IV

A Summary

In this section a technical summary of the above results is given. It can be read independently from the other sections by a reader familiar with the categorical notions of *adjunction* and *monad*. After some preliminaries recalling the basic definitions and facts about algebras and coalgebras, the notion of functorial denotational semantics is introduced; as an example, *basic process algebra* [BW90] is defined denotationally. Next, every functorial denotational semantics is shown to induce an operational dual (and vice versa). (The *Basic Property*.) Next, several results are proved (*Operational is Denotational*, Φ*-algebras are* $\Phi^{@}$*-coalgebras*, *Adequacy Theorem*) which illustrate the *adequacy* of the denotational semantics $\Phi^{@}$ coinduced by a functorial operational semantics Φ. Finally, it is proved that the operational semantics induced by *GSOS rules* [BIM88] is always functorial.

Algebras. The category of the algebras of a monad $T{=}{<}T,\eta,\mu{>}$ on a category $\mathbf{C}$ is denoted by $\mathbf{C}^T$. Its objects are the arrows $h : TX \to X$ of $\mathbf{C}$ such that $h \circ \eta_X = \mathsf{id}_X$ and $h \circ \mu_X = h \circ Th$; its arrows $f : (TX \xrightarrow{h} X) \to (TX' \xrightarrow{h'} X')$ are the arrows $f : X \to X'$ in $\mathbf{C}$ such that $f \circ h = h' \circ Tf$.

The evident forgetful functor $U^T : \mathbf{C}^T \to \mathbf{C}$ has a left adjoint $X \mapsto (T^2X \xrightarrow{\mu_X} TX)$. This adjoint situation is here denoted as follows.

$$
\begin{tikzcd}
X \arrow[r, "\eta_X"] \arrow[dr, "f"'] & TX \arrow[d, "f^\sharp"] \\
 & Y
\end{tikzcd}
\qquad
\begin{tikzcd}
TX \arrow[d, dashed, "f^\sharp"'] & T^2X \arrow[l, "\mu_X"'] \arrow[d, "Tf^\sharp"] \\
Y & TY \arrow[l, "h"]
\end{tikzcd}
\tag{4}
$$

(In the sequel, $f^\sharp$ does always denote the above left adjunct of f wrt the adjunction. The uniqueness of $f^\sharp$ is exploited here to prove several equalities between arrows.) The monad defined by this adjunction is trivially equal to the original monad T, hence *every monad is defined by its algebras*.

Given a *signature* Σ and a cocomplete category $\mathbf{C}$ with finite products, one can define an endofunctor (with the same name) on $\mathbf{C}$ as the coproduct $\Sigma X = \coprod_\sigma X^{\mathrm{arity}(\sigma)}$ indexed by the operators σ of the signature. Then the Σ-algebras $h : \Sigma X \to X$ form a category $\mathbf{C}^\Sigma$ with as arrows $f : (\Sigma X \xrightarrow{h} X) \to (\Sigma X' \xrightarrow{h'} X')$ the arrows $f : X \to X'$ in $\mathbf{C}$ such that $f \circ h = h' \circ \Sigma f$.

Also the forgetful functor $U^\Sigma : \mathbf{C}^\Sigma \to \mathbf{C}$ has a left adjoint and, moreover, it is *monadic*, ie, if T is the monad arising from this adjunction, then there is an isomorphism of categories $\mathbf{C}^\Sigma \cong \mathbf{C}^T$ making the following diagram commute.

$$\begin{tikzcd} \mathbf{C}^\Sigma \arrow[r, phantom, "\cong"] \arrow[dr, "U^\Sigma"'] & \mathbf{C}^T \arrow[d, "U^T"] \\ & \mathbf{C} \end{tikzcd} \tag{5}$$

For $\mathbf{C} = \mathbf{Set}$, TX is the usual set of terms inductively defined by the operators in Σ and the variables $x \in X$. In particular, T at the empty set 0 is the set of closed terms. In other words, $T0$ is the carrier of the initial Σ-algebra $\Sigma T0 \cong T0$ (*Lambek's lemma*: initial algebras are always isomorphisms [SP82]) and, in general, TX is the carrier of the initial $(X+\Sigma)$-algebra $X+\Sigma TX \cong TX$. The unit $\eta_X = \mathsf{inl}_X : X \to TX$ at X is the formal insertion of the variables $x \in X$ in the terms $t \in TX$ and the multiplication $\mu_X : T^2X \Rightarrow TX$ is the 'inductive extension' of the right injection inr_X. Thus, for instance, $\mu_X(\sigma(\eta_X t_1, \ldots, \eta_X t_n)) = \sigma(t_1, \ldots, t_n)$. To ease the notation, η and μ is often omitted from the terms.

