Initial Algebra and Final Coalgebra Semantics for Concurrency

Jan Rutten and Daniele Turi

CWI
P.O. Box 94079, 1090 GB Amsterdam
({Jan.Rutten, Daniele.Turi}@cwi.nl)

Abstract. The aim of this paper is to relate *initial algebra semantics* and *final coalgebra semantics.* It is shown how these two approaches to the semantics of programming languages are each others dual, and some conditions are given under which they coincide. More precisely, it is shown how to derive initial semantics from final semantics, using the initiality and finality to ensure their equality. Moreover, many facts about *congruences* (on algebras) and (generalized) *bisimulations* (on coalgebras) are shown to be dual as well.

Keywords: Algebra, coalgebra, initiality, finality, Σ-algebra, labelled transition system, congruence, bisimulation, induction, coinduction.

Contents

Daniele Turi's research was supported by the *Stichting Informatica Onderzoek in Nederland* within the context of the project "Non-well-founded sets and semantics of programming languages". (Project no. 612-316-402.)

0 Introduction

The aim of this paper is to relate *initial algebra semantics* and *final coalgebra semantics.* It is shown how these two approaches to the semantics of programming languages are each others dual, and some conditions are given under which they coincide. More precisely, it is shown how to derive initial semantics from final semantics, using the initiality and finality to ensure their equality. Moreover, many facts about *congruences* (on algebras) and (generalized) *bisimulations* (on coalgebras) are shown to be dual as well.

Initial algebra semantics is a well-established technique in the study of programming languages, while final coalgebra semantics is a more recent one. In initial semantics, a meaning is assigned to programs in a *compositional* manner. In final semantics, the attention is rather focussed on describing the *observational* behavior of programs; once it has been decided what should be considered as observable, programs which are observationally equivalent are identified.

Initial semantics exploits the fact that the collection of terms of a given signature Σ forms an initial Σ-algebra. The semantics is determined by fixing another Σ-algebra as a semantic domain, in which the function symbols of Σ are interpreted. The semantic mapping of the terms into this semantic domain is canonically given by initiality, and is compositional with respect to the signature.

Likewise, final semantics exploits the fact that, given a notion of observation, say G, the elements of a final G-coalgebra are equal if and only if they are observationally equivalent (w.r.t. G). Once a (G-)coalgebra structure is given to the terms of the language, the semantic mapping is again canonically given, but now by finality instead of initiality. This semantic mapping has the property of identifying terms if and only if they are observationally equivalent. (The interest of observational equivalences for programming languages arose in connection with the study of concurrent and non-deterministic languages, where—in contrast with sequential (imperative) languages with their clear input/output behavior—various kinds of observations are meaningful.)

Let us give some further explanation. In initial semantics, an endofunctor Σ^* is associated with a signature Σ; then a Σ-algebra is simply a set X and a function

$$\alpha : \Sigma^*(X) \rightarrow X.$$

More generally, for any endofunctor F on an arbitrary category, an F-algebra is an object C and an arrow

$$\alpha : F(C) \rightarrow C.$$

Dually, a coalgebra of an endofunctor G is an object C and an arrow

$$\beta : C \rightarrow G(C).$$

Certain endofunctors are suitable for formalizing observations. In this paper, the following two functors are used. The (covariant) endofunctor $\mathcal{P}_f(A \times \text{-})$, which assigns to a set S the collection of all finite subsets of $A \times S$, is used to describe *strong bisimulation equivalence*. Its coalgebras are in one-to-one correspondence with (finitely branching) labelled transition systems, by viewing a transition relation R on $S \times A \times S$ (for a set S of states and a set A of labels) as a (non-deterministic) function

$$\beta : S \rightarrow \mathcal{P}_f(A \times S).$$

Secondly, the functor $1 + A \otimes^r \text{-}$, defined on the category of (join) semi-lattices is used to describe *trace equivalence*. (Here $A \otimes^r \text{-}$ denotes, for a given set A, the *right tensor* product, which will be introduced here.) The coalgebras of this functor correspond to a proper subclass of transition systems called *linear*, because they have a semi-lattice structure.

We shall formulate some general conditions under which an initial semantics can be derived from a final one. A crucial step in the construction is reminiscent of a technique used (although for apparently different reasons) in the semantics of the lambda calculus: the extension of the collection of terms (over a given signature) with the elements of the semantic domain, regarded as constants. Another important step will be made under the assumption that bisimulation is a congruence. As pointed out in [GV92], this can be ensured by considering only transition systems that are defined by means of a *transition system specification* ([Plo81b]), in which the axioms and rules are of a restricted syntactic format.

First an initial semantics for strong bisimulation is derived (using the functor $\mathcal{P}_f(A \times \text{-})$). Next the construction of an initial semantics from final semantics is formulated more generally for arbitrary categories and functors (the reverse direction is briefly discussed as well). Then it is applied to obtain semantics for trace equivalence (using the functor $1 + A \otimes^r \text{-}$).

The initial semantics, which is canonically constructed here, turns out to be—for certain specific signatures—the same as already existing denotational models. The initial semantics of the example in Section 3.6 coincides with (a variant of) a compositional model given in [BM88]. Similarly, the linear semantics of Section 5 is essentially the compositional model from [HP79]. In both papers, observational and compositional semantics are defined independently, and their equivalence is proved next using some fixed-point argument (in metric and ordered spaces, respectively). Interestingly, such fixed-point arguments are not needed here, but rather the uniqueness of initial and final arrows is exploited.

In Sections 1 and 2, the definitions and properties needed for the above construction are given in all detail. They are of some interest for their own sake, since the definitions and facts about coalgebras will simply be dual versions of similar definitions and facts about algebras. Let us mention the following examples. The definitions of congruence and (generalized) bisimulation are dual. Homomorphisms of algebras are precisely those functions whose graph is a congruence; dually, homomorphisms of coalgebras are those functions whose graph is a bisimulation. The kernel of a homomorphism of algebras is a congruence, and the kernel of a homomorphism of coalgebras (for most functors) is a bisimulation. As a last example, the equality relation on an initial algebra is the smallest congruence, yielding an *induction* principle. Dually, the equality relation on a final coalgebra is the greatest bisimulation, which can be seen as a *coinduction* principle.

Thus, since the world of algebras and in particular that of Σ-algebras has been intensively studied (think of universal algebra), these (and other) correspondences used in this paper pave the way for a more systematic exploitation of results about algebras in the study of coalgebras.

(Note that some care is needed, however. In particular, the duality between algebras and coalgebras involves the reversal of the arrow between an object and its image under the functor application. At the same time, the direction of the homomorphisms between algebras and coalgebras is the same (with respect to the underlying category).)

Further References and Related Work

Given its widespread use, references to initial semantics are probably superfluous. Let us just mention [GTW78] as an early reference, and [MG85] for an overview. For final semantics, instead, let us try to give a more detailed account.

One of the main features of final semantics is that it is independent of the specific way in which the semantic domains are constructed: It is defined in terms of their 'universal' properties only. Traditionally, semantic domains have been constructed in a recursive manner by using sets with some additional structure, like partial orders or metric spaces. (See, e.g., [SP82, Ken87, AR89]. See also [ArM82] for an early reference on final coalgebras of functors on sets.) A construction of semantic domains in terms of sets with no additional structure occurs in [Acz88]; however, a *non-standard* set theory is used in which sets may be non-well-founded. In the same book, the final coalgebra of a powerset functor appears as a model of this theory. Furthermore it is used for giving both an observational and a compositional semantics for the language *CCS*. (The observational semantics is with respect to strong bisimulation; for the compositional semantics, an adhoc method is used rather than a general methodology.) Later, in [AM89], more attention is given to final coalgebras in the category of (ordinary) sets. Moreover, the notion of (generalized) bisimulation of a functor is introduced there. In [Bar93], the results of [AM89] are expanded. (The existence of a final coalgebra of the functor $\mathcal{P}_f(A\times -)$ is proved in the present paper using a theorem from [Bar93].)

In our previous paper [RT93], a first step is made towards a generalization of the above notions to a, say universal, semantics based on final coalgebras. Properties of arbitrary categories of coalgebras are studied there, and the above mentioned approaches to the construction of semantic domains are put into a unifying framework.

Recently, in [TJ93], it has been shown how to express trace equivalence and applicative bisimulation (in the sense of [Abr90]) in terms of coalgebras. The former is the same as the final semantics for trace equivalence used in the present paper. The latter is a reformulation in terms of final semantics of Abramsky's observational semantics for his lazy lambda calculus; it is given in an 'order-enriched' setting. (See [Fio93, Rut93, Pit92] for related work in order-enriched categories.)

The idea of deriving compositional models from observational semantics based on transition system specifications is already described in [DG87] and [Bad87]. A more general construction is given in [Rut92], which is the starting point for the present paper; here we abstract from the specific observational equivalence used there (bisimulation) by means of final semantics. Moreover, we exploit the generality of this formulation for applying the same method also to trace equivalence.

How to Read this Paper

Sections and remarks marked by three stars, as in *Important****, are intended for second reading.

Sections 1, 2 and 3—with the exception of remarks and subsections marked by three stars—do not presuppose any knowledge of category theory. Everything is formulated in the category of sets and functions (at the price of omitting certain generalizations, which will be described elsewhere), and the exposition is, at many places, very concrete and detailed. Instead, Sections 4 and 5 use several constructions and results from category theory.

Readers with some categorical background might want to read the first two sections more quickly. They are also invited to make the generalizations that are left implicit there. Section 2.3 might be of particular interest since it, together with Section 2.4, provides a bridge between the construction of final coalgebras as given in [AM89] and the one in [Bar93]. Section 5 might be a good example of the generality of the final semantics approach.

1 Algebras

The well-known notion of *algebra of an (endo-)functor* is used to describe the familiar concept of Σ-algebras. There are two main reasons for choosing this somewhat abstract way of presentation here. Firstly, it will allow for a transparent and precise formulation and proof of our main constructions (in Sections 3 and 5). Secondly, the notion of algebra is dual to that of coalgebra, which—as we shall argue—is very suitable for describing transition systems and their properties. Also the notions of congruence and bisimulation will turn out to be each others dual. As a consequence, certain facts and proofs in the world of coalgebras are simply the dual versions of their counterparts in the dual world of algebras. Since in particular Σ-algebras are studied in a renowned field of research such as *universal algebra*—and in fact, the observations in this section will come as no surprise to anyone with some basic knowledge thereof—the exploitation in the present paper of the duality between algebras and coalgebras can furthermore be seen as a first step towards a more systematic exploitation of results about algebras in the world of coalgebras.

We shall work with the category *Set* consisting of sets and functions, and shall consider functors from this category to itself. (One might already from the beginning want to keep in mind, that almost all what follows equally well applies to arbitrary categories and functors. In Section 5, a different category will be considered.)

Let $F : Set \rightarrow Set$ be a functor. Such a functor is called an *endofunctor* on Set. It maps sets to sets, and functions between sets to functions between their images, in such a way that composition of functions is preserved and identity functions are mapped to identity functions.

Definition 1.1 An *F-algebra* is a pair (A, α), consisting of a set A and a function $\alpha : F(A) \rightarrow A$. A *homomorphism* $f : (A, \alpha) \rightarrow (B, \beta)$ *between F-algebras* (A, α) and (B, β) is a function $f : A \rightarrow B$ satisfying $f \circ \alpha = \beta \circ F(f)$:

$$\begin{tikzcd}
F(A) \arrow[r, "F(f)"] \arrow[d, "\alpha"'] \arrow[dr, phantom, "*"] & F(B) \arrow[d, "\beta"] \\
A \arrow[r, "f"'] & B
\end{tikzcd}$$

(Such an f will also be called an F-homomorphism.) Composition of two homomorphisms f and g between F-algebras is defined by $g \circ f$, the function composition of f and g, and yields again a homomorphism. The collection Set^F of F-algebras and algebra homomorphisms constitutes a category. □

An *initial* object in a category is an object A such that for any other object B there exists a unique arrow from A to B. Thus an F-algebra (A, α) is initial if for any other F-algebra (B, β) there exists a *unique* homomorphism $\mathcal{I} : (A, \alpha) \rightarrow (B, \beta)$.

The following result is classical.

Theorem 1.2 *Initial F-algebras (A, α) are fixed points of F; that is, $\alpha : F(A) \rightarrow A$ is an isomorphism.*

Proof: Let (A, α) be an initial F-algebra. Then also $(F(F(A)), F(\alpha))$ is an F-algebra and by initiality of (A, α), there exists a homomorphism $\mathcal{I} : (A, \alpha) \rightarrow (F(F(A)), F(\alpha))$,

$$\begin{tikzcd}
F(A) \arrow[r, "F(\mathcal{I})"] \arrow[d, "\alpha"'] \arrow[dr, phantom, "*"] & F(F(A)) \arrow[d, "F(\alpha)"] \\
A \arrow[r, "\mathcal{I}"'] & F(A)
\end{tikzcd}$$

Since the following diagram commutes trivially,

$$\begin{tikzcd}
F(F(A)) \arrow[r, "F(\alpha)"] \arrow[d, "F(\alpha)"'] \arrow[dr, phantom, "*"] & F(A) \arrow[d, "\alpha"] \\
F(A) \arrow[r, "\alpha"'] & A,
\end{tikzcd}$$

it follows that $\alpha \circ \mathcal{I} : (A, \alpha) \rightarrow (A, \alpha)$ is a homomorphism. Because also 1_A is a homomorphism from (A, α) to itself, it follows by the initiality of (A, α) that $1_A = \alpha \circ \mathcal{I}$. Moreover,

$$\begin{aligned} \mathcal{I} \circ \alpha &= F(\alpha) \circ F(\mathcal{I}) \\ &= F(\alpha \circ \mathcal{I}) \\ &= F(1_A) \\ &= 1_{F(A)}, \end{aligned}$$

showing that α is an isomorphism. □

If one sees categories and functors as generalizations of preordered sets and monotone functions, algebras of a functor correspond to pre-fixed points. Initial algebras correspond to least fixed points.

Next the notion of a congruence is introduced (cf. [Man76]).

Definition 1.3 An *F-congruence* between two F-algebras (A, α) and (B, β) is a relation R on $A \times B$ such that it can be extended to a subalgebra of the product of (A, α) and (B, β). That is, R is an F-algebra (R, γ), with $\gamma : F(R) \to R$ as given below, such that its projections $\pi_1 : R \to A$ and $\pi_2 : R \to B$ (defined by $\pi_1((a,b)) = a$ and $\pi_2((a,b)) = b$, for $(a,b) \in A \times B$) are homomorphisms of F-algebras:

$$\begin{tikzcd} F(A) \arrow[d, "\alpha"'] & F(R) \arrow[l, "F(\pi_1)"'] \arrow[r, "F(\pi_2)"] \arrow[d, "\gamma"] \arrow[dl, phantom, "*"] \arrow[dr, phantom, "*"] & F(B) \arrow[d, "\beta"] \\ A & R \arrow[l, "\pi_1"] \arrow[r, "\pi_2"'] & B \end{tikzcd}$$

Note that there is at most one such function γ making the diagram above commute: It is determined by the requirement that for any $x \in F(R)$,

$$\gamma(x) = \langle \alpha \circ F(\pi_1)(x), \beta \circ F(\pi_2)(x) \rangle.$$

□

Note that the above definition of congruence does not require R to be an equivalence, as opposed to the standard definition of a congruence on a Σ-algebra (see Example 1.4).

1.1 Σ-Algebras

In this section, we shall first define, for a given signature Σ, a functor Σ^* on *Set*. Then the definitions (of algebra and congruence) given above for arbitrary endofunctors on *Set* will be applied to Σ^*. We shall see that the resulting Σ^*-algebras and Σ^*-homomorphisms are precisely the familiar Σ-algebras and the homomorphisms between Σ-algebras. Further, a congruence on a Σ-algebra will turn out to be an instance of the (slightly more general) notion of Σ^*-congruence. Finally, the usual Σ-algebra of (closed) terms is shown to be an initial Σ^*-algebra.

Let (Σ, r) be a single-sorted signature (ranked alphabet), consisting of a set Σ of function symbols, and a ranking function $r : \Sigma \to N$, which assigns to each function symbol $f \in \Sigma$ a natural number $r(f)$, called the rank (or arity) of f. The functor $\Sigma^* : Set \to Set$ is a functor defined as follows. For a set X,

$$\Sigma^*(X) \equiv \coprod_{f \in \Sigma} X^{r(f)},$$

where $\coprod$ denotes the disjoint sum (coproduct), X^0 is a singleton set (final object) $1 = \{*\}$ and

$$X^k = \{\langle x_1, \dots, x_k\rangle \mid x_1 \in X, \dots, x_k \in X\},$$

if $k > 0$. (Note that here and in the sequel, the symbol $\equiv$ is used as "defining equality", meaning that the left side is being defined, whereas the right side is assumed to be already known.) Equivalently,

$$\Sigma^*(X) = \bigcup_{f\in\Sigma} \{f\} \times X^{r(f)}.$$

(Elements $(f, \langle x_1, \dots, x_{r(f)}\rangle)$ of $\{f\} \times X^{r(f)}$ will be denoted by $f\langle x_1, \dots, x_{r(f)}\rangle$.) A function $h : X \to Y$ is mapped by Σ^* to the function $\Sigma^*(h) : \Sigma^*(X) \to \Sigma^*(Y)$, which is defined, for any $f\langle x_1, \dots, x_{r(f)}\rangle \in \Sigma^*(X)$, by

$$\Sigma^*(h)(f\langle x_1, \dots, x_{r(f)}\rangle) \equiv f\langle h(x_1), \dots, h(x_{r(f)})\rangle.$$

Consider a Σ^*-algebra (X, α). Since the function $\alpha : \Sigma^*(X) \to X$ has a (disjoint) sum as domain, the restriction of α to each of the components of this sum determines a family of functions

$$\{f_X : X^{r(f)} \to X \mid f \in \Sigma\},$$

by putting, for any $f \in \Sigma$ and $\langle x_1, \dots, x_{r(f)}\rangle \in X^{r(f)}$,

$$f_X(\langle x_1, \dots, x_{r(f)}\rangle) \equiv \alpha(f\langle x_1, \dots, x_{r(f)}\rangle).$$

(Note that the argument of f_X is an element of $X^{r(f)}$, whereas the argument of α is an element of $\{f\} \times X^{r(f)}$.) Conversely, reading this definition right to left shows that any such family determines a function α from $\Sigma^*(X)$ to X. Thus the Σ^*-algebras are precisely the usual Σ-algebras.

Let (X, α) and (Y, β) be two Σ^*-algebras. A function $h : X \to Y$ is a homomorphism of Σ^*-algebras:

$$\begin{tikzcd}
\Sigma^*(X) \arrow[r, "\Sigma^*(h)"] \arrow[d, "\alpha"'] \arrow[dr, phantom, "*"] & \Sigma^*(Y) \arrow[d, "\beta"] \\
X \arrow[r, "h"'] & Y
\end{tikzcd}$$

if and only if, for any $f \in \Sigma$ and $\langle x_1, \dots, x_{r(f)}\rangle \in X^{r(f)}$,

$$h(f_X(\langle x_1, \dots, x_{r(f)}\rangle)) = f_Y(\langle h(x_1), \dots, h(x_{r(f)})\rangle).$$

This follows from the observation that in the following sequence of equations,

$$\begin{aligned}
h(f_X(\langle x_1, \dots, x_{r(f)}\rangle)) &= h(\alpha(f\langle x_1, \dots, x_{r(f)}\rangle)) \\
&= \beta(\Sigma^*(h)(f\langle x_1, \dots, x_{r(f)}\rangle)) \\
&= \beta(f\langle h(x_1), \dots, h(x_{r(f)})\rangle) \\
&= f_Y(\langle h(x_1), \dots, h(x_{r(f)})\rangle),
\end{aligned}$$

the second equality holds if and only if h is a Σ^*-homomorphism (and the other equalities hold always). Since a homomorphism of Σ-algebras is usually defined as a function h satisfying

$$h(f_X(\langle x_1, \dots, x_{r(f)}\rangle)) = f_Y(\langle h(x_1), \dots, h(x_{r(f)})\rangle),$$

it follows that the notions of Σ^*-homomorphism and Σ-homomorphism coincide.

