Centrum voor Wiskunde en Informatica

REPORTRAPPORT

SEN

Software Engineering

Software ENgineering

SEN

GSOS for probabilistic transition systems

F. Bartels

REPORT SEN-R0221 NOVEMBER 30, 2002

CWI is the National Research Institute for Mathematics and Computer Science. It is sponsored by the Netherlands Organization for Scientific Research (NWO).
CWI is a founding member of ERCIM, the European Research Consortium for Informatics and Mathematics.

CWI's research has a theme-oriented structure and is grouped into four clusters. Listed below are the names of the clusters and in parentheses their acronyms.

Probability, Networks and Algorithms (PNA)

Software Engineering (SEN)

Modelling, Analysis and Simulation (MAS)

Information Systems (INS)

P.O. Box 94079, 1090 GB Amsterdam (NL)
Kruislaan 413, 1098 SJ Amsterdam (NL)
Telephone +31 20 592 9333
Telefax +31 20 592 4199

ISSN 1386-369X

GSOS for Probabilistic Transition Systems

Falk Bartels
CWI
P.O. Box 94079, 1090 GB Amsterdam, The Netherlands

ABSTRACT

We introduce $PGSOS$, an operator specification format for (reactive) probabilistic transition systems which bears similarity to the known GSOS format for labelled (nondeterministic) transition systems. Like the standard one, the format is well behaved in the sense that on all models bisimilarity is a congruence and the up-to-context proof principle is valid. Moreover, guarded recursive equations involving the specified operators have unique solutions up to bisimilarity. These results generalize well-behavedness results given in the literature for specific operators that turn out to be definable by our format.

PGSOS arose from the following procedure: Turi and Plotkin proposed to model specifications in the (standard) GSOS format as natural transformations of a type they call $abstract\ GSOS$. This formulation allows for simple proofs of several well-behavedness properties, such as bisimilarity being a congruence on all models of such a specification. First, we give a full proof of Turi and Plotkin's claim about the correspondence of abstract GSOS and standard GSOS for labelled transition systems. Next, we instantiate their categorical framework to yield a specification format for probabilistic transition systems. The main contribution of the present paper is the derivation of the PGSOS format as a rule-style representation of the natural transformations obtained this way. We benefit from the fact that some parts of our argument for the nondeterministic case can be reused. The well-behavedness results for abstract GSOS immediately carry over to the new concrete format.

2000 Mathematics Subject Classification: 68Q60, 68Q85
1998 ACM Computing Classification System: F.1.1, F.3.2, G.3
Keywords and Phrases: coalgebra, probabilistic transition systems, transition system specification, congruence formats, abstract GSOS.
Note: Research supported by the NWO project ProMACS

Table of Contents

1. Introduction

In theoretical computer science one often deals with systems that carry an algebraic as well as a behavioural structure. For example this is the case when an operational semantics is assigned to the terms of a programming language. As another example – which is dual in some sense – one may want to equip a given domain of behaviours with operators. The algebraic and behavioural structure are interrelated: the semantics of a composed program for instance is usually determined by the semantics of its components.

Labelled (image finite) transition systems (LTS) are frequently used as semantic models. At any moment such a system is in some state p taken from a set of possible states P. We sometimes call the transition system in this state just the *process* p. The process p may or may not be able to react to a given input label a from an input alphabet L. In the first case, this would cause the system to move to a new state, say p', which is chosen nondeterministically out of a finite set of possible successor states of p for the label a. We depict these possibilities by

$$p \xrightarrow{a} p' \quad \text{and} \quad p \stackrel{a}{\not\rightarrow}$$

respectively.

The states of a system are often regarded *internal* and invisible from the outside. All an observer can see is which input labels are enabled and which are not. For an enabled label he can of course continue experimenting with the successor states. When two states are not distinguishable by such experiments we call them *behaviourally equivalent*. Behavioural equivalence for LTS can be established by showing that the states are related by some *(strong) bisimulation*. Therefore we will often alternatively talk about *bisimilar* processes.

Operators acting on the state set of an LTS can be specified by *structural operational rules*, a format relating the first steps in the behaviour of a composed process to the behaviour of its components. As an example, consider the sequential composition of two processes specified by the following rules

$$\frac{x \xrightarrow{a} x'}{x.y \xrightarrow{a} x'.y} \qquad \frac{x \stackrel{l}{\not\rightarrow} \ (\forall l \in L) \quad y \xrightarrow{a} y'}{x.y \xrightarrow{a} y'} \qquad \text{(each for all } a \in L) \tag{1.1}$$

For any two states p and q in a transition system $\langle P, \alpha \rangle$ this defines that $p.q$ allows *precisely* the transitions arising in the conclusion after we substitute p for x, q for y, and any states in P for x' and y' such that the corresponding premises are satisfied in the given transition system.

A number of questions naturally arise about such a specification: First of all, the rules should uniquely determine the behaviour of $p.q$ for any two processes p and q. And moreover, this behaviour should solely depend on the behaviour of p and q, i.e. for any two processes $\hat{p}$ and $\hat{q}$ which are bisimilar to p and q respectively, we want that $p.q$ and $\hat{p}.\hat{q}$ are bisimilar as well. In other words, we want bisimilarity to be a *congruence* for the resulting operators.

It turns out that one can guarantee these and other properties by restricting oneself to specifications where all rules are of a certain format. These formats usually restrict the depth of the terms one may put as the source or target of the transitions in the premises or conclusion. For some of these places this may be the application of precisely one operator to variables, or just a variable. The format may furthermore disallow look-ahead, i.e. a chaining of premises, or negative premises, i.e. premises requiring that certain transitions are not possible. A number of such formats have extensively been studied in the literature (for an overview see e.g. [AFV01]). A popular example is the GSOS format [BIM95], on which we will focus in this paper. It covers the above example specification and is known to be well-behaved in a number of ways. It has for instance the two properties mentioned above: any GSOS specification uniquely determines the behaviour of the composed processes and bisimilarity is a congruence on each of its models. Moreover the specified operators are suitable for the up-to-context proof principle [San98], and guarded recursive specifications involving them have unique solutions (up to bisimilarity).

There is by now a rich body of work published on this issue, mostly concerning nondeterministic transition systems. However, these systems are not suitable for all applications. Often one needs to represent further aspects, like timed or probabilistic behaviour. Consequently, more complex types of systems that incorporate these features are nowadays studied. Still there is little known about well-behaved specification formats in such settings.

As a step in this direction we consider *probabilistic transition systems (PTS)*: as before, a state p in such a system may or may not be able to process a given input label $a \in L$, and when it can do so, it moves to one out of finitely many potential successor states. This time the actual successor is not chosen nondeterministically, but according to a given probability distribution. We write

$$p \xrightarrow{a[u]} p' \quad \text{for } u \in [0,1]$$

when it ends up in the state p' with probability u.

The above describes just one out of several possible ways to incorporate probabilistic behaviour into transition systems. We took it from the work of Larsen and Skou [LS91], who also introduced a notion of probabilistic bisimilarity for PTS. Elsewhere, PTS are referred to as the *reactive model* of probabilistic processes [vGSS95] as opposed to a *generative model*, where to each state one also assigns a probability distribution on the labels (which should then rather be viewed as output labels). Other authors consider a more complex setting where nondeterministic and probabilistic choice are incorporated as independent concepts (see [JLY01] for an overview). In this setting our systems appear as the special case where the nondeterminism disappears and they are therefore called *deterministic* in loc. cit. We want to stress that the results for PTS we are about to describe are derived in such a way that large parts of the argument can easily be adapted to the other types of systems as well.

As before, we are interested in operator specifications for PTS. As an example, we again consider the sequential composition. It is specified by the following transition rules:

$$\frac{x \xrightarrow{a[r]} x'}{x.y \xrightarrow{a[r]} x'.y} \qquad \frac{x \overset{l}{\nrightarrow} \;(l \in L) \quad y \xrightarrow{a[r]} y'}{x.y \xrightarrow{a[r]} y'} \qquad \text{(each for all } a \in L \text{ and } r \in [0,1]) \tag{1.2}$$

The same questions as in the nondeterministic case arise here as well: Do the rules uniquely determine behaviours? If so, are the resulting operators well behaved? For example, we again want (probabilistic) bisimilarity to be a congruence for them.

Specification formats guaranteeing such properties would be helpful in the setting of PTS as well. The above example may suggest that such formats can easily be given, since the transition rules appear similar to the ones for nondeterministic systems. But note that the transitions here are of a rather different nature. Assigning probabilities does not just mean to consider labels of a slightly more complex type, as we will explain later. This is confirmed by the fact that up to our knowledge no such formats have been proposed yet, although well-behavedness of concrete specifications is considered in the literature (see e.g. van Glabbeek et al. [vGSS95]).

In this paper, we introduce a probabilistic version of the GSOS format, which we call *PGSOS*, give a number of example specifications, and state that the format has similar well-behavedness properties as its nondeterministic correspondent: bisimilarity is a congruence on all models, an up-to-context technique for bisimilarity proofs is available, and guarded recursive specifications involving the specified operators have solutions which are unique up to bisimilarity.

The paper is divided in two parts: in the first (Section 3 through Section 5) we introduce LTS and PTS, recall the GSOS format and introduce PGSOS, give examples and state properties. In a second, technical part we explain how the format together with its properties was derived using (co)algebraic methods:

It arose from an abstract categorical account of operator specification formats by Turi and Plotkin [TP97]. Among other things, they generalize the GSOS format for LTS to the *abstract GSOS* format for coalgebras of an arbitrary Set-functor B. Such a functor describes the type of system under consideration and a specification in abstract GSOS is a natural transformation between two functors constructed from B. It turns out that the abstract framework allows elegant and relatively simple proofs for several well-behavedness properties of the specification format. It remains to be shown that the abstract format is indeed related to some concrete rule shape. Turi and Plotkin state that when one takes a functor B appropriate for modelling LTS, abstract GSOS indeed corresponds to the known GSOS format. But this fact is not proved in detail in loc. cit.

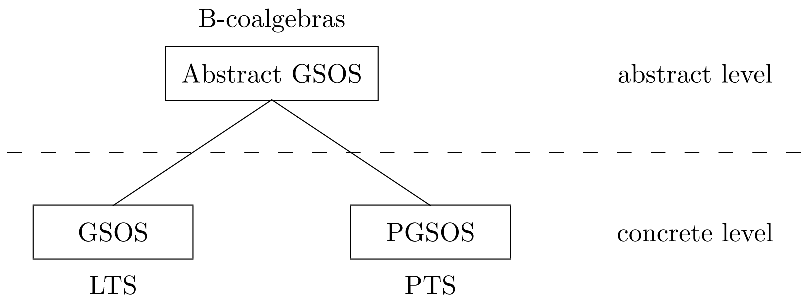


Figure 1: PGSOS arises as an instance of abstract GSOS.

We fill this gap in Section 7. Our proof establishes the correspondence by first decomposing the type of natural transformation under consideration in a number of steps. Then an elementary representation theorem is developed for the natural transformations of the simplest type encountered (c.f. Theorem 7.6). A stepwise extension of this result to the more complex types eventually yields a representation corresponding to GSOS specifications. Through this correspondence, GSOS inherits the well-behavedness results proved in the abstract framework.

The idea is now to use the same approach to obtain a format for PTS. Therefore we first describe PTS as coalgebras of an appropriate functor B. Instantiating abstract GSOS with this functor yields a class of natural transformations which can be viewed as well-behaved specifications for the probabilistic setting. The natural transformations in this class are then again characterised in terms of transition rules in a certain format, which can practically be used to write down specifications. The idea is pictured in Figure 1.

The advantage of our modular proof is that a similar decomposition as in the nondeterministic case can be carried out in the setting of PTS, so the first part of the proof can basically be reused. The elementary representation theorem needed this time, which we consider the main technical result of this paper (c.f. Theorem 8.6), is considerably harder to prove though. This result may be interesting in its own right.

Our argument establishes a correspondence between PGSOS specifications as introduced in the first part and abstract GSOS instantiated with the functor B we used to model PTS. Through this correspondence we obtain a number of well-behavedness results for the new format as special cases of properties that have been shown for the abstract framework. These include the statements that bisimilarity is a congruence on every model of a PGSOS specification, that the bisimulation up-to-context proof technique is valid for them, and that every guarded recursive specification has a solution in some model of a PGSOS specification, and that this solution is determined up to (probabilistic) bisimilarity.

This technical report is the full version of the extended abstract presented at CMCS 2002 [Bar02]. It adds the full treatment of the nondeterministic setting as well as several proofs for the probabilistic case, like the one of the representation result mentioned above (Theorem 8.6).

2. Preliminaries and notation

We use the categorical notions of a functor, natural transformation, and initial/final object. We mostly work in Set, the category of sets and total functions. We write 1_{C} for a final object in a category C, which in Set is the singleton set $1 = \{*\}$ (in this case we usually drop the subscript). The unique morphism from any object to the final one is denoted by $!_X : X \to 1_{\mathsf{C}}$.

By $\prod_{i\in I} X_i$ and $\coprod_{i\in I} X_i$ we denote the I-indexed categorical product and coproduct with projections $\pi_j : (\prod_{i\in I} X_i) \to X_j$ and injections $\iota_j : X_j \to \coprod_{i\in I} X_i$. For arrows $f_i : X \to Y_i$ and $g_i : Y_i \to Z$ ($i \in I$) we write the pairing as $\langle f_i\rangle_{i\in I} : X \to \prod_{i\in I} Y_i$ and the case analysis as $[g_i]_{i\in I} : \coprod_{i\in I} Y_i \to Z$. In Set, by $(x_i)_{i\in I} \in \prod_{i\in I} X_i$ we denote the unique element with $\pi_j\big((x_i)_{i\in I}\big) = x_j$ for all $j \in I$. We will sometimes drop the subscript $i \in I$ if it is reasonably clear from the context. In case $I = \{1, \ldots, m\}$ for some $m \in \mathbb{N}$ we further write $\vec{x} := \langle x_1, \ldots, x_m\rangle \in X_1 \times \cdots \times X_m := \prod_{i\in I} X_i$.

The image and inverse image of a function $f : X \to Y$ are written as $f^{-1}(y) := \{x \in X \mid f(x) = y\}$ for $y \in Y$ and $f[X'] = \{f(x) \in Y \mid x \in X'\}$ for $X' \subseteq X$. The non-negative real numbers are denoted by $\mathbb{R}^+_0$. The support of a function $\mu : X \to \mathbb{R}^+_0$ is defined to be the set $\mathtt{supp}(\mu) := \{x \in X \mid \mu(x) > 0\} \subseteq X$. For such functions μ and $X' \subset X$ we further overload the bracket notation to mean $\mu[X'] := \sum_{x\in X'} \mu(x)$. This is done in situations only where the sum is defined. We will use the notation

$$\sum\big(\, \mu(x) \;\big|\; x \in X'\big) := \sum_{x\in X'} \mu(x)$$

if the description of the set $X' \subseteq X$ is such that the expression on the right hand side would be unwieldy. Furthermore, for $r \in [0, 1]$ we abbreviate $1 - r$ to $\bar{r}$.

To update or extend a function $u : X \to Y$ by one value, we write

$$u[d := y] : X \cup \{d\} \to Y \quad \text{with} \quad u[d := y](x) := \begin{cases} y & \text{if } x = d, \\ u(x) & \text{otherwise.} \end{cases}$$

3. Nondeterministic and probabilistic transition systems

In this section we define nondeterministic as well as probabilistic transition systems. For both we assume a set L of input labels to be fixed.

Definition 3.1 *A* **labelled transition systems (LTS)** *is a pair $\langle P, \alpha\rangle$ consisting of a set of states P and a transition function $\alpha : P \times L \to \mathcal{P}_\omega P$, where for any set X we define the finite powerset construction $\mathcal{P}_\omega$ to be*

$$\mathcal{P}_\omega X := \big\{X' \subseteq X \;\big|\; X \text{ is finite}\big\}.$$

A pair $\langle\langle P, \alpha\rangle, p\rangle$ of an LTS $\langle P, \alpha\rangle$ and a state $p \in P$ is called a **(nondeterministic) process**. *We will sometimes leave the LTS implicit and just talk about a process p.*

At any moment, a process $\langle\langle P, \alpha\rangle, p\rangle$ receives input labels from L. If $\alpha(p, a)$ for some input $a \in L$ is empty, we say that in state p the system rejects a or that a is *disabled*. Otherwise, the label a is *enabled* and the process responds to it by making a move to one of the states in $\alpha(p, a)$, the potential a-successor states of p. In case there is more than one, the choice of the actual a-successor $p' \in \alpha(p, a)$ is made *nondeterministically*.

A PTS $\langle P, \alpha\rangle$ is a similar type of systems where the choice of the successor states is made *probabilistically*.

Definition 3.2 *A* **probabilistic transition system (PTS)** *is a pair $\langle P, \alpha\rangle$ of a set of states P and a transition function $\alpha : P \times L \to \mathcal{D}_\omega P$, where $\mathcal{D}_\omega$ constructs (possibly empty) probability distributions with finite support, namely*

$$\mathcal{D}_\omega X := \big\{\mu : X \to \mathbb{R}^+_0 \;\big|\; \mathtt{supp}(\mu) \text{ is finite}, \mu[X] \in \{0, 1\}\big\},$$

where $\mathtt{supp}(\mu) := \{x \in X \mid \mu(x) > 0\}$ and $\mu[X'] := \sum_{x\in X'} \mu(x)$. A pair $\langle\langle P, \alpha\rangle, p\rangle$ of a PTS $\langle P, \alpha\rangle$ and a state $p \in P$ is called a **(probabilistic) process**. *We will again sometimes leave the PTS implicit.*

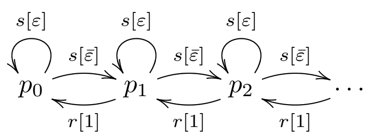
When a probabilistic process $\langle\langle P, \alpha\rangle, p\rangle$ receives the label $a \in L$, it becomes the process $p' \in P$ with probability $\alpha(p,a)(p')$. This probability is positive for at most finitely many states p' and if it is zero for all states (i.e. $\alpha(p,a)[P] = 0$) then the label a is *disabled* in p.

We use the following arrow notation for a nondeterministic or probabilistic process $\langle\langle P, \alpha\rangle, p\rangle$ respectively in case no confusion about α is likely to arise:

LTS			PTS		
$p \stackrel{a}{\not\longrightarrow}$	for	$\alpha(p,a) = \emptyset$	$p \stackrel{a}{\not\longrightarrow}$	for	$\alpha(p,a)[P] = 0$
$p \stackrel{a}{\longrightarrow}$	for	$\alpha(p,a) \neq \emptyset$	$p \stackrel{a}{\longrightarrow}$	for	$\alpha(p,a)[P] = 1$
$p \stackrel{a}{\longrightarrow} p'$	for	$p' \in \alpha(p,a)$	$p \stackrel{a[r]}{\longrightarrow} p'$	for	$\alpha(p,a)(p') = r$

We usually do not draw arrows with a zero probability.

Example 3.3 *As an example, we consider what could be called a* **lossy bag***: a system that can perform store (s) and remove (r) operations, where the number of removals is limited to the number of previous storages. But the system is lossy in the sense that a store operation fails to actually add something to the bag with a given probability* $\varepsilon \in [0,1]$. *We model the bag as a probabilistic process* p_0 *in a PTS* $\langle P, \alpha_P\rangle$ *for the set of labels* $L := \{s, r\}$. *The set of states is* $P := \{p_i \mid i \in \mathbb{N}\}$, *where* p_i *is the state of the system with* i *items in storage. A store event can always be processed and will increase the number of stored items by one if everything works fine. But with probability* ε *an error occurs and the number stays the same. A remove event is possible if there is at least one item stored and it will decrease the number of stored items by one. Graphically, we have the following system, where we abbreviate* $1 - \varepsilon$ *to* $\bar{\varepsilon}$*:*



3.1 Bisimulation

We often assume that the states of a system are internal and cannot be accessed as such. One can just experiment with the system and observe whether a given action is enabled or disabled. If it was enabled, one can continue to analyse the successor state. Processes that cannot be distinguished this way are called *behaviourally equivalent*. For LTS and PTS this equivalence can be established using the notion of a *bisimulation*.

Definition 3.4 *A* **(strong) bisimulation** *between two LTS* $\langle P, \alpha_P\rangle$ *and* $\langle Q, \alpha_Q\rangle$ *is a relation* $R \subseteq P \times Q$ *such that for all* $\langle p, q\rangle \in R$ *and* $a \in L$ *we have that*

$$p \stackrel{a}{\longrightarrow} p' \quad \text{implies} \quad q \stackrel{a}{\longrightarrow} y \quad \text{for some } y \in Q \text{ with } \langle p', y\rangle \in R, \text{ and}$$
$$q \stackrel{a}{\longrightarrow} q' \quad \text{implies} \quad p \stackrel{a}{\longrightarrow} x \quad \text{for some } x \in P \text{ with } \langle x, q'\rangle \in R.$$

The greatest bisimulation between two LTS is called **bisimilarity** *and denoted by* $\sim$.

It is easy to see that the greatest bisimulation always exists and that it is an equivalence relation when we take the same system for $\langle P, \alpha_P\rangle$ and $\langle Q, \alpha_Q\rangle$. Two processes are behaviourally equivalent just in case they are bisimilar.

Bisimilarity for PTS is slightly more complicated. The definition can be simplified a bit when it is restricted to equivalence relations, as done e.g. by Larsen and Skou [LS91]. We prefer to work with a general notion, mainly because the relations arising in examples are not always equivalences (or it is at least not clear that they are).

Definition 3.5 *A* **(probabilistic) bisimulation** *between two PTS* $\langle P, \alpha_P\rangle$ *and* $\langle Q, \alpha_Q\rangle$ *is a relation* $R \subseteq P \times Q$ *such that for all* $\langle p, q\rangle \in R$ *and* $a \in L$ *we have that there exists a distribution* $\mu \in \mathcal{D}_\omega R$ *such that*

$$p \xrightarrow{a[r]} p' \quad \text{just in case} \quad r = \sum \big(\mu(\langle p', y\rangle) \;\big|\; y \in Q \text{ with } \langle p', y\rangle \in R\big),$$
$$q \xrightarrow{a[r]} q' \quad \text{just in case} \quad r = \sum \big(\mu(\langle x, q'\rangle) \;\big|\; x \in P \text{ with } \langle x, q'\rangle \in R\big).$$

The greatest bisimulation between two PTS is called **(probabilistic) bisimilarity** *and again denoted by* $\sim$.

As for LTS, a greatest bisimulation between two PTS can be shown to exist, and this bisimilarity relation coincides with behavioural equivalence.

4. The GSOS format for LTS

We now turn to operator specification formats. Before we introduce the new format for PTS we recall correspondent in the nondeterministic setting and state properties of it. We assume that a signature $\Sigma = (\Sigma_n)_{n\in\mathbb{N}}$ is fixed, where for any $n \in \mathbb{N}$ we view an element $\sigma \in \Sigma_n$ as an operator symbol with arity n. This signature is *finitary* in the sense that every operator symbol has a finite arity, but we do not restrict the overall number of symbols under consideration.

Definition 4.1 *Given a signature* $\Sigma = (\Sigma_n)_{n\in\mathbb{N}}$ *and a set* X *we denote by* $\mathrm{T}X$ *the set of* **terms** *for the signature* Σ *with variables in* X. *This is the smallest set containing* X *such that for all* $n \in \mathbb{N}$ *and* $\sigma \in \Sigma_n$ *whenever* $t_1, \ldots, t_n \in \mathrm{T}X$ *then also* $\sigma(t_1, \ldots, t_n) := \langle \sigma, \langle t_1, \ldots, t_n\rangle\rangle \in \mathrm{T}X$. *By* $\mathtt{vars}(t)$ *we denote the set of variables from* X *occurring in a terms* $t \in \mathrm{T}X$ *(i.e.* $\mathtt{vars}(x) := \{x\}$ *for* $x \in X$ *and* $\mathtt{vars}(\sigma(t_1, \ldots, t_n)) := \mathtt{vars}(t_1) \cup \cdots \cup \mathtt{vars}(t_n)$*).*

Definition 4.2 *A* **GSOS rule** *has the shape*

$$\frac{\begin{array}{lll} x_i \xrightarrow{b} & b \in R_i,\ 1 \le i \le n & (i) \\ x_i \overset{b}{\not\longrightarrow} & b \in P_i,\ 1 \le i \le n & (ii) \\ x_{i_j} \xrightarrow{l_j} y_j & 1 \le j \le m & (iii) \end{array}}{\sigma(x_1, \ldots, x_n) \xrightarrow{a} t}$$

where

- $\sigma \in \Sigma_n$ *for some* $n \in \mathbb{N}$ *is the* type *of the rule,*
- $x_1, \ldots, x_n$ *are distinct argument state variables (we set* $X := \{x_1, \ldots, x_n\}$*),*
- *for* $1 \le i \le n$, $R_i, P_i \subseteq L$ *with* $R_i \cap P_i = \emptyset$ *are the sets of* requested *and* prohibited *labels for the i-th argument,*
- $y_1, \ldots, y_m$ *for some* $m \in \mathbb{N}$ *are distinct* successor state variables *such that* $Y \cap X = \emptyset$ *for* $Y := \{y_1, \ldots, y_m\}$, *where each* y_j *is tagged as a successor of argument* $i_j \in \{1, \ldots, n\}$ *for a requested label* $l_j \in R_{i_j}$ $(1 \le j \le m)$,
- $a \in L$ *is the* label *of the rule,*
- $t \in \mathrm{T}(X \cup Y)$ *such that* $Y \subseteq \mathtt{vars}(t)$ *is the* target *of the rule.*

The premises of type (ii) are called *negative*, the others are *positive*. Moreover, we refer to the premises of type (iii) as *reference premises*, the others are *applicability premises*. This presentation of a GSOS rule differs from the standard one in the literature in that we use positive applicability premises (i.e. those of type (i)). Usually one would replace them by reference premises pointing to fresh variables not used in the target. We introduced positive applicability premises in order to be able to disclose unused variables, which are troublesome for our purposes, as the treatment of probabilistic systems will make apparent. We usually omit a positive applicability premise when its presence is enforced by a reference premise (since we assumed $l_j \in R_{i_j}$).