In the sequel, also $\langle\Sigma, E\rangle$-algebras are considered, ie Σ-algebras which satisfy some equations E on the operators derivable from the signature. The forgetful functor from the corresponding category $\mathbf{Set}^{\langle\Sigma,E\rangle}$ has a left adjoint and it is monadic, hence

$$\mathbf{Set}^{\langle\Sigma,E\rangle} \cong \mathbf{Set}^T \tag{6}$$

for the corresponding monad T. For instance, consider semi-lattices with a least element, ie let Σ contain only a binary operator $\vee$ and a constant $\perp$ and let E be the associativity, commutativity, and associativity axioms for $\vee$ and the unit axiom for $\perp$ wrt $\vee$. Then $\mathbf{Set}^{\langle\Sigma,E\rangle}$ is isomorphic to the category of algebras of the monad $<\mathcal{P}_{\mathrm{fi}}, \{\text{-}\}, \bigcup>$, where $\check{\mathcal{P}}X$ is the set of finite subsets of X.

If the operators of Σ are the constructs of a programming language then, an algebra $h : TY \to Y$ of the corresponding *syntactical* monad T is a *denotational* model of the language and it induces an *initial algebra semantics* [GTW78], namely the unique arrow $h^\# : T0 \to Y$ from the initial algebra $\mu_0 : T^2 0 \to T0$ to $h : TY \to Y$:

$$h^\#(\sigma(t_1, \ldots, t_n)) = h(\sigma(h^\# t_1, \ldots, h^\# t_n))$$

Coalgebras. Dually, let $\mathbf{C}_B$ denote the category of coalgebras of an endofunctor B on $\mathbf{C}$, having as objects arrows $k : X \to BX$ in $\mathbf{C}$ and as arrows $f : (X \xrightarrow{k} BX) \to (X' \xrightarrow{k'} BX')$ those arrows $f : X \to X'$ in $\mathbf{C}$ such that $f \circ k' = Bf \circ k$.

Every *finitely branching* labelled transition system [Plo81b]

$$\langle X, \{\xrightarrow{a}\}_{a \in \mathsf{Act}}\rangle$$

can be seen as a coalgebra $k : X \to BX$ of the *behaviour* endofunctor $BX = \mathcal{P}_{\mathrm{fi}}(\mathsf{Act} \times X)$ on $\mathbf{Set}$ [Acz88]:

$$x \xrightarrow{a} x' \iff <a, x'> \in k(x) \tag{7}$$

Notice that although the category of B-coalgebras has the same objects as the standard category of transition systems [WN95], the arrows are different.

The final B-coalgebra $D1 \cong B(D1)$ exists [AM89] and, correspondingly, every *operational* model, ie coalgebra (ie transition system) $[\![\text{-}]\!] : TX \to BTX$ on the syntax, coinduces a *final coalgebra semantics* [RT93], namely the unique arrow $[\![\text{-}]\!]^{@} : TX \to D1$ from $[\![\text{-}]\!]$ to the final coalgebra. Up to the isomorphism $D1 \cong B(D1)$,

$$[\![t]\!]^{@} = \{ < a, [\![t']\!]^{@} > \mid < a, t' > \in [\![t]\!] \}$$

The existence of this final coalgebra is a corollary of the fact that the forgetful functor $U_B : \mathbf{Set}_B \to \mathbf{Set}$ has a *right* adjoint [Bar93]. In general, for any endofunctor on a complete category $\mathbf{C}$, if the forgetful functor $U_B : \mathbf{C}_B \to \mathbf{C}$ has a right adjoint then it is *comonadic*, ie the coalgebras of the corresponding comonad $D = < D, \varepsilon, \delta >$ are isomorphic to the B-coalgebras:

$$\mathbf{C}_B \cong \mathbf{C}_D \tag{8}$$

(A coalgebra of the comonad D is an arrow $k : X \to DX$ in $\mathbf{C}$ such that $\varepsilon_X \circ k = \mathsf{id}_X$ and $\delta_X \circ k = Dk \circ k$.) Correspondingly, the forgetful functor $U_D : \mathbf{C}_D \to \mathbf{C}$ has a right adjoint $X \mapsto (DX \xrightarrow{\delta_X} D^2X)$:

$$\begin{tikzcd} & X \arrow[dl, "f"'] \arrow[d, "f^\flat"] \\ X & DY \arrow[l, "\varepsilon_Y"] \end{tikzcd} \qquad \begin{tikzcd} X \arrow[r, "k"] \arrow[d, dashed, "f^\flat"'] & DX \arrow[d, "Df^\flat"] \\ DY \arrow[r, "\delta_Y"'] & D^2Y \end{tikzcd} \tag{9}$$