1.1.1 Σ^*-Congruences

The notion of Σ^*-congruence generalizes the standard notion of a congruence: Let (X, α) be a Σ^*-algebra, and let $R \subseteq X \times X$ be a relation on X. It is a simple exercise to verify that R can be extended (in a unique way) to a Σ^*-congruence (R, γ):

$$\begin{tikzcd}
\Sigma^*(X) \arrow[d, "\alpha"'] \arrow[dr, phantom, "*"] & \Sigma^*(R) \arrow[l, "\Sigma^*(\pi_1)"'] \arrow[r, "\Sigma^*(\pi_2)"] \arrow[d, "\gamma"] \arrow[dr, phantom, "*"] & \Sigma^*(X) \arrow[d, "\alpha"] \\
X & R \arrow[l, "\pi_1"] \arrow[r, "\pi_2"'] & X
\end{tikzcd}$$

if and only if R is a congruence relation in the traditional sense; that is, for all $f \in \Sigma$ and sequences $\langle x_1, \ldots, x_{r(f)} \rangle$ and $\langle x'_1, \ldots, x'_{r(f)} \rangle$ in $X^{r(f)}$,

if $(x_i, x'_i) \in R$, for $i \in \{1, \ldots, r(f)\}$,

then $(f_X(\langle x_1, \ldots, x_{r(f)} \rangle), f_X(\langle x'_1, \ldots, x'_{r(f)} \rangle)) \in R$.

Example 1.4 Consider the signature $\{s, 0\}$, consisting of a unary function symbol s (for successor) and a constant 0. Let $N = \{0, 1, 2, \ldots\}$ be the set of natural numbers, and let $\phi : N + 1 \to N$ be defined by $\phi(n) \equiv n + 1$, for $n \in N$ and $\phi(*) \equiv 0$. Then (N, ϕ) is an example of a $\{s, 0\}^*$-algebra.

As an example of a congruence relation on N, consider the set

$$E \equiv \{(n, m) \in N \times N \mid n + m \text{ is even } \},$$

together with the function $\gamma : E + 1 \to E$ mapping (n, m) in E to $(n + 1, m + 1)$, again in E, and $*$ to $(0, 0)$.

The following example shows that a congruence need not always be an equivalence relation. Let (X, α) be the $\{s, 0\}^*$-algebra consisting of a three element set $X = \{x, y, z\}$ and the function $\alpha : (X + 1) \to X$ that is the identity on X and maps $*$ to x. Next consider the relation $R = \{(x, x), (x, y), (y, z)\}$ on X, which forms a $\{s, 0\}^*$-congruence on (X, α) together with the function $\gamma : (R + 1) \to R$, which is defined as the identity on R and maps $*$ to (x, x). Now R is an example of a congruence relation that is neither reflexive, nor symmetric, nor transitive. □

Definition 1.5 The *kernel* of a function $h : X \to Y$ is the set

$$K_h \equiv \{(x, x') \in X \times X \mid h(x) = h(x')\}.$$

The *graph* G_h of h is defined as

$$G_h \equiv \{(x, y) \in X \times Y \mid h(x) = y\}.$$

The next two propositions state that the kernel and the graph of a Σ^*-homomorphism are congruences. In fact they hold for arbitrary functors. As we shall see in Section 2, they have a dual counterpart in the world of coalgebras, where the kernel and the graph of a homomorphism of *coalgebras* is a bisimulation (Propositions 2.7 and 2.8).

Proposition 1.6 *Let (X, α) and (Y, β) be two Σ^*-algebras, and let $h : X \to Y$ be a function. If h is a Σ^*-homomorphism then its kernel K_h is a Σ^*-congruence.* □

A proof of this proposition, and an example showing that its converse is false, are easily found. For the latter, consider the algebra (N, ϕ) of the natural numbers (see Example 1.4), and the function $h : N \rightarrow N$ that takes $n \in N$ to $n+1$: the kernel of h is the identity relation on N, which is always a congruence, but h is not a homomorphism from (N, ϕ) to itself.

In this respect, graphs are better behaved.

Proposition 1.7 *A function $h : X \rightarrow Y$ is a homomorphism of Σ^*-algebras (X, α) and (Y, β) if and only if its graph G_h is a Σ^*-congruence (G_h, γ) between (X, α) and (Y, β).*

Proof: Let (X, α) and (Y, β) be two Σ^*-algebras, and let $h : X \rightarrow Y$ be a function. Define a function γ on $\Sigma^*(G_h)$, for any $f\langle(x_1, h(x_1)), \ldots, (x_{r(f)}, h(x_{r(f)}))\rangle$ in $\Sigma^*(G_h)$, by

$$\gamma(f\langle(x_1, h(x_1)), \ldots, (x_{r(f)}, h(x_{r(f)}))\rangle) \equiv$$

$$\langle\alpha(f\langle x_1, \ldots, x_{r(f)}\rangle), \; \beta(f\langle h(x_1), \ldots, h(x_{r(f)})\rangle)\rangle.$$

Note that the righthand side of the above equation is an element of G_h if and only if

$$h(\alpha(f\langle x_1, \ldots, x_{r(f)}\rangle)) = \beta(f\langle h(x_1), \ldots, h(x_{r(f)})\rangle).$$

Since the latter term is equal to

$$\beta(\Sigma^*(h)(f\langle x_1, \ldots, x_{r(f)}\rangle)),$$

it follows that (G_h, γ) is a congruence if and only if h is a Σ^*-homomorphism: $h \circ \alpha = \beta \circ \Sigma^*(h)$. □

1.1.2 Initial Semantics

An *initial* Σ^**-algebra* (T, ϕ) is given by the usual free construction of terms over Σ: the set T can be constructed as the union of a sequence of sets $(T_n)_n$ given by $T_0 \equiv \emptyset$ and for $n \geq 0$,

$$T_{n+1} \equiv \{f(t_1, \ldots, t_{r(f)}) \mid f \in \Sigma \text{ and } t_i \in T_n \text{ for } i = 1, \ldots, r(f)\}.$$

(If $r(f) = 0$ then $f(t_1, \ldots, t_0)$ should be read as f. This implies that the set T_n, for any n contains all constants. As a consequence, it can be proved inductively that T_n is contained in T_{n+1}.) The function $\phi : \Sigma^*(T) \rightarrow T$ is defined, for any $f \in \Sigma$ and $f\langle t_1, \ldots, t_{r(f)}\rangle \in \{f\} \times T^{r(f)}$, by

$$\phi(f\langle t_1, \ldots, t_{r(f)}\rangle) \equiv f(t_1, \ldots, t_{r(f)}).$$

Note that ϕ is indeed an isomorphism. Similarly, the set T_X of terms over Σ with variables in a given set X, is obtained as the union of a sequence $(V_n)_n$ with $V_0 = \emptyset$ and, for $n \geq 0$,

$$V_{n+1} = X \cup \{f(t_1, \ldots, t_{r(f)}) \mid f \in \Sigma \text{ and } t_i \in V_n \text{ for } i = 1, \ldots, r(f)\}.$$

The set T_X is an initial algebra of the functor $\Sigma^*_X : Set \rightarrow Set$ defined, on sets S, by

$$\Sigma^*_X(S) \equiv X + \Sigma^*(S).$$

Now consider an arbitrary Σ^*-algebra (X, α). By the initiality of (T, ϕ) there exists a unique homomorphism $\mathcal{I} : (T, \phi) \rightarrow (X, \alpha)$,

$$\begin{tikzcd}
\Sigma^*(T) \arrow[r, "\Sigma^*(\mathcal{I})"] \arrow[d, "\phi"'] \arrow[dr, phantom, "*"] & \Sigma^*(X) \arrow[d, "\alpha"] \\
T \arrow[r, "\mathcal{I}"'] & X,
\end{tikzcd}$$

called the *initial semantics* for X. By the commutativity of the diagram above, it is *compositional*, satisfying for all $f \in \Sigma$ and $\langle t_1, \ldots, t_{r(f)} \rangle$ in $T^{r(f)}$,

$$\mathcal{I}(f(t_1, \ldots, t_{r(f)})) = f_X(\mathcal{I}(t_1), \ldots, \mathcal{I}(t_{r(f)})).$$

Example 1.4, continued: The construction of an initial $\{s, 0\}^*$-algebra yields the set of terms

$$I = \{*, s(*), s(s(*)), \ldots\},$$

together with a function ι from $\Sigma^*(I)$ to I. Clearly, this initial algebra (I, ι) is isomorphic to (N, ϕ), which therefore is initial as well. (In general, many initial algebras may exist but they are all isomorphic.)

For a simple example of initial semantics, consider the $\{s, 0\}^*$-algebra (A, α) given by $A \equiv \{o, e\}$ (for *odd* and *even*), and $\alpha : A + 1 \rightarrow A$ mapping o, e, and $*$ to e, o, and e, respectively. The initial semantics $\mathcal{I} : N \rightarrow A$ then maps even natural numbers to e and odd natural numbers to o.

□

1.2 Smallest Congruences and Induction***

Let $F : Set \rightarrow Set$ be a functor. Let (A, α) be an initial F-algebra and let (R, γ) be an F-congruence on (A, α), with projections $\pi_1, \pi_2 : (R, \gamma) \rightarrow (A, \alpha)$. By the initiality of (A, α) there exists a (unique) homomorphism $i : (A, \alpha) \rightarrow (R, \gamma)$. Again by initiality, $\pi_1 \circ i = 1_A = \pi_2 \circ i$. This implies, for any $a \in A$ that $i(a) = \langle a, a \rangle$ is in R. Thus the equality relation $=_A$ on A is contained in R. Since $=_A$ itself is an F-congruence on (A, α), we have proved the following theorem. It is dual to Theorem 2.4, which states that the equality relation $=_A$ on a final F-coalgebra (A, α) is the greatest F-bisimulation on (A, α).

Theorem 1.8 *For an initial F-algebra (A, α), the equality relation $=_A$ on A is the smallest F-congruence,*

$$=_A = \bigcap \{R \subseteq A \times A \mid R \textit{ is an } F\textit{-congruence on } (A, \alpha)\ \}.$$

This theorem can be interpreted as a principle of *induction*, as is illustrated by the following.

Example 1.4, continued: Applying Theorem 1.8 to the initial $\{s, 0\}^*$-algebra N, the natural numbers, yields: for every $R \subseteq N \times N$ such that $\langle 0, 0 \rangle \in R$ and such that, for all $\langle m, n \rangle \in N \times N$, if $\langle m, n \rangle \in R$ then $\langle m+1, n+1 \rangle \in R$, we have $=_N \subseteq R$. It is easy to see that this is equivalent to the well-known principle of mathematical induction: for all $P \subseteq N$,

$$\text{if } 0 \in P \text{ and } (\forall n \in N,\ n \in P \Rightarrow n+1 \in P) \text{ then } P = N.$$

2 Coalgebras

A coalgebra of a functor is defined as the dual of an algebra. Labelled transition systems correspond to the coalgebras of a certain functor (because of a well-known bijection between relations and non-deterministic functions). Similarly, bisimulations are coalgebras ([AM89]). (See also [Ken87] for an early reference.) In [RT93], these ideas are further developed and systematically employed in giving semantics (called *final* semantics since it is based on the notion of final coalgebra) to (generalized) transition systems. In [TJ93], this framework is applied to linear semantics (which will be treated in Section 5) and to the lazy lambda calculus.

In this section, the main definitions and theorems of [RT93] are recalled, now formulated for the category *Set*. (But again, most of it applies to arbitrary categories.) Furthermore, the category of coalgebras of one particular functor on sets is investigated in great detail. All of the definitions and theorems, formulated for arbitrary functors on *Set*, are next instantiated for this functor, yielding familiar notions. In particular, the coalgebras of this functor exactly correspond to the standard labelled transition systems. Many properties of such systems—some new, some already known—are formulated and proved in an elegant way, by using some basic properties of coalgebras and coalgebra homomorphisms.

Consider a functor $F : Set \rightarrow Set$.

Definition 2.1 An *F-coalgebra* is a pair (A, α), consisting of a set A and a function $\alpha : A \rightarrow F(A)$.

A *homomorphism of F-coalgebras* $f : (A, \alpha) \rightarrow (B, \beta)$ (or F-homomorphism) is a function $f : A \rightarrow B$ satisfying $F(f) \circ \alpha = \beta \circ f$:

$$\begin{tikzcd}
A \arrow[r, "f"] \arrow[d, "\alpha"'] \arrow[dr, phantom, "*"] & B \arrow[d, "\beta"] \\
F(A) \arrow[r, "F(f)"'] & F(B)
\end{tikzcd}$$

Composition of two homomorphisms f and g between between F-coalgebras is defined by $g \circ f$, and yields again a homomorphism. The collection Set_F of F-coalgebras and coalgebra homomorphisms constitutes a category. □

Definition 2.2 An *F-bisimulation* between two F-coalgebras (A, α) and (B, β) is a relation $R \subseteq A \times B$ that can be extended to an F-coalgebra (R, γ), for some $\gamma : R \rightarrow F(R)$, such that its projections $\pi_1 : R \rightarrow A$ and $\pi_2 : R \rightarrow B$ are homomorphisms of F-coalgebras:

$$\begin{tikzcd}
A \arrow[d, "\alpha"'] & R \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] \arrow[d, "\gamma"] \arrow[dl, phantom, "*"] \arrow[dr, phantom, "*"] & B \arrow[d, "\beta"] \\
F(A) & F(R) \arrow[l, "F(\pi_1)"] \arrow[r, "F(\pi_2)"'] & F(B)
\end{tikzcd}$$

Note that in general there may be more than one such function γ. □

An F-coalgebra (A,α) is *final* if for any other F-coalgebra (B,β) there exists a *unique* homomorphism $f : (B,\beta) \to (A,\alpha)$. It is *weakly final* if there exists at least one such homomorphism.

2.1 Basic Facts

The following theorem is the dual of Theorem 1.2.

Theorem 2.3 *Final F-coalgebras (A,α) are fixed points of F; that is, $\alpha : A \to F(A)$ is an isomorphism.* □

Final F-coalgebras are of particular interest because of the following property. For any F-coalgebra (A,α), let $\sim_A$ be defined as the union of its F-bisimulations:

$$\sim_A \equiv \bigcup\{R \subseteq A \times A \mid R \text{ is an } F\text{-bisimulation on } (A,\alpha)\}.$$

Two elements a and a' in A with $a \sim_A a'$ are called *(F-)bisimilar*. (For most functors, $\sim_A$ is itself a bisimulation relation. A sufficient condition is that F weakly preserves kernels. Cf. Remark 2.5 and Section 2.3.) The following theorem (from [RT93]) is easily derived from a similar result in [AM89].

Theorem 2.4 *A final F-coalgebra (A,α) is* strongly extensional*: For all $a, a' \in A$,*

$$\text{if } a \sim_A a' \text{ then } a = a'.$$

Since the equality relation $=_A$ on any F-coalgebra (A,α) can be readily seen to be a bisimulation, this implies $=_A = \sim_A$; that is,

$$=_A = \bigcup\{R \subseteq A \times A \mid R \text{ is an } F\text{-bisimulation on } (A,\alpha)\}.$$

Proof: Immediate from the fact that the two projections $\pi_1, \pi_2 : (R,\gamma) \to (A,\alpha)$, of any F-bisimulation (R,γ) on (A,α), are equal by the finality of (A,α). □

Note that the above theorem is dual to Theorem 1.8. It can be seen as a proof principle—called the principle of *coinduction*: in order to prove the equality of two elements, it suffices to establish the existence of a bisimulation between them.

Recall that the kernel of a homomorphism between Σ^*-algebras is a congruence (Proposition 1.6). To prove the dual fact that the kernel of a homomorphism of F-coalgebras is an F-bisimulation, a condition on the functor F is needed. A sufficient condition is that F weakly preserves kernels. All familiar functors, which are defined using constants, products, sums and powerset constructions, satisfy this condition. In particular, all functors used in this paper do.

Remark 2.5 *** *The functor F* weakly preserves kernels *if $K_{F(f)}$ can be injectively mapped into $F(K_f)$. If $K_{F(f)} \cong F(K_f)$ then F is said to* preserve kernels. *It is not difficult to show that the function γ defined for $(a,a') \in K_f$ by $\gamma(a,a') \equiv (\alpha(a),\alpha(a'))$, maps into $K_{F(f)}$. If F weakly preserves kernels this actually defines a function into $F(K_f)$, and (K_f,γ) is a bisimulation on (A,α). A more general, categorical fact underlying this observation is that for endofunctors F (on a category C) that preserve pullbacks, the forgetful functor from C_F to C creates pullbacks.* □

The fact that the kernel of an F-coalgebra homomorphism is a bisimulation (for functors F that weakly preserve kernels), is used to prove the following.

Theorem 2.6 *([RT93]) Let F weakly preserve kernels. Let (A,α) be a final F-coalgebra and (B,β) be any F-coalgebra. Let $\mathcal{F}$ be the unique homomorphism from (B,β) to (A,α). For all $b,b' \in B$,*

$$b \sim_B b' \text{ if and only if } \mathcal{F}(b) = \mathcal{F}(b').$$

Proof: The implication from left to right follows from the fact that for an F-bisimulation (R,γ) on (B,β), $\mathcal{F} \circ \pi_1, \mathcal{F} \circ \pi_2 : (R,\gamma) \to (A,\alpha)$ both are homomorphisms to the final F-coalgebra (A,α). The converse is immediate from the assumption that F weakly preserves kernels, by which $K_{\mathcal{F}}$ is an F-bisimulation on (B,β). □

2.2 Labelled Transition Systems

In this section, the above definitions and theorems will be applied to one particular functor. We shall see that its coalgebras correspond to labelled transition systems, and that the definition of F-bisimulation yields the familiar notion of strong bisimulation.

Let A be a given (possibly infinite) set. Let $\mathcal{P}(A \times \cdot) : Set \to Set$ be the functor defined, on sets S, by

$$\mathcal{P}(A \times S) \equiv \{V \subseteq A \times S\};$$

$\mathcal{P}(A \times \cdot)$ maps a function $f : S \to T$ to the function $\mathcal{P}(A \times f) : \mathcal{P}(A \times S) \to \mathcal{P}(A \times T)$, which is defined, for any $V \in \mathcal{P}(A \times S)$, by

$$\mathcal{P}(A \times f)(V) \equiv \{\langle a, f(s)\rangle \in A \times T \mid \langle a, s\rangle \in V\}.$$

The coalgebras of this functor are in one-to-one correspondence with *labelled transition systems* over A: that is, triples $\langle S, A, \rightarrow\rangle$ consisting of a set S of states, the set A of labels, and a transition relation $\rightarrow \subseteq S \times A \times S$. (As usual, we write $s \xrightarrow{a} s'$ for $\langle s, a, s'\rangle \in \rightarrow$.) For to any $\mathcal{P}(A \times \cdot)$-coalgebra (S,α), a labelled transition system $\langle S, A, \rightarrow\rangle$ can be assigned (one-to-one) by putting, for $s, s' \in S$ and $a \in A$,

$$s \xrightarrow{a} s' \Leftrightarrow \langle a, s'\rangle \in \alpha(s).$$

2.2.1 Bisimulation

The $\mathcal{P}(A \times \cdot)$-bisimulations between two $\mathcal{P}(A \times \cdot)$-coalgebras (S,α) and (T,β) are precisely the usual strong bisimulations between transition systems ([Par81, Mil89]): a relation $R \subseteq S \times T$ can be extended (not necessarily uniquely) to a $\mathcal{P}(A \times \cdot)$-bisimulation (R,γ),

$$\begin{tikzcd}
S \arrow[d, "\alpha"'] & & R \arrow[ll, "\pi_1"'] \arrow[rr, "\pi_2"] \arrow[d, "\gamma"] & & T \arrow[d, "\beta"] \\
\mathcal{P}(A \times S) & * & \mathcal{P}(A \times R) \arrow[ll, "\mathcal{P}(A \times \pi_1)"] \arrow[rr, "\mathcal{P}(A \times \pi_2)"'] & * & \mathcal{P}(A \times T)
\end{tikzcd}$$

if and only if R is a strong bisimulation between S and T, seen as transition systems; that is, for all $s \in S$ and $t \in T$ with $(s,t) \in R$:

(1) if $s \xrightarrow{a} s'$, for some $s' \in S$,
then $t \xrightarrow{a} t'$ for some $t' \in T$ with $(s',t') \in R$;

(2) if $t \xrightarrow{a} t'$, for some $t' \in T$,
then $s \xrightarrow{a} s'$ for some $s' \in S$ with $(s', t') \in R$.

The implication from left to right follows from the observation that the commutativity of the left and right squares in the above diagram implies conditions (1) and (2), respectively.

For the converse, consider a bisimulation relation R on $S \times T$ satisfying clauses (1) and (2). As is well known, a bisimulation relation can be turned into a transition system by defining, for (s,t) and (s',t') in R,

$$(s,t) \xrightarrow{a} (s',t') \equiv s \xrightarrow{a} s' \text{ and } t \xrightarrow{a} t'.$$

This transition system, which has the set R for its set of states, can be turned into a $\mathcal{P}(A \times \cdot)$-bisimulation (R, γ) in the canonical way described at the beginning of Section 2.2: define $\gamma : R \to \mathcal{P}(A \times R)$, for $(s,t) \in R$, by

$$\gamma((s,t)) \equiv \{\langle a, (s',t')\rangle \mid (s,t) \xrightarrow{a} (s',t') \text{ (and } (s',t') \in R\text{)}\}.$$

It follows from (1) and (2) that (R, γ) is a $\mathcal{P}(A \times \cdot)$-bisimulation, that is, that $\pi_1 : (R, \gamma) \to (S, \alpha)$ and $\pi_2 : (R, \gamma) \to (T, \beta)$ are homomorphisms.

