is a $\tau_1$-orbit of $c'$ to prefix with $\langle l, c' \rangle$?

If $\tau_1(c') = \emptyset$, then $\langle\ \rangle$ is a $\tau_1$-orbit of $c'$. If $\tau_1(c') \neq \emptyset$, then we would again expect that there is at least one $\tau_1$-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 $\tau$ and $c$, there is a $\tau$-orbit of $c$. And indeed, this statement is equivalent to the *Axiom of Dependent Choice*.

We will need the following lemma:

**Lemma 4.17.** *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 $\tau$-orbit of $c'$, then there is $e \in \mathsf{Seq}(L \times C)$ such that $e$ is a $\tau$-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 $\tau$-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 $\tau$-orbit of $c''$. Thus, by the induction hypothesis, there is $e \in \mathsf{Seq}(L \times C)$ such that $e$ is a $\tau$-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 $\tau$-orbit of $c'$, then there is $e \in \mathsf{Seq}(L \times C)$ such that $e$ is a $\tau$-orbit of $c$. □

**Theorem 4.18.** *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 $\tau$-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 $s\ R\ s'$, then there is an infinite sequence $d$ over $S$ such that for every $n \in \omega$, $\mathsf{head}\,\mathsf{tail}^n\ d\ R\ \mathsf{head}\,\mathsf{tail}^{n+1}\ d$.*

*Proof.* Suppose that (a) is true.

Assume a non-empty set $S$.

Let $l$ be a label in $L$.

Let $\tau$ be a class function from $S$ to $\mathsf{Pow}(L \times S)$ such that for every $s \in S$,

$$\tau(s) = \{\langle l, s' \rangle \mid s\ R\ s'\}.$$

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 $\tau$-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 $\tau$-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 $\tau$-orbit of $c'$. Thus, by Lemma 4.17, there is $e \in \mathsf{Seq}(L \times C)$ such that $e$ is a $\tau$-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 $\tau$-orbit of $c'$. Thus, by Lemma 4.17, there is $e \in \mathsf{Seq}(L \times C)$ such that $e$ is a $\tau$-orbit of $c$.

Thus, by generalization, (a) is true. □

Here, we accept the Axiom of Dependent Choice, and so we will take Theorem 4.18(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 4.19.** *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 $\tau_1$-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 $\mathsf{gen}$ as an operator from cooperations of $L$-labelled transition coalgebras to cooperations of $L$-labelled execution coalgebras, then we can read Proposition 4.19 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 $\eta$:

**Proposition 4.20.** *The following are true:*

*(a)* $\eta(C) \circ \mathsf{gen}\,\tau = \tau$*;*

*(b) if $\varepsilon$ 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 $\tau$-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 $\varepsilon$ 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 $\tau$-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 $\varepsilon$ is *consistent* with $\tau$ 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 4.21.** *The following are equivalent:*

*(a) $e$ is a $\tau$-orbit of $c$;*

*(b) there is a class function $\varepsilon : C \to \mathsf{Pow}\,\mathsf{Seq}(L \times C)$ such that $\varepsilon$ is consistent with $\tau$, and $e \in \varepsilon(c)$.*

*Proof.* Suppose that (a) is true.

Let $\varepsilon$ be a class function from $C$ to $\mathsf{Pow}\,\mathsf{Seq}(L \times C)$ 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, $\varepsilon$ is consistent with $\tau$.

Thus, (b) is true.

Conversely, suppose (b) is true,

Then clause (i) of the property of being a $\tau$-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 $\tau$-orbit of $c$ is true.

Therefore, (a) is true. □

The following is immediate:

**Corollary 4.22.** *If $\varepsilon$ is consistent with $\tau$, then for any $c \in C$,*

$$\varepsilon(c) \subseteq (\mathsf{gen}\,\tau)(c).$$

The following is now straightforward:

**Corollary 4.23.** $\mathsf{gen}\,\tau$ *is consistent with* $\tau$.

*Proof.* Assume $c \in C$ and $e \in (\mathsf{gen}\,\tau)(c)$.

By Theorem 4.21, there is a class function $\varepsilon : C \to \mathsf{Pow}\,\mathsf{Seq}(L \times C)$ such that $\varepsilon$ is consistent with $\tau$, 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 4.22, $e' \in (\mathsf{gen}\,\tau)(c')$.