To every endofunctor B corresponds a notion of B-*bisimulation* [AM89] which, for $BX = \mathcal{P}_{\mathrm{fi}}(\mathsf{Act} \times X)$, specializes to the ordinary bisimulation [Par81]. Final coalgebras are *internally fully-abstract* in the sense that their greatest B-bisimulation (exists and) is an equality relation; moreover, if B (like the above behaviour) preserves weak pullbacks, then the kernel pair of the final coalgebra semantics is the greatest B-bisimulation (on the B-coalgebra under consideration) [RT93]. One can prove that the final coalgebra of the behaviour $BX = \mathcal{P}_{\mathrm{fi}}(\mathsf{Act} \times X)$ is the set of rooted finitely branching trees quotiented by its greatest bisimulation.

In general, for an endofunctor B to qualify as a *behaviour* its corresponding notion of bisimulation should be a significant notion of *observational equivalence*; moreover, it should preserve weak pullbacks, and the forgetful functor $U_B : \mathbf{C}_B \to \mathbf{C}$ should have a right adjoint. The corresponding comonad D is then an *observational comonad*.

Functorial Denotational Semantics

Given an observational comonad $D = < D, \varepsilon, \delta >$ and a syntactical monad $T = < T, \eta, \mu >$, a **functorial denotational semantics** is a comonad Ψ **lifting** the comonad D to the T-algebras:

$$\begin{tikzcd} \mathbf{C}^T \arrow[r, "\Psi"] \arrow[d, "U^T"'] & \mathbf{C}^T \arrow[d, "U^T"] \\ \mathbf{C} \arrow[r, "D"'] & \mathbf{C} \end{tikzcd}$$

That is, Ψ is a triple $< \Psi, \widetilde{\varepsilon}, \widetilde{\delta} >$ such that

$$\begin{array}{rcl} U^T\Psi & = & DU^T : \mathbf{C}^T \to \mathbf{C} \\ U^T\widetilde{\varepsilon} & = & \varepsilon_{U^T} : DU^T \Rightarrow U^T \\ U^T\widetilde{\delta} & = & \delta_{U^T} : DU^T \Rightarrow D^2U^T \end{array}$$

In other words, $U^T : \mathbf{C}^T \to \mathbf{C}$ is a 'map of monads'.

The second and third equation imply that the counit $\widetilde{\varepsilon}$ and comultiplication $\widetilde{\delta}$ of Ψ are the same as those of $D = < D, \varepsilon, \delta >$, because of the very definition of coalgebra arrows. Therefore:

$$\Psi = < \Psi, \varepsilon, \delta >$$

One can check that the three equations and the fact that the triple $< P, \varepsilon, \delta >$ is a comonad imply that also the triple $\Psi = < \Psi, \varepsilon, \delta >$ is a comonad. Also, the first equation is equivalent to Ψ being an **action** of T on $DU^T : \mathbf{C}^T \to \mathbf{C}$, ie a natural transformation

$$\Psi : TDU^T \Rightarrow DU^T$$

such that, for every T-algebra $h : TX \to X$, $\Psi h : TDX \to DX$ is also a T-algebra. (See, eg, [BW85] for the equivalence between liftings and actions.) Then, the second and third equations are equivalent to the fact that, for every $h : TX \to X$, the following diagram commutes.

$$\begin{tikzcd} TX \arrow[d, "h"'] & TDX \arrow[l, "T\varepsilon_X"'] \arrow[r, "T\delta_X"] \arrow[d, "\Psi h"] & TD^2X \arrow[d, "\Psi^2 h"] \\ X & DX \arrow[l, "\varepsilon_X"] \arrow[r, "\delta_X"'] & D^2X \end{tikzcd}$$

That is,

$$\varepsilon_X \circ \Psi h = h \circ T\varepsilon_X \tag{10}$$

$$\delta_X \circ \Psi h = \Psi^2 h \circ T\delta_X \tag{11}$$

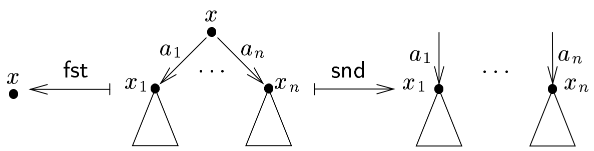
As an example, consider the following functorial denotational semantics for *basic process algebra* [BW90]. The base category $\mathbf{C}$ is $\mathbf{Set}$. The syntactical monad T is the one freely generated by the constructs $\Sigma = \{\mathsf{nil}, a\,.\,, \mathsf{or}\,\}$, ie