Now that we have seen the correspondence between strong bisimulations and $\mathcal{P}(A \times \cdot)$-bisimulations, the two notions will be used in what follows interchangeably.

The following two propositions are the duals of Propositions 1.6 and 1.7. They are formulated for the functor $\mathcal{P}(A \times \cdot)$ but also hold for arbitrary functors (that weakly preserve kernels).

Proposition 2.7 *Let (S, α) and (T, β) be two $\mathcal{P}(A \times \cdot)$-coalgebras, and let $f : S \to T$ be a function. If f is a $\mathcal{P}(A \times \cdot)$-homomorphism then its kernel K_f is a $\mathcal{P}(A \times \cdot)$-bisimulation.* □

The proof is easy and therefore omitted. Its converse does not hold (and a counter example is again easily found).

Proposition 2.8 *A function $f : S \to T$ is a homomorphism of coalgebras from (S, α) to (T, β) if and only if its graph G_f is a bisimulation.*

Proof: First note that the graph G_f is a bisimulation if and only if the following two conditions are satisfied: for all $s \in S$,

(1) if $s \xrightarrow{a} s'$, for some $s' \in S$, then $f(s) \xrightarrow{a} f(s')$;

(2) if $f(s) \xrightarrow{a} t$, for some $t \in T$, then there exists $s' \in S$ with

$s \xrightarrow{a} s'$ and $f(s') = t$.

Further note that, by definition, the function f is a homomorphism if and only if the following diagram commutes,

$$\begin{tikzcd}
S \arrow[r, "f"] \arrow[d, "\alpha"'] \arrow[dr, phantom, "*"] & T \arrow[d, "\beta"] \\
\mathcal{P}(A \times S) \arrow[r, "\mathcal{P}(A \times f)"'] & \mathcal{P}(A \times T)
\end{tikzcd}$$

which is equivalent to the equality of the following two sets, for every $s \in S$:

$$\beta \circ f(s) = \{\langle a, t\rangle \in A \times T \mid f(s) \xrightarrow{a} s'\},$$

$$\mathcal{P}(A{\times}f) \circ \alpha(s) = \{\langle a, f(s)\rangle \in A \times T \mid s \xrightarrow{a} s'\}.$$

(Recall the correspondence between transition relations and coalgebras.) Now the theorem follows from the observation that conditions (1) and (2) above are equivalent to the inclusions of $\mathcal{P}(A{\times}f) \circ \alpha(s)$ in $\beta \circ f(s)$, and of $\beta \circ f(s)$ in $\mathcal{P}(A{\times}f) \circ \alpha(s)$, respectively. □

A function $f : S \to T$ satisfying condition (1) is sometimes called a *morphism of transition systems*. For a category of labelled transition systems, in which—a variant of—such morphisms are taken as the arrows, see [WN93]. If f satisfies both (1) and (2) it is sometimes called a *saturating morphism* ([AD89]). Bisimulations like G_f are called *functional.*

Remark 2.9 **** Both Proposition 2.8 and Proposition 1.7 could be given simple categorical proofs by viewing the graph of a function $f : S \to T$ as the pullback of f with the identity function on T. Such proofs could then be easily seen to be each others dual.* □

With what above, one can prove that $\mathcal{P}(A \times \cdot)$-homomorphisms satisfy yet another useful property.

Theorem 2.10 *Let $f : (S, \alpha) \to (T, \beta)$ be a homomorphism of $\mathcal{P}(A \times \cdot)$-coalgebras. For any bisimulation $R \subseteq S \times S$, the set*

$$R^f \equiv \{(f(s), f(s')) \in T \times T \mid (s, s') \in R\}$$

is a bisimulation on T. Conversely, for any bisimulation $R \subseteq T \times T$, the set

$$R_f \equiv \{(s, s') \in S \times S \mid (f(s), f(s')) \in R\}$$

is a bisimulation on S. Thus $\mathcal{P}(A \times \cdot)$-homomorphisms are bisimulation preserving and reflecting.

Proof: The (relational) inverse of a bisimulation and the (relational) composition of two bisimulations yields again a bisimulation. Then the theorem follows from

$$R^f = (G_f)^{-1} \circ R \circ G_f \text{ and } R_f = G_f \circ R \circ (G_f)^{-1}$$

(where the composition of relations should be read from left to right). □

Although it has not been stated, a similar property holds for congruences and algebras.

2.2.2 Final Semantics

Clearly, there does not exist a final coalgebra for the functor $\mathcal{P}(A \times \cdot) : Set \to Set$: any final coalgebra is a fixed point, and the functor $\mathcal{P}(A \times \cdot)$ does not have any fixed points; for, the assumption that X, for any set X, is isomorphic to $\mathcal{P}(A{\times}X)$ leads to a contradiction, since the cardinality of the latter is (for non-empty A) strictly bigger than the cardinality of X.

Therefore we shall consider a restriction of the functor $\mathcal{P}(A \times \cdot)$, for which there *does* exist a final coalgebra. It is the functor $\mathcal{P}_f(A \times \cdot) : Set \to Set$, which is defined, on sets S, by

$$\mathcal{P}_f(A{\times}S) \equiv \{V \subseteq A \times S \mid V \text{ is finite}\};$$

on functions, $\mathcal{P}_f(A \times \cdot)$ is defined as before. The coalgebras of this functor are in one-to-one correspondence with labelled transition systems $\langle S, A, \rightarrow\rangle$ that are *finitely branching*: for all $s \in S$ the set $\{\langle a, s'\rangle \in A \times S \mid s \xrightarrow{a} s'\}$ is finite. Note that all the observations made in Section 2.2 about labelled transition systems and coalgebras of $\mathcal{P}(A \times \cdot)$, also apply to finitely branching labelled transition systems and coalgebras of $\mathcal{P}_f(A \times \cdot)$.

In [Bar93], it is shown that there exists a final coalgebra (P, ψ) for the functor $\mathcal{P}_f(A \times \cdot)$. The set P—the elements of which will be called *processes*—can be obtained by first constructing the collection of all finitely branching ordered (possibly infinitely deep) trees with labels from A, and next taking the set of all $\mathcal{P}_f(A \times \cdot)$-bisimulation equivalence classes of such trees. (Recall that $\mathcal{P}_f(A \times \cdot)$-bisimulation coincides with the usual notion of (strong) bisimulation.) Since the construction of the final coalgebra (P, ψ) has some interest of its own, and since it turns out that the original construction in [Bar93] can be somewhat simplified, we shall describe it in some detail in Section 2.4.

Remark 2.11 *** *The family of labelled transition systems $\langle S, A, \rightarrow\rangle$ that are* image finite*—for all $a \in A$ and $s \in S$, the set $\{s' \in S \mid s \xrightarrow{a} s'\}$ is finite—can be similarly described as the category of coalgebras of the functor $A \rightarrow \mathcal{P}_f(\cdot)$, which maps a set S to the set of all functions from A to the set of all finite subsets of S. Also this functor, which occurs for the first time (in a metric setting) in [Bre93], can be shown to have a final coalgebra in Set.* □

Let (S, α) be any $\mathcal{P}_f(A \times \cdot)$-coalgebra (that is, finitely branching transition system), and let $\mathcal{F} : (S, \alpha) \rightarrow (P, \psi)$ be the unique homomorphism given by finality of (P, ψ), called

Branching Final Semantics:

$$\begin{tikzcd}
S \arrow[r, "\mathcal{F}"] \arrow[d, "\alpha"'] \arrow[dr, phantom, "*"] & P \arrow[d, "\psi"] \\
\mathcal{P}_f(A{\times}S) \arrow[r, "\mathcal{P}_f(A{\times}\mathcal{F})"'] & \mathcal{P}_f(A{\times}P)
\end{tikzcd}$$

By the commutativity of this diagram, for $s \in S$,

$$\mathcal{F}(s) = \psi^{-1}(\ \{\langle a, \mathcal{F}(s)\rangle \in A \times P \mid \langle a, s'\rangle \in \alpha(s)\}\ );$$

equivalently,

$$\mathcal{F}(s) = \psi^{-1}(\ \{\langle a, \mathcal{F}(s')\rangle \in A \times P \mid s \xrightarrow{a} s'\}\ ).$$

One can easily verify that the functor $\mathcal{P}_f(A \times \cdot)$ preserves kernels: for any function $f : S \rightarrow T$, the kernel of $\mathcal{P}_f(A{\times}f)$ is equal to $\mathcal{P}_f(A{\times}K_f)$. Hence Theorem 2.6 applies to $\mathcal{F}$: for all $s, s' \in S$,

$$s \sim_S s' \text{ if and only if } \mathcal{F}(s) = \mathcal{F}(s').$$

For the implication from right to left it is sufficient that $K_{\mathcal{F}}$ is a bisimulation, which follows from Proposition 2.7 (rather than deriving this from the fact that $\mathcal{P}_f(A \times \cdot)$ preserves kernels).

2.3 Quotients of Coalgebras***

Some basic properties of quotients of coalgebras are discussed, in order to arrive at a characterization of final semantics in terms of canonical quotients with respect to the greatest bisimulation. First the functor $\mathcal{P}_f(A\times\cdot)$ is treated, next arbitrary functors. Much of what follows in this subsection is an expansion of similar results in [AM89], where coalgebras of endofunctors on a category of classes are discussed.

Let (S,α) be a $\mathcal{P}_f(A\times\cdot)$-coalgebra and (R,γ) a $\mathcal{P}_f(A\times\cdot)$-bisimulation on (S,α). For convenience it is assumed that R is an equivalence relation (if not, the smallest equivalence containing R has to be taken in the construction below). We define, for any $s\in S$,

$$[s]_R \equiv \{s'\in S \mid (s,s')\in R\} \text{ and } S_R \equiv \{[s]_R \mid s\in S\};$$

moreover two functions $\pi_R : S\to S_R$ and $\alpha_R : S_R\to\mathcal{P}_f(A\times S_R)$ are defined by $\pi_R(s)\equiv [s]_R$ and

$$\alpha_R([s]_R) \equiv \{\langle a,[s'']_R\rangle \in A\times S_R \mid s' \xrightarrow{a} s'' \text{ for some } s'\in[s]_R\}.$$

Note that α_R is the unique function from S_R to $\mathcal{P}_f(A\times S_R)$ making the right side of the diagram below commute:

$$\begin{tikzcd}
R \arrow[r, shift left, "\pi_1"] \arrow[r, shift right, "\pi_2"'] \arrow[d, "\gamma"'] \arrow[dr, phantom, "*"] & S \arrow[r, "\pi_R"] \arrow[d, "\beta"] \arrow[dr, phantom, "*"] & S_R \arrow[d, "\alpha_R"] \\
\mathcal{P}_f(A\times R) \arrow[r, shift left, "\mathcal{P}_f(A\times\pi_1)"] \arrow[r, shift right, "\mathcal{P}_f(A\times\pi_2)"'] & \mathcal{P}_f(A\times S) \arrow[r, "\mathcal{P}_f(A\times\pi_R)"'] & \mathcal{P}_f(A\times S_R)
\end{tikzcd}$$

Applying the above to the greatest bisimulation $\sim_S$ on S yields the coalgebra $(S_\sim,\alpha_\sim)$. Interestingly, it is strongly extensional: for if R is a bisimulation on $S_\sim$, then, by Theorem 2.10,

$$\{(s,t)\in S\times S \mid ([s]_\sim,[t]_\sim)\in R\}$$

is a bisimulation on S, implying $\pi_\sim(s)=\pi_\sim(t)$, that is $[s]_\sim=[t]_\sim$. Thus R is a subset of the identity relation on $S_\sim$.

As a corollary of this, $(S_\sim,\alpha_\sim)$ satisfies the 'uniqueness half' of the definition of finality: let (T,β) be any $\mathcal{P}_f(A\times\cdot)$-coalgebra, and let $f,g:(T,\beta)\to(S_\sim,\alpha_\sim)$ be two homomorphisms. Since $(S_\sim,\alpha_\sim)$ is strongly extensional, and since the set $\{(f(t),g(t))\in S_\sim\times S_\sim \mid t\in T\}$ is a bisimulation on $S_\sim$ (it is equal to $(G_f)^{-1}\circ G_g$), it follows that $f=g$.

By the finality of (P,ψ), the following triangle commutes:

$$\begin{tikzcd}
(S,\alpha) \arrow[r, "\pi_\sim"] \arrow[dr, "\mathcal{F}"'] & (S_\sim,\alpha_\sim) \arrow[d, "\mathcal{F}_\sim"] \\
& (P,\psi),
\end{tikzcd}$$

where $\mathcal{F}$ and $\mathcal{F}_\sim$ are the final semantics for S and $S_\sim$. By the strong extensionality of $(S_\sim,\alpha_\sim)$ and the fact that kernels of homomorphisms are bisimulations, it follows that

$\mathcal{F}_{\sim}$ is injective. This tells us that the final semantics $\mathcal{F}$ maps S onto a subset of P that is isomorphic to $S_{\sim}$, and which therefore can be thought of as the canonical quotient of S.

The above can be used for the construction of an easy proof of the folklore theorem that for any two bisimilar transition systems (S, α) and (T, β), there exists a third transition system (U, δ) to which both of them reduce (see [Sif84] and [Bad93]). (Here a *reduction* is a $\mathcal{P}_f(A \times \cdot)$-homomorphism that is surjective; for instance, the quotient mapping $\pi_{\sim}$ is a reduction.) The proof will be easy because it uses (the canonical quotient given by) the final semantics.

(Note that in general it is not possible, given two bisimilar transition systems S and T, to reduce S to T or vice versa; for a simple example consider the two transition systems determined by

$$S = \{s_1, s_2, s_3\}, \text{ and transitions } \{s_1 \xrightarrow{a} s_2,\ s_2 \xrightarrow{a} s_1,\ s_3 \xrightarrow{b} s_3\ \};$$

$$T = \{t_1, t_2, t_3\}, \text{ and transitions } \{t_1 \xrightarrow{a} t_1,\ t_2 \xrightarrow{b} t_3,\ t_3 \xrightarrow{b} t_2,\ \}.)$$

So consider two transition systems (S, α) and (T, β), and suppose (R, γ) is a bisimulation between them, with $\pi_1 : (R, \gamma) \to (S, \alpha)$ and $\pi_2 : (R, \gamma) \to (T, \beta)$ surjective. Let $\mathcal{F}_S : (S, \alpha) \to (P, \psi)$ be the final semantics for S and $\mathcal{F}_T : (T, \beta) \to (P, \psi)$ be the final semantics for T. Let U be defined as

$$\begin{aligned} U &\equiv \mathcal{F}_S(S) \\ &(= \{\mathcal{F}_S(s) \in P \mid s \in S\}), \end{aligned}$$

and let δ be the restriction of ψ to U. The fact that $\mathcal{F}_S$ is a homomorphism implies that δ is a function from U to $\mathcal{P}_f(A \times U)$. Hence (U, δ) is a $\mathcal{P}_f(A \times \cdot)$-coalgebra. By the finality of (P, ψ), the following diagram commutes,

$$\begin{tikzcd}
(R,\gamma) \arrow[r, "\pi_2"] \arrow[d, "\pi_1"'] \arrow[dr, phantom, "*"] & (T,\beta) \arrow[d, "\mathcal{F}_T"] \\
(S,\alpha) \arrow[r, "\mathcal{F}_S"'] & (P,\psi)
\end{tikzcd}$$

Because π_1 and π_2 are surjective, this implies that $\mathcal{F}_S(S) = \mathcal{F}_T(T)$. This shows that (S, α) and (T, β) both reduce to (U, δ):

$$(S, \alpha) \xrightarrow{\mathcal{F}_S} (U, \delta) \xleftarrow{\mathcal{F}_T} (T, \beta).$$

Another proof (which is essentially the one from [Sif84]) can be given without using the fact that a final coalgebra exists. It consists of taking the push-out of π_1 and π_2: the quotient of the disjoint union of S and T modulo the smallest equivalence relation containing R (cf. [Bad93]).

Much of the above applies also to arbitrary functors $F : \mathcal{C} \to \mathcal{C}$. Let (A, α) be an F-coalgebra. A canonical quotient $\pi_{\sim} : A \to A_{\sim}$ can be defined as the (generalized) coequalizer (if, as in Set, it exists) of all bisimulation projections on (A, α) (thus satisfying, for all F-bisimulations $\pi_1, \pi_2 : (R, \gamma) \to (A, \alpha)$, $\pi_{\sim} \circ \pi_1 = \pi_{\sim} \circ \pi_2$). Since the forgetful functor from $\mathcal{C}_F$ to $\mathcal{C}$ creates colimits (such as coequalizers), $A_{\sim}$ can be extended uniquely to an F-coalgebra $(A_{\sim}, \alpha_{\sim})$ such that $\pi_{\sim} : (A, \alpha) \to (A_{\sim}, \alpha_{\sim})$ is an F-homomorphism. Taking the kernel of this projection $\pi_{\sim}$ yields, if F weakly preserves kernel pairs, an

F-bisimulation on (A,α), which is by construction the greatest bisimulation. As above, $(A_\sim,\alpha_\sim)$ is strongly extensional, and satisfies the 'uniqueness half' of the definition of finality. (In fact, these two notions are equivalent for functors weakly preserving kernel pairs.)

2.4 Final Coalgebras for Endofunctors on Sets***

In [Bar93], a method is given for computing final coalgebras for certain functors $F : Set \rightarrow Set$ that are not ω-continuous. It is repeated here (with a somewhat simpler proof) and next applied to the functor $\mathcal{P}_f(A\times\cdot) : Set \rightarrow Set$. As we shall see, the construction of a final coalgebra out of a weakly final one (the last step in the proof) can be nicely characterized in terms of bisimulation.

First let us recall a classical theorem on the construction of final coalgebras. (It is formulated as the dual of the so-called Basic Lemma from [SP82].) Let $\mathcal{C}$ be a category with final object 1 and let $F : \mathcal{C} \rightarrow \mathcal{C}$ be a functor. Let Δ be the following chain,

$$\begin{tikzcd} 1 & F(1) \arrow[l, "!"'] & F^2(1) \arrow[l, "F(!)"'] & \cdots \arrow[l, "F^2(!)"'] \end{tikzcd}$$

(with ! the unique arrow into the final object $1 = \{0\}$). Suppose that both $\mu : D \rightarrow \Delta$ and $F(\mu) : F(D) \rightarrow F(\Delta)$ are limiting cones. Then (D,δ) is a final F-coalgebra, where $\delta : D \rightarrow F(D)$ is the mediating arrow given by the fact that μ (minus its first arrow) is also a cone from D to $F(\Delta)$.

For instance, let A be a given set and consider the functor $A\times\cdot : Set \rightarrow Set$ which maps sets X to the Cartesian product $A\times X$ (and works on functions as one would expect). Constructing the chain Δ as above yields for $A\times\cdot$

$$\begin{tikzcd} 1 & A \arrow[l, "f_0"'] & A^2 \arrow[l, "f_1"'] & \cdots \arrow[l, "f_2"'] \end{tikzcd}$$

where A^n consists of sequences of elements in A of length n, and f_n takes a sequence of length $n+1$ and yields a sequence of length n by removing its last element. The set

$$D \equiv \{(x_n)_n \in \prod_n A^n \mid x_n \in A^n \text{ and } f_n(x_{n+1}) = x_n \}$$

(together with functions $\pi_n : D \rightarrow A^n$ mapping sequences to their n-th component) is a limit for Δ, called the *projective* limit. Since $A\times\cdot$ is continuous, $A\times D$ is a limit for $A\times\Delta$. It follows that there exists a function $\delta : D \rightarrow A\times D$ such that (D,δ) is a final L-coalgebra—δ maps a sequence $(0,\ \langle a_1,0\rangle,\ \langle a_1,a_2,0\rangle,\ \ldots)$ to the pair $\langle a_1,(0,\ \langle a_2,0\rangle,\ \ldots)\rangle$. Clearly, D is (isomorphic to) the set of all infinite sequences over A. Similarly, taking L' as the variant of L that takes sets X to $1+(A\times X)$, one obtains the collection of finite and infinite sequences over A.

Another example—to be used below—is the functor $R : Set \rightarrow Set$ defined, for a set X, by

$$\begin{aligned} R(X) &\equiv \sum_{0\leq n<\omega} (A\times X)^n \\ &(= 1+(A\times X)+(A\times X)^2+\cdots). \end{aligned}$$

The above construction yields a final R-coalgebra (T,τ), consisting of all finitely branching, labelled (over A), ordered (possibly infinitely deep) trees.