A GSOS specification is a set of GSOS rules satisfying a size restriction which accounts for the image finiteness assumption we imposed on LTS. The following notion is introduced in order to express this condition.

Definition 4.3 *A tuple $E_1, \ldots, E_n \subseteq L$ is a* **trigger** *of a GSOS rule*

$$\frac{\begin{array}{ll} x_i \xrightarrow{b} & b \in R_i,\ 1 \leq i \leq n \\ x_i \stackrel{b}{\not\rightarrow} & b \in P_i,\ 1 \leq i \leq n \\ x_{i_j} \xrightarrow{l_j} y_j & 1 \leq j \leq m \end{array}}{\sigma(x_1, \ldots, x_n) \xrightarrow{a} t}$$

if $R_i \subseteq E_i$ and $P_i \cap E_i = \emptyset$ for all $1 \leq i \leq n$.

For $1 \leq i \leq n$ the set E_i is supposed to hold the enabled transitions for the process supplied as the i-th argument of σ. The above rule is triggered, if for each argument all requested and non of the prohibited transitions are enabled.

Definition 4.4 *A* **GSOS specification** *is a set $\mathcal{R}$ of GSOS rules such that for all $\sigma \in \Sigma_n$, $a \in L$, and $E_1, \ldots, E_n \subseteq L$ only finitely many rules with type σ and label a in $\mathcal{R}$ are triggered by $E_1, \ldots, E_n$.*

Often, a GSOS specification is used to specify one particular LTS, or rather, to equip the set of terms without variables with a transition function. Here we will adopt a broader notion of a model of a GSOS specification. The term model above will reappear later as the *initial* one.

Definition 4.5 *A* **model** *of a GSOS specification $\mathcal{R}$ is a triple $\langle P, (\sigma_P), \alpha\rangle$ consisting of an LTS $\langle P, \alpha\rangle$ and a collection of operators $\sigma_P : P^n \to P$ for each $n \in \mathbb{N}$ and $\sigma \in \Sigma_n$, such that for all $n \in \mathbb{N}$, $\sigma \in \Sigma_n$, and $p_1, \ldots, p_n \in P$ the transitions α assigns to $\sigma_P(p_1, \ldots, p_n) \in P$ are precisely those derivable as instances of the rules in $\mathcal{R}$.*

An instantiation of a rule

$$\frac{\begin{array}{ll} x_i \xrightarrow{b} & b \in R_i,\ 1 \leq i \leq n \\ x_i \stackrel{b}{\not\rightarrow} & b \in P_i,\ 1 \leq i \leq n \\ x_{i_j} \xrightarrow{l_j} y_j & 1 \leq j \leq m \end{array}}{\sigma(x_1, \ldots, x_n) \xrightarrow{a} t}$$

in $\mathcal{R}$ is determined by states $p_1, \ldots, p_n, q_1, \ldots q_m \in P$ and it yields the derivation

$$\frac{\begin{array}{ll} p_i \xrightarrow{b} & b \in R_i, 1 \leq i \leq n \\ p_i \stackrel{b}{\not\rightarrow} & b \in P_i, 1 \leq i \leq n \\ p_{i_j} \xrightarrow{l_j} q_j & 1 \leq j \leq m \end{array}}{\sigma_P(p_1, \ldots, p_n) \xrightarrow{a} [\![\ t[x_i := p_i,\ y_j := q_j]\ ]\!]_P}$$

where $t[x_i := p_i, y_j := q_j]$ is the term that results by replacing in t each x_i by p_i and y_j by q_j for $1 \leq i \leq n$ and $1 \leq j \leq m$, and $[\![t']\!]_P \in P$ for $t' \in \mathrm{T}P$ is the evaluation of t' by applying the appropriate operators from (σ_P). (Note that the arrows in a GSOS rule are just symbols, whereas the arrows in the instance of the rule are the transitions allowed by the LTS $\langle P, \alpha\rangle$).

The models of a GSOS specification are well-behaved in many respects. In order to express some of those properties we define the notions of a congruence, a bisimulation up-to-context, and a guarded recursive specification.

Definition 4.6 *Let $\langle P, (\sigma_P), \alpha_P\rangle$ and $\langle Q, (\sigma_Q), \alpha_Q\rangle$ be models of a GSOS specification $\mathcal{R}$. A relation $R \subseteq P \times Q$ is a congruence for the two models if for all $n \in \mathbb{N}$ and $\sigma \in \Sigma_n$*

$$\langle p_1, q_1\rangle, \ldots, \langle p_n, q_n\rangle \in R \quad \textit{implies} \quad \langle \sigma_P(p_1, \ldots, p_n), \sigma_Q(q_1, \ldots, q_n)\rangle \in R.$$

The **congruence closure** *of a relation $R \subseteq P \times Q$ is the smallest congruence containing R.*

Definition 4.7 *A* **bisimulation up-to-context** *between two models $\langle P, (\sigma_P), \alpha_P\rangle$ and $\langle Q, (\sigma_Q), \alpha_Q\rangle$ of a GSOS specification $\mathcal{R}$ is a relation $R \subseteq P \times Q$ such that for all $\langle p, q\rangle \in R$ and $a \in L$ we have that*

$$p \xrightarrow{a} p' \quad \textit{implies} \quad q \xrightarrow{a} y \quad \textit{for some } y \in Q \textit{ with } \langle p', y\rangle \in \bar{R}, \textit{ and}$$
$$q \xrightarrow{a} q' \quad \textit{implies} \quad p \xrightarrow{a} x \quad \textit{for some } x \in P \textit{ with } \langle x, q'\rangle \in \bar{R},$$

where $\bar{R}$ is the congruence closure of R.

Definition 4.8 *A* **(nondeterministic) guarded recursive specification** *is a pair $\langle P, Tr\rangle$ consisting of a set of variables X and a set of transitions*

$$Tr \subseteq \left\{ x \xrightarrow{a} t \;\middle|\; x \in X,\ a \in L,\ t \in \mathrm{T}X \right\}$$

such that for all $x \in X$ and $a \in L$ the set Tr contains finitely many transitions from x with label a only. A **solution** *of $\langle X, Tr\rangle$ in a model $\langle P, (\sigma_P), \alpha\rangle$ of a GSOS specification $\mathcal{R}$ is given by an assignment of variables $h : X \to P$ such that for all $x \in X$, $a \in L$, and $q \in P$*

$$h(x) \xrightarrow{a} q \quad \textit{just in case} \quad (x \xrightarrow{a} t) \in Tr \quad \textit{for some } t \in \mathrm{T}X \textit{ with } [\![\, t[y := h(y)] \,]\!]_P = q.$$

In the literature, the term *guarded recursive equations* is used for a set of equations of the shape

$$x = t \quad (x \in X,\ t \in \mathrm{T}X, t \text{ guarded}),$$

for a suitable notion of *guardedness*. This is some syntactical restriction on the terms t guaranteeing that the immediate transitions of the process denoted by t can be derived without knowing the instantiation of the variables occurring in it. To this end, one identifies operators that define an initial transition (like the action prefixing $a.x$ describing a process that can make an a-transition to move to state x) and demands that every variable is preceded (guarded) by at least one application of such an operator. Our definition is a slightly more general encoding of the same idea, since it does not require the presence (and identification) of the operators above.

Models of a GSOS specification are well behaved in many respects. Amongst others, they have the following properties.

Proposition 4.9 *Let $\langle P, (\sigma_P), \alpha_P\rangle$ and $\langle Q, (\sigma_Q), \alpha_Q\rangle$ be models of a GSOS specification $\mathcal{R}$.*

1. *The congruence closure of any bisimulation R between $\langle P, (\sigma_P), \alpha_P\rangle$ and $\langle Q, (\sigma_Q), \alpha_Q\rangle$ is a bisimulation again. In particular, the bisimilarity relation $\sim \subseteq P \times Q$ itself is a congruence.*

2. *Every bisimulation up-to-context between $\langle P, (\sigma_P), \alpha_P\rangle$ and $\langle Q, (\sigma_Q), \alpha_Q\rangle$ is contained in some (standard) bisimulation. This enables the following bisimulation up-to-context proof principle: to prove $p \sim q$ it suffices to find a bisimulation up-to-context R with $\langle p, q\rangle \in R$.*

3. *Every guarded recursive specification $\langle X, Tr\rangle$ has a solution in some model of $\mathcal{R}$. Furthermore, such a solution is determined up to bisimilarity.*

With the development in Section 7 these properties and others will follows from corresponding facts about the abstract framework by Turi and Plotkin [TP97]. The first statement is well known. The other two may be new. The bisimulation up-to-context proof principle was studied by Sangiorgi [San98], who proves that it is valid for specifications in the more restrictive DeSimone format. He also gives an example for an operator specification for which the principle is not valid. The example is beyond GSOS, since it involves a chaining of premises (*look-ahead*) as exemplified in the rule below.

$$\frac{x \xrightarrow{a} y \quad y \xrightarrow{a} z}{\sigma(x) \xrightarrow{a} \sigma(z)}$$

5. The PGSOS format for PTS

In this section we introduce a specification format for PTS. It bears similarity with the GSOS format above and is therefore called *PGSOS* for *probabilistic GSOS*. We start by considering again the specification of a sequential composition from the introduction, which consisted of the following transition rules.

$$\frac{x \xrightarrow{l[r]} x'}{x.y \xrightarrow{l[r]} x'.y} \qquad \frac{x \stackrel{l'}{\not\rightarrow} \ (l' \in L) \qquad y \xrightarrow{l[r]} y'}{x.y \xrightarrow{l[r]} y'} \qquad \text{(both for all } l \in L \text{ and } r \in [0,1])$$

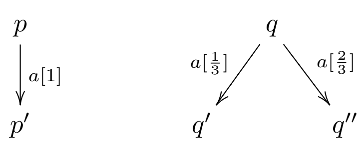
At first sight one may be tempted to view this as a specification for a (nondeterministic) system with labels from the set $\{\ l[r] \ \mid l \in L,\ r \in [0,1]\}$ and propose to use the corresponding instance of the GSOS format for this setting. But the situation is not as simple as that.

First, for a specification to have models, we need to ensure that the generated transitions yield a probability distribution indeed. So we need a criterion to guarantee that the probabilities of all generated transitions sum up to one if there are any. This will lead to a new global constraint on the sets of rules in a PGSOS specification.

Second, we have to realise that we cannot fix the probabilities for the transitions in the premises. To illustrate this point, we consider the specification rules below. They are meant to define an operator δ that removes all transitions with probability less than one, i.e. all "nondeterministic" transitions.

$$\frac{x \xrightarrow{l[1]} x'}{\delta(x) \xrightarrow{l[1]} \delta(x')} \quad \text{(for all } l \in L) \tag{5.1}$$

To see that this specification is troublesome, assume that $\langle P, \delta_P, \alpha \rangle$ is a model of it which contains the following two processes.



(5.2)

Note that p and q are bisimilar: for both, the only enabled label is a, and the a-transition leads to an inert state with probability one (it is easy to check that the relation $R = \{\langle p, q\rangle, \langle p', q'\rangle, \langle p', q''\rangle\}$ is a probabilistic bisimulation). Still, $\delta_P(p)$ and $\delta_P(q)$ are not bisimilar, because $\delta_P(p)$ can do an a-transition while $\delta_P(q)$ cannot. So no operator δ_P on $\langle P, \alpha \rangle$ satisfying the specification preserves bisimilarity, one of our basic requirements for the format to be found.

Generally, using an argument similar to the one above we can see that rules with premises demanding an absolute probability for a transition cause problems. This point is taken care of by our definition of a PGSOS rule, in which the probabilities in the premises are treated as variables.

Definition 5.1 *A rule in* **PGSOS** *has the shape*

$$\frac{\begin{array}{ll} x_i \xrightarrow{b} & b \in R_i,\ 1 \leq i \leq n \\ x_i \xrightarrow{b}\!\!\!\!\!\not\;\; & b \in P_i,\ 1 \leq i \leq n \\ x_{i_j} \xrightarrow{l_j[z_j]} y_j & 1 \leq j \leq m \end{array}}{\sigma(x_1, \ldots, x_n) \xrightarrow{a[w \cdot \prod_j z_j]} t}$$

where

- $\sigma \in \Sigma_n$ *for some* $n \in \mathbb{N}$ *is the* type *of the rule,*
- $x_1, \ldots, x_n$ *are distinct argument state variables (we set* $X := \{x_1, \ldots, x_n\}$*),*
- $R_i, P_i \subseteq L$ *with* $R_i \cap P_i = \emptyset$ *are the sets of* requested *and* prohibited *labels for the* i*-th argument* x_i *(*$1 \leq i \leq n$*),*
- $y_1, \ldots, y_m$ *for some* $m \in \mathbb{N}$ *are distinct* successor state variables *such that* $Y \cap X = \emptyset$ *for* $Y := \{y_1, \ldots, y_m\}$*, where each* y_j *is tagged as a successor of argument* $i_j \in \{1, \ldots, n\}$ *for a requested label* $l_j \in R_{i_j}$ *(*$1 \leq j \leq m$*),*
- $z_1, \ldots, z_m$ *are distinct probability variables,*
- $a \in L$ *is the* label *of the rule,*
- $w \in (0, 1]$ *is the* weight *of the rule.*
- $t \in \mathrm{T}(X \cup Y)$ *such that* $Y \subseteq \mathtt{vars}(t)$ *is the* target *of the rule.*

It is easy to see that whenever such a rule is applicable in a given situation, the probabilities of all transitions derivable by it sum up to the weight w of the rule. To make sure that in any situation the accumulated probability of all derivable transitions for the same label is either zero or one, it thus suffices to require that the weights of all applicable rules sum up to zero or one. we applying the notion of a trigger from Def. 4.3 in the obvious way to PGSOS rules also to talk about all possible applicability scenarios.

Definition 5.2 *A* PGSOS specification *is a set* $\mathcal{R}$ *of PGSOS rules such that for all* $n \in \mathbb{N}$*,* $\sigma \in \Sigma_n$*,* $a \in L$*, and* $E_1, \ldots, E_n \subseteq L$ *only finitely many rules with type* σ *and label* a *in* $\mathcal{R}$ *are triggered by* $E_1, \ldots, E_n$*, and in case there are any, the weights of all these rules sum up to* 1*.*

Definition 5.3 *A* **model** *of a PGSOS specification* $\mathcal{R}$ *is a triple* $\langle P, (\sigma_P), \alpha \rangle$ *consisting of a PTS* $\langle P, \alpha \rangle$ *and a collection of functions* $\sigma_P : P^n \to P$ *for all* $n \in \mathbb{N}$ *and* $\sigma \in \Sigma_n$ *such that the following holds: for all* $n \in \mathbb{N}$*,* $\sigma \in \Sigma_n$*,* $a \in L$*, and* $p_1, \ldots, p_n, q \in P$ *we have*

$$\sigma_P(p_1, \ldots, p_n) \xrightarrow{a[u]} q$$

just in case u *is the sum of all contributions to an* a*-transitions from* $\sigma_P(p_1, \ldots, p_n)$ *to* q *that can be derived from different instantiations of the rules in* $\mathcal{R}$*.*

An instantiation of a rule

$$\frac{\begin{array}{ll} x_i \xrightarrow{b} & b \in R_i,\ 1 \leq i \leq n \\ x_i \xrightarrow{b}\!\!\!\!\!\not\;\; & b \in P_i,\ 1 \leq i \leq n \\ x_{i_j} \xrightarrow{l_j[z_j]} y_j & 1 \leq j \leq m \end{array}}{\sigma(x_1, \ldots, x_n) \xrightarrow{a[w \cdot \prod_j z_j]} t}$$

in $\mathcal{R}$ is determined by states $p_1, \ldots, p_n, q_1, \ldots, q_m \in P$ and probabilities $u_1, \ldots, u_m \in (0,1]$ and it yields the derivation

$$\frac{\begin{array}{ll} p_i \xrightarrow{b} & b \in R_i, 1 \le i \le n \\ p_i \overset{b}{\nrightarrow} & b \in P_i, 1 \le i \le n \\ p_{i_j} \xrightarrow{l_j[u_j]} q_j & 1 \le j \le m \end{array}}{\sigma_P(p_1, \ldots, p_n) \xrightarrow{a[w \cdot \prod_j u_j]} [\![\, t[x_i := p_i,\, y_j := q_j] \,]\!]_P}$$

where $[\![t']\!]_P \in P$ for $t' \in \mathrm{T}P$ is the evaluation of t' by applying the appropriate operators from (σ_P). This instance contributes a portion of $w \cdot \prod_j u_j$ to the a-transition from $\sigma_P(p_1, \ldots, p_n)$ to $[\![\, t[x_i := p_i,\, y_j := q_j] \,]\!]_P$.

Before we consider properties of PGSOS specifications, we first give some examples.

5.1 Some examples of PGSOS specifications

To illustrate the PGSOS format, we present the definitions of some basic operators.

1. A constant $0 \in \Sigma_0$ is intended to yield the idle process that cannot do any transitions. We achieve this by giving no rules with type 0.
2. Consider the atomic action constant $a \in \Sigma_0$ for $a \in L$. The associated process should have a as its only enabled label and an a-transition should lead to the state 0 with probability 1. We specify the constant with the following single rule without premises.

$$\frac{}{a \xrightarrow{a[1]} 0}$$

3. Next we specify a probabilistic choice operator $\oplus_r \in \Sigma_2$ for $r \in [0,1]$. For processes x and y we want $x \oplus_r y$ to be a process behaving either as x or as y, depending on the first input label and the probability r. In case the input can only be processed by x, the system should behave like x, and similar for y. If both can react, the decision should be made in favour of x with probability r and otherwise in favour of y. This is captured by the following set of PGSOS rules (for $\bar{r} = 1 - r$):

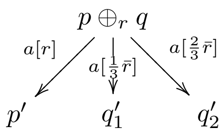
$$\frac{x \xrightarrow{l[z']} x' \quad y \overset{l}{\nrightarrow}}{x \oplus_r y \xrightarrow{l[z']} x'} \qquad \frac{x \overset{l}{\nrightarrow} \quad y \xrightarrow{l[z']} y'}{x \oplus_r y \xrightarrow{l[z']} y'}$$

$$\frac{x \xrightarrow{l[z']} x' \quad y \xrightarrow{l}}{x \oplus_r y \xrightarrow{l[r \cdot z']} x'} \qquad \frac{x \xrightarrow{l} \quad y \xrightarrow{l[z']} y'}{x \oplus_r y \xrightarrow{l[\bar{r} \cdot z']} y'} \qquad \text{(each for all } l \in L\text{)}$$

To see that these rules satisfy the global constraints from Def. 5.2, for all $a \in L$ and $E_1, E_2 \subseteq L$ we have to inspect the rules for $\oplus_r$ and a which are triggered by E_1 and E_2: it is either no rule at all (in case $a \notin E_1 \cup E_2$), one of the upper ones with $l = a$ (in case $a \in (E_1 \setminus E_2) \cup (E_2 \setminus E_1)$, each of which has weight 1, or both lower ones with $l = a$ (if $a \in E_1 \cap E_2$), the weights of which sum up to $r + \bar{r} = 1$.

To illustrate Def. 5.3, we spell out what the requirement on a model $\langle P, (\sigma_P), \alpha \rangle$ amounts to in a concrete case. Let $p, q \in P$ again be the two processes from (5.2). Both can make an a-transition, so the third and fourth rule with $l = a$ are applicable to $p \oplus_r q$ (here and in the following, we will drop the subscript P for the concrete operators. So we just write $\oplus_r : P \times P \to P$). They derive an a-transition which leads to the a-successor of p with probability r and to an a-successor of

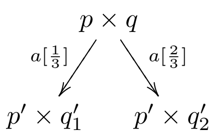
q otherwise. In case this choice is made for q, the conditional probability of moving to q'_i is the same as the probability of moving from q to it.



4. Furthermore, we define the product operator $\times \in \Sigma_2$ such that the process $x \times y$ consists of two components x and y waiting for input side by side. The enabled labels are those that are enabled for each of x and y. On such a label each component will independently make a move according to its own transition probability and the whole process will become the product of the two resulting states. The operation is defined by the following set of rules:

$$\frac{x \xrightarrow{l[u]} x' \qquad y \xrightarrow{l[v]} y'}{x \times y \xrightarrow{l[u \cdot v]} x' \times y'} \qquad \text{(for all } l \in L\text{)}$$

Considering again p and q from (5.2) we get that $p \times q$ is the process below.



Note that we may have $p' \times q'_1 = p' \times q'_2$. In that case the arrows above would actually represent one arrow with probability $\frac{1}{3} + \frac{2}{3} = 1$.

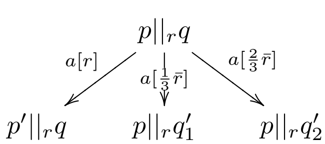
5. For any $r \in [0, 1]$ the (binary) probabilistic parallel composition $x||_r y$ of the two processes x and y is intended to behave as follows: an input label a can be processed if it can by at least one of x or y. The input is always handled by one of them only, the other stays unchanged. If a is enabled for only one process, then this one is taken. If both components are able to deal with the input, then the choice is made probabilistically, where x is chosen with the probability r.

   The operator $||_r \in \Sigma_2$ is specified by the rules below.

$$\frac{x \xrightarrow{l[z']} x' \qquad y \stackrel{l}{\not\longrightarrow}}{x||_r y \xrightarrow{l[z']} x'||_r y} \qquad \frac{x \stackrel{l}{\not\longrightarrow} \qquad y \xrightarrow{l[z']} y'}{x||_r y \xrightarrow{l[z']} x||_r y'}$$

$$\frac{x \xrightarrow{l[z']} x' \qquad y \xrightarrow{l}}{x||_r y \xrightarrow{l[r \cdot z']} x'||_r y} \qquad \frac{x \xrightarrow{l} \qquad y \xrightarrow{l[z']} y'}{x||_r y \xrightarrow{l[\bar{r} \cdot z']} x||_r y'} \qquad \text{(each for all } l \in L\text{)}$$

   Again for p and q from (5.2) we get the following transitions:



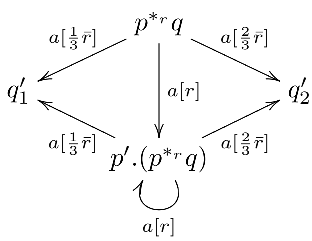
6. All the examples so far were simple in the sense that they did not use terms consisting of more than one operator application as their target. As a more complex example we specify

a probabilistic variant of the Kleene-Star operator $(-)^{*_r}(-) \in \Sigma_2$ for $r \in [0,1]$. It uses the sequential composition from Section 5 (it is easily seen that the rules given there form a PGSOS specification). The operator is specified by the following rules.

$$\frac{x \xrightarrow{l[z]} x' \qquad y \xrightarrow{l}}{x^{*_r}y \xrightarrow{l[r\cdot z]} x'.(x^{*_r}y)} \qquad \frac{x \xrightarrow{l[z]} x' \qquad y \stackrel{l}{\not\rightarrow}}{x^{*_r}y \xrightarrow{l[z]} x'.(x^{*_r}y)}$$

$$\frac{x \xrightarrow{l} \qquad y \xrightarrow{l[z]} y'}{x^{*_r}y \xrightarrow{l[\bar{r}\cdot z]} y'} \qquad \frac{x \stackrel{l}{\not\rightarrow} \qquad y \xrightarrow{l[z]} y'}{x^{*_r}y \xrightarrow{l[z]} y'} \qquad \text{(each for all } l \in L)$$

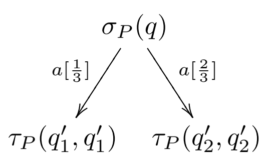
For p and q from (5.2) we get the following picture, where again $p^{*_r}q$ and $p'.(p^{*_r}q)$ may describe the same state.



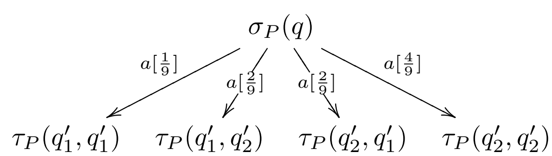
One aspect of the format is not illustrated by the examples above, namely the possibility that more than once a successor of the same argument and label is mentioned in the target of the rule. We give an artificial example to show that in such a situation it makes a difference whether the same or different successor variables are used. Consider the following alternative rules for a signature $\Sigma = (\Sigma_n)_{n\in\mathbb{N}}$ with $\sigma \in \Sigma_1$ and $\tau \in \Sigma_2$:

$$\frac{x \xrightarrow{l[u]} x'}{\sigma(x) \xrightarrow{l[u]} \tau(x', x')} \quad \text{(for all } l \in L) \qquad \text{or} \qquad \frac{x \xrightarrow{l[u]} x_1' \quad x \xrightarrow{l[v]} x_2'}{\sigma(x) \xrightarrow{l[u\cdot v]} \tau(x_1', x_2')} \quad \text{(for all } l \in L)$$

For q from (5.2) the two rules will generate the following a-transitions for $\sigma_P(p)$ respectively:



or



5.2 Properties

To state the properties of the models of a PGSOS specification we have to adapt the notion of a bisimulation up-to-context and a guarded recursive specification to the probabilistic setting.