$$t ::= \mathsf{nil} \mid a\,.\,t \mid t\,\mathsf{or}\,t$$

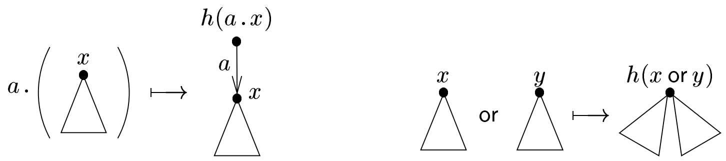
The observational comonad $D = < D, \varepsilon, \delta >$ is cofreely generated by the behaviour $BY = \mathcal{P}_{\mathrm{fi}}(\mathsf{Act} \times Y)$. The set DX is the carrier of the final $(X \times B)$-coalgebra:

$$X \xleftarrow{\varepsilon_X \,=\, \mathsf{fst}_X} DX \cong X \times BDX \xrightarrow{\mathsf{snd}_X} BDX$$

and it is a set of (finitely branching) trees whose nodes are labelled by $x \in X$ and whose arcs are labelled by $a \in \mathsf{Act}$. The operation $\varepsilon_X = \mathsf{fst}_X : DX \to X$ gives the label of the root node for each tree in DX and the other operation $\mathsf{snd}_X : DX \to BDX = \mathcal{P}_{\mathrm{fi}}(\mathsf{Act} \times DX)$ gives the remaining part of the tree (and it coinductively extends to give the counit $\delta : D \Rightarrow D^2$ of the comonad D):



Using (5), one can define Ψ as an action of Σ rather than of T. That is, $\Psi : \Sigma P U^{\Sigma} \Rightarrow D U^{\Sigma}$. Then, for every $h : \Sigma X \to X$, define the action of the constant nil as the tree with only one node and label $h(\mathsf{nil})$, and the action of '$a\,.$' and 'or' as follows.



Formally, using the meta-variables p and q to range over the elements of DTX, for every X, Ψ is defined as follows.

$$\begin{array}{lcl}
\mathsf{nil} & \mapsto & < h(\mathsf{nil}), \emptyset > \\
a\,.\,p & \mapsto & < h(a\,.\,(\mathsf{fst} p)), \{< a, p >\} > \\
p\,\mathsf{or}\,q & \mapsto & < h((\mathsf{fst} p)\,\mathsf{or}\,(\mathsf{fst} q)), (\mathsf{snd} p) \cup (\mathsf{snd} q) >
\end{array}$$

Therefore, the Σ-algebra $\Psi h : \Sigma DX \to DX$ is a pair, whose first component is simply the composite function $h \circ \Sigma\mathsf{fst}_X$, ie, up to (5), the equation (10) holds, because $\mathsf{fst}_X = \varepsilon_X$. Also (11) holds, because both $\delta_X \circ \Psi h$ and $\Psi^2 h \circ \Sigma\delta_X$ fit as the (unique!) pair $< \Psi h, B\delta_X \circ \mathsf{snd}_X \circ \Psi h >$. Therefore, $\Psi = < \Psi, \varepsilon, \delta >$ is a functorial denotational semantics for the above signature Σ and behaviour B.

A Dual Lifting: Functorial Operational Semantics

The definition of **functorial operational semantics** is the dual of the one of functorial denotational semantics: it is a monad $\Phi = <\Phi, \eta, \mu>$ lifting the syntactical monad $T = <T, \eta, \mu>$ to the D-coalgebras. That is, a **coaction**

$$\Phi : TU_D \Rightarrow DTU_D$$

of the comonad D on $TU_D : \mathbf{C}_D \to \mathbf{C}$ such that, for every D-coalgebra $k : X \to DX$, the following diagram commutes.

$$\begin{tikzcd}
X \arrow[r, "\eta_X"] \arrow[d, "k"'] & TX \arrow[d, "\Phi k"'] & T^2X \arrow[l, "\mu_X"'] \arrow[d, "\Phi^2 k"] \\
DX \arrow[r, "D\eta_X"'] & DTX & DT^2X \arrow[l, "D\mu_X"]
\end{tikzcd}$$

The Basic Property. Every functorial denotational semantics Ψ defines a functorial operational semantics whose action $\Psi^{\#} : TU_D \Rightarrow DTU_D$ is defined by means of the adjunction (4) as follows.