The method does not apply to the functor $\mathcal{P}_f(A\times\cdot)$, which is *not* continuous, since $\mathcal{P}_f(\cdot)$ is not. (See also the last remark of this subsection.) Still a final coalgebra exists by the following theorem from [Bar93].

Theorem 2.12 *([Bar93]) Let F and G be two functors from Set to Set. Let $\pi : F \to G$ be a natural transformation: a family of functions $\{\pi_X\}$—one for each set X—with, for any function $f : X \to Y$,*

$$\begin{tikzcd}
F(X) \arrow[r, "F(f)"] \arrow[d, "\pi_X"'] \arrow[dr, phantom, "*"] & F(Y) \arrow[d, "\pi_Y"] \\
G(X) \arrow[r, "G(f)"'] & G(Y)
\end{tikzcd}$$

Suppose π_X is surjective, for any set X (the functor G is then called a quotient of F). If F has a final F-coalgebra, then also G has a final G-coalgebra.

The proof consists of two steps: first it is shown that a final F-coalgebra (S, α) is a *weakly final* G-coalgebra (meaning that from any other G-coalgebra there is at least one homomorphism into (S, α)); secondly, this weakly final G-coalgebra is transformed into a final one.

So let (S, α) be a final F-coalgebra. Then $(S, \pi_S \circ \alpha)$ is a weakly final G-coalgebra. For let (B, β) be an arbitrary G-coalgebra. Since π_B is surjective there exists (using the axiom of choice) a right-inverse $\rho : G(B) \to F(B)$ with $\pi_B \circ \rho = 1_{G(B)}$. By the finality of (S, α) there exists a homomorphism

$$\begin{tikzcd}
B \arrow[r, "f"] \arrow[d, "\rho \circ \beta"'] \arrow[dr, phantom, "*"] & S \arrow[d, "\alpha"] \\
F(B) \arrow[r, "F(f)"'] & F(S)
\end{tikzcd}$$

Combining the above diagram with the one below,

$$\begin{tikzcd}
F(B) \arrow[r, "F(f)"] \arrow[d, "\pi_B"'] \arrow[dr, phantom, "*"] & F(S) \arrow[d, "\pi_S"] \\
G(B) \arrow[r, "G(f)"'] & G(S),
\end{tikzcd}$$

yields that f is a G-homomorphism from (B, β) to $(S, \pi_S \circ \alpha)$ (noting that $\pi_B \circ \rho \circ \beta = \beta$).

Secondly, there are standard techniques for constructing a final object from a weakly final one, which apply to any category. (In [Bar93] two alternatives are mentioned: either take the—generalized—coequalizer of all endomorphisms on the weakly final object, or take the cointersection of all its quotients.) Alternatively (in this particular category of coalgebras), the quotient construction from Section 2.3 can be applied: let $(S_\sim, (\pi_S \circ \alpha)_\sim)$ be the quotient G-coalgebra of $(S, \pi_S \circ \alpha)$ with respect to the greatest G-bisimulation. We saw that it is strongly extensional, which was shown to imply (in fact, it is equivalent) to the 'uniqueness half' of the definition of finality. Because $(S, \pi_S \circ \alpha)$ is weakly final also $(S_\sim, (\pi_S \circ \alpha)_\sim)$ is weakly final. Thus $(S_\sim, (\pi_S \circ \alpha)_\sim)$ is a final G-coalgebra, which concludes the proof of the theorem.

It can be used to show that the functor $\mathcal{P}_f(A \times \cdot)$ has a final coalgebra as follows. Recall the definition of the (tree constructing functor) R above, and consider the family of functions $\pi_X : R(X) \to \mathcal{P}_f(A\times X)$, for any set X, defined by $\pi_X(0) \equiv \emptyset$ and, for $\langle (a_1, x_1), \ldots, (a_n, x_n)\rangle \in (A \times X)^n$,

$$\pi_X(\langle (a_1, x_1), \ldots, (a_n, x_n)\rangle) \equiv \{(a_1, x_1), \ldots, (a_n, x_n)\}.$$

This defines a surjective natural transformation $\pi : R \to \mathcal{P}_f(A \times \cdot)$. Because R has a final coalgebra (T, τ), Theorem 2.12 yields the existence of a final $\mathcal{P}_f(A \times \cdot)$-coalgebra (P, ψ), which is obtained as the quotient of (T, τ) with respect to the greatest $\mathcal{P}_f(A \times \cdot)$-bisimulation.

In conclusion, we give an explicit description of the elements in P (which are, by construction, equivalence classes). Consider the chain

$$\begin{tikzcd} 1 & \mathcal{P}_f(A \times \cdot)(1) \arrow[l, "!"'] & \mathcal{P}_f(A \times \cdot)^2(1) \arrow[l, "\mathcal{P}_f(A \times \cdot)(!)"'] & \cdots \arrow[l, "\mathcal{P}_f(A \times \cdot)^2(!)"'] \end{tikzcd}$$

Write B_n for $\mathcal{P}_f(A \times \cdot)^n(1)$ and π_n for $\mathcal{P}_f(A \times \cdot)^n(!)$, and note that B_n contains all finitely branching trees modulo bisimulation, of depth at most n; the function $\pi_n : B_{n+1} \to B_n$ maps a tree of depth $n + 1$ to one of depth n by removing all nodes at depth $n + 1$. As before, this chain has a (projective) limit,

$$B \equiv \{(X_n)_n \in \prod_n B_n \mid X_n \in B_n \text{ and } \pi_n(X_{n+1}) = X_n \}.$$

Note that it follows from the general construction of final coalgebras, described at the beginning of this subsection, that if the functor $\mathcal{P}_f(A \times \cdot)$ were continuous—which it is not—then B would be a final $\mathcal{P}_f(A \times \cdot)$-coalgebra. However, the final $\mathcal{P}_f(A \times \cdot)$-coalgebra P constructed above can be seen to be (isomorphic to) a *subset* of B, as follows.

Let $X \in B_n$, for some $n \geq 0$, and define for every $k \geq 0$ the number $\epsilon_k(X)$ as the number of nodes in X up to depth k (a formal definition would be easy). A projective sequence $(X_n)_n$ in B is *k-stable* if the sequence

$$(\epsilon_k(X_0), \epsilon_k(X_1), \epsilon_k(X_2), \ldots)$$

becomes eventually constant. The intuition is that from that moment on, the elements in the chain all have a fixed number of nodes up to depth k. Now one can prove the following:

$$P \cong \{(X_n)_n \in B \mid (X_n)_n \text{ is } k\text{-stable for every } k \geq 0 \}.$$

Note that those elements in B that are (representing) infinitely branching trees are not in (the isomorphic image of) P, like the sequence $(Y_n)_n$ given, for $n \geq 0$, by

$$Y_n \equiv X_0 \cup \cdots \cup X_n, \text{ with } X_0 \equiv \emptyset \text{ and } X_{n+1} \equiv \{\langle a, X_n\rangle\}.$$

(This example can be turned into a formal proof of the fact that $\mathcal{P}_f(A \times \cdot)$ is not continuous.)

3 From Final Coalgebra to Initial Algebra Semantics for Strong Bisimulation

Let (Σ, r) be a ranked alphabet and let (T, ϕ) be the initial Σ^*-algebra defined in Section 1: T is the set of all closed terms over Σ. Consider a finitely branching transition system

$\langle T, A, \to \rangle$ or, equivalently a $\mathcal{P}_f(A \times \cdot)$-coalgebra (T, α) for T. It was shown in Section 2 that there exists, by the finality of the $\mathcal{P}_f(A \times \cdot)$-coalgebra (P, ψ), a unique coalgebra homomorphism $\mathcal{F} : (T, \alpha) \to (P, \psi)$:

$$\begin{tikzcd}
T \arrow[r, "\mathcal{F}"] \arrow[d, "\alpha"'] \arrow[dr, phantom, "*"] & P \arrow[d, "\psi"] \\
\mathcal{P}_f(A\times T) \arrow[r, "\mathcal{P}_f(A\times\mathcal{F})"'] & \mathcal{P}_f(A\times P),
\end{tikzcd}$$

called the final semantics for T, with for all $s, t \in T$,

$s \sim_T t$ if and only if $\mathcal{F}(s) = \mathcal{F}(t)$.

In this section, the set P will be turned into a Σ^*-algebra structure (P, β) by constructing a function $\beta : \Sigma^*(P) \to P$ from the transition system specification (see below) for α. The initiality of (T, ϕ) then gives the existence of a unique Σ^*-homomorphism $\mathcal{I} : (T, \phi) \to (P, \beta)$:

$$\begin{tikzcd}
\Sigma^*(T) \arrow[r, "\Sigma^*(\mathcal{I})"] \arrow[d, "\phi"'] \arrow[dr, phantom, "*"] & \Sigma^*(P) \arrow[d, "\beta"] \\
T \arrow[r, "\mathcal{I}"'] & P
\end{tikzcd}$$

(the initial semantics for T), satisfying for all $f \in \Sigma$ and $\langle t_1, \ldots, t_{r(f)} \rangle \in T^{r(f)}$,

$$\mathcal{I}(f(t_1, \ldots, t_{r(f)})) = f_P(\langle \mathcal{I}(t_1), \ldots, \mathcal{I}(t_{r(f)}) \rangle).$$

(Recall that, for $\langle p_1, \ldots, p_{r(f)} \rangle \in P^{r(f)}$, $f_P(\langle p_1, \ldots, p_{r(f)} \rangle)$ is—by definition—equal to $\beta(f\langle p_1, \ldots, p_{r(f)} \rangle)$.)

Moreover, the construction will be such that $\mathcal{I} = \mathcal{F}$:

$$\begin{tikzcd}
\Sigma^*(T) \arrow[r, "\Sigma^*(\mathcal{I})"] \arrow[d, "\phi"'] \arrow[dr, phantom, "*"] & \Sigma^*(P) \arrow[d, "\beta"] \\
T \arrow[r, "\mathcal{I} = \mathcal{F}"] \arrow[d, "\alpha"'] \arrow[dr, phantom, "*"] & P \arrow[d, "\psi"] \\
\mathcal{P}_f(A\times T) \arrow[r, "\mathcal{P}_f(A\times\mathcal{F})"'] & \mathcal{P}_f(A\times P)
\end{tikzcd}$$

Thus a compositional description of the final semantics is obtained. The initial semantics is often called *denotational*, because of the emphasis on the assignment of denotations $\mathcal{I}(s)$ to statements s. On the other hand, the final semantics $\mathcal{F}$ is often called *operational*, since it is based directly on a transition system.

The initial semantics, which will be canonically constructed here, turns out to be—for certain specific signatures—the same as already existing denotational models. The initial semantics of the example in Section 3.6 coincides with (a variant of) a denotational model given in [BM88]. Similarly, the linear semantics of the next section is essentially the denotational model from [HP79]. In both papers, operational and denotational semantics are defined independently, and their equivalence is proved next using some fixed-point argument (in metric and ordered spaces, respectively). Interestingly, such fixed-point arguments are not needed here; the equality of $\mathcal{I}$ and $\mathcal{F}$ is a direct consequence of the finality of their co-domain P.

3.1 Processes as Terms

The crux of the construction is the definition of a set T_P of *mixed terms*, consisting of the set of terms over the extended signature $\Sigma + P$: the original signature Σ to which all processes $p \in P$ have been added as constants (thus $r(p) \equiv 0$). Formally, T_P is an initial algebra of the functor $(\Sigma + P)^*$, which is defined in the same way as Σ^*.

This technique of extending the collection of syntactic entities (terms) with semantic entities (processes), was introduced in the context of semantics for transition systems in [Rut92]. It is well known in the world of models for the lambda calculus, where elements d of a model are included into the collection of lambda terms as constants $\underline{d}$.

Consider the signature $\Sigma = \{0, s, +\}$, with 0 a constant, s a unary and $+$ a binary function symbol. Examples of terms in T are $s(s(0))$ and $s(0) + s(0 + s(0))$; two mixed terms in T_P are (with $p \in P$) $s(s(p))$ and $s(0) + s(p + s(0))$.

It follows from the definition of the functors Σ^* and $(\Sigma + P)^*$ that, for any set S,

$$(\Sigma + P)^*(S) \cong \Sigma^*(S) + P.$$

Since T_P is a fixed point of $(\Sigma + P)^*$, this implies $T_P \cong \Sigma^*(T_P) + P$, giving the existence of two functions

$$\begin{tikzcd}
\Sigma^*(T_P) \arrow[d, "\kappa"'] & \\
T_P & P. \arrow[l, "\iota"]
\end{tikzcd}$$

which are defined straightforwardly. By the initiality of T, there exists a unique homomorphism $\mathcal{I}_P : (T, \phi) \to (T_P, \kappa)$,

$$\begin{tikzcd}
\Sigma^*(T) \arrow[r, "\Sigma^*(\mathcal{I}_P)"] \arrow[d, "\phi"'] \arrow[dr, phantom, "*"] & \Sigma^*(T_P) \arrow[d, "\kappa"] \\
T \arrow[r, "\mathcal{I}_P"'] & T_P
\end{tikzcd}$$

The function $\mathcal{I}_P$ is merely the inclusion of the set T of terms into the set T_P of mixed terms.

It is also possible to provide T_P with a suitable $\mathcal{P}_f(A \times \cdot)$*-coalgebra* structure. For that, we shall have to consider the way in which the transition relation $\to$ in $\langle T, A, \to \rangle$ (equivalently, α in (T, α)) has been defined.

3.2 Transition System Specifications

A common way of defining a transition relation $\to$ is to specify a collection of *rules* that are used for proving that a triple $\langle t, a, t'\rangle \in T \times A \times T$ is in $\to$. When the collection of states of the transition system is well structured—here it is the set T of terms over the signature Σ—the form of these rules often reflects this structure. Such an approach is therefore called *structural operational semantics* ([Plo81b]).

A *transition systems specification* for the signature Σ (with labels in the set A) is a collection $\mathcal{R}$ of rules of the form

$$\frac{\{t_i \xrightarrow{a_i} t_i' : i \in I\}}{t \xrightarrow{a} t'}$$

where I is some set of indices, a and a_i (for $i \in I$) are in A, and t, t' and t_i, t_i' (for $i \in I$) are in T_X, the set of terms over Σ with variables in a given set X (see Section 1). The expressions $t_i \xrightarrow{a_i} t_i'$ are called premises and $t \xrightarrow{a} t'$ is called the conclusion of this rule. If I is empty then the rule is called an axiom.

Such a transition system specification induces a transition relation $\to \subseteq T \times A \times T$ as usual: $\to$ contains those triples that are provable from the rules in $\mathcal{R}$. (See, for instance, [Rut92] for a formal definition. We shall see an example below.)

For the rest of this section, it is assumed that the (transition relation of the) transition system $\langle T, A, \to\rangle$ is induced by a transition system specification $\mathcal{R}$.

Now it is possible to define a transition system $\langle T_P, A, \to_P\rangle$ for the set T_P of mixed terms as follows. Let the specification $\mathcal{R}_P$ be defined as

$$\mathcal{R}_P \equiv \mathcal{R} \cup \{p \xrightarrow{a} q \mid \langle a, q\rangle \in \psi(p)\}.$$

(Recall that $\psi : P \to \mathcal{P}_f(A \times P)$.) That is, all possible transitions in (P, ψ)—which is itself a transition system—are added to $\mathcal{R}$ as axioms. Next let $\to_P$ be the transition relation induced by $\mathcal{R}_P$. Then T_P can be turned into a $\mathcal{P}_f(A \times \cdot)$-coalgebra

$$\begin{tikzcd}
T_P \arrow[d, "\alpha_P"'] \\
\mathcal{P}_f(A \times T_P)
\end{tikzcd}$$

by defining α_P from $\to_P$ in the familiar way: for $a \in A$ and $t, t' \in T_P$,

$$\langle a, t'\rangle \in \alpha_P(t) \Leftrightarrow t \xrightarrow{a}_P t'.$$

We should like (T_P, α_P) to be such that it can be seen as a *conservative extension* of both (T, α) and (P, ϕ); that is,

$$\begin{tikzcd}
T \arrow[r, "\mathcal{I}_P"] \arrow[d, "\alpha"'] \arrow[dr, phantom, "*_1"] & T_P \arrow[d, "\alpha_P"] \arrow[dr, phantom, "*_2"] & P \arrow[l, "\iota"'] \arrow[d, "\psi"] \\
\mathcal{P}_f(A \times T) \arrow[r, "\mathcal{P}_f(A \times \mathcal{I}_P)"'] & \mathcal{P}_f(A \times T_P) & \mathcal{P}_f(A \times P) \arrow[l, "\mathcal{P}_f(A \times \iota)"]
\end{tikzcd}$$

As we shall see in Section 3.5, the commutativity of both squares above can be guaranteed making some mild

assumptions (1 and 2)

on the form of the rules in $\mathcal{R}$.

By finality of (P,ψ) there exists a unique homomorphism $\mathcal{F}_P : (T_P,\alpha_P) \rightarrow (P,\psi)$,

$$\begin{tikzcd}
T_P \arrow[r, "\mathcal{F}_P"] \arrow[d, "\alpha_P"'] \arrow[dr, phantom, "*"] & P \arrow[d, "\psi"] \\
\mathcal{P}_f(A \times T_P) \arrow[r, "\mathcal{P}_f(A \times \mathcal{F}_P)"'] & \mathcal{P}_f(A \times P)
\end{tikzcd}$$

Note that $\mathcal{F}_P \circ \mathcal{I}_P = \mathcal{F}$, since both $\mathcal{F}_P \circ \mathcal{I}_P$ and $\mathcal{F}$ are homomorphisms to the final $\mathcal{P}_f(A \times \cdot)$-coalgebra (P,ψ); for the same reason, $\mathcal{F}_P \circ \iota = 1_P$ (with 1_P is the identity function on P).

3.3 Turning P into a Σ^*-Algebra

The next diagram collects the constructions we have described so far:

$$\begin{tikzcd}
\Sigma^*(T) \arrow[r, "\Sigma^*(\mathcal{I}_P)"] \arrow[d, "\phi"'] \arrow[dr, phantom, "*"] & \Sigma^*(T_P) \arrow[r, shift right, "\Sigma^*(\mathcal{F}_P)"'] \arrow[d, "\kappa"] & \Sigma^*(P) \arrow[l, shift right, "\Sigma^*(\iota)"'] \\
T \arrow[r, "\mathcal{I}_P"] \arrow[d, "\alpha"'] \arrow[dr, phantom, "*_1"] & T_P \arrow[r, shift right, "\mathcal{F}_P"'] \arrow[d, "\alpha_P"] \arrow[dr, phantom, "*"] & P \arrow[l, shift right, "\iota"'] \arrow[d, "\psi"] \\
\mathcal{P}_f(A \times T) \arrow[r, "\mathcal{P}_f(A \times \mathcal{I}_P)"'] & \mathcal{P}_f(A \times T_P) \arrow[r, "\mathcal{P}_f(A \times \mathcal{F}_P)"'] & \mathcal{P}_f(A \times P)
\end{tikzcd}$$

Together with the fact that $\mathcal{F}_P \circ \iota = 1_P$, this diagram contains all the observations that have been made above.

As the last step in the construction, we define the missing arrow in the diagram: let $\beta : \Sigma^*(P) \rightarrow P$ be given by

$$\beta \equiv \mathcal{F}_P \circ \kappa \circ \Sigma^*(\iota).$$

The function β corresponds, as usual, to a family of functions

$$\{f_P : P^{r(f)} \rightarrow P \mid f \in \Sigma\},$$

given, for $f \in \Sigma$ and $\langle p_1, \ldots, p_{r(f)} \rangle \in P^{r(f)}$, by

$$\begin{aligned}
f_P(\langle p_1, \ldots, p_{r(f)} \rangle) &\equiv \beta(f\langle p_1, \ldots, p_{r(f)} \rangle) \\
&= \mathcal{F}_P \circ \kappa \circ \Sigma^*(\iota)(f\langle p_1, \ldots, p_{r(f)} \rangle) \\
&= \mathcal{F}_P \circ \kappa(\ f\langle \iota(p_1), \ldots, \iota(p_{r(f)}) \rangle\ ) \\
&= \mathcal{F}_P \circ \kappa(\ f\langle p_1, \ldots, p_{r(f)} \rangle\ ) \\
&= \mathcal{F}_P(\ f(p_1, \ldots, p_{r(f)})\ ).
\end{aligned}$$

Thus the value of $f_P(\langle p_1, \dots, p_{r(f)} \rangle)$ is obtained by first embedding $\langle p_1, \dots, p_{r(f)} \rangle$ into the collection of mixed terms T_P by an application of $\kappa \circ \Sigma^*(\iota)$. (Recall that ι is the embedding of P into T_P.) Thus the mixed term $f(p_1, \dots, p_{r(f)})$ is obtained: It constitutes the motivating example for the 'processes as terms' approach. Finally, the result is obtained by the application of $\mathcal{F}_P$, which is the final semantics for mixed terms.