Definition 5.4 *A* **(probabilistic) bisimulation up-to-context** *between two models $\langle P, (\sigma_P), \alpha_P\rangle$ and $\langle Q, (\sigma_Q), \alpha_Q\rangle$ of a PGSOS specification $\mathcal{R}$ is a relation $R \subseteq P \times Q$ such that for all $\langle p, q\rangle \in R$ and $a \in L$ there is a distribution $\mu \in \mathcal{D}_\omega \bar{R}$ where $\bar{R}$ is the congruence closure of R such that*

$$p \xrightarrow{a[r]} p' \quad \textit{just in case} \quad r = \sum\big(\mu(\langle p', y\rangle) \;\big|\; y \in Q \textit{ with } \langle p', y\rangle \in \bar{R}\big),$$

$$q \xrightarrow{a[r]} q' \quad \textit{just in case} \quad r = \sum\big(\mu(\langle x, q'\rangle) \;\big|\; x \in P \textit{ with } \langle x, q'\rangle \in \bar{R}\big).$$

Definition 5.5 *A* **(probabilistic) guarded recursive specification** *is a pair $\langle P, Tr\rangle$ consisting of a set of variables X and a set of transitions*

$$Tr \subseteq \left\{ x \xrightarrow{a[u]} t \;\middle|\; x \in X,\ a \in L,\ u \in (0,1],\ t \in \mathrm{T}X \right\}$$

such that for all $x \in X$ and $a \in L$ the set Tr contains finitely many transitions from x with label a only, the probabilities u of which sum up to 1 if there are any. A **solution** *of $\langle X, Tr\rangle$ in a model $\langle P, (\sigma_P), \alpha\rangle$ of a PGSOS specification $\mathcal{R}$ is given by an assignment of variables $h : X \to P$ such that for all $x \in X$, $a \in L$, and $q \in P$*

$$h(x) \xrightarrow{a[r]} q \quad \textit{just in case} \quad r = \sum\big(u \mid (x \xrightarrow{a[u]} t) \in Tr,\ [\![\, t[y := h(y)] \,]\!]_P = q\big).$$

Models of a PGSOS specification are well behaved in the following sense:

Proposition 5.6 *Let $\langle P, (\sigma_P), \alpha_P\rangle$ and $\langle Q, (\sigma_Q), \alpha_Q\rangle$ be models of a GSOS specification $\mathcal{R}$.*

1. *The congruence closure of a probabilistic bisimulation R between $\langle P, (\sigma_P), \alpha_P\rangle$ and $\langle Q, (\sigma_Q), \alpha_Q\rangle$ is a bisimulation again. In particular, the bisimilarity relation $\sim \subseteq P \times Q$ itself is a congruence.*
2. *Every probabilistic bisimulation up-to-context between $\langle P, (\sigma_P), \alpha_P\rangle$ and $\langle Q, (\sigma_Q), \alpha_Q\rangle$ is contained in some probabilistic bisimulation. This yields the following principle: to prove $p \sim q$ it suffices to find a probabilistic bisimulation up-to-context R with $\langle p, q\rangle \in R$.*
3. *Every probabilistic guarded recursive specification $\langle X, Tr\rangle$ has a solution in some model of $\mathcal{R}$. Furthermore, such a solution is determined up to bisimilarity.*

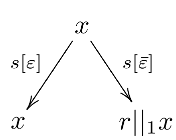
Our experiments indicate that in the probabilistic setting the bisimulation up-to-context technique is less useful than in the nondeterministic setting. The reason seems to be that the additional information about transition probabilities helps in distinguishing processes, so that less process equivalences hold. As an example, notice that with our definition of a PTS and probabilistic choice for any $u, v \in (0,1)$ there are no values $u', v' \in [0,1]$ such that we have

$$x \oplus_u (y \oplus_v z) \sim (x \oplus_{u'} y) \oplus_{v'} z$$

for all states x, y, z in any model of the specification. The bisimulation up-to-context proof principle is less successful here because its application usually requires such laws to hold in order to rewrite given process terms into a format that makes the common context visible.

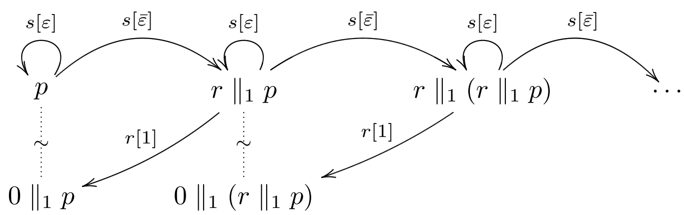
The definition principle using guarded recursive equations however is valuable, as the following simple example is supposed to demonstrate.

Example 5.7 *We can now alternatively specify the* **lossy bag** *from Example 3.3 as a state x in some probabilistic transition system with the following behaviour: it can perform a store action (s) which keeps it unchanged with probability ε or otherwise leads to a state behaving like x except that it can do one additional remove action (r) at an arbitrary moment in the future. Using the operators specified in Section 5.1 this can be expressed by the guarded recursive specification $\langle\{x\}, Tr\rangle$ where the set Tr contains the two transitions drawn below.*



Proposition 5.6 (3) says that this specification has solutions which are all bisimilar. Such a solution is given by a model $\langle P, (\sigma_P), \alpha\rangle$ of the operators from Section 5.1 and a state $p \in P$ (the state that x is

mapped on) which exhibits the behaviour shown below. The operators appearing in the picture denote the interpretations of the operator symbols in the model under consideration (the transitions from the states in the lower row are omitted).



The states p and $0 \parallel_1 p$ (as well as $r \parallel_1 p$ and $0 \parallel_1 (r \parallel_1 p)$ and so forth) are not necessarily identical, but they are bisimilar. From this we conclude that the state p in any such solution is bisimilar to the state p_0 from Example 3.3.

6. The abstract GSOS format

Up to now we just stated some of the properties of PGSOS without giving proofs. We now start a second, more technical part, which will explain that the format was derived in such a way that these results as well as those in Proposition 4.9 arise as instances of a more general framework.

We show that GSOS as well as PGSOS specifications are instances of an abstract account of operator specification formats introduced by Turi and Plotkin [TP97]. The approach is based on the fact that various kinds of transition systems – including LTS and PTS – can uniformly be described as *coalgebras for a functor* B, where the functor captures the type of system under consideration. On the same level of abstraction, the signatures considered earlier give rise to functors Σ such that interpretations of the operators in the signature correspond to *algebras for the functor* Σ. Turi and Plotkin observed that operator specifications in some of the congruence formats give rise to natural transformations ρ of a certain type involving the two functors above (and others derived from them), and that some of the well-behavedness results of the formats can nicely be proved on this abstract level. Here we will concentrate on their abstract modelling of GSOS rules, which they call *abstract GSOS*. By instantiating the framework with appropriate functors B, one obtains well-behaved formats for different types of transition systems. Those are of course still expressed as natural transformations of a certain shape and are thus not practically usable as such. One needs to characterise the natural transformations in concrete terms, like for instance by means of transition rules. We do so in Sections 7 and 8, where we prove that the resulting natural transformations indeed correspond to specifications in GSOS and PGSOS respectively. Through these results, the concrete formats inherit the well-behavedness properties of abstract GSOS. Figure 2 shows an outline of the approach, which is a refined version of Figure 1 from the introduction.

We start in this section by recalling basic coalgebraic notions to model state based systems. We explain that LTS and PTS are instances of this framework. For a deeper introduction into the theory of (co)algebras we refer the reader to the tutorial/overview articles of Jacobs and Rutten [JR96, Rut00]. Moreover, we give a brief introduction into the abstract specification format introduced by Turi and Plotkin [TP97]. In the following two sections we show that GSOS and PGSOS specifications form concrete representations of specifications in the abstract framework when instantiated for LTS and PTS respectively.

6.1 Transition systems as coalgebras

Dynamical systems such as transition systems, automata, or models of modal or epistemic logic can abstractly be described as coalgebras of a functor B, where B determines the type of behaviour under consideration.

Definition 6.1 *For a* Set*-functor* B *a* B**-coalgebra** *is a pair* $\langle P, \alpha \rangle$ *consisting of a set* P *and a function* $\alpha : P \to \mathrm{B}P$*. We will sometimes call* P *the* carrier *and* α *the* structure *or* operation *of the*

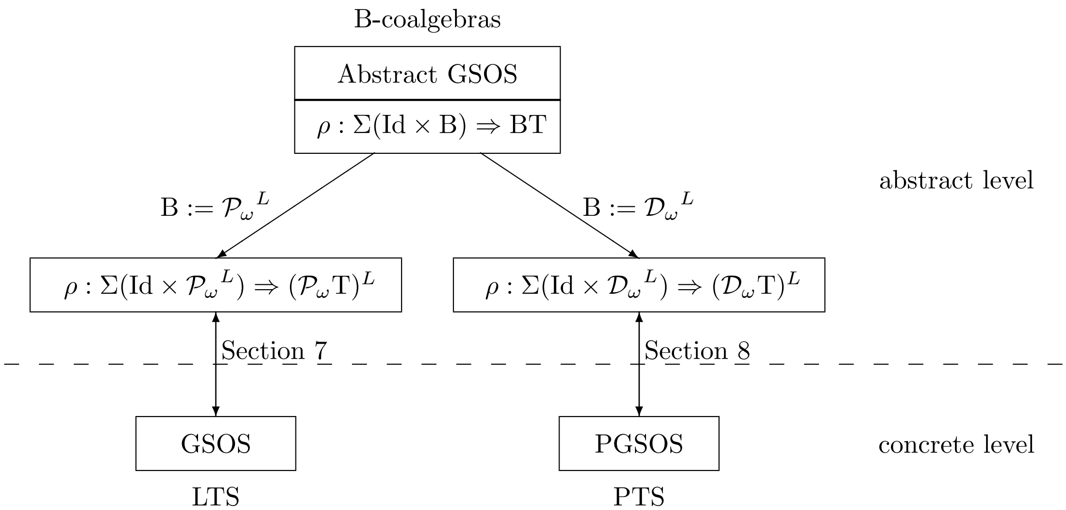


Figure 2: Outline of the approach.

coalgebra. A pair $\langle\langle P, \alpha\rangle, p\rangle$ consisting of a coalgebra $\langle P, \alpha\rangle$ and a designated state $p \in P$ is called a **process***. A* **homomorphism** *between two* B*-coalgebras $\langle P, \alpha_P\rangle$ and $\langle Q, \alpha_Q\rangle$ is a function $h : P \to Q$ satisfying*

$$\mathrm{B}h \circ \alpha_P = \alpha_Q \circ h.$$

All B*-coalgebras together with their homomorphisms form the category $\mathsf{Coalg}_\mathrm{B}$. A* **final B-coalgebra** *is a final object in $\mathsf{Coalg}_\mathrm{B}$, i.e. a* B*-coalgebra such that there exists precisely one homomorphism from any* B*-coalgebra to it.*

In order to model LTS and PTS as coalgebras, we turn the construction of powersets and probability distributions into functors.

Definition 6.2 *We define $\mathcal{P}_\omega$ to be the* **finite powerset functor***, i.e. the* Set*-functor defined for any set X and any function $f : X \to Y$ as*

$$\begin{aligned} \mathcal{P}_\omega X &:= \big\{X' \subseteq X \mid X' \textit{ is finite}\big\}, \\ (\mathcal{P}_\omega f)(X') &:= \big\{f(x) \mid x \in X'\big\}. \end{aligned}$$

Furthermore, we denote by $\mathcal{P}_\omega^+$ the **nonempty finite powerset functor***, i.e. the restriction of $\mathcal{P}_\omega$ to nonempty subsets.*

Definition 6.3 *Let the* **(possibly empty, simple) probability distribution functor** *$\mathcal{D}_\omega : \mathsf{Set} \to \mathsf{Set}$ be the functor defined for every set X, function $f : X \to Y$, and element $y \in Y$ as*

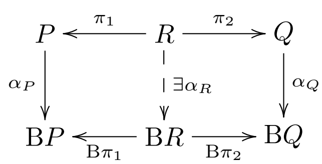
$$\begin{aligned} \mathcal{D}_\omega X &:= \big\{\mu : X \to \mathbb{R}_0^+ \mid \mathsf{supp}(\mu) \textit{ is finite}, \mu[X] \in \{0,1\}\big\}, \\ (\mathcal{D}_\omega f)(\mu) &:= y \mapsto \mu[f^{-1}(y)]. \end{aligned}$$

(Remember that for $X' \subseteq X$ and $\mu : X \to \mathbb{R}_0^+$ we defined $\mu[X'] := \sum_{x \in X'} \mu(x)$, in case the sum exists.) By $\mathcal{D}_\omega^+$ we denote the restriction of $\mathcal{D}_\omega$ to such $\mu \in \mathcal{D}_\omega P$ with $\mu[X] = 1$ (or, equivalently, $\mathsf{supp}(\mu) \neq \emptyset$). An element $\mu \in \mathcal{D}_\omega^+ X$ is called a **simple probability distribution over** *X.*

Writing the transition function $\alpha : P \times L \to \mathcal{P}_\omega P$ of an LTS $\langle P, \alpha\rangle$ equivalently as a function of the type $P \to (\mathcal{P}_\omega P)^L$, we see that LTS are coalgebras for the functor ${\mathcal{P}_\omega}^L$. In the same way we get that PTS are coalgebras for the functor ${\mathcal{D}_\omega}^L$.

The notions of a nondeterministic and probabilistic bisimulation from Def. 3.4 and Def. 3.5 can be generalized to arbitrary B-coalgebras.

Definition 6.4 (cf. [AM89]) *A* **bisimulation** *between two* B*-coalgebras* $\langle P, \alpha_P\rangle$ *and* $\langle Q, \alpha_Q\rangle$ *is a relation* $R \subseteq P \times Q$ *such that there exists a* B*-coalgebra operation* $\alpha_R : R \to \mathrm{B}R$ *making the projections* $\pi_1 : R \to P$ *and* $\pi_2 : R \to Q$ *homomorphisms from* $\langle R, \alpha_R\rangle$ *to* $\langle P, \alpha_P\rangle$ *and* $\langle Q, \alpha_Q\rangle$ *respectively.*



The greatest bisimulation between two coalgebras is denoted by $\sim$ *and is called* **bisimilarity**.

A greatest bisimulation always exists[1], and it can be seen to be the union of all bisimulations. The definition of bisimilarity induces the following proof principle: in order to show that two processes are bisimilar, it suffices to exhibit any bisimulation between the respective coalgebras which relates the two states.

It can easily be checked that the general notion of a bisimulation instantiates to nondeterministic and probabilistic bisimulation when we instantiate B with the respective functors $(\mathcal{P}_\omega)^L$ and $(\mathcal{D}_\omega)^L$ from above.

6.2 Composition operators as algebras

We now express signatures and the operators interpreting them categorically. Again we assume that there is a finitary (single-sorted) signature $\Sigma = (\Sigma_n)_{n\in\mathbb{N}}$, where $\sigma \in \Sigma_n$ is viewed as an operator symbol with arity n. To a set of states P we want to associate an interpretation $(\sigma_P : P^n \to P)_{n\in\mathbb{N},\sigma\in\Sigma_n}$ that contains a function with the corresponding arity for each operator symbol in the signature. We can combine all these functions into one function $\beta : \Sigma P \to P$ where we now view Σ as the following construction:

$$\Sigma X := \coprod_{n\in\mathbb{N}} \Sigma_n \times X^n = \big\{\sigma(x_1, \dots, x_n) \mid n \in \mathbb{N};\ \sigma \in \Sigma_n;\ x_1, \dots, x_n \in X\big\}.$$

For better readability the tuple $\langle\sigma, \langle x_1, \dots, x_n\rangle\rangle \in \Sigma X$ is again written like a function application. We write $\sigma^\beta : P^n \to P$ for the component of a combined function $\beta : \Sigma P \to P$ corresponding to $\sigma \in \Sigma_n$.

The construction of the sets ΣX extends to a functor $\Sigma : \mathsf{Set} \to \mathsf{Set}$ by setting for any function $f : X \to Y$

$$\Sigma f := \coprod_{n\in\mathbb{N}} \mathtt{id}_{\Sigma_n} \times f^n = \big[\sigma(x_1, \dots, x_n) \mapsto \sigma\big(f(x_1), \dots, f(x_n)\big)\big].$$

This makes the interpretation $(\sigma_P : P^n \to P)_{n\in\mathbb{N},\sigma\in\Sigma_n}$ an algebra of a functor.

Definition 6.5 *For a* Set*-functor* Σ *a* Σ**-algebra** *is a pair* $\langle P, \beta\rangle$ *consisting of set* P *and a function* $\beta : \Sigma P \to P$*. A* **homomorphism** *between two* Σ*-algebras* $\langle P, \beta_P\rangle$ *and* $\langle Q, \beta_Q\rangle$ *is a function* $h : P \to Q$ *satisfying*

$$h \circ \beta_P = \beta_Q \circ \Sigma h.$$

[1]This is true here since we restrict ourselves to working in the category Set.

All Σ-algebras together with their homomorphisms form the category Alg^{Σ}. An initial Σ-algebra is an initial object in Alg^{Σ}, i.e. a Σ-algebra such that there exists precisely one homomorphism from it to any Σ-algebra.

For functors Σ arising from a finitary signature as above we obtain an initial Σ-algebra as follows: the carrier set is given by the set of terms without variables, i.e. $\mathrm{T}\emptyset$, and the structure is usual building of terms.

In the following we will use the fact that the construction of terms from Def. 4.1 also extends to a functor.

Definition 6.6 *For a signature $\Sigma = (\Sigma_n)_{n\in\mathbb{N}}$ we define $\mathrm{T} : \mathsf{Set} \to \mathsf{Set}$ to be the* **term functor** *that maps a set X to the set $\mathrm{T}X$ of Σ-terms with variables in X, i.e. the smallest set such that*

$$X \subseteq \mathrm{T}X \quad \textit{and} \quad \Sigma\mathrm{T}X \subseteq \mathrm{T}X.$$

For $f : X \to Y$ the function $\mathrm{T}f : \mathrm{T}X \to \mathrm{T}Y$ replaces each variable $x \in X$ occurring in a term $t \in \mathrm{T}X$ by $f(x) \in Y$, i.e. for $x \in X$, $n \in \mathbb{N}$, $\sigma \in \Sigma_n$, and $t_i \in \mathrm{T}X$ $(1 \le i \le n)$ we set

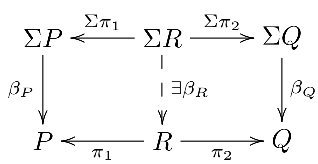
$$(\mathrm{T}f)(x) := f(x) \quad \textit{and} \quad (\mathrm{T}f)\big(\sigma(t_1,\dots,t_n)\big) := \sigma\big((\mathrm{T}f)(t_1),\dots,(\mathrm{T}f)(t_n)\big).$$

Moreover, for a Σ-algebra operation $\beta : \Sigma P \to P$ we define the **term evaluation** $[\![.]\!]_\beta : \mathrm{T}P \to P$ *by*

$$[\![x]\!]_\beta := x \quad \textit{and} \quad [\![\sigma(t_1,\dots,t_n)]\!]_\beta := \sigma\big([\![t_1]\!]_\beta,\dots,[\![t_n]\!]_\beta\big).$$

The definition of a congruence from Def. 4.6 can be lifted to the categorical setting as well.

Definition 6.7 *A* **congruence** *between two Σ-algebras $\langle P, \beta_P\rangle$ and $\langle Q, \beta_Q\rangle$ is a relation $R \subseteq P \times Q$ such that there exists a Σ-algebra operation $\beta_R : \Sigma R \to R$ making the projections $\pi_1 : R \to P$ and $\pi_2 : R \to Q$ algebra homomorphisms from $\langle R, \beta_R\rangle$ to $\langle P, \beta_P\rangle$ and $\langle Q, \beta_Q\rangle$ respectively.*



The congruence closure of a relation R between the carriers of two Σ-algebras again is the smallest congruence relation containing R.

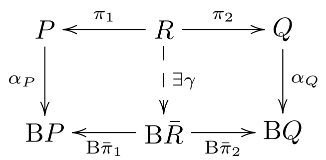
6.3 Bialgebras

Putting algebras and coalgebras together we can model a transition system with composition operators as a bialgebra.

Definition 6.8 *Given two Set-functors Σ and B, a* **$\langle\Sigma, \mathrm{B}\rangle$-bialgebra** *is a triple $\langle P, \beta, \alpha\rangle$ consisting of a set P and two functions $\beta : \Sigma P \to P$ and $\alpha : P \to \mathrm{B}P$, i.e. a Σ-algebra and a B-coalgebra structure on a common carrier. A* **homomorphism** *between two bialgebras $\langle P, \beta_P, \alpha_P\rangle$ and $\langle Q, \beta_Q, \alpha_Q\rangle$ is a function $h : P \to Q$ which is an algebra homomorphism from $\langle P, \beta_P\rangle$ to $\langle Q, \beta_Q\rangle$ as well as a coalgebra homomorphism from $\langle P, \alpha_P\rangle$ to $\langle Q, \alpha_Q\rangle$. All $\langle\Sigma, \mathrm{B}\rangle$-bialgebras together with their homomorphisms form the category $\mathsf{Bialg}^{\Sigma}_{\mathrm{B}}$. An* **initial (final) $\langle\Sigma, \mathrm{B}\rangle$-bialgebra** *is an initial (final) object in $\mathsf{Bialg}^{\Sigma}_{\mathrm{B}}$.*

We will sometimes talk about a bisimulation between two bialgebras, by which we mean a bisimulation between the included coalgebras. Similarly, a congruence between bialgebras is one for the contained algebra operations. Furthermore, we can generalize the notions of a nondeterministic and probabilistic bisimulation up-to-context from Def. 4.7 and Def. 5.4 to $\langle\Sigma, \mathrm{B}\rangle$-bialgebras.

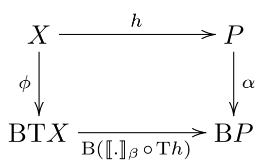
Definition 6.9 (cf. [San98]) *A relation $R \subseteq P \times Q$ is a* **bisimulation up-to-context** *between two $\langle \Sigma, \mathrm{B}\rangle$-bialgebras $\langle P, \beta_P, \alpha_P\rangle$ and $\langle Q, \beta_Q, \alpha_Q\rangle$ if there exists a mapping $\gamma : R \to \mathrm{B}\bar{R}$ making the diagram below commute, where $\bar{R}$ with projections $\bar{\pi}_1 : \bar{R} \to P$ and $\bar{\pi}_2 : \bar{R} \to Q$ is the congruence closure of R with respect to the Σ-algebras $\langle P, \beta_P\rangle$ and $\langle Q, \beta_Q\rangle$.*



In order to show that two states are bisimilar, it is often easier to find a suitable bisimulation up-to-context then an ordinary bisimulation. To use the former in a bisimilarity proof, we need a result saying that every bisimulation up-to-context between the bialgebras under consideration is contained in some standard bisimulation, as we have given it in Propositions 4.9 (2) and 5.6 (2) for the special case of models of a GSOS and PGSOS specification respectively. Later we will present a generalization of this result.

The definitions of a nondeterministic and probabilistic guarded recursive specification from Definitions 4.8 and 5.5 can be generalized as follows.