Thus, by generalization, $\mathsf{gen}\,\tau$ is consistent with $\tau$. □

Corollary 4.22 is the proof rule that we referred to earlier, which is basically an instance of the *coinduction proof technique* described in [47]. 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 $\mathsf{Pow}\,\mathsf{Seq}(L \times C)$: for every class function $\varepsilon_1, \varepsilon_2 : C \to \mathsf{Pow}\,\mathsf{Seq}(L \times C)$, 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, $\varepsilon$ is consistent with $\tau$ if and only if $\varepsilon$ 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 4.22 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 [47] 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 $\mathsf{Pow}\,\mathsf{Seq}(L \times C)$ is not a complete lattice under inclusion, and so neither is the induced ordering of class functions from $C$ to $\mathsf{Pow}\,\mathsf{Seq}(L \times C)$. 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 [47] is just another variation of the general finality theme of Section 2.6. The same, of course, is true for the more *ad hoc* proof rule of Corollary 4.22.

For a historical account on the emergence of coinduction in computer science, we refer to [60].

We have now finally reached our generability characterization theorem.

**Theorem 4.24.** *$\varepsilon$ 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*

$$\mathsf{head}\,\mathsf{tail}^k\, s = \mathsf{head}\,\mathsf{tail}^k\, e,$$

*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 $\varepsilon$ 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 4.23, $e' \in (\mathsf{gen}\,\tau)(c)$, 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 4.23, $\langle\langle l, c'\rangle\rangle \cdot e'_2 \in (\mathsf{gen}\,\tau)(c)$, 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 $\tau$-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 4.23, $\tau(c) = \emptyset$, and hence, $e = \langle\ \rangle$. Thus, (d) is true.

By Theorem 4.18 and the Axiom of Dependent Choice, (e) is true.

Conversely, suppose that (a), (b), (c), (d), and (e) are true.

We prove that $\varepsilon = \mathsf{gen}(\eta(C) \circ \varepsilon)$.

Assume $c \in C$ and $e \in \varepsilon(c)$.

If $e = \langle\ \rangle$, then, by (d), $\varepsilon(c) = \{\langle\ \rangle\}$, and thus, $\eta(C)(\varepsilon(c)) = \emptyset$.

Otherwise, there is $l$, $c'$, and $e'$ such that

$$e = \langle \langle l, c' \rangle \rangle \cdot e'.$$

Thus, by definition of $\eta$, $\langle l, c' \rangle \in \eta(C)(\varepsilon(c))$, and by (a), $e' \in \varepsilon(c')$.

Thus, by generalization, $\varepsilon$ is consistent with $\eta(C) \circ \varepsilon$, and by Corollary 4.22, for any $c \in C$,

$$\varepsilon(c) \subseteq (\mathsf{gen}(\eta(C) \circ \varepsilon))(c).$$

Assume $c \in C$ and $e \in (\mathsf{gen}(\eta(C) \circ \varepsilon))(c)$.

If $e = \langle\ \rangle$, then $\eta(C)(\varepsilon(c)) = \emptyset$. Thus, by (e), $e \in \varepsilon(c)$.

Otherwise, there is $l$, $c'$, and $e'_2$ such that $\langle l, c' \rangle \in \eta(C)(\varepsilon(c))$ 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 $\omega$ such that $\mathsf{tail}^{n+1}\, e = \langle\ \rangle$.

Then there is $l'$ and $c''$ such that $\eta(C)(\varepsilon(c'')) = \emptyset$ 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, $\varepsilon$ is generable. □

Clause (a) of Theorem 4.24 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 [25]).

Each of these five properties has come about in connection with a different cause of failure of the converse of Theorem 4.13, and hence of Theorem 4.10 and Corollary 4.11. 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 $\varepsilon$ is generable.