Now the initiality of (T, ϕ) gives the existence of the initial semantics that has been announced at the beginning of this section:

$$\begin{tikzcd}
\Sigma^*(T) \arrow[r, "\Sigma^*(\mathcal{I})"] \arrow[d, "\phi"'] \arrow[dr, phantom, "*"] & \Sigma^*(P) \arrow[d, "\beta"] \\
T \arrow[r, "\mathcal{I}"'] & P
\end{tikzcd}$$

3.4 Equality of the Initial Semantics $\mathcal{I}$ and the Final Semantics $\mathcal{F}$

It still has to be proved that $\mathcal{I}$ and $\mathcal{F}$ are equal. Although this need not be true in general, we shall see in a moment that the requirement that $\mathcal{P}_f(A \times \cdot)$-bisimulation is a congruence forms a sufficient condition for this equality.

A proof of $\mathcal{I} = \mathcal{F}$ is as follows. It is sufficient to prove that the homomorphism of $\mathcal{P}_f(A \times \cdot)$-coalgebras $\mathcal{F}_P : (T_P, \alpha_P) \to (P, \psi)$ is also a homomorphism of Σ^*-algebras $\mathcal{F}_P : (T_P, \kappa) \to (P, \beta)$; that is,

$$\begin{tikzcd}
\Sigma^*(T_P) \arrow[r, "\Sigma^*(\mathcal{F}_P)"] \arrow[d, "\kappa"'] \arrow[dr, phantom, "*_3"] & \Sigma^*(P) \arrow[d, "\beta"] \\
T_P \arrow[r, "\mathcal{F}_P"'] & P,
\end{tikzcd}$$

since it implies that $\mathcal{I} = \mathcal{F}_P \circ \mathcal{I}_P$, both $\mathcal{I}$ and $\mathcal{F}_P \circ \mathcal{I}_P$ being algebra homomorphisms between the initial Σ^*-algebra (T, ϕ) and (P, β). Since also $\mathcal{F}_P \circ \mathcal{I}_P = \mathcal{F}$ the desired equality then follows,

$$\mathcal{I} = \mathcal{F}_P \circ \mathcal{I}_P = \mathcal{F}.$$

So let us investigate how the above commutativity $(*_3)$ can be established. Consider the following elementary lemma. (Recall that for a function $m : X \to Y$, the kernel K_m is defined as the set $\{(x, x') \in X \times X \mid m(x) = m(x')\}$.)

Lemma 3.1 *Consider the following diagram of sets and functions:*

$$\begin{tikzcd}
A \arrow[r, shift left, "f"] \arrow[dr, "h"'] & B \arrow[l, shift left, "g"] \\
& C
\end{tikzcd}$$

If $f \circ g = 1_B$ and $K_f \subseteq K_h$ then $h \circ g \circ f = h$:

$$\begin{tikzcd}
A \arrow[r, "f"] \arrow[dr, "h"'] & B \arrow[d, "h \circ g"] \\
 & C
\end{tikzcd}$$

Proof: For all $a \in A$, $f \circ g \circ f(a) = 1_B \circ f(a) = f(a)$, thus $(g \circ f(a), a) \in K_f$. By assumption, this implies $(g \circ f(a), a) \in K_h$, yielding the result. □

Clearly we want to apply this lemma to the following functions:

$$\begin{tikzcd}
\Sigma^*(T_P) \arrow[r, shift right, "\Sigma^*(\mathcal{F}_P)"'] \arrow[d, "\kappa"] & \Sigma^*(P) \arrow[l, shift right, "\Sigma^*(\iota)"'] \\
T_P \arrow[r, "\mathcal{F}_P"'] & P
\end{tikzcd}$$

Note that the first condition of Lemma 3.1, $\Sigma^*(\mathcal{F}_P) \circ \Sigma^*(\iota) = 1_{\Sigma^*(P)}$, is fulfilled since by assumption 2 made above, $\mathcal{F}_P \circ \iota = 1_P$. The following lemma describes when the second condition holds.

Lemma 3.2 *Let (A, γ) be a Σ^*-algebra, B a set, and $l : A \rightarrow B$ a function:*

$$\begin{tikzcd}
\Sigma^*(A) \arrow[r, "\Sigma^*(l)"] \arrow[d, "\gamma"'] & \Sigma^*(B) \\
A \arrow[r, "l"'] & B
\end{tikzcd}$$

Then K_l is a Σ^*-congruence on (A, γ) if and only if $K_{\Sigma^*(l)} \subseteq K_{l \circ \gamma}$.

A proof of this lemma can be found in Section 3.7.

Because the kernel $K_{\mathcal{F}_P}$ of $\mathcal{F}_P$ is equal to the greatest bisimulation relation $\sim_{T_P}$ on T_P (Theorem 2.6), the above lemma—and consequently Lemma 3.1—can be applied when $\sim_{T_P}$ is a congruence (on (T_P, κ)). As we shall see in Section 3.5, one way of proving that bisimilarity is a congruence is to make again an

assumption (3)

on the format of the rules in $\mathcal{R}$. Thus, under this assumption, an application of Lemma 3.1 yields

$$\begin{tikzcd}
\Sigma^*(T_P) \arrow[r, "\Sigma^*(\mathcal{F}_P)"] \arrow[d, "\kappa"'] \arrow[dr, phantom, "*_3"] & \Sigma^*(P) \arrow[d, "\mathcal{F}_P \circ \kappa \circ \Sigma^*(\iota)"] \\
T_P \arrow[r, "\mathcal{F}_P"'] & P.
\end{tikzcd}$$

Summarizing the above, we have the following.

Theorem 3.3 *Assume (*$\mathcal{R}$ *to be such) that*

$$\begin{tikzcd}
T \arrow[r, "\mathcal{I}_P"] \arrow[d, "\alpha"'] \arrow[dr, phantom, "*_1"] & T_P \arrow[d, "\alpha_P"] \arrow[dr, phantom, "*_2"] & P \arrow[l, "\iota"'] \arrow[d, "\psi"] \\
\mathcal{P}_f(A \times T) \arrow[r, "\mathcal{P}_f(A \times \mathcal{I}_P)"'] & \mathcal{P}_f(A \times T_P) & \mathcal{P}_f(A \times P) \arrow[l, "\mathcal{P}_f(A \times \iota)"]
\end{tikzcd}$$

Let $\beta \equiv \mathcal{F}_P \circ \kappa \circ \Sigma^*(\iota)$*. Suppose that (*$\mathcal{R}$ *is such that* $\sim_{T_P}$ *is a congruence, by which)*

$$\begin{tikzcd}
\Sigma^*(T_P) \arrow[r, "\Sigma^*(\mathcal{F}_P)"] \arrow[d, "\kappa"'] \arrow[dr, phantom, "*_3"] & \Sigma^*(P) \arrow[d, "\beta"] \\
T_P \arrow[r, "\mathcal{F}_P"'] & P
\end{tikzcd}$$

Then

$$\begin{tikzcd}
\Sigma^*(T) \arrow[r, "\Sigma^*(\mathcal{I}_P)"] \arrow[d, "\phi"'] \arrow[dr, phantom, "*"] & \Sigma^*(T_P) \arrow[r, "\Sigma^*(\mathcal{F}_P)"] \arrow[d, "\kappa"] \arrow[dr, phantom, "*_3"] & \Sigma^*(P) \arrow[d, "\beta"] \\
T \arrow[r, "\mathcal{I}_P"] \arrow[d, "\alpha"'] \arrow[dr, phantom, "*_1"] & T_P \arrow[r, "\mathcal{F}_P"] \arrow[d, "\alpha_P"] \arrow[dr, phantom, "*"] & P \arrow[d, "\psi"] \\
\mathcal{P}_f(A \times T) \arrow[r, "\mathcal{P}_f(A \times \mathcal{I}_P)"'] & \mathcal{P}_f(A \times T_P) \arrow[r, "\mathcal{P}_f(A \times \mathcal{F}_P)"'] & \mathcal{P}_f(A \times P);
\end{tikzcd}$$

in other words,

$\mathcal{F}_P \circ \mathcal{I}_P : (T, \alpha) \to (P, \psi)$ *is a homomorphism of* $\mathcal{P}_f(A \times \cdot)$*-coalgebras, and*

$\mathcal{F}_P \circ \mathcal{I}_P : (T, \phi) \to (P, \beta)$ *is a homomorphism of* Σ^**-algebras.*

By initiality of (T, ϕ) *and finality of* (P, ψ)*, it follows that*

$\mathcal{I} = \mathcal{F}_P \circ \mathcal{I}_P = \mathcal{F}.$

□

3.5 Assumptions on the Format of $\mathcal{R}$

Consider a rule R,

$$\frac{\{t_i \xrightarrow{a_i} t_i' : i \in I\}}{t \xrightarrow{a} t'}$$

(Recall that terms in R are elements of T_X, the set of terms over the signature Σ with variables from X.) The *bound* variables occurring in R are inductively defined as the ones that either occur in t or occur in a term t_i', for some $i \in I$, for which t_i only contains bound variables. The rule R is called *pure* if all variables that occur in it are bound. A transition system specification is called pure if it contains only pure rules. (Cf. [GV92] and [Gla93].)

In [Rut92], the following fact is proved. If $\mathcal{R}$ is pure (called inductive there), then

$$\begin{tikzcd}
T \arrow[r, "\mathcal{I}_P"] \arrow[d, "\alpha"'] \arrow[dr, phantom, "*_1"] & T_P \arrow[d, "\alpha_P"] \\
\mathcal{P}_f(A \times T) \arrow[r, "\mathcal{P}_f(A \times \mathcal{I}_P)"'] & \mathcal{P}_f(A \times T_P)
\end{tikzcd}$$

This takes care of assumption 1 above.

The following rule is not pure,

$$\frac{x \xrightarrow{a} x'}{a \xrightarrow{a} a}$$

and illustrates what can go wrong when not all rules are pure: if Σ is the singleton set containing only the constant a, and if $\mathcal{R}$ consists of only the rule above, then $\alpha(a)$ is empty, whereas $\alpha_P(\mathcal{I}_P(a)) = \alpha_P(a)$ is not, due to the presence of process(es) $p \in P$ with $\langle a, p' \rangle \in \psi(p)$ (for some $p' \in P$).

A rule R is an *x-rule* if the antecedent of the conclusion consists of a variable. Forbidding x-rules in $\mathcal{R}$ implies that all rules that can be used in the derivation of transitions (in $\mathcal{R}_P$) for processes $p \in P$ (now seen as elements of T_P, formally $\iota(p)$), are contained in

$$\{p \xrightarrow{a} q \mid \langle a, q \rangle \in \psi(p)\}.$$

Since these specify precisely the transitions of the coalgebra (P, ψ), it follows that

$$\begin{tikzcd}
T_P \arrow[d, "\alpha_P"] \arrow[dr, phantom, "*_2"] & P \arrow[l, "\iota"'] \arrow[d, "\psi"] \\
\mathcal{P}_f(A \times T_P) & \mathcal{P}_f(A \times P) \arrow[l, "\mathcal{P}_f(A \times \iota)"]
\end{tikzcd}$$

Thus the absence of x-rules is sufficient to validate assumption 2.

In [GV92], a sufficient condition is given to ensure that the bisimilarity equivalence induced by a specification $\mathcal{R}$ is a congruence.

Definition 3.4 A rule R is in *tyft*-format if it has the following form

$$\frac{\{t_i \xrightarrow{a_i} y_i : i \in I\}}{f(x_1, \ldots, x_{r(f)}) \xrightarrow{a} t},$$

and it is in *tyxt*-format if it is of the form

$$\frac{\{t_i \xrightarrow{a_i} y_i : i \in I\}}{x \xrightarrow{a} t}$$

where $f \in \Sigma$; the terms t and t_i, for $i \in I$ are in T_X; and all of the variables in $\{x_1, \ldots, x_{(r(f)}\} \cup \{y_i : i \in I\}$, in the first case, and all of the variables in $\{x\} \cup \{y_i : i \in I\}$, in the second case, are pairwise distinct variables in X. The transition system specification $\mathcal{R}$ is in *tyft/tyxt*-format if all its rules are either in *tyft* or in *tyxt* format. □

Theorem 3.5 *([GV92]) If $\mathcal{R}$ is in pure tyft/tyxt-format then the bisimilarity relation corresponding to the transition system induced by $\mathcal{R}$ is a congruence.*

It is therefore sufficient for assumption 3 above to hold, that $\mathcal{R}_P$ is in pure *tyft/tyxt*-format. Since the axioms we have added to $\mathcal{R}$ in the definition of $\mathcal{R}_P$ have the right format, this amounts to requiring $\mathcal{R}$ itself to be in pure *tyft/tyxt*-format.

Taking the conjunction of all conditions needed to make assumptions 1, 2 and 3 valid, we find in summary that a sufficient condition for Theorem 3.3 to hold is that $\mathcal{R}$ is in pure *tyft*-format. Moreover the transition relation induced by $\mathcal{R}$ should be finitely branching.

Note, in conclusion, that in [GV92] it has been observed that a rule in pure *tyxt*-format can always be translated into an equivalent set of rules in pure *tyft*-format, by making for every $f \in \Sigma$ a new copy of the rule, in which the variable x is replaced everywhere by the term $f(x_1, \ldots, x_{r(f)})$ (the variables $x_1, \ldots, x_{r(f)}$ should not yet occur in the old rule).

3.6 An Example

Consider the signature $\Sigma_B \equiv Act \cup \{\epsilon, \delta\} \cup RecVar \cup \{\cdot, +, \|\}$, consisting of a set Act of atomic actions, two special symbols ϵ and δ, a set $(X \in)RecVar$ of recursion variables, and three operators $\cdot$, $+$, and $\|$. (The signature Σ_B is called Basic Process Algebra with ϵ and δ (see, e.g., [GV92]), here extended with recursion and parallel composition.) All elements are constants, but for the latter three, which are binary operators. The interpretation of $\cdot$, for concatenation, $+$, for nondeterministic choice, and $\|$, for parallel composition, is as usual. Let $A = Act \cup \{\surd\}$. The label $\surd$ is used to indicate termination. A transition system specification $\mathcal{R}_B$ for Σ_B is defined as follows. It uses terms over the signature Σ_B with variables in $\{x, x', y, y'\}$. For every $a \in A$, there is an axiom

$$a \xrightarrow{a} \epsilon;$$

from ϵ, one final transition is possible (there will be no transitions from δ),

$$\epsilon \xrightarrow{\surd} \delta.$$

For $X \in RecVar$, there is the following rule:

$$\frac{s_X \xrightarrow{a} y}{X \xrightarrow{a} y}$$

where s_X is a given term in T_B, the set of terms over Σ_B, which can be seen as the body of the recursion variable X. As usual, these statements are required to be *guarded* in X (excluding statements like $(X \cdot a) + b$), such that the resulting transition system will be finitely branching. (Taking $(X \cdot a) + b$ for s_X, there would be infinitely many transitions possible from X.)

Further there are the following rules,

$$\frac{x \xrightarrow{a} x'}{x + y \xrightarrow{a} x'} \qquad \frac{x \xrightarrow{a} x'}{y + x \xrightarrow{a} x'} \qquad \frac{x \xrightarrow{a} x',\ a \neq \surd}{x \cdot y \xrightarrow{a} x' \cdot y} \qquad \frac{x \xrightarrow{\surd} x' \ \ y \xrightarrow{a} y'}{x \cdot y \xrightarrow{a} y'}$$

$$\frac{x \xrightarrow{a} x'}{x \parallel y \xrightarrow{a} x' \parallel y} \qquad \frac{x \xrightarrow{a} x'}{y \parallel x \xrightarrow{a} y \parallel x'}$$

All of the above rules are in pure *tyft*-format.

Let $< T_B, A, \rightarrow >$ be the transition system induced by $\mathcal{R}_B$. The final semantics $\mathcal{F} : T_B \rightarrow P$ satisfies (omitting here and below the isomorphism ψ), for $s \in T_B$,

$$\mathcal{F}(s) = \{\langle a, \mathcal{F}(s') \rangle \in A \times P \mid s \xrightarrow{a} s'\}.$$

For the initial semantics $\mathcal{I} : T_B \rightarrow P$, we have, for $a \in A$, $s, t \in T_B$,

$$\begin{aligned}
\mathcal{I}(a) &= a_P = \{\langle a, \mathcal{I}(\epsilon) \rangle\} \\
\mathcal{I}(\epsilon) &= \epsilon_P = \{\langle \surd, \mathcal{I}(\delta) \rangle\} \\
\mathcal{I}(\delta) &= \delta_P = \emptyset \\
\mathcal{I}(s + t) &= \mathcal{I}(s) \ +_P \ \mathcal{I}(t) \\
\mathcal{I}(s \cdot t) &= \mathcal{I}(s) \ \cdot_P \ \mathcal{I}(t) \\
\mathcal{I}(s \parallel t) &= \mathcal{I}(s) \ \parallel_P \ \mathcal{I}(t) \\
\mathcal{I}(X) &= \mathcal{I}(s_X)
\end{aligned}$$

It follows from the definition of β that the function $\parallel_P$, for instance, satisfies, for $p, q \in P$,

$$\begin{aligned}
p \ \parallel_P q &= \mathcal{F}_P(p \parallel q) \\
&= \{\langle a, \mathcal{F}_P(t) \rangle \in A \times P \mid \ (p \parallel q) \xrightarrow{a}_P t\} \\
&= \{\langle a, \mathcal{F}_P(p' \parallel q) \rangle \in A \times P \mid \ p \xrightarrow{a}_P p'\} \ \cup \\
& \ \{\langle a, \mathcal{F}_P(p \parallel q') \rangle \in A \times P \mid \ q \xrightarrow{a}_P q'\} \\
&= \{\langle a, p' \ \parallel_P \ q \rangle \in A \times P \mid \ \langle a, p' \rangle \in p\} \ \cup \\
& \ \{\langle a, p \ \parallel_P q' \rangle \in A \times P \mid \ \langle a, q' \rangle \in q\}.
\end{aligned}$$

Thus $\parallel_P$ turns out to be the—in the world of denotational semantics for concurrency—familiar parallel composition of processes. It was introduced first in [BZ82] (in the context of metric spaces), where it was defined directly on a collection of processes (very similar to our P), without making use of transition systems.

3.7 A Proof of the Extension Lemma***

Lemma 3.2 can be proved as follows. Assume that (K_l, δ) is a Σ^*-congruence on (A, γ). The kernel $K_{\Sigma^*(l)}$ of the function $\Sigma^*(l)$ can be easily seen to satisfy

$$\begin{aligned}
K_{\Sigma^*(l)} &= \{ \ (f\langle a_1, \ldots, a_{r(f)} \rangle, f\langle a'_1, \ldots, a'_{r(f)} \rangle) \in \Sigma^*(A) \times \Sigma^*(A) \mid \\
& \ f \in \Sigma \text{ and } (a_i, a'_i) \in K_l, \text{ for } i = 1, \ldots, r(f) \ \}.
\end{aligned}$$

Define the function $\lambda : K_{\Sigma^*(l)} \to \Sigma^*(K_l)$, for a pair

$$(f\langle a_1, \ldots, a_{\tau(f)}\rangle, f\langle a_1', \ldots, a_{\tau(f)}'\rangle) \in \Sigma^*(A) \times \Sigma^*(A),$$

by

$$\lambda(f\langle a_1, \ldots, a_{\tau(f)}\rangle, f\langle a_1', \ldots, a_{\tau(f)}'\rangle) \equiv f\langle (a_1, a_1'), \ldots, (a_{\tau(f)}, a_{\tau(f)}')\rangle.$$

(This actually defines an isomorphism, showing that the functor Σ^* preserves kernels.) Now the lemma follows from the commutativity of the the following diagram (where ρ_1 and ρ_2 are the projections belonging to $K_{\Sigma^*(l)}$):

$$\begin{tikzcd}
K_{\Sigma^*(l)} \arrow[r, equal] \arrow[d, "\rho_1"'] \arrow[dr, phantom, "*"] & K_{\Sigma^*(l)} \arrow[r, equal] \arrow[d, "\lambda"] \arrow[dr, phantom, "*"] & K_{\Sigma^*(l)} \arrow[d, "\rho_2"] \\
\Sigma^*(A) \arrow[d, "\gamma"'] \arrow[dr, phantom, "*"] & \Sigma^*(K_l) \arrow[l, "\Sigma^*(\pi_1)"'] \arrow[r, "\Sigma^*(\pi_2)"] \arrow[d, "\delta"] \arrow[dr, phantom, "*"] & \Sigma^*(A) \arrow[d, "\gamma"] \\
A \arrow[d, "l"'] & K_l \arrow[l, "\pi_1"] \arrow[r, "\pi_2"'] \arrow[d, phantom, "*"] & A \arrow[d, "l"] \\
B \arrow[rr, equal] & & B
\end{tikzcd}$$

For the upper two squares, this can be easily checked; the two squares in the middle commute by the definition of Σ^*-congruence; and the rectangle below commutes by the definition of kernel. Together this implies the commutativity of the outer rectangle: $l \circ \gamma \circ \rho_1 = l \circ \gamma \circ \rho_2$, thus $K_{\Sigma^*(l)} \subseteq K_{l \circ \gamma}$.