Definition 6.10 *We define a* **guarded recursive specification** *to be a pair $\langle X, \phi\rangle$ consisting of a set of variables X and a function $\phi : X \to \mathrm{BT}X$. A* **solution in a $\langle \Sigma, \mathrm{B}\rangle$-bialgebra** *$\langle P, \beta, \alpha\rangle$ is a mapping $h : X \to P$ of the variables to the carrier of the bialgebra such that the diagram below commutes.*



6.4 Operator specification in abstract GSOS

We now sketch a modelling of operator specifications as natural transformations proposed by Turi and Plotkin [TP97]. To motivate the idea in a simplified setup, we consider a parallel composition for LTS given by the following transition rules.

$$\frac{x \xrightarrow{a} x'}{x \parallel y \xrightarrow{a} x' \parallel y} \qquad \frac{y \xrightarrow{a} y'}{x \parallel y \xrightarrow{a} x \parallel y'} \qquad \text{(each for all } a \in L) \tag{6.1}$$

If this is the only operator under consideration, we talk about the signature $\Sigma = (\Sigma_n)_{n\in\mathbb{N}}$ with $\Sigma_2 = \{\parallel\}$ and $\Sigma_n = \emptyset$ for $n \neq 2$, so for the resulting functor we have $\Sigma \simeq (\mathrm{Id})^2$. Turning the rules into a set notation we get that a $\langle \Sigma, (\mathcal{P}_\omega)^{\mathrm{L}}\rangle$-bialgebra $\langle P, \beta, \alpha\rangle$ is a model for the specification if for all $p, q \in P$ and $a \in L$ we have

$$\alpha(p \parallel^\beta q)(a) = \big\{p' \parallel^\beta q \;\big|\; p' \in \alpha(p)(a)\big\} \cup \big\{p \parallel^\beta q' \;\big|\; q' \in \alpha(q)(a)\big\}.$$

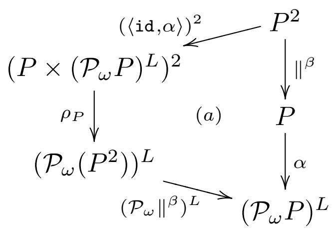
All these equations can be combined in the following single equation

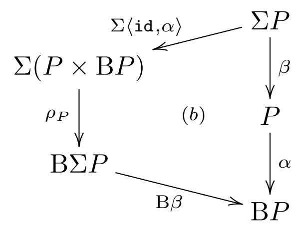
$$\alpha \circ \parallel^\beta = \big(\mathcal{P}_\omega \parallel^\beta\big)^L \circ \rho_P \circ \big(\langle \mathtt{id}, \alpha\rangle\big)^2,$$

where $\rho : \big(\mathrm{Id} \times (\mathcal{P}_\omega)^L\big)^2 \Rightarrow \big(\mathcal{P}_\omega(\mathrm{Id}^2)\big)^L$ is the natural transformation given for all sets X, elements $x, y \in X$, and functions $\phi, \psi \in (\mathcal{P}_\omega X)^L$ by

$$\rho_X\big(\langle x, \phi\rangle, \langle y, \psi\rangle\big) := \big[a \mapsto \big\{\langle x', y\rangle \mid x' \in \phi(a)\big\} \cup \big\{\langle x, y'\rangle \mid y' \in \psi(a)\big\}\big].$$

The above equation is pictured in diagram (a) below.





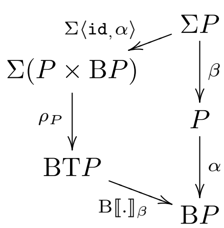
Generalizing this observation to arbitrary signatures Σ and behaviour functors B we would consider natural transformations $\rho : \Sigma(\mathrm{Id} \times B) \Rightarrow \mathrm{B}\Sigma$ as specifications. They characterise the class of all $\langle \Sigma, \mathrm{B} \rangle$-bialgebras $\langle P, \beta, \alpha \rangle$ making diagram (b) above commute.

We can increase the expressiveness of the approach by replacing the use of Σ in the codomain of the natural transformation ρ from above by T from Def. 6.6 (and one application of β by $[\![.]\!]_\beta$ in the corresponding diagram). This yields the following definition:

Definition 6.11 *Let* $\Sigma = (\Sigma_n)_{n \in \mathbb{N}}$ *be a signature and* B *a functor. A* **specification in abstract GSOS** *is a natural transformation*

$$\rho : \Sigma(\mathrm{Id} \times \mathrm{B}) \Rightarrow \mathrm{BT}.$$

A **model** *of a specification* ρ *in abstract GSOS is a* $\langle \Sigma, \mathrm{B} \rangle$*-bialgebra* $\langle P, \beta, \alpha \rangle$ *making the diagram below commute.*



The full subcategory of $\mathsf{Bialg}^{\Sigma}_{\mathrm{B}}$ *containing all models of* ρ *is denoted by* ρ-Bialg.

An element $\sigma(p_1, \ldots, p_n) \in \Sigma P$ is mapped to $\alpha\big(\sigma^\beta(p_1, \ldots, p_n)\big)$ by the path $\alpha \circ \beta$ in the above diagram. In the setting of LTS for instance – i.e. with $\mathrm{B} = (\mathcal{P}_\omega)^L$ – the latter is a description of the outgoing transitions of $\sigma^\beta(p_1, \ldots, p_n)$. For $\langle P, \beta, \alpha \rangle$ to be a model of a specification ρ in abstract GSOS, the composition $\mathrm{B}[\![.]\!]_\beta \circ \rho_P \circ \Sigma\langle \mathtt{id}, \alpha \rangle$ given by the left path should yield the same transitions. The function ρ_P in the middle receives as an input the operator symbol $\sigma \in \Sigma_n$ under consideration as well as the actual arguments $p_1, \ldots, p_n$ each together with the description $\alpha(p_i)$ of its outgoing transitions ($1 \leq i \leq n$). Based on this, ρ_P can declare the successors of $\sigma^\beta(p_1, \ldots, p_n)$ as terms in the given signature with elements of P in the variable positions, which are then iteratively evaluated by β. As a consequence of naturality, ρ_P can plug only those elements of P into the resulting terms that it received in its input, which were the arguments p_i and their immediate successors. Moreover, it can access them as black boxes only, that is, no inspection is possible (like for instance an equality check on different arguments).

Intuitively, this interpretation bears some similarity with the GSOS rules from Definition 4.2: Such a rule also declares an outgoing transition for some $\sigma(p_1, \ldots, p_n)$; its premises concern immediate successors of the arguments p_i; and the resulting transition leads to a state described as a term for the given signature in which the p_i and their successors may appear. We will prove in the next section that this correspondence indeed holds, which is the reason why the natural transformations ρ are called *specifications is abstract GSOS.*

Turi and Plotkin [TP97] actually consider this format as a special case of a more general framework, which is phrased in terms of distributive laws of monads over comonads. Lenisa et alii [LPW00] consecutively found that specifications in abstract GSOS are actually distributive laws of a monad over a copointed functor. In this setting it is possible to prove the results listed below, which can be found in the literature. We do not repeat the proofs here, because they require the introduction of quite some terminology which is not central to the main focus of this paper.

Proposition 6.12 *For a signature $\Sigma = (\Sigma_n)_{n\in\mathbb{N}}$ and a behaviour functor* B *let ρ be a specification in abstract GSOS.*

1. *If the functor* B *has a final coalgebra $\langle\Omega,\omega\rangle$, then there is a unique Σ-algebra structure $\beta_\rho : \Sigma\Omega \to \Omega$ such that $\langle\Omega,\beta_\rho,\omega\rangle$ is a model of ρ. Moreover, it is a final model, i.e. a final element in ρ-*Bialg*.*
2. *The dual statement is true for the initial Σ-algebra.*
3. *The congruence closure of any bisimulation between two models of ρ is a bisimulation again. As a consequence, the greatest bisimulation between two models is itself a congruence.*[2]
4. *Every bisimulation up-to-context between two models of ρ is contained in some ordinary bisimulation. This yields the following proof principle: to show that two states p and q in two models are bisimilar, it suffices to find a bisimulation up-to-context R with $\langle p,q\rangle \in R$.*
5. *Every guarded recursive specification $\langle X, Tr\rangle$ has a solution in some model of ρ. Moreover, such solutions are determined up to bisimilarity, i.e. if $h_P : X \to P$ and $h_Q : X \to Q$ are two solutions in the models $\langle P,\beta_P,\alpha_P\rangle$ and $\langle Q,\beta_Q,\alpha_Q\rangle$ respectively, then $h_P(x)$ and $h_Q(x)$ are bisimilar for all $x \in X$.*

(Variants of) the first three items are proved by Turi and Plotkin [TP97]. The last two items follow from previous work of ours [Bar03].

In the following two section we show that GSOS and PGSOS specifications correspond to specifications in abstract GSOS for the functors B appropriate for LTS and PTS respectively. With these results we obtain Propositions 4.9 and 5.6 as special cases of the above statement.

7. Deriving GSOS from the abstract framework

In this section we show that the GSOS specifications from Def. 4.4 are representations of the natural transformations that arise when we instantiate the abstract GSOS framework from Def. 6.11 with the functor $\mathrm{B} := (\mathcal{P}_\omega)^L$ modelling LTS. Remember that these are natural transformations of the type

$$\rho : \Sigma\big(\mathrm{Id} \times (\mathcal{P}_\omega)^{\mathrm{L}}\big) \Rightarrow (\mathcal{P}_\omega \mathrm{T})^L, \tag{7.1}$$

where Σ is the functor arising from the signature $\Sigma = (\Sigma_n)_{n\in\mathbb{N}}$ and T is the term functor from Def. 6.6.

We will proceed as follows: First, the natural transformations above are in a sequence of steps explained in terms of less and less complex ones. For this purpose we employ a number of simple lemmata about equivalences of natural transformations, which we state and prove in Appendix A. The main types of natural transformation encountered during the decomposition are listed in the left column of the table in Figure 3. Then we derive a representation result for the bottom most one, which brings us to the right column. Third, by going back up in the list we compose representations for the natural transformations on the higher levels of the table, reaching GSOS specifications in the end. The details of the outlined development will be explained next.

[2] Note that this statement holds without assuming that B weakly preserves pullbacks and has a final coalgebra. Turi and Plotkin make these assumptions in their corresponding result, because they base their proof on the construction of the greatest bisimulation as a pullback of the final homomorphisms. We found that it is sufficient to know that a greatest bisimulation exists, which is always the case in Set as we mentioned already.

	Natural transformation		Representation
(7.1)	$\rho : \Sigma(\mathrm{Id} \times (\mathcal{P}_\omega)^{\mathrm{L}}) \Rightarrow (\mathcal{P}_\omega \mathrm{T})^L$		GSOS specification (Def. 4.4)
	$\Downarrow$		$\Uparrow$
(7.4)	$\nu^{n,E} : (\mathrm{Id})^n \times (\mathcal{P}^+_\omega)^E \Rightarrow \mathcal{P}_\omega \mathrm{T}$	(7.22)	$\left\{ \frac{y_j \in X'_{\tau_j} \quad (1 \le j \le k)}{t \in \nu^{n,E}(\langle x_1, \ldots, x_n\rangle, (X'_e))} \right\}$ *finite*
	$\Downarrow$		$\Uparrow$
(7.8)	$\xi^m : (\mathcal{P}^+_\omega)^E \Rightarrow \mathcal{P}^+_\omega(\mathrm{Id}^m)$	(7.20)	$\left\{ \frac{y_j \in X'_{\tau_j} \quad (1 \le j \le k)}{\langle y_{o_1}, \ldots, y_{o_m}\rangle \in \xi^m((X'_e))} \right\}$ *finite, nonempty*
	$\Downarrow$		$\Uparrow$
(7.11)	$\zeta^{\vec{e}}_{(X_e)} : \prod_{e \in E} \mathcal{P}^+_\omega X_e \Rightarrow \mathcal{P}^+_\omega(X_{e_1} \times \cdots \times X_{e_m})$	Cor. $\Rightarrow$ 7.5	$M^{\vec{e}} \in \mathcal{P}^+_\omega(\mathtt{Par}[m]_{\preceq \vec{e}})$
	$\Updownarrow$		$\Updownarrow$
(7.13)	$\zeta : \mathcal{P}^+_\omega \Rightarrow \mathcal{P}^+_\omega(\mathrm{Id}^m)$	Thm. $\Rightarrow$ 7.6	$M \in \mathcal{P}^+_\omega(\mathtt{Par}[m])$

Figure 3: The outline of our approach ((X'_e) abbreviates $(X'_e)_{e \in E}$).

7.1 Top-down: decomposing the natural transformations under consideration

First of all, by Lemma A.1 and the adjunction $\mathrm{Id} \times \mathrm{L} \dashv (\mathrm{Id})^L$ natural transformations (7.1) are in one-to-one correspondence with those of the shape

$$\tilde{\rho} : \underbrace{\Sigma\big(\mathrm{Id} \times (\mathcal{P}_\omega)^{\mathrm{L}}\big) \times \mathrm{L}}_{=:\mathrm{F}} \Rightarrow \mathcal{P}_\omega \mathrm{T}. \tag{7.2}$$

With Lemma A.2 we can write the functor F as $\coprod_{z \in \mathrm{F}1} \mathrm{F}|_z$ so that $\tilde{\rho}$ above can by Lemma A.3 be described by a family of natural transformations

$$(\nu^z : \mathrm{F}|_z \Rightarrow \mathcal{P}_\omega \mathrm{T})_{z \in \mathrm{F}1}. \tag{7.3}$$

We shall now derive a workable description of the individual natural transformations ν^z for $z \in \mathrm{F}1$. With $\mathcal{P}_\omega 1 = \{\emptyset, 1\} \simeq 2$ we get that the functor F from the domain of our natural transformations in (7.2) maps the singleton set 1 to

$$\begin{aligned} \mathrm{F}1 &= \Sigma(1 \times (\mathcal{P}_\omega 1)^L) \times L \\ &\simeq \Sigma(2^L) \times L = \{\langle \sigma(E_1, \ldots, E_n), a\rangle \mid n \in \mathbb{N}, \sigma \in \Sigma_n, E_1, \ldots, E_n \subseteq L, a \in L\}. \end{aligned}$$

The isomorphism is given by $\langle \sigma(\langle *, \theta_1\rangle, \ldots, \langle *, \theta_n\rangle), a\rangle \in \mathrm{F}1 \mapsto \langle \sigma(E_1, \ldots, E_n), a\rangle \in \Sigma(2^L) \times L$ where $b \in E_i$ just in case $\theta_i(b) = 1 \in \{\emptyset, 1\} = \mathcal{P}_\omega 1$. For simplicity we will use elements from the latter set to describe those of F1 without making the isomorphism explicit. For every set X and $z = \langle \sigma(E_1, \ldots, E_n), a\rangle \in \mathrm{F}1$ we calculate

$$\begin{aligned} \mathrm{F}|_z X &:= (\mathrm{F}!_X)^{-1}(z) \\ &= \{\langle \sigma(\langle x_1, \theta_1\rangle, \ldots, \langle x_n, \theta_n\rangle), a\rangle \mid \\ &\qquad x_i \in X,\ \theta_i \in (\mathcal{P}_\omega X)^L \text{ s.t. } \forall b \in L : \theta_i(b) \neq \emptyset \iff b \in E_i,\ 1 \le i \le n\} \\ &\simeq \prod_{i=1}^n (X \times (\mathcal{P}^+_\omega X)^{E_i}). \\ &\simeq X^n \times (\mathcal{P}^+_\omega X)^E, \end{aligned}$$

where $E := E_1 + \cdots + E_n$. So we will study the natural transformations ν^z from (7.3) as examples of natural transformations of the following type for $n \in \mathbb{N}$ and a set E.

$$\nu^{n,E} : (\mathrm{Id})^n \times (\mathcal{P}^+_\omega)^E \Rightarrow \mathcal{P}_\omega \mathrm{T} \tag{7.4}$$

With Lemma A.5 these natural transformations are equivalent to those of the type

$$\tilde{\nu}^{n,E} : (\mathcal{P}^+_\omega)^E \Rightarrow \mathcal{P}_\omega \mathrm{T}(\mathrm{N} + \mathrm{Id}), \tag{7.5}$$

where we set $N := \{1, \ldots, n\}$.

To be able to apply Lemma A.4 for the next step we write the functor $\mathcal{P}_\omega \mathrm{T}(\mathrm{N} + \mathrm{Id})$ as a coproduct according to the following statement:

Lemma 7.1 *For functors* $\mathrm{G}^i : \mathsf{C} \to \mathsf{Set}$ *($i \in I$) we have*

$$\mathcal{P}_\omega(\coprod_{i \in I} \mathrm{G}^i) \simeq \coprod_{M \in \mathcal{P}_\omega I} (\prod_{i \in M} \mathcal{P}^+_\omega \mathrm{G}^i).$$

Proof: For all sets X we have an equivalence of sets

$$\mathcal{P}_\omega(\coprod_{i \in I} \mathrm{G}^i X) \simeq \coprod_{M \in \mathcal{P}_\omega I} (\prod_{i \in M} \mathcal{P}^+_\omega \mathrm{G}^i X)$$

given from left to right by $X' \mapsto \iota_M((X'_i)_{i \in M})$ where

$$M := \{i \in I \mid X' \cap \iota_i[\mathrm{G}^i X] \neq \emptyset\} \quad \text{and} \quad X'_i = \{\alpha \in \mathrm{G}^i X \mid \iota_i(\alpha) \in X'\}.$$

The equivalence easily extends to one between functors.

□

With Lemma A.2 we get

$$\mathrm{T}(\mathrm{N} + \mathrm{Id}) \simeq \coprod_{t \in \mathrm{T}(N+1)} \big(\mathrm{T}(\mathrm{N} + \mathrm{Id})\big)|_t \simeq \coprod_{t \in \mathrm{T}(N+1)} \mathrm{Id}^{|t|_*}. \tag{7.6}$$

For the second equivalence let us analyse what the functor $\big(\mathrm{T}(\mathrm{N} + \mathrm{Id})\big)|_t$ for $t \in \mathrm{T}(N+1)$ looks like. An element $t_X \in \big(\mathrm{T}(\mathrm{N} + \mathrm{Id})\big)|_t X = \big(\mathrm{T}(\mathtt{id}_N + !_X)\big)^{-1}(t)$ differs from t only in that the occurrences of $* \in 1$ in the variable positions are replaced by arbitrary elements from X. Since we use a finitary signature, the variable $*$ occurs in finitely many places in t only, and we will write $|t|_* \in \mathbb{N}$ for this number. Therefore t_X is determined by the elements $x_1, \ldots, x_{|t|_*} \in X$ that are put into these positions.

Applying Lemma 7.1 to the representation in (7.6) yields

$$\mathcal{P}_\omega \mathrm{T}(\mathrm{N} + \mathrm{Id}) \simeq \coprod_{M \in \mathcal{P}_\omega \mathrm{T}(N+1)} (\prod_{t \in M} \mathcal{P}^+_\omega(\mathrm{Id}^{|\mathrm{t}|_*})).$$

So Lemma A.4 and A.3 (b) say that a natural transformation $\tilde{\nu}^{n,E}$ from (7.5) is given by

$$M \in \mathcal{P}_\omega \mathrm{T}(N+1) \quad \text{and} \quad \big(\xi^t : (\mathcal{P}^+_\omega)^E \Rightarrow \mathcal{P}^+_\omega(\mathrm{Id}^{|\mathrm{t}|_*})\big)_{t \in M} \tag{7.7}$$

We will now continue to analyse the natural transformations ξ^t appearing in this representation, which are of the type

$$\xi^m : (\mathcal{P}^+_\omega)^E \Rightarrow \mathcal{P}^+_\omega(\mathrm{Id}^m) \tag{7.8}$$

for some $m \in \mathbb{N}$. The shape that we will transform these natural transformations into next looks more complicated at first sight, but it is nevertheless preferable because it makes the following information explicit: For a set X with nonempty, finite subsets $X'_e \subseteq X$ ($e \in E$) we may have $\vec{x} \in \xi_X^m\big((X'_e)\big)$ with $x_i \in X'_{e_1}$ as well as $x_i \in X'_{e_2}$ for some $1 \leq i \leq m$ and $e_1, e_2 \in E$ with $e_1 \neq e_2$. To understand the structure of the natural transformation, we would like to know from which of the two sets x_i was actually taken. To this end, we artificially separate the sets from which the X'_e are drawn, i.e. we put $X'_e \subseteq X_e$ for sets $(X_e)_{e\in E}$ and change the type of the elements in the resulting tuples from X to the disjoint union of all X_e, so that we can read off the origin of each element. This brings us to the world of functors from Set^E to Set.

More precisely, we apply Lemma A.6 to find that ξ^m from (7.8) is equivalent to a natural transformation

$$\tilde{\xi}^m_{(X_e)_{e\in E}} : \prod_{e\in E} \mathcal{P}^+_\omega X_e \Rightarrow \mathcal{P}^+_\omega\big((\coprod_{e\in E} X_e)^m\big) : \mathsf{Set}^E \to \mathsf{Set} \tag{7.9}$$

The functor describing the codomain of $\tilde{\xi}^m$ can be manipulated as follows

$$\mathcal{P}^+_\omega\big((\coprod_{e\in E} X_e)^m\big) \simeq \mathcal{P}^+_\omega\big(\coprod_{\vec{e}\in E^m} (X_{e_1} \times \cdots \times X_{e_m})\big) \simeq \coprod_{\tilde{M}\in\mathcal{P}^+_\omega(E^m)} \big(\prod_{\vec{e}\in\tilde{M}} \mathcal{P}^+_\omega(X_{e_1} \times \cdots \times X_{e_m})\big),$$

where the first equivalence uses distributivity and the second (a variant of) Lemma 7.1. With the last representation we can again apply Lemma A.4 and Lemma A.3 (b) to find that $\tilde{\xi}^m$ corresponds to

$$\tilde{M} \in \mathcal{P}^+_\omega(E^m) \quad \text{along with} \quad \big(\zeta^{\vec{e}}_{(X_e)_{e\in E}} : \prod_{e\in E} \mathcal{P}^+_\omega X_e \Rightarrow \mathcal{P}^+_\omega(X_{e_1} \times \cdots \times X_{e_m})\big)_{\vec{e}\in\tilde{M}}. \tag{7.10}$$

For the individual natural transformations $\zeta^{\vec{e}}$ we will develop a direct representation result next.

7.2 A representation theorem

Fix $m \in \mathbb{N}$, a set E, and $\vec{e} \in E^m$. In this section we will prove that any natural transformation

$$\zeta^{\vec{e}}_{(X_e)_{e\in E}} : \prod_{e\in E} \mathcal{P}^+_\omega X_e \Rightarrow \mathcal{P}^+_\omega(X_{e_1} \times \cdots \times X_{e_m}) \tag{7.11}$$

as occurring in (7.10) arises as the point-wise union of certain basic ones, which are constructed as in the following example:

Example 7.2 *With $m = 4$, $E = \{1, 2\}$, and $\vec{e} = \langle 1, 1, 1, 2\rangle$ we deal with natural transformations*

$$\beta_{\langle X_1, X_2\rangle} : \mathcal{P}^+_\omega X_1 \times \mathcal{P}^+_\omega X_2 \Rightarrow \mathcal{P}^+_\omega(X_1 \times X_1 \times X_1 \times X_2).$$

Definitions of the following type turn out to specify natural transformations: for all sets X_1 and X_2 and nonempty, finite subsets $X'_1 \subseteq X_1$ and $X'_2 \subseteq X_2$ set

$$\beta_{\langle X_1, X_2\rangle}(X'_1, X'_2) := \big\{\langle x, y, y, z\rangle \mid x, y \in X'_1;\ z \in X'_2\big\}.$$

Using a more intuitive notation, we could alternatively specify that $\beta_{\langle X_1, X_2\rangle}(X'_1, X'_2)$ is the smallest set satisfying the following derivation rule:

$$\frac{x \in X'_1 \quad y \in X'_1 \quad z \in X'_2}{\langle x, y, y, z\rangle \in \beta_{\langle X_1, X_2\rangle}(X'_1, X'_2)}$$

In the following we will generalize this definition of a basic natural transformation. It will describe the tuples $\vec{x} \in \beta_{\langle X_1, X_2\rangle}(X'_1, X'_2)$ as those with $x_i \in X'_{e_i}$ for all $1 \leq i \leq 4$ such that the elements in the second and third position are equal.

To define the set of natural transformations constructed in the above way formally, we introduce some notation allowing us to talk about vectors which have the same elements in certain positions.

Definition 7.3 • *By* $\mathtt{Par}[m]$ *we denote the set of all* **partitions of** $\{1,\ldots,m\}$*, i.e. all sets* Γ *of nonempty, disjoint subsets of* $\{1,\ldots,m\}$ *such that* $\bigcup\Gamma = \{1,\ldots,m\}$.