**Theorem 4.25.** *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 Theorem 4.10, $h$ 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 $\varepsilon'_2$ be a class function from $C_2$ to $\mathsf{Pow}\,\mathsf{Seq}(L \times C_2)$ such that for any $c_2 \in C_2$,

$$\varepsilon'_2(c_2) = \{e_2 \mid \text{there is } c_1 \in C_1 \text{ and } e_1 \in \varepsilon_1(c_1) \\ \text{such that } h(c_1) = c_2 \text{ and } (\mathsf{Seq}(L \times h))(e_1) = e_2\}.$$

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 4.23, 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, $\varepsilon'_2$ is consistent with $\eta(C_2) \circ \varepsilon_2$. Then, by Corollary 4.22, 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 4.23, 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 4.23 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 $\omega$ 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 $n \in \omega$,

$$W(n+1) = \{\langle\langle l', c_1''\rangle, n+1\rangle \mid \text{there is } \langle\langle l, c_1'\rangle, n\rangle \in W(n) \text{ such that } \langle l', c_1''\rangle \in (\eta(C_1) \circ \varepsilon_1)(c_1') \text{ and } \mathsf{head}\,\mathsf{tail}^{n+1}\, e_2 = \langle l', h(c_1'')\rangle\}.$$

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 4.23, 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 $s \; R \; s'$. Then, by the Axiom of Dependent Choice, there is an infinite sequence $d$ over $S$ such that for every $n \in \omega$, $\mathsf{head}\,\mathsf{tail}^n\, d \; R \; \mathsf{head}\,\mathsf{tail}^{n+1}\, d$.

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 $\langle l', c_1'' \rangle \in (\eta(C_1) \circ \varepsilon_1)(c_1')$ 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 $W$, $\langle l, c_1' \rangle \in (\eta(C_1) \circ \varepsilon_1)(c_1)$. 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 2.6 and 4.25:

**Corollary 4.26.** *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 $T : S \leftrightarrow L \times S$.

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 4.27.** $\mathcal{E}_T(E) = \mathsf{rel}\, \mathcal{G}_{\mathsf{fun}\, T}(\mathsf{fun}\, E)$.

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 [47] is directly applicable.

The following follows from Proposition 4.27 and the fact that $\mathsf{gen}\, \mathsf{fun}\, T$ is the greatest fixed point of $\mathcal{G}_{\mathsf{fun}\, T}$:

**Proposition 4.28.** $\mathsf{exec}\, T = \mathsf{rel}\, \mathsf{gen}\, \mathsf{fun}\, T$.

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 3.3 and 4.28:

**Proposition 4.29.** *$E$ is generable if and only if $\mathsf{fun}\, E$ is generable.*

The following is immediate from Proposition 3.3, 4.12, 4.28, and 4.29:

**Proposition 4.30.** *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 4.29:

**Proposition 4.31.** *$\langle S, E\rangle$ 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 3.3(b), 3.7, 4.6, and 4.12, and Corollary 4.26:

**Theorem 4.32.** *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 $\langle S_2, \mathsf{trans}\, E_2\rangle$.*

Proposition 4.30 and Theorem 4.32 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.

We write $L$-$\mathbf{LEC}_{\mathrm{gen}}$ for the category whose objects are all the generable $L$-labelled execution systems, and arrows all the homomorphisms from one generable $L$-labelled execution system to another.

The following is immediate from Proposition 4.20 and Theorem 4.25:

**Theorem 4.33.** *$L$-$\mathbf{LEC}_{\mathrm{gen}}$ and $L$-$\mathbf{LTC}$ are isomorphic.*

Thus, for all practical purposes, generable labelled execution coalgebras are equivalent to labelled transition coalgebras.

In light of this equivalence, Theorem 4.24 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 4.24.

# 5 Related work

As already argued in the introduction, what Milner advocated was a dichotomy between causation and observation in the theory of concurrency. But the expansion law, and its reckless use with agents with infinite behaviours, skewed that dichotomy into a controversy between so-called "true concurrency" and interleaving semantics. And in the midst of that controversy, interleaving came to be thought of as no more than bounded indeterminacy. However, the real cause of this was not the expansion law per se, but the use of labelled transition systems as models of agent behaviour. After all, interleaving is an operation on executions, not transitions.

Rather than generalizing transition systems into executions ones, and obtain a modelling structure that can do justice to the notion of interleaving, and ultimately to the observational view, efforts were steered toward decorating the former with all kinds of different pieces of information that would alleviate the various deficiencies of that misrepresented notion of interleaving (e.g., see [16, 23, 21, 17, 18]). And more often that not, the result was a kind 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 [22], [20]), were not concerned with organizing them into structures and looking at their branching properties, and anyway, seem to have received only scant attention.

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 4.1) 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 [26, p. 152]). They do not, we think. But despite the voiced arguments for

a separation between implementation and correctness issues in reasoning about concurrent programs (e.g., see [19]), transitions remain in the lead role.