Conversely, suppose that the outer rectangle of the diagram above commutes. (The upper two squares always commute.) It follows that

$$l \circ \gamma \circ \Sigma^*(\pi_1) = l \circ \gamma \circ \Sigma^*(\pi_2),$$

which shows that the function $\langle \gamma \circ \Sigma^*(\pi_1), \gamma \circ \Sigma^*(\pi_2)\rangle$ maps from $\Sigma^*(K_l)$ to K_l, thus turning K_l into a Σ^*-congruence on (A, γ). □

4 From $\mathcal{F}$ to $\mathcal{I}$ in a General Setting

In this section a rather straightforward categorical abstraction of the above 'from $\mathcal{F}$ to $\mathcal{I}$' construction is made. The resulting general scheme will be instantiated in Section 5 to trace equivalence semantics. A dual construction, say 'from $\mathcal{I}$ to $\mathcal{F}$', will also be described in this section, although possible semantical applications are not addressed in this paper.

4.1 From *Set* to Arbitrary Categories

The category *Set* generalizes to an arbitrary category $\mathcal{C}$. That is, objects and arrows rather than sets and functions.

The endo-functors $\Sigma^*, \mathcal{P}_f(A \times \text{-}) : Set \to Set$ generalize to arbitrary endo-functors

$$F, G : \mathcal{C} \to \mathcal{C}.$$

The categories of algebras and coalgebras of endo-functors in Set have evident corresponding notions in $\mathcal{C}$. Thus $\mathcal{C}^F$ and $\mathcal{C}_G$ in place of Set^F and Set_G, respectively.

For the definition of F-congruences and G-bisimulations one needs to abstract from relations as subsets and consider relations as subobjects. That is, a *relation* R on two objects A and B of $\mathcal{C}$ is a subobject of $A \times B$. (See §V.7 in [Lan71] for the definition of subobjects.) Notice that we need $\mathcal{C}$ to have finite products in order to define relations this way. Also, recall that relations (as subobjects) are partially ordered (write $R \leq R'$) and that, if $\mathcal{C}$ has arbitrary pullbacks, the intersection of subobjects (hence of relations) is well-defined. Under additional hypotheses, also the union is well-defined.

The *kernel pair* (see §III.4 in [Lan71]) K_f of an arbitrary arrow $f : A \to B$ in $\mathcal{C}$ generalizes the notion of a kernel of a function; it is a subobject of $A \times A$. Lemma 3.1 generalizes to arbitrary categories, by simply putting $K_f \leq K_h$ in place of $K_f \subseteq K_h$. The same holds for the Extension Lemma 3.2 with an arbitrary endo-functor F in place of Σ^*.

4.2 From $\mathcal{F}$ to $\mathcal{I}$

Let F and G be endo-functors on a category $\mathcal{C}$ as described above and assume the existence of the following. An initial F-algebra

$$\phi : FT \xrightarrow{\sim} T;$$

a final G-coalgebra

$$\psi : P \xrightarrow{\sim} GP;$$

and an initial $(P + F)$-algebra $\mathcal{T}_P \cong P + F\mathcal{T}_P$

$$\begin{tikzcd}
F\mathcal{T}_P \arrow[d, "\kappa"'] & \\
\mathcal{T}_P & P \arrow[l, "\iota"]
\end{tikzcd}$$

By initiality, the following diagram commutes.

$$\begin{tikzcd}
FT \arrow[r, "F\mathcal{I}_P"] \arrow[d, "\phi\,\wr"'] & F\mathcal{T}_P \arrow[d, "\kappa"] \\
T \arrow[r, "\mathcal{I}_P"'] & \mathcal{T}_P
\end{tikzcd}$$

Now, for two coalgebras $\alpha : T \to GT$ and $\alpha_P : \mathcal{T}_P \to G\mathcal{T}_P$ such that the diagram

$$\begin{tikzcd}
T \arrow[r, "\mathcal{I}_P"] \arrow[d, "\alpha"'] & \mathcal{T}_P \arrow[d, "\alpha_P"] & P \arrow[l, "\iota"'] \arrow[d, "\psi\,\wr"'] \\
GT \arrow[r, "G\mathcal{I}_P"'] & G\mathcal{T}_P & GP \arrow[l, "G\iota"]
\end{tikzcd}$$

commutes, one obtains the following commutative diagram.

$$\begin{tikzcd}
FT \arrow[r, "F\mathcal{I}_P"] \arrow[d, "\phi"', "\wr"] & F\mathcal{T}_P \arrow[r, shift right, "F\mathcal{F}_P"'] \arrow[d, "\kappa"] & FP \arrow[l, shift right, "F\iota"'] \\
T \arrow[r, "\mathcal{I}_P"] \arrow[d, "\alpha"'] & \mathcal{T}_P \arrow[r, shift right, "\mathcal{F}_P"'] \arrow[d, "\alpha_P"] & P \arrow[l, shift right, "\iota"'] \arrow[d, "\psi"', "\wr"] \\
GT \arrow[r, "G\mathcal{I}_P"'] & G\mathcal{T}_P \arrow[r, shift right, "G\mathcal{F}_P"'] & GP \arrow[l, shift right, "G\iota"']
\end{tikzcd} \tag{1}$$

Theorem 4.1 *Under the hypotheses of this section, the composition*

$\mathcal{F}_P \circ \mathcal{I}_P : T \to P$ is both initial and final

if $K_{\mathcal{F}_P}$ is an F-congruence on $(\mathcal{T}_P, \kappa)$. □

In the proof of the above theorem, P is turned into an F-algebra as before, by defining $\beta : FP \to P$ by $\beta \equiv \mathcal{F}_P \circ \kappa \circ F\iota$.

4.3 From $\mathcal{I}$ to $\mathcal{F}$ ***

The above construction can be (easily) dualized in order to obtain $\mathcal{F}$ from $\mathcal{I}$. The semantic importance of this dual construction is left to be discussed elsewhere. Let us just make the necessary dualizations explicit and mention that it might be possible to apply it for deriving transition systems from denotational definitions.

In addition to an initial F-algebra and a final G-coalgebra, one now needs a final $(T \times G)$-coalgebra $P_T \cong T \times GP_T$, with projections

$$\begin{tikzcd}
T & P_T \arrow[l, "\pi"] \arrow[d, "\rho"] \\
 & GP_T
\end{tikzcd}$$

By finality, the following diagram commutes.

$$\begin{tikzcd}
P_T \arrow[r, "\mathcal{F}_T"] \arrow[d, "\rho"'] & P \arrow[d, "\psi"', "\wr"] \\
GP_T \arrow[r, "G\mathcal{F}_T"'] & GP
\end{tikzcd}$$

The two algebras needed are now of the form $\alpha : FP \to P$ (say, the denotations) and $\alpha_T : FP_T \to P_T$, and such that the diagram

$$\begin{tikzcd}
FT \arrow[d, "\phi"', "\wr"] & FP_T \arrow[l, "F\pi"'] \arrow[r, "F\mathcal{F}_T"] \arrow[d, "\alpha_T"] & FP \arrow[d, "\alpha"] \\
T & P_T \arrow[l, "\pi"] \arrow[r, "\mathcal{F}_T"'] & P
\end{tikzcd}$$

commutes. Pasting these diagrams together one obtains the following commutative diagram.

$$\begin{tikzcd}
FT \arrow[r, shift left, "F\mathcal{I}_T"] \arrow[d, "\phi"', "\wr"] & FP_T \arrow[l, shift left, "F\pi"] \arrow[r, "F\mathcal{F}_T"] \arrow[d, "\alpha_T"] & FP \arrow[d, "\alpha"] \\
T \arrow[r, shift left, "\mathcal{I}_T"] & P_T \arrow[l, shift left, "\pi"] \arrow[r, "\mathcal{F}_T"'] \arrow[d, "\rho"] & P \arrow[d, "\psi"', "\wr"] \\
GT \arrow[r, shift left, "G\mathcal{I}_T"] & GP_T \arrow[l, shift left, "G\pi"] \arrow[r, "G\mathcal{F}_T"'] & GP
\end{tikzcd}$$

The composition $\beta \equiv G\pi \circ \rho \circ \mathcal{I}_T$ is then the coalgebraic structure to be added to the initial algebra T.

5 From $\mathcal{F}$ to $\mathcal{I}$ for Trace Equivalence

The derivation of $\mathcal{I}$ from $\mathcal{F}$ will be now studied for *trace equivalence*. The final arrow $\mathcal{F}$ will be the *linear final semantics* given in [TJ93]. The category involved will be the (monoidal closed) category of *semi-lattices*. The use of semi-lattices in trace equivalence semantics dates back at least to [HP79]. The basic observation there is that the computational operations of non-deterministic choice and sequential composition are best modelled as the *join* operation on a semi-lattice and as the *tensor product* between semi-lattices, respectively.

Next, we shall:

- recall the definition of the category SL of semi-lattices;
- review the *linear final semantics* given in [TJ93] (for this we shall need *right tensor products*);
- extend the endofunctor Σ^* on sets to the category of semi-lattices (for this we shall need *tensor products*) and show it has an initial algebra;
- apply the construction of Section 4.2 to the language in Section 3.6 and obtain a compositional trace equivalence semantics for it.

5.1 Semi-Lattices in *Set*

Semi-lattices are pairs $(L, \sqcup)$, with L a set and $\sqcup : L \times L \to L$ a function satisfying the following axioms. For all x, y, z in L,

$$\begin{aligned} x \sqcup (y \sqcup z) &= (x \sqcup y) \sqcup z \\ x \sqcup y &= y \sqcup x \\ x \sqcup x &= x. \end{aligned}$$

The *join* $\sqcup$ of a semi-lattice $(L, \sqcup)$ induces the following order on it. For every x and y in L,

$$x \leq y \equiv x \sqcup y = y \tag{2}$$

Semi-lattices form a category[1], called SL, by taking as arrows $f : (L, \sqcup) \to (K, \vee)$ those functions $f : L \to K$ such that, for every pair (x, y) in $L \times L$,

$$f(x \sqcup y) = f(x) \vee f(y).$$

In the sequel, we shall simply write L for a semi-lattice, living its join implicit. When we want to refer to the set underlying L, we shall write $|L|$. Notice that this operation of 'forgetting' is functorial and we shall also use the symbol U to denote this forgetful functor from SL to Set.

5.1.1 Free Semi-Lattices

Let $\hat{\mathcal{P}}$ be the restriction of the functor $\mathcal{P}_f$ (defined in Section 2) to non-empty sets. It is a standard fact that $\hat{\mathcal{P}}$ is left adjoint to the forgetful functor $U : SL \to Set$. Formally, $\hat{\mathcal{P}}$ is an endofunctor on Set, but in the sequel we shall use the same symbol also for the functor from Set to SL; for any set A, the join of two elements X and Y of $\hat{\mathcal{P}}A$ (thus X and Y are non-empty finite subsets of A) is simply their set-theoretic binary union. The unit of the adjunction $\hat{\mathcal{P}} \dashv U$ maps every element of a set into the singleton set containing it:

$$\{\cdot\}_A : A \to \hat{\mathcal{P}}A \qquad\qquad a \longmapsto \{a\}.$$

One way of stating (and proving) that $\hat{\mathcal{P}} : Set \to SL$ is left adjoint to $U : SL \to Set$ is as follows.

Proposition 5.1 *For every set A and semi-lattice L, and for every function $f : A \to |L|$ there exists a unique linear function $f^\sharp : \hat{\mathcal{P}}A \to L$ such that $f = f^\sharp \circ \{\cdot\}_A$. Pictorially:*

$$\begin{tikzcd}
 & Set & & SL \\
A \arrow[r, "\{\cdot\}_A"] \arrow[dr, "f"'] & \hat{\mathcal{P}}A \arrow[d, dashed, "f^\sharp"] & & \hat{\mathcal{P}}A \arrow[d, dashed, "f^\sharp"] \\
 & {|L|} & & L
\end{tikzcd}$$

Proof. Immediate from the linearity of f and the fact that every non-empty and finite subset is a finite union of singleton sets. □

[1] In general, one can define semi-lattices 'internally' in any category with binary products. For instance, in [HP79], semi-lattices in the category of cpo's and strict continuous functions are used.

The semi-lattice $\hat{\mathcal{P}}A$ is called the *free semi-lattice* on A.

An alternative description of the same adjunction is given by the following bijection.

$$\frac{A \xrightarrow{\quad f \quad} |L|}{\hat{\mathcal{P}}A \xrightarrow{\quad f^{\sharp} \quad} L} \tag{3}$$

Since the functor $\hat{\mathcal{P}} : Set \longrightarrow SL$ has a right adjoint it preserves colimits (see §V.5 in [Lan71]). In particular it preserves coproducts and, since SL is cocomplete (see Corollary 5.3), one has, for all sets A and B,

$$\hat{\mathcal{P}}A + \hat{\mathcal{P}}B \;\cong\; \hat{\mathcal{P}}(A+B). \tag{4}$$

(Notice that this isomorphism holds in SL but not in Set.) The same holds for the initial object:

$$0 \;\cong\; \hat{\mathcal{P}}0. \tag{5}$$

Clearly, the 0 on the left hand side is in SL, while the one on the right hand side is in Set. Thus the empty semi-lattice is the initial object in SL. The dual fact also holds: the one-element semi-lattice is the final object in SL and

$$1 \;\cong\; \hat{\mathcal{P}}1. \tag{6}$$

5.1.2 Completeness and Co-Completeness***

Since the forgetful functor $U : Set \longrightarrow SL$ is monadic, an alternative presentation of SL is as the category $Set^{\mathbf{T}}$ of algebras of the monad $\mathbf{T} = \langle \hat{\mathcal{P}}, \{\text{-}\}, \cup \rangle$ given by the above adjunction (see §VI.2 in [Lan71]):

Proposition 5.2 $\qquad SL \cong Set^{\mathbf{T}}$

Then the following corollary is immediate (see Proposition 9.3.4 in [BW85]).

Corollary 5.3 *The category SL is (small-) cocomplete.*

By the completeness of Set and the fact that the functor $U : SL \longrightarrow Set$ creates limits (see §VI.2 in [Lan71]), one has also the dual fact:

Corollary 5.4 *The category SL is (small-) complete.*

5.2 Linear Final Semantics

In this section we shall review the final semantics for trace equivalence – called *linear final semantics* – given in [TJ93].

Traces (both finite and infinite) on an alphabet of actions A can be categorically described as elements of the (unique up to isomorphism) final coalgebra of the following endofunctor on Set (cf. Section 2.4),

$$1 + A \times \text{-} \; : Set \longrightarrow Set\,.$$

In trace equivalence semantics, the meaning of a program is usually given as a *set* of traces. By using a suitable endofunctor on semi-lattices instead, namely

$$1 + A \otimes^{r} \text{-} \;\; : SL \longrightarrow SL\,, \tag{7}$$

we shall be able to give a final coalgebra description of those sets of traces of interest for semantics. The functor $\otimes^r$ appearing in (7) is of type

$$\otimes^r : Set \times SL \to SL \qquad (8)$$

and is obtained by precomposing the first component of the *tensor product* of semi-lattices[2] with the free constructor $\hat{\mathcal{P}}$ on semi-lattices. One way to describe this *'right' tensor product* $\otimes^r$, independently from the 'full' tensor product $\otimes$, is as 'classifying' *right-linear* functions in the following sense. Given a set A and two semi-lattices L and K, a right-linear function $f : A\times |L| \to |K|$ is a function which is linear in the right component. That is, for every a in A and all l, l' in L,

$$f(a, l \sqcup l') = f(a, l) \sqcup f(a, l').$$

A semi-lattice, say $A \otimes^r L$, classifies right-linear functions if there exists a right-linear function

$$\otimes^r : A\times |L| \to |A \otimes^r L|$$

which is universal in the sense that, for every right-linear function $f : A\times |L| \to |K|$, there exists a unique *linear* function $f^{\natural}$ such that the following diagram commutes.

$$\begin{tikzcd}
A\times|L| \arrow[r, "\otimes^r"] \arrow[dr, "f"'] & {|A\otimes^r L|} \arrow[d, dashed, "f^{\natural}"] & & A\otimes^r L \arrow[d, dashed, "f^{\natural}"] \\
& {|K|} & & K
\end{tikzcd} \qquad (9)$$

(Right-Linear) (Linear)

The existence of such a universal right-linear function is shown in Section 5.5 where it is also shown that it gives rise to a functor of type as in (8). (The idea of a right tensor is already present in [Hen93], but here we let A be an object from *Set* rather than from *SL*.)

An important property of this right tensor is the following. For every two sets A and B,

$$A \otimes^r \hat{\mathcal{P}} B \cong \hat{\mathcal{P}}(A\times B). \qquad (10)$$

(For its proof, see Corollary 5.13.)

5.2.1 A Linear Final Coalgebra

Theorem 5.5 *There exists a final coalgebra $\psi : P \xrightarrow{\sim} 1 + A \otimes^r P$ of the endofunctor*

$$1 + A \otimes^r - : SL \to SL.$$

Moreover, P is the limit of the following ω^{op}-chain.

$$1 \xleftarrow{\ !\ } 1 + A\otimes^r 1 \xleftarrow{1+A\otimes^r !} 1 + A\otimes^r(1 + A\otimes^r 1) \longleftarrow \cdots \qquad (11)$$

[2] The use of tensor products in trace equivalence semantics dates back at least to [HP79].

Proof. Let μ be the universal cone given by the above limit and put

$$\nu \equiv 1 + A \otimes^r \mu.$$

This ν is a cone from $1 + A \otimes^r$- to the diagram in (11), which gives a unique linear function

$$\xi : 1 + A \otimes^r P \rightarrow P$$

such that $\nu = \mu \circ \xi$. The claim is that there exists a linear function $\psi : P \rightarrow 1 + A \otimes^r P$ such that $\psi = \xi^{-1}$. In order to see this, let us first have a more concrete understanding of the limit P. Put, for every p in P,

$$p[n] \equiv \mu_n(p). \tag{12}$$

Next, let F_n and f_n denote the n-th object and the n-th arrow in the above diagram, respectively. By (10), (6) and (4), one has

$$F_n \cong \hat{\mathcal{P}}(\textstyle\coprod_{k \leq n} A^k) \tag{13}$$

By the standard construction of limits by products and equalizers (see §V.2 in [Lan71]), the above limit P can be concretely described as follows.

$$|P| \cong \{x \in \prod_{n<\omega} \hat{\mathcal{P}}(\textstyle\coprod_{k \leq n} A^k) \mid \pi_n(x) = f_n(\pi_{n+1}(x))\} \tag{14}$$

The join of P is inherited from the product $\prod_{n<\omega} \hat{\mathcal{P}}(\coprod_{k\leq n} A^k)$, on which the join is defined pointwise.

Firstly, notice that, for every p in P and every a in A, if a belongs to $p[1]$, then, for every $n \geq 1$, there exists a w in F_{n-1} such that $< a, w >$ is in $p[n]$.[3] One can then define a partial function

$$-_a : P \rightharpoonup P \tag{15}$$

as follows. Regard an infinite tuple as function from the natural numbers; then,

$$p_a = \begin{cases} n \longmapsto \{w \mid < a, w > \in p[n]\} & \text{if } a \in p[1] \\ \text{undefined} & \text{otherwise} \end{cases}$$

Secondly, notice that, if 0 belongs to $p[1]$, then it belongs to every $p[n]$.

We can now define, for every p in P,

$$\psi(p) \equiv \begin{cases} 0 \sqcup \bigsqcup_{a \in p[1]} a \otimes p_a & \text{if } 0 \in p[1] \\ \bigsqcup_{a \in p[1]} a \otimes p_a & \text{otherwise} \end{cases}$$

One can then check that ψ is linear and that it is the inverse of ξ. □

5.2.2 Linear Transition Systems

In Section 2, a one-to-one correspondence is given between finitely branching labelled transition systems (over A) and coalgebras of the endo-functor $\mathcal{P}_f(A \times -)$ on *Set*. The existence of a final coalgebra for that functor yields a 'branching' final semantics for any transition system. Here it is shown that something similar – although of a 'linear' rather

[3] Actually, to be formal, we should write $inr(a)$ and $inr(a, w)$, because the $p[i]$'s are subsets of the coproduct $1 + (A + \ldots + A^i)$ with injections inl: $1 \rightarrow 1 + (A + \ldots + A^i)$ and inr: $(A + \ldots + A^i) \rightarrow 1 + (A + \ldots + A^i)$. But this would make the notation too cumbersome in the sequel.

than of a branching nature – can be done for the endo-functor $1 + A \otimes^{r} \text{-}$ on SL when applied to free semi-lattices.