- *For* $\Gamma \in \mathtt{Par}[m]$ *and* $1 \leq i \leq m$ *we denote by* $[i]_\Gamma$ *the* **equivalence class of** i **in** Γ*, which is the unique* $c \in \Gamma$ *such that* $i \in c$.
- *We write* $\sim_\Gamma$ *for the equivalence relation on* $\{1,\ldots,m\}$ *induced by the partition* $\Gamma \in \mathtt{Par}[m]$*, i.e.* $i \sim_\Gamma j$ *just in case* $[i]_\Gamma = [j]_\Gamma$*. (Since partitions and equivalence relations are in one-to-one correspondence, we can define one in terms of the other, as we will do below).*
- *There is an order of partitions defined for* $\Gamma, \Gamma' \in \mathtt{Par}[m]$ *as* $\Gamma \preceq \Gamma'$ *if and only if* $\sim_\Gamma \subseteq \sim_{\Gamma'}$*, which means that for all* $1 \leq i,j \leq m$ *we have that* $i \sim_\Gamma j$ *implies* $i \sim_{\Gamma'} j$*. We write* $\Gamma \prec \Gamma'$ *if* $\Gamma \preceq \Gamma'$ *and* $\Gamma \neq \Gamma'$.
- *Given a vector* $\vec{x} \in X^m$ *we define the* **partition** $\mathtt{par}(\vec{x}) \in \mathtt{Par}[m]$ **induced by** $\vec{x}$ *to satisfy* $i \sim_{\mathtt{par}(\vec{x})} j$ *just in case* $x_i = x_j$.
- *For* $\Gamma \in \mathtt{Par}[m]$ *and* $c \in \Gamma$ *we write* $c\!\downarrow\, \in \{1,\ldots,m\}$ *for an arbitrary element in* c*. This notation will be used in cases only where no ambiguity arises. As an example, for* $\vec{x} \in X^m$*,* $\Gamma \in \mathtt{Par}[m]$ *with* $\Gamma \preceq \mathtt{par}(\vec{x})$*, and* $c \in \Gamma$ *we might write* $x_{c\downarrow}$*. This is unambiguous because for* $i,j \in \{1,\ldots,m\}$ *we have*

$$i,j \in c \;\Rightarrow\; i \sim_\Gamma j \;\Rightarrow\; i \sim_{\mathtt{par}(\vec{x})} j \;\Rightarrow\; x_i = x_j.$$

Generalizing the construction in Example 7.2, a partition $\Gamma \in \mathtt{Par}[m]$ determines a natural transformation, say $\beta^{\vec{e},\Gamma}$, of the type (7.11) as follows: $\beta^{\vec{e},\Gamma}_{(X_e)}\big((X'_e)\big)$ contains all tuples $\vec{x}$ such that each component x_i is in the corresponding subset X'_{e_i} and moreover $\vec{x}$ carries identical elements in positions related by Γ, i.e. $\Gamma \preceq \mathtt{par}(\vec{x})$. We have to be careful with the typing though: we may not prescribe that two positions i and j should hold the same element if they have different types, i.e. if $e_i \neq e_j$. So whenever $i \sim_\Gamma j$ we require $e_i = e_j$, which is $\Gamma \preceq \mathtt{par}(\vec{e})$ (otherwise, the resulting transformations would neither be natural nor would $\beta^{\vec{e},\Gamma}_{(X_e)}\big((X'_e)\big) \neq \emptyset$ be guaranteed). This idea leads to the following formal definition:

Definition 7.4 *Define*

$$\mathtt{Par}[m]_{\preceq\vec{e}} := \{\Gamma \in \mathtt{Par}[m] \mid \Gamma \preceq \mathtt{par}(\vec{e})\}.$$

For $\Gamma \in \mathtt{Par}[m]_{\preceq\vec{e}}$ *define the* **basic natural transformation**

$$\beta^{\vec{e},\Gamma}_{(X_e)_{e\in E}} : \prod_{e\in E} \mathcal{P}^+_\omega X_e \Rightarrow \mathcal{P}^+_\omega(\mathrm{X}_{\mathrm{e}_1} \times \cdots \times \mathrm{X}_{\mathrm{e}_\mathrm{m}})$$

for sets X_e*, subsets* $X'_e \in \mathcal{P}^+_\omega X_e$ $(e \in E)$*, and* $\vec{x} \in X_{e_1} \times \cdots \times X_{e_m}$ *as*

$$\vec{x} \in \beta^{\vec{e},\Gamma}_{(X_e)}\big((X'_e)\big) \iff \Gamma \preceq \mathtt{par}(\vec{x}) \;\wedge\; \forall i \in \{1,\ldots,m\} :\; x_i \in X'_{e_i}.$$

The tuples $\vec{x}$ in $\beta^{\vec{e},\Gamma}_{(X_e)}\big((X'_e)\big)$ are generated as follows: for each $c \in \Gamma$ an element y_c is chosen from $X'_{e_{c\downarrow}}$ and put in all positions $i \in c$ of $\vec{x}$. This can be expressed by the following schematic rule:

$$\frac{y_c \in X'_{e_{c\downarrow}}\,(c \in \Gamma)}{\langle y_{[1]_\Gamma},\ldots,y_{[m]_\Gamma}\rangle \in \beta^{\vec{e},\Gamma}_{(X_e)}\big((X'_e)\big)} \tag{7.12}$$

Note that with $\Gamma = \big\{\{1\},\{2,3\},\{4\}\big\}$ this schema instantiates to a rule equivalent to the one in Example 7.2.

Our main representation result for the nondeterministic setting below states that all natural transformations $\zeta^{\vec{e}}$ as in (7.11) arise as (point-wise) unions of the basic transformations $\beta^{\vec{e},\Gamma}$.

Corollary 7.5 *Every natural transformation $\zeta^{\vec{e}}$ as in* (7.11) *can be written as*

$$\zeta^{\vec{e}} = \bigcup_{\Gamma\in M^{\vec{e}}} \beta^{\vec{e},\Gamma} \quad \textit{for some} \quad M^{\vec{e}} \in \mathcal{P}^+_\omega(\mathtt{Par}[m]_{\preceq\vec{e}}).$$

To simplify the presentation, we will prove Corollary 7.5 in the special case $E \simeq 1$ only. In this case there is a unique $\vec{e} = \langle *,\ldots,*\rangle \in E^m$ (which yields $\mathtt{Par}[m]_{\preceq\vec{e}} = \mathtt{Par}[m]$) and we will drop the corresponding superscripts, e.g. in $\beta^{\vec{e},\Gamma}$ from Def. 7.4. So we prove the following theorem.

Theorem 7.6 *Every natural transformation*

$$\zeta : \mathcal{P}^+_\omega \Rightarrow \mathcal{P}^+_\omega(\mathrm{Id}^m) \tag{7.13}$$

can be written as

$$\zeta = \bigcup_{\Gamma\in M} \beta^{\Gamma} \quad \textit{for some} \quad M \in \mathcal{P}^+_\omega(\mathtt{Par}[m]),$$

where for $\Gamma \in \mathtt{Par}[m]$ the natural transformation $\beta^\Gamma : \mathcal{P}^+_\omega \Rightarrow \mathcal{P}^+_\omega(\mathrm{Id}^m)$ is given by (cf. Def. 7.4)

$$\vec{x} \in \beta^\Gamma_X(X') \iff \Gamma \preceq \mathtt{par}(\vec{x}) \ \wedge\ \forall i \in \{1,\ldots,m\} : \ x_i \in X'. \tag{7.14}$$

We do not claim that Corollary 7.5 follows as such from this statement. It rather results from a straightforward extension of the proof we are about to develop. This extension essentially introduces some bureaucracy to keep track of the typing. Since this complicates the presentation without adding considerable insight, we decided to restrict ourselves to showing the treatment of the special case.

Before we approach the proof of Theorem 7.6 we remark that the mentioned representation is not unique, due to the following fact about the natural transformations β^Γ, which immediately follows from their definition.

Lemma 7.7 *For $\Gamma, \Gamma' \in \mathtt{Par}[m]$ with $\Gamma \preceq \Gamma'$ we have $\beta^{\Gamma'} \subseteq \beta^\Gamma$, where the subset relation is to be read point-wise, i.e. $\beta^{\Gamma'}_X(X') \subseteq \beta^\Gamma_X(X')$ for all sets X and $X' \in \mathcal{P}^+_\omega X$.*

Let M be the representation from Theorem 7.6. With the above lemma, for $\Gamma, \Gamma' \in M$ with $\Gamma \prec \Gamma'$ the union on the right hand side of the equation in Theorem 7.6 would stay the same if we removed Γ' from M. We will therefore call Γ' *redundant* in this setting. This means that the union is solely determined by the minimal elements of M. On the other hand, it is easy to verify that the resulting natural transformations differ for two sets with different minimal elements. So the representation is unique up to the inclusion or omission of redundant partitions. This remark holds for the more general case of Corollary 7.5 as well.

For the proof of Theorem 7.6 we need the following lemma.

Lemma 7.8 *Let ζ be a natural transformation as in* (7.13). *For a set X and $X' \in \mathcal{P}^+_\omega X$ we have that $\vec{x} \in \zeta_X(X')$ implies $x_i \in X'$ for all $1 \le i \le m$.*

Proof: Let $\mathtt{in} : X' \hookrightarrow X$ be the subset inclusion and consider the following naturality square:

$$\begin{array}{ccc} \mathcal{P}^+_\omega X' & \xrightarrow{\ \zeta_{X'}\ } & \mathcal{P}^+_\omega(X'^m) \\ {\scriptstyle \mathcal{P}^+_\omega \mathtt{in}}\downarrow & & \downarrow{\scriptstyle \mathcal{P}^+_\omega(\mathtt{in}^m)} \\ \mathcal{P}^+_\omega X & \xrightarrow[\ \zeta_X\ ]{} & \mathcal{P}^+_\omega(X^m) \end{array} \qquad \begin{array}{ccccc} X' & \overset{\zeta_{X'}}{\longmapsto} & \zeta_{X'}(X') & \ni & \vec{x}' \\ {\scriptstyle \mathcal{P}^+_\omega \mathtt{in}}\downarrow & & \downarrow{\scriptstyle \mathcal{P}^+_\omega(\mathtt{in}^m)} & & \downarrow{\scriptstyle \mathtt{in}^m} \\ X' & \underset{\zeta_X}{\longmapsto} & \zeta_X(X') & \ni & \vec{x} \end{array}$$

We can read off that for every $\vec{x} \in \zeta_X(X')$ there has to be $\vec{x}' \in \zeta_{X'}(X')$ with $\vec{x} = \mathtt{in}^m(\vec{x}')$. We get $x_i = \mathtt{in}(x_i') = x_i' \in X'$ for all i as wanted.

□

Proof: [Theorem 7.6] We claim that the statement holds for $M := \{\Gamma \in \mathtt{Par}[m] \mid \beta^\Gamma \subseteq \zeta\}$, which is to say that

$$\zeta = \bigcup\{\beta^\Gamma \mid \Gamma \in \mathtt{Par}[m], \beta^\Gamma \subseteq \zeta\}$$

Since the other inclusion is immediate, we need to show $\zeta \subseteq \bigcup\{\beta^\Gamma \mid \Gamma \in \mathtt{Par}[m], \beta^\Gamma \subseteq \zeta\}$ only, which is to say that for any set X, subset $X' \in \mathcal{P}^+_\omega X$, and $\vec{x} \in \zeta_X(X')$ we have to find $\Gamma \in \mathtt{Par}[m]$ such that $\beta^\Gamma \subseteq \zeta$ and $\vec{x} \in \beta^\Gamma_X(X')$. We show that we can take $\Gamma = \mathtt{par}(\vec{x})$. From $\vec{x} \in \zeta_X(X')$ it follows with Lemma 7.8 that $x_i \in X'$ for all i. With $\mathtt{par}(\vec{x}) \preceq \mathtt{par}(\vec{x})$ this yields $\vec{x} \in \beta^{\mathtt{par}(\vec{x})}_X(X')$ as needed (cf. (7.14)). It remains to be shown that $\beta^{\mathtt{par}(\vec{x})} \subseteq \zeta$.

Below we will treat the case that X, X', and $\vec{x}$ are such that $\mathtt{par}(\vec{x})$ is minimal with respect to the order $\prec$. By this we mean that there are no Y, $Y' \in \mathcal{P}^+_\omega Y$, and $\vec{y} \in \zeta_Y(Y')$ with $\mathtt{par}(\vec{y}) \prec \mathtt{par}(\vec{x})$. Otherwise, we choose Y, Y', and $\vec{y}$ as above such that $\mathtt{par}(\vec{y})$ is minimal and carry out the argument below for them instead to obtain $\beta^{\mathtt{par}(\vec{y})} \subseteq \zeta$. With Lemma 7.7 we have $\beta^{\mathtt{par}(\vec{x})} \subseteq \beta^{\mathtt{par}(\vec{y})}$ and thus $\beta^{\mathtt{par}(\vec{x})} \subseteq \zeta$ as needed.

To prove $\beta^{\mathtt{par}(\vec{x})} \subseteq \zeta$ under the minimality assumption, we show that for all sets Y and $Y' \in \mathcal{P}^+_\omega Y$ we have $\beta^{\mathtt{par}(\vec{x})}_Y(Y') \subseteq \zeta_Y(Y')$. Take any $\vec{y} \in \beta^{\mathtt{par}(\vec{x})}_Y(Y')$, i.e. $\vec{y} \in Y^m$ with $y_i \in Y'$ for all i and $\mathtt{par}(\vec{x}) \preceq \mathtt{par}(\vec{y})$. We derive $\vec{y} \in \zeta_Y(Y')$ as follows:

For $Z := X' \times Y'$ we find

$$\begin{aligned}
\vec{x} \in \zeta_X(X') \quad &\iff \quad \vec{x} \in \underbrace{\zeta_X\big((\mathcal{P}^+_\omega \pi_1)(Z)\big)}_{\overset{\text{nat. }\zeta}{=} (\mathcal{P}^+_\omega(\pi_1^m))(\zeta_Z(Z))} \\
&\iff \quad \exists \vec{z} \in \zeta_Z(Z) : \vec{x} = \pi_1^m(\vec{z}) \\
&\iff \quad \exists \vec{w} \in (Y')^m : \overrightarrow{\langle x, w\rangle} \in \zeta_Z(Z) \\
&\overset{(*)}{\iff} \quad \overrightarrow{\langle x, y\rangle} \in \zeta_Z(Z) \\
&\implies \quad \underbrace{\pi_2^m(\overrightarrow{\langle x, y\rangle})}_{=\vec{y}} \in \underbrace{\big(\mathcal{P}^+_\omega(\pi_2^m)\big)\big(\zeta_Z(Z)\big)}_{\overset{\text{nat. }\zeta}{=} \zeta_Y((\mathcal{P}^+_\omega \pi_2)(Z)) = \zeta_Y(Y')} \\
&\iff \quad \vec{y} \in \zeta_Y(Y'),
\end{aligned}$$

where $\overrightarrow{\langle x, w\rangle} := \big\langle \langle x_1, w_1\rangle, \ldots, \langle x_m, w_m\rangle \big\rangle$.

The implication "$\Longrightarrow$" in step $(*)$ remains to be explained: We easily find $\mathtt{par}(\overrightarrow{\langle x, w\rangle}) \preceq \mathtt{par}(\vec{x})$, but with $\overrightarrow{\langle x, w\rangle} \in \zeta_Z(Z)$ the above minimality assumption on $\vec{x}$ rules out that $\mathtt{par}(\overrightarrow{\langle x, w\rangle})$ is strictly smaller than $\mathtt{par}(\vec{x})$. So we find $\mathtt{par}(\overrightarrow{\langle x, w\rangle}) = \mathtt{par}(\vec{x})$, which implies $\mathtt{par}(\vec{x}) \preceq \mathtt{par}(\vec{w})$. Together with the assumption $\mathtt{par}(\vec{x}) \preceq \mathtt{par}(\vec{y})$ this means that $x_i = x_j$ implies $w_i = w_j$ as well as $y_i = y_j$. With this observation the function $f : Z \to Z$ which exchanges $\langle x_i, w_i\rangle$ and $\langle x_i, y_i\rangle$ for all $i \in \{1, \ldots, m\}$ is well defined (in the sense that whenever multiple cases in the definition apply, then they all determine the same result) by

$$f(x, y) := \begin{cases} \langle x_i, y_i\rangle & \text{if } \langle x, y\rangle = \langle x_i, w_i\rangle \text{ for some } 1 \leq i \leq m, \\ \langle x_i, w_i\rangle & \text{if } \langle x, y\rangle = \langle x_i, y_i\rangle \text{ for some } 1 \leq i \leq m, \\ \langle x, y\rangle & \text{otherwise.} \end{cases}$$

The function f is self inverse and thus bijective, so that we find $(\mathcal{P}^+_\omega f)(Z) = Z$. Knowing this we reason as follows:

$$\overrightarrow{\langle x,w\rangle} \in \zeta_Z(Z) \implies \underbrace{f^m(\overrightarrow{\langle x,w\rangle})}_{=\overrightarrow{\langle x,y\rangle}} \in \underbrace{\big(\mathcal{P}^+_\omega(f^m)\big)\big(\zeta_Z(Z)\big)}_{\overset{\text{nat. }\zeta}{=}\zeta_Z((\mathcal{P}^+_\omega f)(Z))=\zeta_Z(Z)} \iff \overrightarrow{\langle x,y\rangle} \in \zeta_Z(Z).$$

This concludes the proof of Theorem 7.6.

□

7.3 Bottom-up: constructing the rule format

At this point we have completely characterised natural transformations of the type (7.1): Starting with them, natural transformations of a complex type were successively described by (families of) natural transformations of a simpler type, until the format (7.11) was reached, which could be understood in elementary terms. In the overview of Figure 3 we have reached the bottom of the right column. We will now collect the bits and pieces to construct direct representations of the more complicated natural transformations, i.e. we will walk up the table again, this time on the right hand side. At some point it will be convenient to introduce rule notations to express the resulting representations, and in the end we will rediscover GSOS specifications from Def. 4.4.

Plugging the representation of the natural transformations $\zeta^{\vec{e}}$ from Corollary 7.5 into (7.10), we find that a natural transformation $\tilde{\xi}^m$ as in (7.9) can be characterised by a set

$$\tilde{M}^m \in \mathcal{P}^+_\omega(E^m) \quad \text{and sets} \quad \big(M^{\vec{e}} \in \mathcal{P}^+_\omega(\mathtt{Par}[m]_{\preceq\vec{e}})\big)_{\vec{e}\in\tilde{M}^m}$$

We write this more compactly but equivalently as one set

$$M^m = \{\langle\vec{e},\Gamma\rangle \mid \vec{e}\in\tilde{M}^m,\ \Gamma\in M^{\vec{e}}\} \in \mathcal{P}^+_\omega\big\{\langle\vec{e},\Gamma\rangle \mid \vec{e}\in E^m,\ \Gamma\in\mathtt{Par}[m]_{\preceq\vec{e}}\big\}. \tag{7.18}$$

Any such M^m represents the natural transformation

$$\tilde{\xi}^m_{(X_e)_{e\in E}} = \bigcup_{\langle\vec{e},\Gamma\rangle\in M^m} \mathcal{P}^+_\omega(\iota_{e_1}\times\cdots\times\iota_{e_m})\circ\beta^{\vec{e},\Gamma}_{(X_e)_{e\in E}} : \quad \prod_{e\in E}\mathcal{P}^+_\omega \mathrm{X_e} \Rightarrow \mathcal{P}^+_\omega\big((\coprod_{e\in E} X_e)^m\big).$$

Through the correspondence given by Lemma A.6, the same sets M^m characterise the natural transformations ξ^m from (7.8) as

$$\begin{aligned}
\xi^m_X &= \mathcal{P}^+_\omega([\mathtt{id}_X]_{e\in E})^m \circ \tilde{\xi}^m_{(X)_{e\in E}} \\
&= \mathcal{P}^+_\omega([\mathtt{id}_X]_{e\in E})^m \circ \bigcup_{\langle\vec{e},\Gamma\rangle\in M^m} \mathcal{P}^+_\omega(\iota_{e_1}\times\cdots\times\iota_{e_m})\circ\beta^{\vec{e},\Gamma}_{(X)_{e\in E}} \\
&= \bigcup_{\langle\vec{e},\Gamma\rangle\in M^m} \mathcal{P}^+_\omega\big(\underbrace{([\mathtt{id}_X]_{e\in E})^m\circ(\iota_{e_1}\times\cdots\times\iota_{e_m})}_{=\mathtt{id}_{X^m}}\big)\circ\beta^{\vec{e},\Gamma}_{(X)_{e\in E}} \\
&= \bigcup_{\langle\vec{e},\Gamma\rangle\in M^m} \beta^{\vec{e},\Gamma}_{(X)_{e\in E}} : \quad (\mathcal{P}^+_\omega X)^E \Rightarrow \mathcal{P}^+_\omega(X^m).
\end{aligned}$$

After Def. 7.4 we remarked that a basic natural transformations $\beta^{\vec{e},\Gamma}$ can be described by a derivation rule of a certain shape. We will now write this rule using a (finite) set of variables $Y = \{y_1,\ldots,y_k\}$, $o_i \in \{1,\ldots,k\}$, and $\tau_i \in E$ $(1 \le i \le m)$ as

$$\frac{y_j \in X'_{\tau_j} \quad (1 \le j \le k)}{\langle y_{o_1},\ldots,y_{o_m}\rangle \in \beta^{\vec{e},\Gamma}\big((X'_e)\big)} \tag{7.19}$$

It describes $\beta^{\vec{e},\Gamma}$ with $\Gamma = \mathtt{par}(\vec{o})$ and $\vec{e} = \langle\tau_{o_1},\ldots,\tau_{o_m}\rangle$.

Of course the step to this rule representation introduces redundancy. In order to get a unique representation of Γ and $\vec{e}$ – at least up to renaming of variables – we assume that every y_j appears in the conclusion of the rule, i.e. $\{o_1, \dots, o_m\} = \{1, \dots, k\}$.

We will denote the nonempty, finite sets M^m from (7.18) by sets of rules as in (7.19):

$$\xi^m \doteq M^m \doteq \left\{ \frac{y_j \in X'_{\tau_j} \quad (1 \le j \le k)}{\langle y_{o_1}, \dots, y_{o_m} \rangle \in \xi^m\big((X'_e)\big)} \right\}_{\mathit{finite, nonempty}} \tag{7.20}$$

This representation is unique up to the inclusion or omission of redundant rules and the renaming of variables.

Such a set of rules describes the natural transformation ξ^m for which $\vec{x} \in \xi^m_X\big((X'_e)\big)$ just in case this is implied by at least one of the rules in the set.

Each natural transformation ξ^t for $t \in \mathrm{T}(N+1)$ appearing in (7.7) can now be represented by a set of rules as in (7.20) for $m = |t|_*$, where $|t|_*$ again denotes the number of occurrences of $*$ in t. We can easily include the term t into the rule notation: we replace the vector $\langle y_{o_1}, \dots, y_{o_m} \rangle$ by $t_Y \in \mathrm{T}(N+Y)$, where t_Y is the term that arises after replacing the i-th occurrence of $*$ in t by y_{o_i}. To get a representation for $\tilde{\nu}^{n,E}$ from (7.5) we can now just collect all rules for the ξ^t $(t \in M)$ from (7.7), since this encoding of t makes them all distinct. This yields a no longer necessarily nonempty (since M could be empty) but still finite set of rules as below. The condition on the variables for each rule is now that y_j occurs at least once in t_Y for every $1 \le j \le k$:

$$\tilde{\nu}^{n,E} \doteq \left\{ \frac{y_j \in X'_{\tau_j} \quad (1 \le j \le k)}{t_Y \in \tilde{\nu}^{n,E}\big((X'_e)\big)} \right\}_{\mathit{finite}} \tag{7.21}$$

For the step from $\tilde{\nu}^{n,E}$ in (7.5) to $\nu^{n,E}$ in (7.4) the elements from $N = \{1, \dots, n\}$ appearing in each term t_Y are treated as variables, which are to be instantiated with the corresponding arguments when the rule is applied. To reflect this step in the notation, we pick a set $X = \{x_1, \dots, x_n\}$ of n variable names, distinct from those in Y, and replace $i \in \{1, \dots, n\}$ appearing in t_Y by x_i. This yields the following format, where $t_{X,Y} \in \mathrm{T}(X \cup Y)$.

$$\nu^{n,E} \doteq \left\{ \frac{y_j \in X'_{\tau_j} \quad (1 \le j \le k)}{t_{X,Y} \in \nu^{n,E}\big(\langle x_1, \dots, x_n \rangle, (X'_e)\big)} \right\}_{\mathit{finite}} \tag{7.22}$$

We studied the above natural transformations $\nu^{n,E}$ as generalisations of the natural transformations ν^z from (7.3). To describe the family $(\nu^z)_{z \in \mathrm{F}1}$ mentioned there, we will again collect all rules for the individual ν^z. We need to incorporate the information about $z = \langle \sigma(E_1, \dots, E_n), a \rangle \in \mathrm{F}1$ into the rule notation. The rule needs to fire whenever $\tilde{\rho}$ is applied to some $\sigma(\langle p_1, \theta_1 \rangle, \dots, \langle p_n, \theta_n \rangle)$ and the label a such that $\theta_i(b) \neq \emptyset$ just in case $b \in E_i$. To ensure the latter condition we add extra premises. Furthermore X'_{τ_j} is replaced by $\theta_{i_j}(l_j)$ where $\tau_j = \iota_{i_j}(l_j) \in E = E_1 + \dots + E_n$.

$$\tilde{\rho} \doteq \left\{ \frac{\begin{array}{ll} \theta_i(b) \neq \emptyset & b \in E_i,\ 1 \le i \le n \\ \theta_i(b) = \emptyset & b \notin E_i,\ 1 \le i \le n \\ y_j \in \theta_{i_j}(l_j) & 1 \le j \le m \end{array}}{t_{X,Y} \in \tilde{\rho}\big(\sigma(\langle x_1, \theta_1 \rangle, \dots, \langle x_n, \theta_n \rangle), a\big)} \right\}_{\mathit{image\ finite}} \tag{7.23}$$

These sets are *image finite* in the sense that they contain only finitely many rules for each collection $\sigma \in \Sigma_n$, $a \in L$, and $E_1, \dots, E_n \subseteq L$. The same set of rules describes ρ from (7.1), except that we would replace $\tilde{\rho}\big(\sigma(\langle x_1, \theta_1 \rangle, \dots, \langle x_n, \theta_n \rangle), a\big)$ by $\rho\big(\sigma(\langle x_1, \theta_1 \rangle, \dots, \langle x_n, \theta_n \rangle)\big)(a)$.

This representation corresponds to that of a GSOS specifications from Def. 4.4. To see this, we need to modify the formulation in two aspects only:

First, we incorporate into the notation the fact that the pairs $\langle x_i, \theta_i \rangle$ would be instantiated by $\langle p_i, \alpha(p_i) \rangle$ for some $(\mathcal{P}_\omega)^L$-coalgebra (i.e. LTS) $\langle P, \alpha \rangle$ with $p_i \in P$ and that

$$\rho_P\big(\sigma(\langle p_1, \alpha(p_i) \rangle, \ldots, \langle p_n, \alpha(p_n) \rangle)\big)$$

is supposed to describe the outgoing transitions of the state represented by $\sigma(p_1, \ldots, p_n)$ (cf. the definition of a model of ρ in Def. 6.11). So we replace

- a premise $\theta_i(b) \neq \emptyset$ by $x_i \xrightarrow{b}$,
- a premise $\theta_i(b) = \emptyset$ by $x_i \stackrel{b}{\not\rightarrow}$,
- a premise $y_j \in \theta_{i_j}(l_j)$ by $x_{i_j} \xrightarrow{l_j} y_j$,
- the conclusion by $\sigma(x_1, \ldots, x_n) \xrightarrow{a} t_{X,Y}$.