In [31], 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 [31, p. 27]). They also define a concept of *extended bisimulation* for such systems, which is basically the same as our canonically derived concept of bisimulation between labelled executions systems (see Definition 4.5). The focus in [31] is in logic, and specifically, in a generalization of Hennessy-Milner Logic (see [30]) 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 [5]), who is aware of Hennessy's work in [29], a precursor of [31], but apparently, unaware of the work in [31] (see footnote in [5, p. 3]). Aczel's intention is to apply the final universe approach of [4] to the semantics of Milner's SCCS with finite delay (see [44]). 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 [31], but only because the latter are prefix closed. Eventually, these structures are represented as coalgebras over **Class**, and [4, 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 [32]. The starting point is again Milner's SCCS with finite delay, and the structures used are practically the same as in [5]. But the approach is purely categorical. Indeed, the main goal in [32] is showing how much can be done within category theory alone.

Comparing [31], [5], and [32] with our work here, there are two things that we think stand out and would like to mention. First, regarding the general idea underlying the concept of labelled execution system, we find that in all three of [31], [5], and [32], 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 [31], where prefix closure is an added feature, but not so in [5] and [32], where the property is practically built into the structure of a system. And second, regarding 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 [31], [5], and [32].

It should be emphasized that the precedence of [31], [5], and [32] 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. One thing is certain though: 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.

# 6 Conclusion

The purpose of this work was to introduce the concept of labelled execution system, a generalization of that of labelled transition system that we believe can better accommodate the needs of an observational approach to concurrency theory. And as we saw in Section 4.5, in order for the use of labelled execution systems over labelled transition systems to be justified, one or more of the five properties listed in the respective clauses of Theorem 4.24 must be given up. But as we saw in Section 4.3, if we want our systems to be "well behaved", the first two of them must be hold on to. Therefore, if we ignore the rather

uninteresting non-triviality condition, we are left with having to give up limit closure, impossibility of indeterminate termination, or both.

In fact, giving up any of these two properties has its own merit. For example, giving up limit closure enables us to faithfully model the finite delay property, so intrinsically bound to the notion of asynchronous parallelism. And possibility of indeterminate termination provides us with the means of simulating the behaviour of a capricious environment that may at any time cease to produce input stimuli.

Returning to the discussion in our introduction, it is not hard to see how one could use Abrahamson systems to provide a model for Milner's CCS that avoided "explicating parallelism in terms of non-determinism", at least in the stronger sense of the expansion law, and to be sure, distinguished between the two sides of (2). The critical step is of course in the treatment of parallel composition as a *fair merge* operation over the executions of the individual systems under composition, which, however, does not seem to present any particular difficulty (e.g., see [50]), and can be fitted to different notions of fairness, such as, for example, weak and strong fairness (see [10]). But if one is really serious about this endeavour, one must also allow for abstraction, whereby certain actions of an agent become unobservable. And this calls for a suitably weakened version of bisimilarity among labelled execution systems. An interesting question then is whether such a weaker version can be canonically obtained by the coalgebraic methods used here. If possible, this could lead, through Theorem 4.33, to a coalgebraic characterization of weaker versions of bisimilarity among labelled transition systems as well, a goal that has heretofore remained elusive.

Finally, another interesting direction for future work is the stratification of the concept of bisimilarity among labelled execution systems, which can be obtained in a manner completely analogous to the case of labelled transition systems (e.g., see [46, chap. 10.4]). In the latter case, each iteration of the stratification process is associated with a notion of depth down to which certain differences in branching structure can be observed. This is most starkly evident in the classical example used to demonstrate that, in the case of infinitely branching systems, a transfinite number of iterations is needed for the stratification process to converge (e.g., see [60, exam. 2.6]). In the case of labelled execution systems, however, there is no such notion of depth, and that particular example is defused. This is not to say that, in that case, the stratification process does not need a transfinite number of iterations to converge, what can be shown to fail by appropriately embedding the labelled transition systems of [46, prop. 10.5] in the executions of certain suitably constructed labelled execution systems. But the proposed stratification of bisimilarity can be seen as a stratification of the linear-time/branching-time spectrum (e.g., see [62]), offering a natural formal framework for the classification of different semantic notions within that spectrum.

# References


[1] Karl Abrahamson. *Decidability and Expressiveness of Logics of Programs.* PhD thesis, University of Washington at Seattle, 1980.