Firstly, notice that, by (10), (6), and (4),

$$1 + A \otimes^{r} \hat{\mathcal{P}}\text{-} \;\cong\; \hat{\mathcal{P}}(1 + A \times \text{-}) \tag{16}$$

Secondly, coalgebras in *Set* of the form

$$S \longrightarrow \hat{\mathcal{P}}(1 + A \times S)$$

can be seen as (finitely branching) labelled transition systems with a distinguished termination state, say a state 0 not belonging to S. Indeed, one can write these transition systems as tuples $\langle S, A, \rightarrow, \downarrow \rangle$, where the last symbol denotes the following predicate ranging over S: for every s in S,

$$s \downarrow \text{ holds } \iff s \text{ can do a transition into the termination state } 0$$

Thirdly, using the bijection (3) given by the adjunction $\hat{\mathcal{P}} \dashv U$, one obtains the isomorphism

$$\frac{S \xrightarrow{\alpha} \hat{\mathcal{P}}(1 + A \times S)}{\hat{\mathcal{P}}S \xrightarrow{\alpha^{\sharp}} 1 + A \otimes^{r} \hat{\mathcal{P}}S} \tag{17}$$

Therefore, $(1{+}A\otimes^{r}\text{-}\,)$-coalgebras of free semi-lattices are just transition systems transposed across the adjunction and one can infer a linear final semantics for them as follows.

Linear Final Semantics:

$$\begin{tikzcd}
S \arrow[r, "\{\text{-}\}_S"] \arrow[dr, "\alpha"'] & \hat{\mathcal{P}}S \arrow[d, dashed, "\alpha^{\sharp}"] & \hat{\mathcal{P}}S \arrow[rr, "\mathcal{F}"] \arrow[d, dashed, "\alpha^{\sharp}"] & & P \arrow[d, "\psi"', "\wr"] \\
 & \hat{\mathcal{P}}(1{+}A{\times}S) & 1 + A \otimes^{r} \hat{\mathcal{P}}S \arrow[rr, "1 + A \otimes^{r} \mathcal{F}"'] & & 1 + A \otimes^{r} P
\end{tikzcd} \tag{18}$$

One can regard the coalgebra $(\hat{\mathcal{P}}S, \alpha^{\sharp})$ as a *free linear transition system* (where by a transition system we intend now a finitely branching one with a suitable distinguished 'termination state') as follows. For all finite and non-empty subsets S_1, S_2 of S,

$$S_1 \xrightarrow{a} S_2 \iff \forall s' \in S_2\ \exists s \in S_1\ \ s \xrightarrow{a} s'. \qquad S_1 \downarrow \iff \exists s \in S_1\ \ s \downarrow \tag{19}$$

This definition follows from the more general case in which a transition relation and a termination predicate are derived from an arbitrary coalgebra $\beta : L \rightarrow 1 + A \otimes^{r} L$: for every a in A, and all l, l' in L,

$$l \xrightarrow{a} l' \iff a \otimes^{r} l' \leq \beta(l) \qquad l \downarrow \iff 0 \leq \beta(l) \tag{20}$$

In the sequel, when applying $\mathcal{F}$ to an element, we shall write

$[\![\text{-}]\!]$ in place of $\mathcal{F}(\text{-})$. **(notational convention)**

Diagram (18) implies that, up to isomorphism,

$$[\![S_1]\!] = \begin{cases} 0 \sqcup \bigsqcup_{S_1 \xrightarrow{a} S_2} a \otimes^{r} [\![S_2]\!] & \text{if } S_1 \downarrow \\ \bigsqcup_{S_1 \xrightarrow{a} S_2} a \otimes^{r} [\![S_2]\!] & \text{otherwise} \end{cases}$$

In particular, writing s for the singleton state $\{s\}$ in $\langle \hat{\mathcal{P}}S, A, \rightarrow, \downarrow \rangle$, one can derive from (19)

$$s \xrightarrow{a} S' \iff \forall s' \in S' \; s \xrightarrow{a} s' \tag{22}$$

and, exploiting the linearity of the semantic function, one can derive

$$[\![s]\!] = \begin{cases} 0 \sqcup \bigsqcup_{s \xrightarrow{a} s'} a \otimes^r [\![s']\!] & \text{if } s \downarrow \\ \bigsqcup_{s \xrightarrow{a} s'} a \otimes^r [\![s']\!] & \text{otherwise} \end{cases} \tag{linear}$$

Notice that in branching semantics one has

$$[\![s]\!] = \begin{cases} \{0\} \cup \bigcup_{s \xrightarrow{a} s'} \{< a, [\![s']\!] >\} & \text{if } s \downarrow \\ \bigcup_{s \xrightarrow{a} s'} \{< a, [\![s']\!] >\} & \text{otherwise} \end{cases} \tag{branching}$$

The difference between the two semantics becomes apparent when a further iteration step is considered.

Let us come back to the correspondence, given in (20), between arbitrary coalgebras of the functor $1 + A \otimes^r$ - and (linear) transition systems; it will be used later for deriving $\mathcal{I}$ from $\mathcal{F}$ in the linear setting.

Firstly, to any coalgebra $\beta : L \rightarrow 1 + A \otimes^r L$ one can associate the following final 'semantics'. For every l in L,

$$[\![l]\!] = \begin{cases} 0 \sqcup \bigsqcup_{l \xrightarrow{a} l'} a \otimes^r [\![l']\!] & \text{if } l \downarrow \\ \bigsqcup_{l \xrightarrow{a} l'} a \otimes^r [\![l']\!] & \text{otherwise} \end{cases}$$

Secondly, notice that the correspondence in (20) shows how to derive a linear transition system from an arbitrary coalgebra of the functor $1 + A \otimes^r$ - but not the converse; namely, it is not clear to which transition systems $\langle L, A, \rightarrow, \downarrow \rangle$ the correspondence in (20) can be applied in order to obtain a *linear* function $\beta : L \rightarrow 1 + A \otimes^r L$. Certainly, in order for β to be linear, the following three conditions should hold. Their interest lies in particular in the fact that, as it will later be shown, they can be regarded as transition system specification rules, and used to derive a linear transition system from an ordinary specification.

For all l_1, l_1', l_2, l_2' in L,

$$\begin{array}{lll} l_1 \xrightarrow{a} l_1', \;\; l_2 \xrightarrow{a} l_2' & \Rightarrow & l_1 \sqcup l_2 \xrightarrow{a} l_1' \sqcup l_2' \\ l_1 \xrightarrow{a} l_1', \;\; l_2 \not\xrightarrow{a} & \Rightarrow & l_1 \sqcup l_2 \xrightarrow{a} l_1' \\ l_1 \downarrow & \Rightarrow & l_1 \sqcup l_2 \downarrow \end{array} \tag{25}$$

5.3 Σ-Algebras in *SL*

The description of the endofunctor Σ^* on *Set* associated to a signature Σ can easily be made categorical and then instantiated to *SL*. Let (Σ, r) be a ranked alphabet and recall that in *Set*

$$\Sigma^* \cong \coprod_{f \in \Sigma} (\text{-})^{r(f)} \tag{26}$$

The above functor is described in terms of coproducts and *Cartesian* products, but, from semantical consideration (namely, the bilinearity of the operators – but see also the construction in [Plo81a] of the initial Σ-algebra in the category of cpo's with strict continuous functions) it turns out that in general the *tensor product* instead of the Cartesian product has to be used. (In *Set* Cartesian and tensor product coincide – see below.)

The tensor product of two semi-lattices is a functor of the form

$$\otimes : SL \times SL \rightarrow SL$$

which 'classifies' *bilinear* functions. Bilinear functions are functions $f : |L_1| \times |L_2| \rightarrow |L|$ which are linear in each of the two components *separately*. That is, for all l_1, l_1' in L_1 and all l_2, l_2' in L_2,

$$f(l_1 \sqcup l_1', l_2) = f(l_1, l_2) \sqcup f(l_1', l_2) \qquad f(l_1, l_2 \sqcup l_2') = f(l_1, l_2) \sqcup f(l_1, l_2') \tag{27}$$

Equivalently,

$$f(l_1 \sqcup l_1', l_2 \sqcup l_2') = f(l_1, l_2) \sqcup f(l_1', l_2) \sqcup f(l_1, l_2') \sqcup f(l_1', l_2') \tag{28}$$

The tensor product $L_1 \otimes L_2$ of two semi-lattices L_1 and L_2 can be defined in terms of the existence of a bilinear function $\circledast : |L| \times |K| \rightarrow |L \otimes K|$ which is universal among all bilinear functions. That is, for every bilinear function $f : |L_1| \times |L_2| \rightarrow |L|$ there exists a unique *linear* function $f^\natural$ such that the following diagram commutes.

$$\begin{tikzcd}
|L_1| \times |L_2| \arrow[r, "\circledast"] \arrow[dr, "f"'] & |L_1 \otimes L_2| \arrow[d, dashed, "f^\natural"] & & L_1 \otimes L_2 \arrow[d, dashed, "f^\natural"] \\
 & |L| & & L
\end{tikzcd} \tag{29}$$

(Bi-Linear) (Linear)

(For more details, see Section 5.5.)

The tensor product of two free semi-lattices has the following useful property. For every two sets A and B,

$$\hat{\mathcal{P}}A \otimes \hat{\mathcal{P}}B \cong \hat{\mathcal{P}}(A \times B). \tag{30}$$

(For the proof of this fact see Proposition 5.12.)

One can now define powers

$$(\text{-})^n : SL \rightarrow SL$$

of the tensor product inductively as follows. For every natural number n,

$$\begin{aligned} (\text{-})^0 &\equiv 1 \\ (\text{-})^{n+1} &\equiv 1_{SL} \otimes \cdot (\text{-})^n \end{aligned}$$

Here 1_{SL} stands for the identity functor, while 1 is the final object (which is also the neutral object for the tensor).

Using this definition of powers, the expression in (26) can be taken as a definition for an endofunctor $\Sigma^* : SL \rightarrow SL$. For example, if Σ is a signature consisting of one constant, one unary and one binary function symbol, then, for every semi-lattice L,

$$\Sigma^*(L) = 1 + L + L \otimes L.$$

5.3.1 Initial Σ^*-Algebras in *SL*

Proposition 5.6 *The endofunctor $\Sigma^* : SL \to SL$ is cocontinuous, (i.e., it preserves colimits). The same holds for the endofunctor $L + \Sigma^* : SL \to SL$, for any semi-lattice L.*

Proof. The endofunctor $(L+)\ \Sigma^*$ is defined in terms of composition of coproducts, tensor products, (constant) and identity functors. The latter are trivially cocontinuous. Coproducts are colimits and colimits commute with each other; therefore they are cocontinuous. The same hold for tensor products, since, as shown in Section 5.5, they are coequalizers. Finally, the composition of cocontinuous functors is always cocontinuous. □

Corollary 5.7 *The initial algebra of the endofunctor Σ^* exists and is the colimit of the following ω-chain.*

$$\begin{tikzcd} 0 \arrow[r, "!"] & \Sigma^*(0) \arrow[r, "\Sigma^*(!)"] & \Sigma^*(\Sigma^*(0)) \arrow[r, "\Sigma^*(\Sigma^*((!))"] & \cdots \end{tikzcd} \tag{31}$$

Similarly, the initial algebra of the endofunctor $P+\Sigma^*$ is the colimit of

$$\begin{tikzcd} 0 \arrow[r, "!"] & P+\Sigma^*(0) \arrow[r, "P+\Sigma^*(!)"] & P+\Sigma^*(P+\Sigma^*(0)) \arrow[r] & \cdots \end{tikzcd}$$

□

In the sequel we shall use Σ^* both for the endofunctor in *Set* and for that in *SL*. The context should help in avoiding ambiguities. For instance, in the next proposition, the Σ^* on the left hand side of the isomorphism is on *SL* while the one on the right is on *Set*. Instead, the two occurrences of $\hat{\mathcal{P}}$ both refer to the functor from *Set* to *SL*.

Proposition 5.8 $\quad \Sigma^* \circ \hat{\mathcal{P}} \cong \hat{\mathcal{P}} \circ \Sigma^*.$

Proof.

$$\begin{aligned} \Sigma^* \circ \hat{\mathcal{P}} &\cong \coprod_{f\in\Sigma} (\hat{\mathcal{P}}\,\text{-}\,)^{r(f)} \\ &\cong \coprod_{f\in\Sigma} \hat{\mathcal{P}}((\text{-})^{r(f)}) && \text{(by Proposition 5.12)} \\ &\cong \hat{\mathcal{P}}(\coprod_{f\in\Sigma} (\text{-})^{r(f)}) && \text{(by (4))} \\ &\cong \hat{\mathcal{P}} \circ \Sigma^* \end{aligned}$$

□

The above proposition can be used to give a concrete representation of the initial Σ^*-algebra in *SL*. (A Σ^*-algebra in *SL* is a pair consisting of a semi-lattice L and a linear function $\alpha : \Sigma^* L \to L$.)

Proposition 5.9 *Let (T, ϕ) be the initial Σ^*-algebra in Set (see Section 1.1.2). Then the initial Σ^*-algebra in SL is simply $(\hat{\mathcal{P}}T, \hat{\mathcal{P}}\phi)$.*

Proof. By the cocompleteness of *Set* and the fact that the forgetful functor $U : SL \to Set$ *creates filtered colimits* (see §IX.1 in [Lan71]), Corollary 5.7 applies to *SL*. By (5) (for the first element) and Proposition 5.8 (for the iterated steps), Diagram (31) in *SL* can be obtained by applying $\hat{\mathcal{P}}$ to the corresponding diagram in *Set*. By applying the fact that $\hat{\mathcal{P}}$ – as a left adjoint – preserves colimits, one obtains the desired result. □

5.4 From $\mathcal{F}$ to $\mathcal{I}$ in Linear Semantics

The existence of a final coalgebra $P \overset{\sim}{\to} 1 + A \otimes^r P$, and initial algebras $\Sigma^*\hat{\mathcal{P}}T \overset{\sim}{\to} \hat{\mathcal{P}}T$ and $P + \Sigma^* \mathcal{T}_P \overset{\sim}{\to} \mathcal{T}_P$, allows us to apply the formal construction described in Section 4 to the endofunctors Σ^* and $1 + A \otimes^r$ - on *SL*. From the form of the initial Σ^*-algebra (see Propositions 5.9 and 5.8) and from the correspondence (depicted in (18)) between transition systems (in *Set*) and free linear ones (in *SL*), one has the following.

$$\begin{tikzcd}
\mathit{Set} & & \mathit{SL} \\
\Sigma^* T \arrow[d, "\phi", "\wr"'] & & \hat{\mathcal{P}}\Sigma^* T \cong \Sigma^* \hat{\mathcal{P}} T \arrow[d, "\hat{\mathcal{P}}\phi"', "\wr"] \\
T \arrow[r, "\{\text{-}\}_T"] \arrow[dr, "\alpha"'] & \hat{\mathcal{P}}T \arrow[d, dashed, "\alpha^\sharp"] & \hat{\mathcal{P}}T \arrow[d, dashed, "\alpha^\sharp"'] \\
& \hat{\mathcal{P}}(1 + A \times T) & 1 + A \otimes^r \hat{\mathcal{P}}T
\end{tikzcd} \qquad (32)$$

If one can find a coalgebra

$\alpha_P^\sharp : \mathcal{T}_P \to 1 + A \otimes^r \mathcal{T}_P$

such that the diagram

$$\begin{tikzcd}
\hat{\mathcal{P}}T \arrow[r, "\mathcal{I}_P"] \arrow[d, "\alpha^\sharp"'] & \mathcal{T}_P \arrow[d, "\alpha_P^\sharp"] & P \arrow[l, "\iota"'] \arrow[d, "\psi"', "\wr"] \\
1 + A \otimes^r \hat{\mathcal{P}}T \arrow[r, "1 + A \otimes^r \mathcal{I}_P"'] & 1 + A \otimes^r \mathcal{T}_P & 1 + A \otimes^r P \arrow[l, "1 + A \otimes^r \iota"']
\end{tikzcd} \qquad (33)$$

commutes, then also the following diagram – instance of (1) – commutes.

$$\begin{tikzcd}
\hat{\mathcal{P}}\Sigma^* T \cong \Sigma^* \hat{\mathcal{P}} T \arrow[r, "\Sigma^* \mathcal{I}_P"] \arrow[d, "\hat{\mathcal{P}}\phi"', "\wr"] & \Sigma^* \mathcal{T}_P \arrow[d, "\kappa"] \arrow[r, shift right, "\Sigma^* \mathcal{F}_P"'] & \Sigma^* P \arrow[l, shift right, "\Sigma^* \iota"'] \\
\hat{\mathcal{P}}T \arrow[r, "\mathcal{I}_P"] \arrow[d, "\alpha^\sharp"'] & \mathcal{T}_P \arrow[d, "\alpha_P^\sharp"] \arrow[r, shift right, "\mathcal{F}_P"'] & P \arrow[l, shift right, "\iota"'] \arrow[d, "\psi"', "\wr"] \\
1 + A \otimes^r \hat{\mathcal{P}}T \arrow[r, "1 + A \otimes^r \mathcal{I}_P"'] & 1 + A \otimes^r \mathcal{T}_P \arrow[r, shift right, "1 + A \otimes^r \mathcal{F}_P"'] & 1 + A \otimes^r P \arrow[l, shift right, "1 + A \otimes^r \iota"']
\end{tikzcd} \qquad (34)$$

Theorem 4.1 gives then that

$\mathcal{F}_P \circ \mathcal{I}_P : \hat{\mathcal{P}}T \to P$ is both initial and final

if $K_{\mathcal{F}_P}$ is a Σ^*-congruence on (P, β), for $\beta \equiv \mathcal{F}_P \circ \kappa \circ \Sigma^* \iota$.

Let us now see how to derive $\alpha_P^\sharp$ from $\alpha^\sharp$ in case α is defined by means of a transition system specification; some care is needed because of the semi-lattice structure which makes things more involved than in the branching case from Section 3.

Notice that, in general, transition system specifications do not include rules for $\sqcup$. When dealing with free semi-lattices like $\hat{\mathcal{P}}T$, this is not a problem, because one can use the by now standard transposition across the adjunction and work with T, where no join appears. But in the above diagram, the coalgebra $\alpha_P^\sharp : T_P \to 1 + A \otimes^r T_P$ involves the semi-lattice T_P (containing 'processes as terms') which is not free. A solution would be to add to every transition system specifications the conditions given in (25) as rules; that is:

$$\frac{x \xrightarrow{a} x' \quad y \xrightarrow{a} y'}{x \sqcup y \xrightarrow{a} x' \sqcup y'} \qquad \frac{x \xrightarrow{a} x' \quad y \not\xrightarrow{a}}{x \sqcup y \xrightarrow{a} x'} \qquad \frac{x \downarrow}{x \sqcup y \downarrow} \tag{35}$$

This suffices for reducing transitions of terms of the form $a \sqcup b$ to transitions of simpler terms a and b. But problems arise again when dealing with terms in which joins appear after a function symbol, like, e.g., $(a \sqcup b) \parallel (c \sqcup d)$. The solution is suggested by the way the free algebra of terms is constructed, namely by means of tensor products: the function symbols have to be treated bilinearly. Thus, for instance, one can reduce the above term to $(a \parallel c) \sqcup (a \parallel d) \sqcup (b \parallel c) \sqcup (b \parallel d)$.

Given a transition system specification $\mathcal{R}$, after adding to it the rules in (35) and having treated function symbols as bilinear, one still needs to include the 'transitions' of $P \xrightarrow{\sim} 1 + A \otimes^r P$ as axioms; by the correspondence in (20) one derives the following. Recall that, for every p in P, the expressions $p[1]$ and p_a are defined in (12) and (15), respectively. For every p in P and every $a{\otimes}^r p' \leq \psi(p)$ (i.e., $p' \leq p_a$), add the axiom

$$p \xrightarrow{a} p' \, ;$$

for every p such that $0 \leq \psi(p)$ (i.e., $0 \in p[1]$), add

$$p \downarrow .$$

As in the branching case, if $\mathcal{R}$ does not have x-rules, the commutativity of the right square of the diagram in (33) is ensured, because, as previously mentioned, the rules in (35) are already implied by transition systems stemming from coalgebras. As for the two remaining assumptions for our theory to hold, namely the commutativity of the left square of the diagram in (33) and the 'congruence condition', we do leave them to be discussed elsewhere and turn to a specific example – the transition system specification $\mathcal{R}_B$ for the language in Section 3.6. For that example the remaining assumptions can be verified 'by hand'.