This rewrites the individual rules above into the following shape:

$$\frac{\begin{array}{ll} x_i \xrightarrow{b} & b \in E_i, 1 \leq i \leq n \\ x_i \stackrel{b}{\not\rightarrow} & b \notin E_i, 1 \leq i \leq n \\ x_{i_j} \xrightarrow{l_j} y_j & 1 \leq j \leq k \end{array}}{\sigma(x_1, \ldots, x_n) \xrightarrow{a} t_{X,Y}}$$

Second, a GSOS rule (cf. Def. 4.2) mentions the sets R_i and P_i (with $R_i \cap P_i = \emptyset$) of *requested* and *prohibited* labels instead of the sets E_i of *enabled* labels (for $1 \leq i \leq n$). This is just "syntactic sugar" allowing us to abbreviate several rules by one with some of the applicability premises left out. As a result we obtain rules with $R_i \cup P_i \neq L$ for some i which we call *incomplete*. The notion of a trigger from Def. 4.3 is introduced to recover the original sets of rules from such an abbreviation.

The overall result of our development is expressed in the following statement.

Corollary 7.9 *Every specification ρ in abstract GSOS for the behaviour functor $B = (\mathcal{P}_\omega)^L$ modelling LTS (i.e. a natural transformation as in* (7.1)*) can be characterised by a GSOS specification $\mathcal{R}$. This correspondence is one-to-one up to the abbreviation of sets of complete rules by sets containing incomplete ones, the renaming of variables, and the inclusion or omission of redundant rules. Moreover, the models of the GSOS specification $\mathcal{R}$ (cf. Def. 4.5) are precisely the models of ρ (cf. Def. 6.11) for the natural transformation ρ represented by $\mathcal{R}$.*

Corollary 7.9 is essentially the result of Turi and Plotkin [TP97, Theorem 1.1]. Our treatment now provides a detailed and modular proof, parts of which are furthermore reusable in other settings, as we shall see.

More as a byproduct, we have extended the statement from finite to arbitrary sets of labels L, a task which was explicitly mentioned as an open problem in loc. cit. Actually, an extension from image finite transition systems to arbitrary ones is straightforward as well (actually we do not need to do much more than syntactically replacing the finite powerset functor $\mathcal{P}_\omega$ by the unrestricted one $\mathcal{P}$ in the above argument). The restriction to image finiteness is often imposed in order to obtain a final LTS. It turns out not to be essential for the representation of specifications in abstract GSOS as sets of transition rules as such. The other finiteness assumption we are making, namely the one about the arity of the operators in the signature, seems more severe though.

As another advantage, our proof provides a better insight into the type of redundancy contained in the rule notation. In loc. cit. the correspondence of abstract GSOS and GSOS rules was stated "up to equivalence of sets of rules" only.

	Natural transformation		Representation
(8.1)	$\rho : \Sigma(\mathrm{Id} \times (\mathcal{D}_\omega)^{\mathrm{L}}) \Rightarrow (\mathcal{D}_\omega \mathrm{T})^L$		PGSOS specification (Def. 5.2)
	$\Downarrow$		$\Uparrow$
(8.4)	$\nu^{n,E} : (\mathrm{Id})^n \times (\mathcal{D}^+_\omega)^E \Rightarrow \mathcal{D}_\omega \mathrm{T}$	(8.22)	$\left\{ \frac{\phi_j(y_j)=u_j \quad (1 \leq j \leq k)}{\nu^{n,E}(\langle x_1,\ldots,x_n\rangle,(\phi_e))(t) \stackrel{\pm}{=} w \cdot \prod_j u_j} \right\}$ *finite*, $\sum w \in \{0,1\}$
	$\Downarrow$		$\Uparrow$
(8.8)	$\xi^m : (\mathcal{D}^+_\omega)^E \Rightarrow \mathcal{D}^+_\omega(\mathrm{Id}^m)$	(8.20)	$\left\{ \frac{\phi_j(y_j)=u_j \quad (1 \leq j \leq k)}{\xi^m((\phi_e))(\langle y_{o_1},\ldots,y_{o_m}\rangle) \stackrel{\pm}{=} w \cdot \prod_j u_j} \right\}$ *finite*, $\sum w = 1$
	$\Downarrow$		$\Uparrow$
(8.11)	$\zeta^{\vec{e}}_{(X_e)} : \prod_e \mathcal{D}^+_\omega \mathrm{X_e} \Rightarrow \mathcal{D}^+_\omega(X_{e_1} \times \cdots \times X_{e_m})$	Cor. $\Rightarrow$ 8.5	$\mu^{\vec{e}} \in \mathcal{D}^+_\omega(\mathtt{Par}[m]_{\preceq \vec{e}})$
	$\updownarrow$		$\updownarrow$
(8.13)	$\zeta : \mathcal{D}^+_\omega \Rightarrow \mathcal{D}^+_\omega(\mathrm{Id}^m)$	Thm. $\Rightarrow$ 8.6	$\mu \in \mathcal{D}^+_\omega(\mathtt{Par}[m])$

Figure 4: The outline of the approach in the probabilistic setting ($e \in E$).

8. Deriving PGSOS from abstract GSOS

As in the previous section we will now again derive a concrete representation for specifications in abstract GSOS from Def. 6.11, but this time instantiated with the behaviour functor modelling PTS instead of LTS, i.e. with $\mathrm{B} := (\mathcal{D}_\omega)^L$. So we are dealing with natural transformations

$$\rho : \Sigma\big(\mathrm{Id} \times (\mathcal{D}_\omega)^{\mathrm{L}}\big) \Rightarrow (\mathcal{D}_\omega \mathrm{T})^L. \tag{8.1}$$

Structurally they are rather similar to those in (7.1), so one can expect that the development will be similar to the one in Section 7. It turns out that the decomposition is indeed the same as before, as the outline in Figure 4 shows. It differs from the one in the nondeterministic setting (see again Figure 3) in that the occurrences of the functor $\mathcal{P}_\omega$ are replaced by $\mathcal{D}_\omega$ (and $\mathcal{P}^+_\omega$ by $\mathcal{D}^+_\omega$). The probabilistic nature comes into play almost only when we turn to the representation result for the natural transformations at the bottom of the table. The main result here is Theorem 8.6. Its statement closely relates to that of Theorem 7.6, but the proof is considerably more involved. In the end we will see that the desired representation for the natural transformations ρ in (8.1) is given by PGSOS specifications from Def. 5.2.

In the following we explain the details. The presentation will be rather brief whenever the argument is similar to the one from the nondeterministic case, so the reader is advised to consult Section 7 if any of the steps are unclear. To facilitate this we kept the equation numbering in both sections alike.

8.1 Top-down: decomposing the natural transformations under consideration

The natural transformations in (8.1) are in one-to-one correspondence with those of the shape

$$\tilde{\rho} : \underbrace{\Sigma\big(\mathrm{Id} \times (\mathcal{D}_\omega)^{\mathrm{L}}\big) \times \mathrm{L}}_{=:\mathrm{F}} \Rightarrow \mathcal{D}_\omega \mathrm{T}, \tag{8.2}$$

which in turn are equivalent to families of natural transformations

$$(\nu^z : \mathrm{F}|_z \Rightarrow \mathcal{D}_\omega \mathrm{T})_{z \in \mathrm{F}1}. \tag{8.3}$$

We find $\mathcal{D}_\omega 1 = \{0,1\} \simeq 2$, where the elements in the set are the numbers $0, 1 \in \mathbb{R}_0^+$ viewed as functions $1 \to \mathbb{R}_0^+$. This yields

$$\begin{aligned} \mathrm{F}1 &= \Sigma(1 \times (\mathcal{D}_\omega 1)^{\mathrm{L}}) \times \mathrm{L} \\ &\simeq \Sigma(2^L) \times L = \{\langle \sigma(E_1, \dots, E_n), a\rangle \mid n \in \mathbb{N}, \sigma \in \Sigma_n, E_i \subseteq L, a \in L\}. \end{aligned}$$

For $z = \langle \sigma(E_1, \dots, E_n), a\rangle \in \mathrm{F}1$ we calculate

$$\mathrm{F}|_z \simeq (\mathrm{Id})^n \times (\mathcal{D}_\omega^+)^E,$$

where $E := E_1 + \cdots + E_n$. So each natural transformation ν^z from the representation (8.3) is for a suitable number $n \in \mathbb{N}$ and set E equivalent to a natural transformation

$$\nu^{n,E} : (\mathrm{Id})^n \times (\mathcal{D}_\omega^+)^E \Rightarrow \mathcal{D}_\omega \mathrm{T}. \tag{8.4}$$

The latter in turn is, again for $N := \{1, \dots, n\}$, equivalent to one of the type

$$\tilde{\nu}^{n,E} : (\mathcal{D}_\omega^+)^E \Rightarrow \mathcal{D}_\omega \mathrm{T}(\mathrm{N} + \mathrm{Id}). \tag{8.5}$$

At this point, we need the following correspondent of Lemma 7.1.

Lemma 8.1 *For functors* $\mathrm{G}^i : \mathsf{C} \to \mathsf{Set}$ *($i \in I$) we have*

$$\mathcal{D}_\omega\big(\coprod_{i\in I} \mathrm{G}^i\big) \simeq \coprod_{\mu \in \mathcal{D}_\omega I} \big(\prod_{j \in \mathsf{supp}(\mu)} \mathcal{D}_\omega^+ \mathrm{G}^j\big).$$

Proof: For all sets X we have an equivalence of sets

$$\mathcal{D}_\omega\big(\coprod_{i\in I} \mathrm{G}^i X\big) \simeq \coprod_{\mu \in \mathcal{D}_\omega I} \big(\prod_{j \in \mathsf{supp}(\mu)} \mathcal{D}_\omega^+ \mathrm{G}^j X\big)$$

given from left to right by $\phi \mapsto \iota_\mu\big((\phi_j)_{j\in\mathsf{supp}(\mu)}\big)$ where

$$\mu(i) := \phi[\iota_i[\mathrm{G}^i X]] \quad \text{and} \quad \phi_j(\alpha) := \frac{\phi\big(\iota_j(\alpha)\big)}{\mu(j)} \quad \text{for all } j \in \mathsf{supp}(\mu) \text{ and } \alpha \in \mathrm{G}^j X.$$

The equivalence extends from sets to functors.

□

We get

$$\mathcal{D}_\omega \mathrm{T}(\mathrm{N} + \mathrm{Id}) \overset{(7.6)}{\simeq} \mathcal{D}_\omega\big(\prod_{t \in \mathrm{T}(N+1)} (\mathrm{Id}^{|t|_*})\big) \overset{\text{Lemma 8.1}}{\simeq} \coprod_{\mu \in \mathcal{D}_\omega \mathrm{T}(N+1)} \big(\prod_{t\in\mathsf{supp}(\mu)} \mathcal{D}_\omega^+(\mathrm{Id}^{|t|_*})\big),$$

so that with Lemmata A.4 and A.3 (b) we find that any natural transformation $\tilde{\nu}^{n,E}$ from (8.5) can be characterised by

$$\mu \in \mathcal{D}_\omega \mathrm{T}(N+1) \quad \text{and} \quad \big(\xi^t : (\mathcal{D}_\omega^+)^E \Rightarrow \mathcal{D}_\omega^+(\mathrm{Id}^{|t|_*})\big)_{t\in\mathsf{supp}(\mu)}. \tag{8.7}$$

Natural transformations of the type

$$\xi^m : (\mathcal{D}_\omega^+)^E \Rightarrow \mathcal{D}_\omega^+(\mathrm{Id}^m), \tag{8.8}$$

for $m \in \mathbb{N}$ as they appear in the representation (8.7) are equivalent to natural transformations between functors from Set^E to Set of the type

$$\tilde{\xi}^m_{(X_e)_{e\in E}} : \prod_{e\in E} \mathcal{D}_\omega^+ X_e \Rightarrow \mathcal{D}_\omega^+\big((\prod_{e\in E} X_e)^m\big) : \mathsf{Set}^E \to \mathsf{Set}. \tag{8.9}$$

With

$$\mathcal{D}^+_\omega\big((\coprod_{e\in E} X_e)^m\big) \simeq \mathcal{D}^+_\omega\big(\coprod_{\vec{e}\in E^m}(X_{e_1}\times\cdots\times X_{e_m})\big) \simeq \coprod_{\tilde{\mu}\in\mathcal{D}^+_\omega(E^m)}\big(\prod_{\vec{e}\in\mathtt{supp}(\tilde{\mu})}\mathcal{D}^+_\omega(X_{e_1}\times\cdots\times X_{e_m})\big)$$

each of those can be characterised by

$$\tilde{\mu}\in\mathcal{D}^+_\omega(E^m) \quad \text{along with} \quad \big(\zeta^{\vec{e}}_{(X_e)_{e\in E}} : \prod_{e\in E}\mathcal{D}^+_\omega \mathrm{X_e} \Rightarrow \mathcal{D}^+_\omega(X_{e_1}\times\cdots\times X_{e_m})\big)_{\vec{e}\in\mathtt{supp}(\tilde{\mu})}. \tag{8.10}$$

In the next section we will give a direct representation of the natural transformations $\zeta^{\vec{e}}$ above.

8.2 A representation theorem for the probabilistic setting
Fix $m\in\mathbb{N}$, a set E, and $\vec{e}\in E^m$. In this section we will state that any natural transformation

$$\zeta^{\vec{e}}_{(X_e)_{e\in E}} : \prod_{e\in E}\mathcal{D}^+_\omega \mathrm{X_e} \Rightarrow \mathcal{D}^+_\omega(X_{e_1}\times\cdots\times X_{e_m}) \tag{8.11}$$

arises as a convex combination of the following basic ones.

Definition 8.4 *For* $\Gamma\in\mathtt{Par}[m]_{\preceq\vec{e}}$ *define the* **basic natural transformation** $\beta^{\vec{e},\Gamma}$ *of the type in* (8.11) *for sets* X_e*, distributions* $\phi_e\in\mathcal{D}^+_\omega X_e$ *($e\in E$), and* $\vec{x}\in X_{e_1}\times\cdots\times X_{e_m}$ *as*

$$\beta^{\vec{e},\Gamma}_{(X_e)}\big((\phi_e)\big)(\vec{x}) := \begin{cases}\prod_{c\in\Gamma}\phi_{e_{c\downarrow}}(x_{c\downarrow}) & \textit{if } \Gamma\preceq\mathtt{par}(\vec{x}),\\ 0 & \textit{otherwise.}\end{cases}$$

To see the similarity with Definition 7.4 note that we could have written the latter alternatively as

$$\beta^{\vec{e},\Gamma}_{(X_e)}\big((X'_e)\big)(\vec{x}) = \begin{cases}\bigwedge_{c\in\Gamma}X'_{e_{c\downarrow}}(x_{c\downarrow}) & \text{if } \Gamma\preceq\mathtt{par}(\vec{x}),\\ \bot & \text{otherwise.}\end{cases}$$

In the nondeterministic setting, we gave a derivation rule to calculate these sets. To write down similar rules we would introduce additional variables u_c to carry probabilities. Furthermore we would stipulate that all tuples for which the rule cannot be instantiated receive a zero probability (similar to the convention that the rules in the nondeterministic case define the smallest set satisfying them)

$$\frac{\phi_{e_{c\downarrow}}(y_c) = u_c \quad (c\in\Gamma)}{\beta^{\vec{e},\Gamma}_{(X_e)}\big((\phi_e)\big)(\langle y_{[1]_\Gamma},\ldots,y_{[m]_\Gamma}\rangle) = \prod_{c\in\Gamma}u_c} \tag{8.12}$$

Corollary 8.5 *Every natural transformation* $\zeta^{\vec{e}}$ *as in* (8.11) *can be written as*

$$\zeta^{\vec{e}} = \sum_{\Gamma\in\mathtt{supp}(\mu)}\mu(\Gamma)\cdot\beta^{\vec{e},\Gamma} \quad \textit{for some} \quad \mu\in\mathcal{D}^+_\omega(\mathtt{Par}[m]_{\preceq\vec{e}}).$$

The sum above is to be read point-wise, i.e.

$$\big(\sum_{i\in\mathtt{supp}(\mu)}\mu(i)\cdot\beta^i\big)_X(\alpha) := \sum_{i\in\mathtt{supp}(\mu)}\mu(i)\cdot\beta^i_X(\alpha).$$

For the same reason as before we will again consider the special case $E\simeq 1$ only, i.e. we prove the following theorem.

Theorem 8.6 *For $m \in \mathbb{N}$ every natural transformation*

$$\zeta : \mathcal{D}_\omega^+ \Rightarrow \mathcal{D}_\omega^+(\mathrm{Id}^m). \tag{8.13}$$

can be represented as a convex combination of the basic ones, i.e.

$$\zeta = \sum_{\Gamma \in \mathtt{supp}(\mu)} \mu(\Gamma) \cdot \beta^\Gamma \quad \text{for some } \mu \in \mathcal{D}_\omega^+ \mathtt{Par}[m],$$

where for $\Gamma \in \mathtt{Par}[m]$ the natural transformation $\beta^\Gamma : \mathcal{D}_\omega^+ \Rightarrow \mathcal{D}_\omega^+(\mathrm{Id}^m)$ is given by

$$\beta_X^\Gamma(\phi)(\vec{x}) := \begin{cases} \prod_{c \in \Gamma} \phi(x_{c\downarrow}) & \text{if } \Gamma \preceq \mathtt{par}(\vec{x}), \\ 0 & \text{otherwise.} \end{cases} \tag{8.14}$$

It can easily be shown that for $\mu, \mu' \in \mathcal{D}_\omega^+(\mathtt{Par}[m])$ we have

$$\sum_{\Gamma \in \mathtt{supp}(\mu)} \mu(\Gamma) \cdot \beta^\Gamma \;=\; \sum_{\Gamma \in \mathtt{supp}(\mu')} \mu'(\Gamma) \cdot \beta^\Gamma \quad \text{just in case} \quad \mu = \mu',$$

so the representation of ζ by a distribution μ given above is unique. In the probabilistic case there are no redundant partitions!

For the proof we need a few lemmata. Two of them are solely about real valued functions and we moved them to Appendix B.

Lemma 8.7 *Let ζ be a natural transformation as in* (8.13)*, X and Y be sets, $\phi \in \mathcal{D}_\omega^+ X$ and $\psi \in \mathcal{D}_\omega^+ Y$ be distributions, and let $\vec{x} \in X^m$ and $\vec{y} \in Y^m$. We find*

$$\zeta_X(\phi)(\vec{x}) = \zeta_Y(\psi)(\vec{y}) \quad \text{if} \quad \mathtt{par}(\vec{x}) = \mathtt{par}(\vec{y}) \quad \text{and} \quad \phi(x_i) = \psi(y_i) \quad \text{for all} \quad 1 \le i \le m.$$

Proof: Let $\Gamma := \mathtt{par}(\vec{x})$ $(= \mathtt{par}(\vec{y}))$, $Z := \Gamma \cup \{*\}$, $\chi \in \mathcal{D}_\omega^+ Z$ with $\chi(c) := \phi(x_{c\downarrow})$ $(= \psi(y_{c\downarrow}))$ for $c \in \Gamma$ and $\chi(*) := 1 - \chi[\Gamma]$, and let $\vec{z} := \langle [1]_\Gamma, \ldots, [m]_\Gamma \rangle$. With $f : X \to Z$ where

$$f(x) := \begin{cases} [i]_\Gamma & \text{if } x = x_i \text{ for some } i \in \{1, \ldots, m\}, \\ * & \text{otherwise,} \end{cases}$$

we find

$$\begin{array}{rcll} \zeta_X(\phi)(\vec{x}) & = & \zeta_X(\phi)[(f^m)^{-1}(\vec{z})] & \{(f^m)^{-1}(\vec{z}) = \{\vec{x}\}\} \\ & = & \big((\mathcal{D}_\omega^+(f^m))(\zeta_X(\phi))\big)(\vec{z}) & \{\text{Def. } \mathcal{D}_\omega^+\} \\ & = & \zeta_Z\big((\mathcal{D}_\omega^+ f)(\phi)\big)(\vec{z}) & \{\text{nat. } \zeta\} \\ & = & \zeta_Z(\chi)(\vec{z}). & \{(\mathcal{D}_\omega^+ f)(\phi) = \chi\} \end{array}$$

In the same way we obtain $\zeta_Y(\psi)(\vec{y}) = \zeta_Z(\chi)(\vec{z})$, which implies the statement.

□

The above lemma states that the following family of functions is well defined and characterises ζ uniquely:

Definition 8.8 *For $\Gamma \in \mathtt{Par}[m]$ let*

$$\mathbb{C}^\Gamma := \{u : \Gamma \to \mathbb{R}_0^+ \mid u[\Gamma] \le 1\}.$$

Every natural transformation ζ as in (8.13) *induces a family of functions*

$$\big(\gamma^\Gamma : \mathbb{C}^\Gamma \to [0,1]\big)_{\Gamma \in \mathtt{Par}[m]}$$

defined by $\gamma^\Gamma(u) := \zeta_X(\phi)(\vec{x})$ where X, ϕ, and $\vec{x}$ are such that $\Gamma = \mathtt{par}(\vec{x})$ and $u(c) = \phi(x_{c\downarrow})$ for $c \in \Gamma$. (For all $\Gamma \in \mathtt{Par}[m]$ and $u \in \mathbb{C}^\Gamma$ we can find suitable X, ϕ, and $\vec{x}$. Take e.g. $X := \Gamma \cup \{*\}$, $\vec{x} := \langle [1]_\Gamma, \ldots, [m]_\Gamma \rangle$, $\phi := u[* := u[\Gamma]]$.)

It will be handy to talk about ζ in terms of these functions. For later use we check what they look like in the case of our basic transformations: For $\Gamma' \in \mathtt{Par}[m]$ we find that $\beta^{\Gamma'}$ induces a family of functions $\left(\gamma^\Gamma : \mathbb{C}^\Gamma \to [0,1]\right)_{\Gamma \in \mathtt{Par}[m]}$ with

$$\gamma^\Gamma(u) = \begin{cases} \prod_{c' \in \Gamma'} u([c' \downarrow]_\Gamma) = \prod_{c \in \Gamma} u(c)^{|l(\Gamma', c)|} & \text{if } \Gamma' \preceq \Gamma, \\ 0 & \text{otherwise.} \end{cases}$$

where $l(\Gamma', c) := \{c' \in \Gamma' \mid c' \subseteq c\}$.

The functions γ^Γ induced by a natural transformation ζ have the following property:

Lemma 8.9 *For $\Gamma \in \mathtt{Par}[m]$, $d \in \Gamma$, $u : \Gamma \setminus \{d\} \to \mathbb{R}_0^+$, and $r, s \in \mathbb{R}_0^+$ such that $u[d := r + s] \in \mathbb{C}^\Gamma$ we have*

$$\gamma^\Gamma(u[d := r + s]) = \gamma^\Gamma(u[d := r]) + \gamma^\Gamma(u[d := s]) + \sum_{\emptyset \subset d' \subset d} \gamma^{\Gamma(d')}(u[d' := r, (d \setminus d') := s]),$$

where $\Gamma(d') \in \mathtt{Par}[m]$ for $\emptyset \subset d' \subset d$ results from Γ by splitting d into d' and $d \setminus d'$, i.e.