[2] Samson Abramsky. What are the fundamental structures of concurrency? We still don't know! *Electronic Notes in Theoretical Computer Science*, 162:37–41, 2006. Proceedings of the Workshop "Essays on Algebraic Process Calculi" (APC 25).

[3] Peter Aczel. *Non-well-founded Sets.* Number 14 in Lecture Notes. CLSI, 1988.

[4] 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.

[5] Peter Aczel. A semantic universe for the study of fairness. VERY ROUGH AND INCOMPLETE DRAFT, October 1996.

[6] Peter Aczel and Nax Paul Mendler. A final coalgebra theorem. In *Category Theory and Computer Science*, pages 357–365, London, UK, 1989. Springer-Verlag.

[7] Jiří Adámek, Horst Herrlich, and George Strecker. *Abstract and Concrete Categories: The Joy of Cats*. New York, NY, USA, 2004.

[8] 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.

[9] Bowen Alpern and Fred B. Schneider. Defining liveness. *Information Processing Letters*, 21(4):181–185, 1985.

[10] Krzysztof R. Apt and Ernst-Rüiger Olderog. Proof rules and transformations dealing with fairness. *Science of Computer Programming*, 3(1):65–100, 1983.

[11] Jos C. M. Baeten. A brief history of process algebra. *Theoretical Computer Science*, 335(2-3):131–146, 2005.

[12] Jon Barwise and Lawrence Moss. *Vicious Circles*. Number 60 in Lecture Notes. CLSI, 1996.

[13] Martin Berger. An interview with Robin Milner. http://www.informatics.sussex.ac.uk/users/mfb21/interviews/milner/, September 2003.

[14] Garrett Birkhoff. On the structure of abstract algebras. *Mathematical Proceedings of the Cambridge Philosophical Society*, 31(04):433–454, 1935.

[15] Bard Bloom, Sorin Istrail, and Albert R. Meyer. Bisimulation can't be traced. *J. ACM*, 42(1):232–268, 1995.

[16] Gérard Boudol and Ilaria Castellani. Permutation of transitions: An event structure semantics for CCS and SCCS. In J. de Bakker, W. de Roever, and G. Rozenberg, editors, *Linear Time, Branching Time and Partial Order in Logics and Models for Concurrency*, volume 354 of *Lecture Notes in Computer Science*, pages 411–427. Springer Berlin / Heidelberg, 1989.

[17] Gérard Boudol, Ilaria Castellani, Matthew Hennessy, and Astrid Kiehn. Observing localities. *Theoretical Computer Science*, 114(1):31–61, 1993.

[18] Rosvelter João Coelho da Costa and Jean-Pierre Courtiat. A causality-based semantics for CCS. In *Proceedings of the First North American Process Algebra Workshop*, NAPAW '92, pages 200–215, London, UK, 1993. Springer-Verlag.

[19] 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.

[20] Philippe Darondeau. About fair asynchrony. *Theoretical Computer Science*, 37:305–336, 1985.

[21] Philippe Darondeau and Pierpaolo Degano. Causal trees: Interleaving + causality. In Irène Guessarian, editor, *Semantics of Systems of Concurrent Processes*, volume 469 of *Lecture Notes in Computer Science*, pages 239–255. Springer Berlin / Heidelberg, 1990.

[22] 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.

[23] Pierpaolo Degano, Rocco De Nicola, and Ugo Montanari. A partial ordering semantics for CCS. *Theoretical Computer Science*, 75(3):223–262, 1990.

[24] Edsger W. Dijkstra. *A Discipline of Programming*. Prentice-Hall, 1976.

[25] E. Allen Emerson. Alternative semantics for temporal logics. *Theoretical Computer Science*, 26(1-2):121–130, 1983.

[26] E. Allen Emerson and Joseph Y. Halpern. "Sometimes" and "not never" revisited: On branching versus linear time temporal logic. *J. ACM*, 33(1):151–178, January 1986.

[27] 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.

[28] H. Peter Gumm. Elements of the general theory of coalgebras. Lecture Notes for LUATCS'99 at Rand Afrikaans University, Johannesburg, South Africa, 1999.

[29] Matthew Hennessy. Axiomatising finite delay operators. *Acta Informatica*, 21(1):61–88, 1984.