Notice that a further difference from the branching case is here the fact that we now deal with transition systems with a distinguished termination state. In the transition system induced by $\mathcal{R}_B$, the state δ plays such role and transitions of the form $x \xrightarrow{\surd} \delta$ simply correspond to $x \downarrow$. Having settled this last problem, we can now derive the following compositional semantics.

$$\begin{aligned}
\epsilon_P &= [\![\epsilon]\!] \\
&= \surd{\otimes}^r 0 \\
a_P &= [\![a]\!] \\
&= a{\otimes}^r 0 \\
p \cdot_P q &= [\![p \cdot q]\!]
\end{aligned}$$

$$= \begin{cases} q \sqcup \bigsqcup_{a \in p[1]} a \otimes^r [\![p_a \cdot q]\!] & \text{if } p \downarrow \\ \bigsqcup_{a \in p[1]} a \otimes^r [\![p_a \cdot q]\!] & \text{otherwise} \end{cases}$$

Finally, consider, for simplicity, the remaining $p \ +_P \ q$ and $p \ \|_P \ q$ only in the case in which neither $p \downarrow$ nor $q \downarrow$ holds:

$$\begin{aligned} p \ +_P \ q &= [\![p + q]\!] \\ &= \bigsqcup_{a \in p[1]} a \otimes^r [\![p_a]\!] \sqcup \bigsqcup_{a \in q[1]} a \otimes^r [\![q_a]\!] \\ &= p \sqcup q \\ p \ \|_P \ q &= [\![p \parallel q]\!] \\ &= \bigsqcup_{a \in p[1]} a \otimes^r [\![p_a \parallel q]\!] \sqcup \bigsqcup_{a \in q[1]} a \otimes^r [\![p \parallel q_a]\!] \end{aligned}$$

Notice that the language and the transition system specification considered here is very close to the one given in [HP79]. One can then compare the compositional semantics in that paper with the above 'automatically derived' semantics and find that, mutatis mutandis, they are the same.

5.5 Tensor Products in SL ***

In this section tensor and right tensor products are seen first as left adjoints of suitable hom-functors and then as quotients – more precisely, as coequalizers. The former presentation has the advantage of being easily applied and understood also in other categories. The latter is useful for proving the existence of such products.

5.5.1 Tensor Products as Left Adjoints

One way to define tensor products is to use *closed categories* and adjunctions. Informally (for a formal definition see [EK66]), a category $\mathcal{C}$ is *closed* if its hom-sets are themselves objects of the category. That is, for every two objects C_1 and C_2 in $\mathcal{C}$, the set $\mathcal{C}(C_1, C_2)$ of arrows in $\mathcal{C}$ can be regarded as an object of $\mathcal{C}$. This is always the case for categories of algebras, like SL. (See, e.g., [Jac92].)

For every object C in a closed category $\mathcal{C}$, one has an endofunctor $\mathcal{C}(C, \text{ - }) : SL \rightarrow SL$. If this functor has a left adjoint, one can define a tensor product as follows:

$$C \otimes \text{ - } \dashv \mathcal{C}(C, \text{ - })$$

By a standard property of adjunctions with parameters ([Lan71]) this gives rise to a functor $\otimes : \mathcal{C} \times \mathcal{C} \rightarrow \mathcal{C}$.

Clearly, the category *Set* is closed (hom-sets are sets!) and it is well known that, for every set A, $A \times \text{ - } : Set \rightarrow Set$ is left adjoint to the hom-functor. Thus in *Set* tensor and Cartesian products coincide.

A familiar example of a tensor product in semantics is the *smash product* between cpo's. Notice that the space of strict continuous functions between two cpo's is a cpo itself, which shows that the category $CPO_\perp$ is closed. It is possible to prove that, for every cpo C, the smash product $C \otimes$ - is the left adjoint to $CPO_\perp(C, \text{ - })$.

Also SL is a closed category. For every two semi-lattices L and K, the hom-set $SL(K, L)$ inherits a semi-lattice structure as follows. For all linear functions f, g in $SL(K, L)$, and for every k in K, one can put

$$(f \sqcup g)(k) \ \equiv \ f(k) \sqcup g(k), \tag{36}$$

where the join on the left hand side is the join of L. Thus the join in $SL(K,L)$ is defined pointwise and, for every semi-lattice K, $SL(K,\ \text{-}\ )$ is an endofunctor on SL.

For proving the existence of this tensor in SL one can use *Freyd's Adjoint Functor Theorem* (§V.6 in [Lan71]) as well as a concrete quotient construction (see Section 5.5). What the adjunction then gives is the following bijection.

$$\frac{L_2 \otimes L_1 \longrightarrow L}{L_1 \longrightarrow SL(L_2, L)}$$

Summarizing the above, we have the following.

Definition 5.10 For every L in SL,

$$L \otimes \text{-} : SL \to SL$$

is defined as the (unique up to isomorphism) left adjoint of $SL(L,\ \text{-}\ ) : SL \to SL$. By a standard property of adjoints with parameters (§IV.7 in [Lan71]), this gives rise to a bifunctor

$$\otimes : SL \times SL \to SL$$

which is called the *tensor product* between semi-lattices.

One can check that $(SL, \otimes, 1)$ forms a *monoidal closed category*. In particular,

$$L \otimes 1 \cong 1 \otimes L \cong L.$$

The above definition can be easily varied to obtain a right tensor product classifying only 'right-linear' functions, instead of bilinear (i.e., right- and left-linear) ones. The hom-functor

$$Set_{SL}(\ \text{-}\ ,\ \text{-}\ ) : Set^{\mathrm{op}} \times SL \to SL$$

to be used is defined as follows. Given a set A and a semi-lattice L, $Set_{SL}(A,L)$ is the semi-lattice having as elements all functions – thus not only the linear ones! – from the set A to the set underlying L, and as join the pointwise one as in (36). This is well defined, since in (36) only the join of the target L is used.

Definition 5.11 For every A in Set,

$$A \otimes^{r} \text{-} : SL \to SL$$

is defined as the left adjoint of $Set_{SL}(A,\ \text{-}\ ) : SL \to SL$. This gives rise to a bifunctor

$$\otimes^{r} : Set \times SL \to SL$$

which is called the *right tensor product* between sets and semi-lattices.

The above adjunction is a natural isomorphism of the form

$$\frac{A \otimes^{r} L \longrightarrow K}{L \longrightarrow Set_{SL}(A,K)}$$

An alternative way of understanding the right tensor product is in terms of the 'full' tensor product. Indeed, it can be obtained by simply precomposing the left component of the tensor product with the free constructor on semi-lattices:

$$\begin{tikzcd} Set \arrow[d, "\hat{\mathcal{P}}"'] \\ SL \end{tikzcd} \qquad \begin{tikzcd} Set \times SL \arrow[d, "\hat{\mathcal{P}} \times 1_{SL}"'] \arrow[dr, "\otimes^r"] & \\ SL \times SL \arrow[r, "\otimes"'] & SL \end{tikzcd}$$

5.5.2 Tensor Products of Free Semi-Lattices

The tensor product of two free semi-lattices has the following useful property.

Proposition 5.12 *For every two sets A and B,*

$$\hat{\mathcal{P}}A \otimes \hat{\mathcal{P}}B \cong \hat{\mathcal{P}}(A \times B).$$

Proof. By uniqueness (up to isomorphism) of adjoints, it is enough to prove that, for every semi-lattice L, there exists a natural isomorphism between linear functions from $\hat{\mathcal{P}}(A \times B)$ to L and linear functions from $\hat{\mathcal{P}}B$ to $SL(\hat{\mathcal{P}}A, L)$:

$$\frac{\hat{\mathcal{P}}(A \times B) \longrightarrow L}{\hat{\mathcal{P}}B \longrightarrow SL(\hat{\mathcal{P}}A, L)}$$

Because of the adjunction $\hat{\mathcal{P}} \dashv U$, this amounts to the following natural isomorphism between functions in *Set* (thus not necessarily linear):

$$\frac{A \times B \longrightarrow |L|}{B \longrightarrow |SL(\hat{\mathcal{P}}A, L)|}$$

A second application of the above argument yields:

$$\frac{A \times B \longrightarrow |L|}{B \longrightarrow Set(A, |L|)}$$

The latter immediately follows from the adjunction $A \times - \dashv Set(A, -)$. □

Notice that since in *Set* $\times$ coincides with $\otimes$ one could rephrase the above proposition by saying that $\hat{\mathcal{P}}$ (as a monad) preserves tensors. (Cf. [Jac92] – also for a more abstract proof of the above proposition.)

Corollary 5.13 *For every two sets A and B,*

$$A \otimes^r \hat{\mathcal{P}}B \cong \hat{\mathcal{P}}(A \times B).$$

Proof. Immediate from the above proposition and the fact that the right tensor product is obtained by precomposing the first component of the tensor product with $\hat{\mathcal{P}}$. □

5.5.3 Tensor Products as Quotients

Next, we give a direct construction of tensor products based on quotients (more precisely, coequalizers). (Cf. [Lin66] and [LM92].)

Let L and K be two semi-lattices and let l, l' and k, k' be generic elements of L and K, respectively. Let the same symbol $\sqcup$ denote the join of both L and K. In order to obtain $L \otimes K$ from $L \times K$, one needs to put

$$< l \sqcup l', k \sqcup k' > = < l, k > \sqcup < l, k' > \sqcup < l', k > \sqcup < l', k' > \qquad \text{(yes)}$$

but not

$$< l \sqcup l', k \sqcup k' > = < l, k > \sqcup < l', k' > \qquad \text{(no)}$$

as it is the case with the join in the Cartesian product. This can be neatly expressed by means of coequalizers as follows.

Consider the two functions

$$\begin{tikzcd} {|L|^2 \times |K|^2} \arrow[r, shift left, "\theta"] \arrow[r, shift right, "\tau"'] & \hat{\mathcal{P}}(|L| \times |K|) \end{tikzcd} \qquad (39)$$

defined as follows:

$$\begin{aligned} \theta &: < l, l', k, k' > \longmapsto \{< l, k >, < l, k' >, < l', k >, < l', k' >\} \\ \tau &: < l, l', k, k' > \longmapsto \{< l \sqcup l', k \sqcup k' >\} \end{aligned}$$

Take now the coequalizer of their transposes $\theta^\sharp$ and $\tau^\sharp$ across the adjunction $\hat{\mathcal{P}} \dashv U$:

$$\begin{tikzcd} \hat{\mathcal{P}}(|L|^2 \times |K|^2) \arrow[r, shift left, "\theta^\sharp"] \arrow[r, shift right, "\tau^\sharp"'] & \hat{\mathcal{P}}(|L| \times |K|) \arrow[r, "q"] & \text{Coequalizer}(\theta^\sharp, \tau^\sharp) \end{tikzcd} \qquad (40)$$

This coequalizer gives the desired tensor product:

Proposition 5.14 *For $\theta^\sharp$ and $\tau^\sharp$ as in (40),*

$$L \otimes K \;\cong\; \mathit{Coequalizer}(\theta^\sharp, \tau^\sharp)$$

Moreover, by transposing the universal arrow given by the coequalizer (in the other direction than in (3)), one obtains the universal bilinear function described in (29). More formally, let $-^\flat$ be the inverse of the above $-^\sharp$. Then

$$\circledast = q^\flat \; : |L| \times |K| \rightarrow |L \otimes K| \;.$$

□

A quotient construction is possible also for right tensor products. The following few straightforward modifications of the above construction are needed.

Let A be a set and L a semi-lattice. Let a be an element of A and l, l' elements of L. The definition of (39) becomes

$$\begin{aligned} \theta, \tau &: A \times |L|^2 \rightarrow \hat{\mathcal{P}}(A \times |L|) \\ \theta &: < a, l, l' > \longmapsto \{< a, l >, < a, l' >\} \\ \tau &: < a, l, l' > \longmapsto \{< a, l \sqcup l' >\} \end{aligned}$$

and, similarly, Diagram (40) becomes

$$\begin{tikzcd} \hat{\mathcal{P}}(A \times |L|^2) \arrow[r, shift left, "\theta^\sharp"] \arrow[r, shift right, "\tau^\sharp"'] & \hat{\mathcal{P}}(A \times |L|) \arrow[r, "q"] & \text{Coequalizer}(\theta^\sharp, \tau^\sharp) \end{tikzcd} \qquad (41)$$

Proposition 5.15 *For $\theta^\sharp$ and $\tau^\sharp$ as in (41),*

$$A \otimes^r L \cong \mathit{Coequalizer}(\theta^\sharp, \tau^\sharp)$$

Moreover,

$$\otimes^r = q^\flat \, : \, A \times |L| \rightarrow |A \otimes^r L|$$

□

Acknowledgements

Discussions with and suggestions made by the following persons are gratefully acknowledged: Michael Barr, Eric Badouel, Jaco de Bakker, Marcello Bonsangue, Franck van Breugel, Philippe Darondeau, Elena Marchiori, Fer-Jan de Vries, and the members of the Amsterdam Concurrency Group. Special thanks to Bart Jacobs for joint work on linear final semantics.

References

[Abr90] S. Abramsky. The lazy lambda calculus. In D.A. Turner, editor, *Research Topics in Functional Programming*, pages 65–116. Addison-Wesley, 1990.

[Acz88] P. Aczel. *Non-well-founded sets.* Number 14 in CSLI Lecture Notes. Stanford University, 1988.

[AD89] A. Arnold and A. Dicky. An algebraic characterization of transition system equivalences. *Information and Computation*, 82:198–229, 1989.

[ArM82] M.A. Arbib and E.G. Manes. Parameterized data types do not need highly constrained parameters. *Information and Control*, 52:139–158, 1982.

[AM89] P. Aczel and N. Mendler. A final coalgebra theorem. In D.H. Pitt, D.E. Ryeheard, P. Dybjer, A.M. Pitts, and A. Poigné, editors, *Proceedings Category Theory and Computer Science*, volume 389 of *Lecture Notes in Computer Science*, pages 357–365, 1989.

[AR89] P. America and J.J.M.M. Rutten. Solving reflexive domain equations in a category of complete metric spaces. *Journal of Computer and System Sciences*, 39(3):343–375, 1989.

[Bad87] E. Badouel. Une construction systématique de modèles à partir de spécifications opérationnelles structurelles. Technical Report 764, INRIA, Rennes, 1987.

[Bad93] E. Badouel. Models of concurrency. Handout at the fifth ESSLLI, Faculdade de Letras, Universidade de Lisboa, 1993.

[Bar93] M. Barr. Terminal coalgebras in well-founded set theory. *Theoretical Computer Science*, 114(2):299–315, 1993. A revised version has appeared under the title "Terminal coalgebras for endofunctors on sets", McGill University, Montreal, 1993.

[BM88] J.W. de Bakker and J.-J.Ch. Meyer. Metric semantics for concurrency. *BIT*, 28:504–529, 1988.

[Bre93] F. van Breugel. Three metric domains of processes for bisimulation. Report CS-R9335, CWI, Amsterdam, 1993. To appear in Proceedings of the 9th International Conference on Mathematical Foundations of Programming Semantics, LNCS, Springer-Verlag, 1993.

[BW85] M. Barr and C. Wells. *Toposes, Triples and Theories.* Springer-Verlag, 1985.

[BZ82] J.W. de Bakker and J.I. Zucker. Processes and the denotational semantics of concurrency. *Information and Control*, 54:70–120, 1982.

[DG87] P. Darondeau and B. Gamatié. Modelling infinitary behaviours of communicating systems. Technical Report 749, INRIA, Rennes, 1987.

[EK66] S. Eilenberg and G.M. Kelly. Closed categories. In S. Eilenberg et al., editor, *Proc. of La Jolla Conf. on Categorical Algebra*, pages 421–562. Springer-Verlag, 1966.

[Fio93] M. Fiore. A coinduction principle for recursive data types based on bisimulation. In *Proceedings of the Eighth IEEE Symposium on Logic In Computer Science*, 1993.

[Gla93] R.J. van Glabbeek. Full abstraction in structural operational semantics (extended abstract). In M. Nivat, C. Rattray, T. Rus, and G. Scollo, editors, *Proceedings of the Third International Conference on Algebraic Methodology and Software Technology*, pages 77–84. Springer-Verlag, 1993.

[GTW78] J.A. Goguen, J.W. Thatcher, and E.G. Wagner. An initial algebra approach to the specification, correctness and implementation of abstract data types. In R. Yeh, editor, *Current Trends in Programming Methodology*, pages 80–149. Prentice-Hall, 1978.

[GV92] J.F. Groote and F. Vaandrager. Structured operational semantics and bisimulation as a congruence. *Information and Computation*, 100(2):202–260, 1992.

[Hen93] M.C.B. Hennessy. A fully abstract denotational model for higher-order processes. In *Proceedings 8th Annual Symposium on Logic in Computer Science.* IEEE Computer Society Press, 1993.

[HP79] M.C.B. Hennessy and G.D. Plotkin. Full abstraction for a simple parallel programming language. In J. Bečvář, editor, *Proc. 8th Int'l Symp. on Mathematical Foundations of Computer Science*, volume 74 of *Lecture Notes in Computer Science*, pages 108–120. Springer-Verlag, 1979.

[Jac92] B. Jacobs. Semantics of weakening and contraction. To appear in *Annals of Pure and Applied Logic*, 1992.

[Ken87] R.E. Kent. The metric closure powerspace construction. In M. Main, A. Melton, M. Mislove, and D. Schmidt, editors, *Proceedings of the 3rd MFPS*, volume 298 of *Lecture Notes in Computer Science*, pages 173–199. Springer-Verlag, 1987.

[Lan71] S. Mac Lane. *Categories for the Working Mathematician*, volume 5 of *Graduate Texts in Mathematics.* Springer-Verlag, 1971.

[Lin66] F.E.J. Linton. Autonomous equational categories. *J. of Mathematics and Mechanics*, 15(4):637–642, 1966.

[LM92] S. Mac Lane and I. Moerdijk. *Sheaves in geometry and logic: a first introduction to topos theory.* Universitext. Springer-Verlag, 1992.

[Man76] E.G. Manes. *Algebraic theories*, volume 26 of *Graduate Texts in Mathematics.* Springer-Verlag, 1976.

[MG85] J. Meseguer and J.A. Goguen. Initiality, induction, and computability. In M. Nivat and J.C. Reynolds, editors, *Algebraic Methods in Semantics*, pages 459–541. Cambridge University Press, 1985.

[Mil89] R. Milner. *Communication and Concurrency.* Prentice Hall, 1989.

[Par81] D.M.R. Park. Concurrency and automata on infinite sequences. In P. Deussen, editor, *Proceedings 5th GI Conference*, volume 104 of *Lecture Notes in Computer Science*, pages 167–183. Springer-Verlag, 1981.

[Pit92] A.M. Pitts. A co-induction principle for recursively defined domains. Technical Report 252, Computer Laboratory, University of Cambridge, 1992. To appear in *Theoretical Computer Science.*

[Plo81a] G.D. Plotkin. Post-graduate lecture notes in advanced domain theory (incorporating the "Pisa Notes"). Department of Computer Science, University of Edinburgh, 1981.

[Plo81b] G.D. Plotkin. A structured approach to operational semantics. Technical Report DAIMI FN-19, Computer Science Department, Aarhus University, 1981.

[RT93] J.J.M.M. Rutten and D. Turi. On the foundations of final semantics: Non-standard sets, metric spaces, partial orders. In J.W. de Bakker, W.-P. de Roever, and G. Rozenberg, editors, *Proceedings of the REX workshop on Semantics: Foundations and Applications*, volume 666 of *Lecture Notes in Computer Science*, pages 477–530. Springer-Verlag, 1993. FTP-available at `ftp.cwi.nl` as pub/CWIreports/AP/CS-R9241.Z.

[Rut92] J.J.M.M. Rutten. Processes as terms: non-well-founded models for bisimulation. *Mathematical Structures in Computer Science*, 2:257–275, 1992.

[Rut93] J.J.M.M. Rutten. A structural co-induction theorem. Technical Report CS-R9346, CWI, 1993. To appear in *Proceedings of the Ninth Conference on the Mathematical Foundations of Programming Semantics*, Lecture Notes in Computer Science, Springer-Verlag, 1994.

[Sif84] J. Sifakis. Property preserving homomorphisms of transition systems. In E. Clarke and D. Kozen, editors, *Logics of programs*, volume 164 of *Lecture Notes in Computer Science*, pages 458–473. Springer-Verlag, 1984.

[SP82] M.B. Smyth and G.D. Plotkin. The category-theoretic solution of recursive domain equations. *SIAM J. Comput.*, 11:761–783, 1982.

[TJ93] D. Turi and B. Jacobs. On final semantics for applicative and non-deterministic languages. Fifth Biennal Meeting on Category Theory and Computer Science, Amsterdam, September 1993.

[WN93] G. Winskel and M. Nielsen. Models for concurrency. Handout at the TEMPUS Summer School on Algebraic and Categorical Methods in Computer Science, 1993.