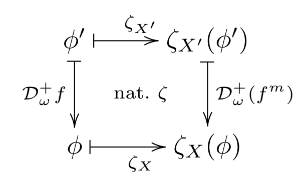
$$\Gamma(d') := (\Gamma \setminus \{d\}) \cup \{d', d \setminus d'\} \prec \Gamma.$$

Proof: The statement follows from the following consideration:

Let Y be a set with $p \notin Y$. Set $X := Y \cup \{p\}$ and let $\phi \in \mathcal{D}_\omega^+ X$ and $\vec{x} \in X^m$ such that p occurs in $\vec{x}$, i.e. $d := \{i \mid x_i = p\} \neq \emptyset$. We can "split" the state p into two, say q_1 and q_2 (for $q_i \notin Y$), and distribute the original probability of p as $\phi(p) = r + s$ on the two copies. This yields $X' := Y \cup \{q_1, q_2\}$ and $\phi' \in \mathcal{D}_\omega^+ X'$ with $\phi'(q_1) := r$, $\phi'(q_2) := s$, and $\phi'(y) = \phi(y)$ for $y \in Y$. From the naturality square of ζ for $f : X' \to X$ with $f(q_i) := p$ and $f(y) := y$ for $y \in Y$ we read off that $\zeta_X(\phi)(\vec{x})$ is the sum of all $\zeta_{X'}(\phi')(\vec{x}')$ where the $\vec{x}'$ arise by replacing in $\vec{x}$ each occurrence of p by either q_1 or q_2. Formally, for $d' \subseteq d$ set

$$\vec{x}^{d'} = \langle x_1^{d'}, \dots, x_m^{d'} \rangle \quad \text{with} \quad x_i^{d'} = \begin{cases} q_1 & \text{if } i \in d', \\ q_2 & \text{if } i \in d \setminus d', \\ x_i & \text{otherwise.} \end{cases}$$

Then we calculate as follows:



$$\begin{aligned} \zeta_X(\phi)(\vec{x}) &= \zeta_X\big((\mathcal{D}_\omega^+ f)(\phi')\big)(\vec{x}) && \{(\mathcal{D}_\omega^+ f)(\phi') = \phi\} \\ &= \big((\mathcal{D}_\omega^+(f^m))(\zeta_{X'}(\phi'))\big)(\vec{x}) && \{\text{nat. } \zeta\} \\ &= \zeta_{X'}(\phi')[(f^m)^{-1}(\vec{x})] && \{\text{def. } \mathcal{D}_\omega^+\} \\ &= \zeta_{X'}(\phi')[\{\vec{x}^{d'} \mid d' \subseteq d\}] && \{f^{-1}(x_i) = \begin{cases} \{p, q\} & \text{if } i \in d, \\ \{x_i\} & \text{otherwise.} \end{cases}\} \\ &= \sum_{d' \subseteq d} \zeta_{X'}(\phi')(\vec{x}^{d'}) \\ &= \zeta_{X'}(\phi')(\vec{x}^\emptyset) + \zeta_{X'}(\phi')(\vec{x}^d) + \sum_{\emptyset \subset d' \subset d} \zeta_{X'}(\phi')(\vec{x}^{d'}). \end{aligned}$$

This idea leads to the statement through an application of Lemma 8.7 to both ends of the computation, together with the observation that for $\Gamma = \mathtt{par}(\vec{x})$ (which yields $d \in \Gamma$) we have $\mathtt{par}(\vec{x}^{\emptyset}) = \Gamma = \mathtt{par}(\vec{x}^{d})$ and $\mathtt{par}(\vec{x}^{d'}) = \Gamma(d')$ for $\emptyset \subset d' \subset d$. (Of course we again need to show that for all suitable Γ and u we can find appropriate X, ϕ, and $\vec{x}$. This can be done as in the proof of Lemma 8.7.)

□

Lemma 8.10 *Let ζ be a natural transformation as in* (8.13) *inducing the family $(\gamma^{\Gamma})_{\Gamma\in\mathtt{Par}[m]}$ from Definition 8.8. For every downwards closed set $M \subseteq \mathtt{Par}[m]$ there exist weights $(\tau_\Gamma \in \mathbb{R}_0^+)_{\Gamma\in M}$ such that for all $\Gamma \in M$ and $u \in \mathbb{C}^{\Gamma}$ we have*

$$\gamma^{\Gamma}(u) = \sum_{\Gamma'\preceq\Gamma} \tau_{\Gamma'} \cdot \prod_{c\in\Gamma} u(c)^{|l(\Gamma',c)|}, \tag{8.15}$$

where again $l(\Gamma', c) := \{c' \in \Gamma' \mid c' \subseteq c\}$.

Proof: The statement is proved by induction on the size of M. For $M = \emptyset$ there is nothing to do. For nonempty M choose a maximal element $\hat{\Gamma} \in M$. Take $(\tau_\Gamma)_{\Gamma\in\hat{M}}$ as given by the induction hypothesis for $\hat{M} := M \setminus \{\hat{\Gamma}\}$. These coefficients satisfy the statement for all $\Gamma \in \hat{M}$ already. We have to find $\tau_{\hat{\Gamma}}$ so that it holds for $\hat{\Gamma}$ as well.

For all $v \in \mathbb{C}^{\hat{\Gamma}}$ defining

$$f(v) := \sum_{\Gamma'\prec\hat{\Gamma}} \tau_{\Gamma'} \cdot \prod_{c\in\hat{\Gamma}} v(c)^{|l(\Gamma',c)|} \quad \text{and} \quad h(v) := \gamma^{\hat{\Gamma}}(v) - f(v),$$

we need to show that there exists a $\tau_{\hat{\Gamma}} \in \mathbb{R}_0^+$ such that

$$h(v) = \tau_{\hat{\Gamma}} \cdot \prod_{c\in\hat{\Gamma}} v(c).$$

The set $\mathbb{C}^{\hat{\Gamma}}$ satisfies the assumption on C in Lemma B.2. Applying the lemma we get that it suffices to show that h is linear in all components. So for any $d \in \hat{\Gamma}$ and $u : \hat{\Gamma} \setminus \{d\} \to \mathbb{R}_0^+$ we need to show that

$$h_u(c \cdot r) := h(u[d := c \cdot r]) = c \cdot h_u(r) \quad \text{for all } c \in [0,1] \text{ and } r \in \mathbb{R}_0^+ \text{ with } u[d := r] \in \mathbb{C}^{\hat{\Gamma}}.$$

The latter condition is satisfied for all $r \in \big[0, u[\hat{\Gamma} \setminus \{d\}]\big]$, and since h_u is bounded (because $\gamma^{\hat{\Gamma}}$ and f are), we can apply Lemma B.1 for this task. With this statement, it remains to be shown that

$$h_u(r + s) = h_u(r) + h_u(s) \quad \text{for all } r, s \in \mathbb{R}_0^+ \text{ such that } r + s \le u[\hat{\Gamma} \setminus \{d\}].$$

Abbreviating as before $\gamma^{\hat{\Gamma}}(u[d := r])$ to $\gamma_u^{\hat{\Gamma}}(r)$ and $f(u[d := r])$ to $f_u(r)$ this is equivalent to

$$\gamma_u^{\hat{\Gamma}}(r + s) - \gamma_u^{\hat{\Gamma}}(r) - \gamma_u^{\hat{\Gamma}}(s) = f_u(r + s) - f_u(r) - f_u(s). \tag{8.16}$$

For the left hand side we compute

$$\begin{aligned}
&\gamma_u^{\hat{\Gamma}}(r + s) - \gamma_u^{\hat{\Gamma}}(r) - \gamma_u^{\hat{\Gamma}}(s) \\
&\overset{\text{Lemma 8.9}}{=} \sum_{\emptyset\subset d'\subset d} \gamma^{\hat{\Gamma}(d')}(u[d' := r, (d \setminus d') := s]) \\
&\overset{I.H.}{=} \sum_{\emptyset\subset d'\subset d} \Bigg(\sum_{\Gamma'\preceq\hat{\Gamma}(d')} \underbrace{\tau_{\Gamma'} \cdot \Bigg(\prod_{c\in\hat{\Gamma}\setminus\{d\}} (u(c))^{|l(\Gamma',c)|} \Bigg)}_{=:\tilde{\tau}_{\Gamma'}} \cdot r^{|l(\Gamma',d')|} \cdot s^{|l(\Gamma',d\setminus d')|} \Bigg) \\
&= \sum_{\Gamma'\prec\hat{\Gamma}} \Bigg(\tilde{\tau}_{\Gamma'} \cdot \sum_{\emptyset\subset d'\subset d, \Gamma'\preceq\hat{\Gamma}(d')} r^{|l(\Gamma',d')|} \cdot s^{|l(\Gamma',d)|-|l(\Gamma',d')|} \Bigg).
\end{aligned}$$

With

$$f_u(x) = \sum_{\Gamma' \prec \hat{\Gamma}} \underbrace{\tau_{\Gamma'} \cdot \Big(\prod_{c \in \hat{\Gamma} \setminus \{d\}} u(c)^{|l(\Gamma',c)|} \Big)}_{=\tilde{\tau}_{\Gamma'}} \cdot x^{|l(\Gamma',d)|} \quad \text{and} \quad (r+s)^k = r^k + s^k + \sum_{j=1}^{k-1} \binom{k}{j} \cdot r^j \cdot s^{k-j}$$

for the right hand side of (8.16) we get

$$f_u(r+s) - f_u(r) - f_u(s) = \sum_{\Gamma' \prec \hat{\Gamma}} \Big(\tilde{\tau}_{\Gamma'} \cdot \sum_{j=1}^{|l(\Gamma',d)|-1} \binom{|l(\Gamma',d)|}{j} \cdot r^j \cdot s^{|l(\Gamma',d)|-j} \Big).$$

So we are done if for all $\Gamma' \prec \hat{\Gamma}$ we can show that the two inner sums are equal, i.e.

$$\sum_{\emptyset \subset d' \subset d, \Gamma' \preceq \hat{\Gamma}(d')} r^{|l(\Gamma',d')|} \cdot s^{|l(\Gamma',d)|-|l(\Gamma',d')|} = \sum_{j=1}^{|l(\Gamma',d)|-1} \binom{|l(\Gamma',d)|}{j} \cdot r^j \cdot s^{|l(\Gamma',d)|-j}. \tag{8.17}$$

Let's investigate what the sum on the left hand side ranges over: For $d' \subseteq d$ we can rewrite the condition $\Gamma' \preceq \hat{\Gamma}(d')$ into $\Gamma' \preceq \hat{\Gamma}$ and $c' \subseteq d'$ or $c' \subseteq d \setminus d'$ for all $c' \in \Gamma'$ with $c' \subseteq d$, i.e. for all $c' \in l(\Gamma', d)$. The first part is implied by our assumption $\Gamma' \prec \hat{\Gamma}$. The second can be stated as $d' = \bigcup C$ for some $C \subseteq l(\Gamma', d)$. The condition $\emptyset \subset d' \subset d$ is satisfied just in case $\emptyset \subset C \subset l(\Gamma', d)$. So with $|l(\Gamma', d')| = |C|$ the sum on the left hand side of (8.17) rewrites to

$$\sum_{\emptyset \subset C \subset l(\Gamma',d)} r^{|C|} \cdot s^{|l(\Gamma',d)|-|C|} = \sum_{j=1}^{|l(\Gamma',d)|-1} \underbrace{\big|\{C \subseteq l(\Gamma',d) \mid |C| = j\}\big|}_{=\binom{|l(\Gamma',d)|}{j}} \cdot r^j \cdot s^{|l(\Gamma',d)|-j}.$$

This completes the proof of (8.17) and thus of (8.16).

We have demonstrated that there is a $\tau_{\hat{\Gamma}} \in \mathbb{R}$ such that equation (8.15) holds for $\gamma^{\hat{\Gamma}}$. It remains to be shown that $\tau_{\hat{\Gamma}} \geq 0$. For $r \in \mathbb{R}_0^+$ let $v_r : \hat{\Gamma} \to \mathbb{R}_0^+$ denote the constant function with $v_r(c) = r$ for all $c \in \hat{\Gamma}$. With $0 < r \leq \frac{1}{|\hat{\Gamma}|}$ we find $v_r \in \mathbb{C}^{\hat{\Gamma}}$. We have

$$0 \leq \gamma^{\hat{\Gamma}}(v_r) = \sum_{\Gamma' \preceq \hat{\Gamma}} \tau_{\Gamma'} \cdot \prod_{c \in \hat{\Gamma}} r^{|l(\Gamma',c)|} = \sum_{\Gamma' \preceq \hat{\Gamma}} \tau_{\Gamma'} \cdot r^{|\Gamma'|} = r^{|\hat{\Gamma}|} \cdot (\tau_{\hat{\Gamma}} + \sum_{\Gamma' \prec \hat{\Gamma}} \tau_{\Gamma'} \cdot r^{|\Gamma'|-|\hat{\Gamma}|}).$$

This implies $\tau_{\hat{\Gamma}} \geq -\sum_{\Gamma' \prec \hat{\Gamma}} \tau_{\Gamma'} \cdot r^{|\Gamma'|-|\hat{\Gamma}|}$. Since $|\Gamma'| > |\hat{\Gamma}|$ for all $\Gamma' \prec \hat{\Gamma}$ we have that the right hand side converges to 0 for $r \to 0$, and so $\tau_{\hat{\Gamma}} \geq 0$ as wanted.

□

Proof: [Theorem 8.6] Just take $\mu(\Gamma) = \tau_\Gamma$ for the values from Lemma 8.10 for $M = \mathtt{Par}[m]$. These weights satisfy the left identity above. It remains to be shown that we get a probability distribution indeed, i.e. that all weights sum up to one. For an arbitrary set X and distribution $\phi \in \mathcal{D}_\omega^+ X$ we have

$$1 = \zeta_X(\phi)[X^m] = \sum_{\Gamma \in \mathtt{Par}[m]} \tau_\Gamma \cdot \underbrace{\beta_X^\Gamma[X^m]}_{=1} = \sum_{\Gamma \in \mathtt{Par}[m]} \tau_\Gamma.$$

□

8.3 Bottom-up: constructing the rule format

We have proved a representation result for the simple natural transformations from the bottom row of the table in Figure 4 and claim that with a straightforward extension of the proof one obtains Corollary 8.5 for the line above. We will extend the representation to the more complex types.

Plugging the representation of the natural transformations $\zeta^{\vec{e}}$ in (8.11) given by Corollary 8.5 into (8.10), we find that a natural transformation $\tilde{\xi}^m$ as in (8.9) can be characterised by a distribution $\tilde{\mu} \in \mathcal{D}^+_\omega(E^m)$ and distributions $\left(\mu^{\vec{e}} \in \mathcal{D}^+_\omega(\mathtt{Par}[m]_{\preceq\vec{e}})\right)_{\vec{e}\in\mathtt{supp}(\tilde{\mu})}$. We write this more compactly as one distribution

$$\mu^m \in \mathcal{D}^+_\omega\big\{\langle\vec{e},\Gamma\rangle \mid \vec{e} \in E^m,\ \Gamma \in \mathtt{Par}[m]_{\preceq\vec{e}}\big\},$$

where

$$\mu^m(\langle\vec{e},\Gamma\rangle) := \begin{cases} \tilde{\mu}(\vec{e}) \cdot \mu^{\vec{e}}(\Gamma) & \text{if } \vec{e} \in \mathtt{supp}(\tilde{\mu}), \\ 0 & \text{otherwise.} \end{cases}$$

This distribution μ^m represents the natural transformation

$$\begin{aligned} \tilde{\xi}^m_{(X_e)_{e\in E}} &= \sum_{\langle\vec{e},\Gamma\rangle\in\mathtt{supp}(\mu^m)} \mu^m(\langle\vec{e},\Gamma\rangle) \cdot \big(\mathcal{D}^+_\omega(\iota_{e_1} \times \cdots \times \iota_{e_m}) \circ \beta^{\vec{e},\Gamma}_{(X_e)_{e\in E}}\big) \\ &: \prod_{e\in E} \mathcal{D}^+_\omega \mathrm{X_e} \Rightarrow \mathcal{D}^+_\omega\big((\coprod_{e\in E} X_e)^m\big). \end{aligned}$$

Through the correspondence given by Lemma A.6, the same distribution μ^m characterises a natural transformation ξ^m from (8.8) as

$$\begin{aligned} \xi^m_X &= \mathcal{D}^+_\omega([\mathtt{id}_X]_{e\in E})^m \circ \tilde{\xi}^m_{(X)} \\ &= \mathcal{D}^+_\omega([\mathtt{id}_X]_{e\in E})^m \circ \left(\sum_{\langle\vec{e},\Gamma\rangle\in\mathtt{supp}(\mu^m)} \mu^m(\langle\vec{e},\Gamma\rangle) \cdot \Big(\mathcal{D}^+_\omega(\iota_{e_1} \times \cdots \times \iota_{e_m}) \circ \beta^{\vec{e},\Gamma}_{(X)}\Big)\right) \\ &= \sum_{\langle\vec{e},\Gamma\rangle\in\mathtt{supp}(\mu^m)} \mu^m(\langle\vec{e},\Gamma\rangle) \cdot \Big(\mathcal{D}^+_\omega\big(\underbrace{([\mathtt{id}_X]_{e\in E})^m \circ (\iota_{e_1} \times \cdots \times \iota_{e_m})}_{=\mathtt{id}_{X^m}}\big) \circ \beta^{\vec{e},\Gamma}_{(X)}\Big) \\ &= \sum_{\langle\vec{e},\Gamma\rangle\in\mathtt{supp}(\mu^m)} \mu^m(\langle\vec{e},\Gamma\rangle) \cdot \beta^{\vec{e},\Gamma}_{(X)} : \quad (\mathcal{D}^+_\omega X)^E \Rightarrow \mathcal{D}^+_\omega(X^m). \end{aligned}$$

Below Def. 8.4 we remarked that a basic natural transformation $\beta^{\vec{e},\Gamma}$ can be described by a derivation rule as in (8.12). To use these rules for the description of ξ^m, we have to incorporate the weight $w = \mu^m(\langle\vec{e},\Gamma\rangle)$ of the contribution of $\beta^{\vec{e},\Gamma}$. Using again a finite set of successor variables $Y = \{y_1, \ldots, y_k\}$ each y_j with an associated type $\tau_j \in E$ and probability variable u_j, and a vector $\vec{y} = \langle y_{o_1}, \ldots, y_{o_m}\rangle \in Y^m$ (with the requirement that every y_j appears in $\vec{y}$) to encode $\vec{e}$ and Γ, this yields a rule as below.

$$\frac{\phi_j(y_j) = u_j \quad (1 \leq j \leq k)}{\xi^m\big((\phi_e)\big)(\langle y_{o_1}, \ldots, y_{o_m}\rangle) \stackrel{+}{=} w \cdot \prod_j u_j} \tag{8.19}$$

We will denote the distribution μ^m characterising a natural transformation ξ^m as in (8.8) as a finite set of such rules. Since the set of rules has to represent a probability distribution, we impose the global constraint that the weights w of all rules should sum up to 1.

$$\xi^m \doteq \mu^m \doteq \left\{ \frac{\phi_j(y_j) = u_j \quad (1 \leq j \leq k)}{\xi^m\big((\phi_e)\big)(\langle y_{o_1}, \ldots, y_{o_m}\rangle) \stackrel{+}{=} w \cdot \prod_j u_j} \right\}_{\textit{finite},\ \sum w=1} \tag{8.20}$$

We write a plus above the equality sign in the conclusion to express that after instantiating one of the rules, the real value calculated in the conclusion does not denote an overall probability, but the rule's contribution to it. The overall probability of a tuple is given by the sum of all contributions derivable from different instances of the rules. The following example is intended to explain how such a set of rules defines a natural transformation.

Example 8.11 *Suppose in the case $E = \{1, 2\}$ and $m = 3$ that ξ^m is represented by the following two rules.*

$$\frac{\phi_1(x) = u \quad \phi_2(z) = v}{\xi^m(\langle\phi_1, \phi_2\rangle)(\langle x, z, z\rangle) \stackrel{+}{=} \frac{1}{5}\, u\, v} \qquad \frac{\phi_1(x) = u \quad \phi_1(y) = v \quad \phi_2(z) = w}{\xi^m(\langle\phi_1, \phi_2\rangle)(\langle x, y, z\rangle) \stackrel{+}{=} \frac{4}{5}\, u\, v\, w}$$

For a set P, states $p, q \in P$, and distributions $\phi_1, \phi_2 \in \mathcal{D}_\omega^+ P$ we calculate the probability of $\langle p, q, q\rangle$ in $\xi^m_P(\phi_1, \phi_2)$. Set $r := \phi_1(p)$, $s := \phi_1(q)$, and $t := \phi_2(q)$. The rules can be instantiated to contribute to the probability of $\langle p, q, q\rangle$ as

$$\frac{\phi_1(p) = r \quad \phi_2(q) = t}{\xi^m_P(\langle\phi_1, \phi_2\rangle)(\langle p, q, q\rangle) \stackrel{+}{=} \frac{1}{5}\, r\, t} \qquad \text{and} \qquad \frac{\phi_1(p) = r \quad \phi_1(q) = s \quad \phi_2(q) = t}{\xi^m_P(\langle\phi_1, \phi_2\rangle)(\langle p, q, q\rangle) \stackrel{+}{=} \frac{4}{5}\, r\, s\, t}$$

We conclude

$$\xi^m_P(\phi_1, \phi_2)(\langle p, q, q\rangle) = \frac{1}{5}\, r\, t + \frac{4}{5}\, r\, s\, t.$$

Remember that – in contrast to the nondeterministic case – the representation of a natural transformation ξ^m by a distribution μ^m over the basic natural transformations is unique. However, the move to the rule notation introduces redundancy, even if we look at the rules up to the renaming of variables. The reason is that we can write down more then one rule to encode the same $\vec{e}$ and Γ. The weights of these rules would add up to the contribution of $\beta^{\vec{e},\Gamma}$ to ξ^m. We call this the *splitting of a rule* and we will not disallow it, since it does not really harm (the above interpretation of the rules for instance still works fine.) So the representation is unique up to the renaming of variables and the splitting of rules.

According to (8.7) the representation of $\tilde{\nu}^{n,E}$ from (8.5) is now given by a distribution $\mu \in \mathcal{D}_\omega \mathrm{T}(N + 1)$ and for each $t \in \mathtt{supp}(\mu)$ a set of rules as in (8.20) with $m = |t|_*$. We again replace the vector $\vec{y}$ in each rule for one t by the term $t_Y \in \mathrm{T}(N + Y)$ that arises after replacing the i-th occurrence of $*$ in t by y_{o_i} for all i. The condition on $\vec{y}$ translates into the postulation that every y_j for $1 \leq j \leq k$ should occur in t_Y at least once.

We can again collect the rewritten rules for all $t \in \mathtt{supp}(\mu)$ into one set, but we have to take the probabilities in μ into account: For $t \in \mathtt{supp}(\mu)$ a rule in the representation of ξ^t with weight w would be adapted to have weight $\mu(t) \cdot w$. This yields a finite set of rules as below with the global condition that all their weights should sum up to 0 (i.e. the set is empty) or 1, since $\mu[\mathrm{T}(N + 1)] \in \{0, 1\}$.

$$\tilde{\nu}^{n,E} \doteq \left\{ \quad \frac{\phi_j(y_j) = u_j \quad (1 \leq j \leq k)}{\tilde{\nu}^{n,E}((\phi_e))(t_Y) \stackrel{+}{=} w \cdot \prod_j u_j} \quad \right\}_{\mathit{finite},\ \sum w \in \{0,1\}} \tag{8.21}$$

For the step from $\tilde{\nu}^{n,E}$ in (8.5) to $\nu^{n,E}$ in (8.4) the elements from $N := \{1, \ldots, n\}$ appearing in each term t_Y are again replaced by distinct variables $\{x_1, \ldots, x_n\} =: X$ different from those in Y. This yields sets of rules as below where $t_{X,Y} \in \mathrm{T}(X + Y)$.

$$\nu^{n,E} \doteq \left\{ \quad \frac{\phi_j(y_j) = u_j \quad (1 \leq j \leq k)}{\nu^{n,E}(\langle x_1, \ldots, x_n\rangle, (\phi_e))(t_{X,Y}) \stackrel{+}{=} w \cdot \prod_j u_j} \quad \right\}_{\mathit{finite},\ \sum w \in \{0,1\}} \tag{8.22}$$

To characterize $\tilde{\rho}$ from (8.2) we collect the descriptions as above of the individual ν^z from (8.3) after including into each rule an encoding of the corresponding $z = \langle \sigma(E_1, \ldots, E_n), a\rangle \in \mathrm{F}1$. To this end we again add premises ensuring that the rule can be used just in case $\tilde{\rho}$ is applied to $\sigma(\langle x_1, \theta_1\rangle, \ldots, \langle x_n, \theta_n\rangle)$ and the label a such that $\theta_i(b)$ is the zero map (i.e. has empty support) just in case $b \notin E_i$. Again, X'_{τ_j} is replaced by $\theta_{i_j}(l_j)$ where $\tau_j = \iota_{i_j}(l_j) \in E = E_1 + \cdots + E_n$. This leads to sets of rules of the type below.

$$\frac{\begin{array}{ll} \mathtt{supp}(\theta_i(b)) \neq \emptyset & b \in E_i, 1 \leq i \leq n \\ \mathtt{supp}(\theta_i(b)) = \emptyset & b \notin E_i, 1 \leq i \leq n \\ \theta_{i_j}(l_j)(y_j) = u_j & 1 \leq j \leq k \end{array}}{\tilde{\rho}\big(\sigma(\langle x_1, \theta_1\rangle, \ldots, \langle x_n, \theta_n\rangle)\big)(a)(t) \stackrel{\pm}{=} w \cdot \prod_j u_j} \tag{8.23}$$

The condition on the original sets of rules translates into the following one: the specification contains finitely many rules only for the same $\sigma \in \Sigma_n$, $a \in L$, and $E_1, \ldots, E_n \subseteq L$, and the weights w of all these rules sum up to 1, if there are any.

This characterization is essentially a PGSOS specification from Def. 5.2, if we syntactically replace

- a premise $\mathtt{supp}(\theta_i(b)) \neq \emptyset$ by $x_i \xrightarrow{b}$,
- a premise $\mathtt{supp}(\theta_i(b)) = \emptyset$ by $x_i \stackrel{b}{\not\longrightarrow}$,
- a premise $\theta_{i_j}(l_j)(y_j) = u_j$ by $x_{i_j} \xrightarrow{l_j[u_j]} y_j$,
- the conclusion by $\sigma(x_1, \ldots, x_n) \xrightarrow{a[w \cdot \prod_j u_j]} t_{X,Y}$,

and allow to abbreviate several complete rules as above by incomplete ones, i.e. by rules where for some x_i and label $b \in L$ neither the positive applicability premise $x_i \xrightarrow{b}$ nor the negative one $x_i \stackrel{b}{\not\longrightarrow}$ is present.

Taken together, we obtained the following result.