[30] Matthew Hennessy and Robin Milner. Algebraic laws for nondeterminism and concurrency. *J. ACM*, 32(1):137–161, January 1985.

[31] Matthew Hennessy and Colin Stirling. The power of the future perfect in program logics. *Information and Control*, 67(1-3):23–52, 1985.

[32] 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.

[33] Bart Jacobs and Jan J. M. M. Rutten. A Tutorial on (Co)Algebras and (Co)Induction. *EATCS Bulletin*, 62:62–222, 1997.

[34] Richard M. Karp and Raymond E. Miller. Parallel program schemata. *Journal of Computer and System Sciences*, 3(2):147–195, 1969.

[35] Yasuo Kawahara and Masao Mori. A small final coalgebra theorem. *Theoretical Computer Science*, 233(1-2):129–145, 2000.

[36] Robert M. Keller. Formal verification of parallel programs. *Commun. ACM*, 19(7):371–384, 1976.

[37] Joachim Lambek. A fixpoint theorem for complete categories. *Mathematische Zeitschrift*, 103(2):151–161, 1968.

[38] Leslie Lamport. Proving the correctness of multiprocess programs. *Software Engineering, IEEE Transactions on*, SE-3(2):125–143, March 1977.

[39] 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.

[40] Leslie Lamport and Fred B. Schneider. Formal foundation for specification and verification. In M. Paul, H. Siegert, M. Alford, J. Ansart, G. Hommel, L. Lamport, B. Liskov, G. Mullery, and F. Schneider, editors, *Distributed Systems*, volume 190 of *Lecture Notes in Computer Science*, pages 203–285. Springer Berlin / Heidelberg, 1985.

[41] Saunders Mac Lane. *Categories for the Working Mathematician*. Number 5 in Graduate Texts in Mathematics. Springer, 2nd edition, 1998.

[42] 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.

[43] Robin Milner. *A Calculus of Communicating Systems*, volume 92 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 1980.

[44] Robin Milner. A finite delay operator in synchronous CCS. Technical Report CSR-116-82, University of Edinburgh, 1982.

[45] Robin Milner. Calculi for synchrony and asynchrony. *Theoretical Computer Science*, 25(3):267–310, 1983.

[46] Robin Milner. *Communication and Concurrency*. Prentice Hall International Series in Computer Science. Prentice-Hall, Upper Saddle River, NJ, USA, 1989.

[47] Robin Milner and Mads Tofte. Co-induction in relational semantics. *Theoretical Computer Science*, 87(1):209–220, 1991.

[48] Edward F. Moore. Gedanken-experiments on sequential machines. *Automata Studies*, 34:129–153, 1956.

[49] Lawrence S. Moss and Norman Danner. On the foundations of corecursion. *Logic Journal of IGPL*, 5(2):231–257, 1997.

[50] 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.

[51] 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.

[52] Benjamin C. Pierce. *Basic Category Theory for Computer Scientists*. Foundations of Computing Series. The MIT Press, 1991.

[53] Gordon D. Plotkin. A structural approach to operational semantics (Aarhus notes). Technical Report DAIMI FN–19, Computer Science Department, Aarhus University, September 1981.

[54] Gordon D. Plotkin. The origins of structural operational semantics. *Journal of Logic and Algebraic Programming*, 60-61:3–15, 2004.

[55] 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.

[56] 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.

[57] Mark Reynolds. An axiomatization of full computation tree logic. *The Journal of Symbolic Logic*, 66(3):1011–1057, 2001.

[58] Jan J. M. M. Rutten. Universal coalgebra: a theory of systems. *Theor. Comput. Sci.*, 249(1):3–80, 2000.

[59] 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.

[60] Davide Sangiorgi. On the origins of bisimulation and coinduction. *ACM Trans. Program. Lang. Syst.*, 31(4):1–41, 2009.

[61] Alfred Tarski. A lattice-theoretical fixpoint theorem and its applications. *Pacific J. of Math.*, 5(2):285–309, 1955.

[62] Rob J. van Glabbeek. The linear time - branching time spectrum. In J. Baeten and J. Klop, editors, *CONCUR '90 Theories of Concurrency: Unification and Extension*, volume 458 of *Lecture Notes in Computer Science*, pages 278–297. Springer Berlin / Heidelberg, 1990.

[63] Hermann Weyl. Mathematics and logic. *The American Mathematical Monthly*, 53(1):2–13, 1946.