Corollary 8.12 *Each specification ρ in abstract GSOS instantiated with the behaviour functor $\mathrm{B} = (\mathcal{D}_\omega)^L$ modelling PTS (i.e. a natural transformation as in (8.1)) can be characterised by a PGSOS specification $\mathcal{R}$. This correspondence is one-to-one up to the abbreviation of sets of complete rules by sets containing incomplete ones, the renaming of variables, and the splitting of rules. Moreover, the models of the PGSOS specification $\mathcal{R}$ (cf. Def. 5.3) are precisely the models of ρ (cf. Def. 6.11) for the natural transformation ρ represented by $\mathcal{R}$.*

With this statement, Proposition 5.6 arises as an instance of Proposition 6.12 about the abstract framework. Note though that in order to obtain this result most of the effort we spent establishing the correspondence of abstract GSOS and PGSOS is not necessary. It would have been sufficient to know that a specification in PGSOS can be captured by a natural transformation ρ as in (8.1). We do not need to prove that all natural transformations ρ arise in such a way, which is actually the hard part. We tackled both directions in order to determine the exact position of PGSOS in Turi and Plotkin's framework.

We have for instance experimented with a format for transition systems showing nondeterministic as well as probabilistic behaviour. As of yet we are not able to prove a similarly strong result for it, but it is not so difficult to show that it is well-behaved by proving that the rules give rise to specifications in the corresponding instance of abstract GSOS.

9. Related and future work

We developed a specification format for (reactive) probabilistic transition systems (PTS) as studied by Larsen and Skou [LS91], who also introduced the corresponding notion of a probabilistic bisimulation.

These systems were studied from a coalgebraic point of view e.g. by de Vink and Rutten [dVR99] and Moss [Mos99].

Larsen and Skou [LS92] furthermore defined a set of basic operators to construct (finite) probabilistic transition systems and stated that probabilistic bisimulation is a congruence for them. A similar set of operators, but this time including recursion, was considered by van Glabbeek, Smolka, and Steffen [vGSS95] (The type of system we treated here is called the *reactive model* in loc. cit. and it is just one out of several types of probabilistic systems considered there.) The congruence result they give is wider in scope than the one by Larsen and Skou in that it reaches infinite systems as well through the use of the recursion operator. Our specification format and thus our congruence statement covers their operators but for the recursion operator, which yields solutions of recursive specifications. In our framework we treated solutions of (guarded) recursive specifications separately, without defining an operator for it.

We are not aware of any proposal for a specification format for probabilistic transition systems ensuring well-behavedness properties. The only step in this direction that we have seen appears in the overview paper by Jonsson, Larsen, and Yi [JLY01], who work with a richer type of system exhibiting nondeterministic as well as probabilistic behaviour. They explain how specifications in the DeSimone format — a format weaker than GSOS — for LTS can be interpreted in the richer setting. But except for a "built-in" probabilistic choice no real probabilistic operator can be defined this way.

The categorical framework generalizing GSOS rules is taken from the work of Turi and Plotkin [TP97], with additions from an article by by Lenisa, Power and Watanabe [LPW00] and our previous work [Bar03]. Turi [Tur97] has worked out concrete examples for several instances of the abstract GSOS format, but no rule format was developed out of these considerations and none of the examples involved probabilistic systems. The idea of using the abstract format for the derivation of novel specification formats for concrete systems has recently also been followed by Marco Kick [Kic02a, Kic02b], who works with timed systems.

The aim of the work reported here was not only to derive a specification format for one particular kind of (probabilistic) system, but also more generally to gain experience in the development of concrete formats out of abstract GSOS. With this approach and the given lemmata one for instance immediately gets a format for *generative* probabilistic transition systems (as defined by van Glabbeek et al. [vGSS95]) as well, and one can make first steps toward an adaptation to systems that include both, nondeterministic and probabilistic choice. We leave the study of the latter type of system – which has received a lot of attention recently – to future work.

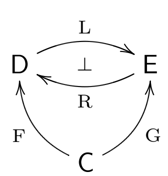
Acknowledgments
I would like to thank my CWI colleagues and the ACG colloquium for helpful discussions. I am particularly indebted to Jan Rutten for his supervision and extensive proof-reading.

Appendix

A. Basic equivalences of natural transformations

In order to decompose the natural transformations arising from the abstract GSOS format, we used some general but simple lemmata, which we state and prove here.

Lemma A.1 *Consider categories and functors as pictured below, where* L *is left adjoint to* R*:*



There is a one-to-one correspondence between natural transformations

$$\nu : \mathrm{F} \Rightarrow \mathrm{RG} \quad \textit{and} \quad \xi : \mathrm{LF} \Rightarrow \mathrm{G}$$

given by $\nu \mapsto \varepsilon \mathrm{G} \circ \mathrm{L}\nu$ *and* $\xi \mapsto \mathrm{R}\xi \circ \eta\mathrm{F}$*, where* $\eta : \mathrm{Id} \Rightarrow \mathrm{RL}$ *and* $\varepsilon : \mathrm{LR} \Rightarrow \mathrm{Id}$ *are the unit and counit of the adjunction.*

Proof: To show that the two constructions are inverses of each other, we calculate using (i) naturality of η and (ii) the adjunction law $\mathrm{R}\varepsilon \circ \eta\mathrm{R} = \mathtt{id}\mathrm{R}$

$$\mathrm{R}(\varepsilon\mathrm{G} \circ \mathrm{L}\nu) \circ \eta\mathrm{F} = \mathrm{R}\varepsilon\mathrm{G} \circ \mathrm{RL}\nu \circ \eta\mathrm{F} \overset{(i)}{=} \mathrm{R}\varepsilon\mathrm{G} \circ \eta\mathrm{RG} \circ \nu = (\mathrm{R}\varepsilon \circ \eta\mathrm{R})\mathrm{G} \circ \nu \overset{(ii)}{=} \mathtt{id}\mathrm{RG} \circ \nu = \nu$$

and similarly, using (i) naturality of ε and (ii) the adjunction law $\varepsilon\mathrm{L} \circ \mathrm{L}\eta = \mathtt{id}\mathrm{L}$

$$\varepsilon\mathrm{G} \circ \mathrm{L}(\mathrm{R}\xi \circ \eta\mathrm{F}) = \varepsilon\mathrm{G} \circ \mathrm{LR}\xi \circ \mathrm{L}\eta\mathrm{F} \overset{(i)}{=} \xi \circ \varepsilon\mathrm{LF} \circ \mathrm{L}\eta\mathrm{F} = \xi \circ (\varepsilon\mathrm{L} \circ \mathrm{L}\eta)\mathrm{F} \overset{(ii)}{=} \xi \circ \mathtt{id}\mathrm{LF} = \xi.$$

□

Lemma A.2 *Let* C *be a category with a final object* 1_{C}*. Every functor* $\mathrm{F} : \mathsf{C} \to \mathsf{Set}$ *can be written as*

$$\mathrm{F} \simeq \coprod_{z \in \mathrm{F}1_{\mathsf{C}}} \mathrm{F}|_z$$

with $\mathrm{F}|_z X := (\mathrm{F}!_X)^{-1}(z)$ *for a* C*-object* X*, and* $\mathrm{F}|_z f : \mathrm{F}|_z X \to \mathrm{F}|_z Y$ *for an arrow* $f : X \to Y$ *is the restriction of* $\mathrm{F}f : \mathrm{F}X \to \mathrm{F}Y$ *to* $\mathrm{F}|_z X$*.*

Proof: For any $f : X \to Y$ and $x \in \mathrm{F}|_z X$ we need to check that $(\mathrm{F}f)(x) \in \mathrm{F}|_z Y$ indeed, but this easily follows from finality:

$$\mathrm{F}!_Y((\mathrm{F}f)(x)) = (\mathrm{F}(!_Y \circ f))(x) = (\mathrm{F}!_X)(x) = z.$$

□

We furthermore used the following special case of the fact that point-wise (co)limits of any type in D yield (co)limits of that type in D^{C}:

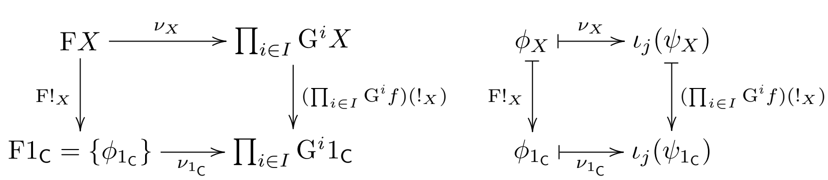
Lemma A.3 *Let* $\mathrm{F}^i, \mathrm{G} : \mathsf{C} \to \mathsf{D}$ *for* $i \in I$ *be functors.*
(a) Let the category D *have* I*-indexed coproducts. There is a one-to-one correspondence between natural transformations* $\nu : \coprod_{i \in I} \mathrm{F}^i \Rightarrow \mathrm{G}$ *and families of natural transformations* $(\nu^i : \mathrm{F}^i \Rightarrow \mathrm{G})_{i \in I}$*.*
(b) Dually, let the category D *have* I*-indexed products. There is a one-to-one correspondence between natural transformations* $\nu : \mathrm{G} \Rightarrow \prod_{i \in I} \mathrm{F}^i$ *and families of natural transformations* $(\nu^i : \mathrm{G} \Rightarrow \mathrm{F}^i)_{i \in I}$*.*

Lemma A.4 *Let* C *be a category with a final object* 1_{C} *and let* $\mathrm{F}, \mathrm{G}^i : \mathsf{C} \to \mathsf{Set}$ *(*$i \in I$*) be functors with* $\mathrm{F}1_{\mathsf{C}} \simeq 1$*. Every natural transformation*

$$\nu : \mathrm{F} \Rightarrow \coprod_{i \in I} \mathrm{G}^i$$

factors as $\nu = \iota_j \circ \nu^j$ *for some* $j \in I$ *and natural transformation* $\nu^j : \mathrm{F} \Rightarrow \mathrm{G}^j$*, where* $\iota_j : \mathrm{G}^j \Rightarrow \coprod_{i \in I} \mathrm{G}^i$ *is the coproduct injection.*

Proof: Let $j \in I$ be such that $\nu_{1_{\mathsf{C}}}(\phi_{1_{\mathsf{C}}}) = \iota_j(\psi_{1_{\mathsf{C}}})$ for some $\psi_{1_{\mathsf{C}}} \in \mathrm{G}^j 1_{\mathsf{C}}$, where $\phi_{1_{\mathsf{C}}}$ is the unique element of $\mathrm{F}1_{\mathsf{C}}$. It suffices to show that for all sets X and $\phi_X \in \mathrm{F}X$ we have that $\nu_X(\phi_X) = \iota_j(\psi_X)$ for some $\psi_X \in \mathrm{G}^j X$. This is equivalent to saying that $(\coprod_{i \in I} \mathrm{G}^i !_X)\nu_X(\phi_X) = \iota_j(\psi'_{1_{\mathsf{C}}})$ for some $\psi'_{1_{\mathsf{C}}} \in \mathrm{G}^j 1_{\mathsf{C}}$, where $!_X : X \to 1_{\mathsf{C}}$ is the unique map given by finality of 1_{C}. But this is the case since by naturality of ν we have $(\coprod_{i \in I} \mathrm{G}^i !_X)(\nu_X(\phi_X)) = \nu_{1_{\mathsf{C}}}(\mathrm{F}!_X(\phi_X)) = \nu_{1_{\mathsf{C}}}(\phi_{1_{\mathsf{C}}}) = \iota_j(\psi_{1_{\mathsf{C}}})$.



□

Lemma A.5 *Let* $\mathrm{F}, \mathrm{G} : \mathsf{Set} \to \mathsf{Set}$ *be functors and let* A *be a set. There is a one-to-one correspondence between natural transformations*

$$\nu : (\mathrm{Id})^A \times \mathrm{F} \Rightarrow \mathrm{G} \quad \textit{and} \quad \xi : \mathrm{F} \Rightarrow \mathrm{G}(\mathrm{A} + \mathrm{Id})$$

given by $\nu \mapsto \xi^\nu$ *and* $\xi \mapsto \nu^\xi$ *defined for any set* X, $\alpha \in \mathrm{F}X$, *and* $f : A \to X$ *as*

$$\xi^\nu_X(\alpha) := \nu_{A+X}(\iota_1, (\mathrm{F}\iota_2)(\alpha)) \quad \textit{and} \quad \nu^\xi_X(f, \alpha) := (\mathrm{G}[f, \mathtt{id}_X] \circ \xi_X)(\alpha).$$

Proof: It is easy to check that the two constructions define natural transformations. Moreover, they are each others inverses, as the calculations below for all sets X, $\alpha \in \mathrm{F}X$, and $f : A \to X$ show. Using $(*)$ naturality ξ we have

$$\begin{aligned}
\xi^{\nu^\xi}_X(\alpha) &= \nu^\xi_{A+X}(\iota_1, \mathrm{F}\iota_2(\alpha)) \\
&= (\mathrm{G}[\iota_1, \mathtt{id}_{A+X}] \circ \xi_{A+X} \circ \mathrm{F}\iota_2)(\alpha) \\
&\overset{(*)}{=} (\mathrm{G}[\iota_1, \mathtt{id}_{A+X}] \circ \mathrm{G}(\mathtt{id}_A + \iota_2) \circ \xi_X)(\alpha) \\
&= (\mathrm{G}\underbrace{[\iota_1, \iota_2]}_{=\mathtt{id}_{A+X}} \circ \xi_X)(\alpha) \\
&= \xi_X(\alpha).
\end{aligned}$$

With $(*)$ the naturality of ν we find

$$\begin{aligned}
\nu^{\xi^\nu}_X(f, \alpha) &= (\mathrm{G}[f, \mathtt{id}_X] \circ \xi^\nu_X)(\alpha) \\
&= (\mathrm{G}[f, \mathtt{id}_X] \circ \nu_{A+X})(\iota_1, \mathrm{F}\iota_2(\alpha)) \\
&\overset{(*)}{=} (\nu_X \circ ([f, \mathtt{id}_X]^A \times \mathrm{F}[f, \mathtt{id}_X]))(\iota_1, \mathrm{F}\iota_2(\alpha)) \\
&= \nu_X(\underbrace{([f, \mathtt{id}_X]^A)(\iota_1)}_{[f, \mathtt{id}_X] \circ \iota_1 = f}, (\mathrm{F}(\underbrace{[f, \mathtt{id}_X] \circ \iota_2}_{=\mathtt{id}_X}))(\alpha)) \\
&= \nu_X(f, \alpha).
\end{aligned}$$

□

Lemma A.6 *Let* C *and* D *be categories with* I*-indexed coproducts and products respectively and let* $\mathrm{F}^i, \mathrm{G} : \mathsf{C} \to \mathsf{D}$ *for* $i \in I$ *be functors. There is a one-to-one correspondence between natural transformations of the type*

$$\nu : \prod_{i \in I} \mathrm{F}^i \Rightarrow \mathrm{G} \quad \textit{and} \quad \xi_{(X_i)_{i \in I}} : \prod_{i \in I} \mathrm{F}^i X_i \Rightarrow \mathrm{G}(\coprod_{i \in I} X_i).$$

The correspondence is given by $\nu \mapsto \nu_\Lambda \circ \prod_{i \in I} \mathrm{F}^i \iota_i$ *and* $\xi \mapsto \mathrm{G}[\mathrm{Id}]_{i \in I} \circ \xi_\Delta$, *where* $\Delta : \mathsf{C} \to \mathsf{C}^I$ *is the diagonal functor mapping* X *to* $(X)_{i \in I}$ *and* $\Lambda : \mathsf{C}^I \to \mathsf{C}$ *is its left adjoint, i.e. the functor mapping the tuple* $(X_i)_{i \in I}$ *to the coproduct* $\coprod_{i \in I} X_i$.

More precisely, we should have written the natural transformation ξ as

$$\xi : \prod_{i \in I} \mathrm{F}^i \pi_\mathrm{i} \Rightarrow \mathrm{G}(\coprod_{i \in I} \pi_\mathrm{i}),$$

where $\pi_i : \mathsf{C}^I \to \mathsf{C}$ for $i \in I$ is the projection functor mapping $(X_j)_{j \in I}$ to X_i. We prefer the above notation since we deem it more readable.

Proof: The statement follows from the dual of Lemma A.1 when instantiated with Δ and its left adjoint, which exists by the assumption that C has I-indexed coproducts.

□

B. Simple statements about real valued functions

Below we present two facts about real valued functions that we used in the proof of Theorem 8.6.

Lemma B.1 *For $u \in \mathbb{R}_0^+$ let $f : [0,u] \to \mathbb{R}$ be a function with a bounded range satisfying*

$$f(r+s) = f(r) + f(s)$$

for all $r, s \in \mathbb{R}_0^+$ such that $r + s \in [0,u]$. Then for all $r \in [0,u]$ and $c \in [0,1]$ we find

$$f(c \cdot r) = c \cdot f(r).$$

Proof: By induction on p we easily get that for all $p \in \mathbb{N}$ and $r \in \mathbb{R}_0^+$ with $r, p \cdot r \in [0,u]$ we have $f(p \cdot r) = p \cdot f(r)$, which further implies $f(r/q) = f(r)/q$ for all $q \in \mathbb{N}$ with $q > 0$ and $r \in [0,u]$. So the statement is true for $c = p/q$, i.e. for rational c. For an arbitrary c choose a sequence of rational numbers $(c_n)_{n\in\mathbb{N}}$ with $c_n \leq c$ and $c_n \to c$ for $n \to \infty$. We calculate

$$c \cdot f(r) = (\lim_{n\to\infty} c_n) \cdot f(r) = \lim_{n\to\infty} (c_n \cdot f(r)) = \lim_{n\to\infty} f(c_n \cdot r) \overset{(*)}{=} f(c \cdot r).$$

For the step $(*)$ we instantiate the following calculation with $d_n = c_n \cdot r$ and $d = c \cdot r$: for any sequence $(d_n)_{n\in\mathbb{N}}$ and $d \in [0,u]$ with $d_n \to d$ for $n \to \infty$ and $d_n \leq d$ we have

$$f(d) = \lim_{n\to\infty} f\big(d_n + (d - d_n)\big) = \lim_{n\to\infty} \big(f(d_n) + f(d - d_n)\big) = \lim_{n\to\infty} f(d_n) + \underbrace{\lim_{n\to\infty} f(d - d_n)}_{=0}.$$

To see that the last addend is zero indeed, note that $d - d_n$ converges to zero. Now the identity follows from the general fact that $f(e_n) \to 0$ for $e_n \to 0$. This is because otherwise there exists $\varepsilon > 0$ such that arbitrary close to zero we can still find values $e \in \mathbb{R}_0^+$ with $\varepsilon < |f(e)|$, which contradicts our assumption on f being bounded. To see this, take any bound $b > 0$. Let $k = \lceil \frac{b}{\varepsilon} \rceil$ and choose $e \in [0, u/k]$ such that $\varepsilon < |f(e)|$. This implies $k \cdot e \in [0,u]$ and $b \leq k \cdot \varepsilon < k \cdot |f(e)| = |f(k \cdot e)|$.

□

Lemma B.2 *For a finite set M let $f : C \to \mathbb{R}$ be a function on a set $C \subseteq (\mathbb{R}_0^+)^M$ such that for all $i \in M$, $\vec{v} \in (\mathbb{R}_0^+)^M$, and $c \in [0,1]$ we have that $\vec{v} \in C$ implies $\vec{v}[i := c \cdot v_i] \in C$ and $f(\vec{v}[i := c \cdot v_i]) = c \cdot f(\vec{v})$. Then there exists $\tau \in \mathbb{R}$ with*

$$f(\vec{v}) = \tau \cdot \prod_{i\in M} v_i \quad \text{for all } \vec{v} \in C.$$

We bother to prove this rather obvious statement only because of the nonstandard domain restriction.

Proof: Choose $\vec{u} \in C$ such that $u_i > 0$ for all $i \in M$. (For all $\vec{u} \in C$ with $u_i = 0$ for some i the assumption easily implies $f(\vec{u}) = 0$, so there is nothing to show in case all $\vec{u} \in C$ have at least one zero component.) Set

$$\tau := \frac{f(\vec{u})}{\prod_{i\in M} u_i}.$$

For any $\vec{v} \in C$ with $I := \{i \in M \mid v_i > u_i\}$ by applying the assumption $|I|$ and $|M \setminus I|$ times respectively we get

$$(\prod_{i\in I} \frac{u_i}{v_i}) \cdot f(\vec{v}) = f(\min(\vec{u}, \vec{v})) = (\prod_{i\in M\setminus I} \frac{v_i}{u_i}) \cdot f(\vec{u})$$

where by $\min(\vec{u}, \vec{v})$ we denote the point-wise minimum of the two vectors. This implies

$$f(\vec{v}) = (\prod_{i \in M} \frac{v_i}{u_i}) \cdot f(\vec{u}) = \tau \cdot \prod_{i \in M} v_i.$$

We use the step via $\min(\vec{u}, \vec{v})$ to make sure that we do not run out of the domain of f on our way.

□

References

[AFV01] Luca Aceto, Wan Fokkink, and Chris Verhoef. Structural operational semantics. In Bergstra et al. [BPS01], pages 197–292.

[AM89] Peter Aczel and Nax Mendler. A final coalgebra theorem. In D.H. Pitt, D.E. Rydeheard, P. Dybjer, A.M. Pitts, and A. Poigné, editors, *Proc. 3*$^{\text{rd}}$ *CTCS*, volume 389 of *Lecture Notes in Computer Science*, pages 357–365. Springer-Verlag, Berlin, 1989.

[Bar02] Falk Bartels. GSOS for probabilistic transition systems (extended abstract). In Larry Moss, editor, *Proc. Coalgebraic Methods in Computer Science (CMCS 2002)*, volume 65 of *Electronic Notes in Theoretical Computer Science*. Elsevier Science Publishers, June 2002.

[Bar03] Falk Bartels. Generalised coinduction. *Mathematical Structures in Computer Science*, 13(1), 2003. To apppear, available via the author's homepage at `http://www.cwi.nl/~bartels/`.

[BIM95] Bard Bloom, Sorin Istrail, and Albert R. Meyer. Bisimulation can't be traced. *Journal of the ACM*, 42(1):232–268, January 1995.

[BPS01] Jan A. Bergstra, Alban Ponse, and Scott A. Smolka, editors. *Handbook of Process Algebra*. Elsevier, 2001.

[dVR99] Erik de Vink and Jan Rutten. Bisimulation for probabilistic transition systems: A coalgebraic approach. *Theoretical Computer Science*, 221, 1999.

[JLY01] Bengt Jonsson, Kim G. Larsen, and Wang Yi. Probabilistic extensions of process algebras. In Bergstra et al. [BPS01], pages 685–710.

[JR96] Bart Jacobs and Jan Rutten. A tutorial on (co)algebras and (co)induction. *Bulletin of the EATCS*, 62:222–259, 1996.

[Kic02a] Marco Kick. Bialgebraic modelling of timed processes. In P. Widmayer, F. Triguero, R. Morales, M. Hennessy, S. Eidenbenz, and R. Conejo, editors, *Proceedings ICALP'02*, volume 2380 of *Lecture Notes in Computer Science*. Springer Verlag, 2002. Also available from `http://www.dcs.ed.ac.uk/home/mk`.

[Kic02b] Marco Kick. Rule formats for timed processes. In *Proceedings CMCIM'02*, volume 68(1) of *Electronic Notes in Theoretical Computer Science*. Elsevier, 2002.

[LPW00] Marina Lenisa, John Power, and Hiroshi Watanabe. Distributivity for endofunctors, pointed and co-pointed endofunctors, monads and comonads. In Horst Reichel, editor, *Proc. Coalgebraic Methods in Computer Science (CMCS 2000)*, volume 33 of *Electronic Notes in The-*

oretical Computer Science, pages 233–263. Elsevier Science Publishers, 2000.

[LS91] Kim G. Larsen and Arne Skou. Bisimulation through probabilistic testing. *Information and Computation*, 94(1):1–28, September 1991.

[LS92] Kim G. Larsen and Arne Skou. Compositional verification of probabilistic processes. In W. R. Cleaveland, editor, *CONCUR '92: Third International Conference on Concurrency Theory*, volume 630 of *Lecture Notes in Computer Science*, pages 456–471, Stony Brook, New York, 1992. Springer-Verlag.

[Mos99] Lawrence S. Moss. Coalgebraic logic. *Annals of Pure and Applied Logic*, 96(1–3):277–317, 1999. Note that layout problems were corrected in volume 99 (1999).

[Rut00] Jan Rutten. Universal coalgebra: A theory of systems. *Theoretical Computer Science*, 249(1):3–80, October 2000.

[San98] Davide Sangiorgi. On the bisimulation proof method. *Journal of Mathematical Structures in Computer Science*, 8:447–479, 1998.

[TP97] Daniele Turi and Gordon D. Plotkin. Towards a mathematical operational semantics. In *Proc. 12*$^{\text{th}}$ *LICS Conf.*, pages 280–291. IEEE, Computer Society Press, 1997.

[Tur97] Daniele Turi. Categorical modelling of structural operational rules: case studies. In E. Moggi and G. Rosolini, editors, *Proc. 7*$^{\text{th}}$ *CTCS Conf.*, volume 1290 of *LNCS*, pages 127–146. Springer-Verlag, 1997.

[vGSS95] Rob J. van Glabbeek, Scott A. Smolka, and Bernhard Steffen. Reactive, generative and stratified models of probabilistic processes. *Information and Computation*, 121(1):59–80, August 1995.