$$\begin{tikzcd}
X \arrow[r, "\eta_X"] \arrow[d, "k"'] & TX \arrow[d, "\Psi^{\#} k"'] \\
DX \arrow[r, "D\eta_X"'] & DTX
\end{tikzcd}
\qquad
\begin{tikzcd}
TX \arrow[d, dashed, "\Psi^{\#} k = (D\eta_X \circ k)^{\sharp}"'] & T^2X \arrow[l, "\mu_X"'] \arrow[d, "T\Psi^{\#} k"] \\
DTX & TDTX \arrow[l, "\Psi\mu_X"]
\end{tikzcd}$$

That is,

$$\Psi^{\#} = (D\eta \circ \_)^{\sharp}$$

Dually, every functorial operational semantics Φ defines, by means of (9) a functorial denotational semantics

$$\Phi^{@} = (\_ \circ T\varepsilon)^{\flat}$$

Proof. Naturality follows from universality. Next, $\Psi^{\#}k \circ \mu_X$ is equal to $D\mu_X \circ \Psi^{\#2}k$ because both fit as the (unique!) T-algebra arrow

$$(\Psi^{\#}k)^{\sharp} : (T^2X \xrightarrow{\mu_{TX}} TX) \to (TDTX \xrightarrow{\Psi k} DTX)$$

obtained by taking the left adjunct of $\Psi^{\#}k$ wrt the adjunction (4). Similarly, $\Psi^{\#}k : TX \to DTX$ is a D-coalgebra: to prove that $\varepsilon_{TX} \circ \Psi^{\#}k$ is equal to the identity id_{TX} on TX notice that both fit as $\eta_X^{\sharp}$, since $\eta_X = \eta_X \circ \mathsf{id}_X = \eta_X \circ (\varepsilon_X \circ k)$; and to prove that $D\Psi^{\#}k \circ \Psi^{\#}k$ is equal to $\delta_{TX} \circ \Psi^{\#}k$ notice that both fit as $(D^2\eta_X \circ (Dk \circ k))^{\sharp} = (D^2\eta_X \circ (\delta_X \circ k))^{\sharp}$. $\square$

For the denotational semantics Ψ in the above example one has the following induced operational semantics $\Psi^{\#}$. Using the isomorphisms (5) (to move from Σ- to

T-actions) and (8) (to move from D- to B-coactions), obtain $\Psi^\# : TU_B \Rightarrow BTU_B$. Consider, for simplicity, the case of the 'empty' coalgebra $0 : 0 \to B0$ given by the initial function into $B0$ and put

$$[\![\text{-}]\!]_\Psi = \Psi^\#(0) : T0 \to BT0$$

Up to (7), this is the transition system induced by Ψ on the closed program $t \in T0$ of basic process algebra. Spelling out the details, one can obtain

$$\begin{array}{lcl} [\![\mathsf{nil}]\!]_\Psi & = & \emptyset \\ [\![a\,.\,t]\!]_\Psi & = & \{<a, t>\} \\ [\![t_1 \,\mathsf{or}\, t_2]\!]_\Psi & = & [\![t_1]\!]_\Psi \cup [\![t_2]\!]_\Psi \end{array}$$

Using (7), this really yields basic process algebra:

$$\begin{array}{l} \mathsf{nil} \not\to \\ a\,.\,t \xrightarrow{a} t \\ t_1 \,\mathsf{or}\, t_2 \xrightarrow{a} t \quad \text{if } t_1 \xrightarrow{a} t \text{ or } t_2 \xrightarrow{a} t \end{array}$$

Operational is Denotational

The mapping $\Phi \mapsto \Phi^@$ is a bijection between operational monads and denotational comonads with $\Psi \mapsto \Psi^\#$ as inverse:

$$\begin{array}{ccc} \Phi & \begin{tikzcd} \mathbf{C}_D \arrow[r, "\Phi"] \arrow[d, "U_D"'] & \mathbf{C}_D \arrow[d, "U_D"] \\ \mathbf{C} \arrow[r, "T"'] & \mathbf{C} \end{tikzcd} & \Psi^\# \\ \big\downarrow & \overline{\overline{\qquad\qquad\qquad}} & \big\uparrow \\ \Phi^@ & \begin{tikzcd} \mathbf{C}^T \arrow[r, "\Psi"] \arrow[d, "U^T"'] & \mathbf{C}^T \arrow[d, "U^T"] \\ \mathbf{C} \arrow[r, "D"'] & \mathbf{C} \end{tikzcd} & \Psi \end{array}$$

Proof. Everything in sight in the following diagram commutes.

$$\begin{tikzcd} TX \arrow[rrr, "\Phi k"] \arrow[d, equal] & & & DTX \\ TX \arrow[r, "T\eta_X"] \arrow[d, "Tk"'] \arrow[rrru, bend right=90, looseness=2, "(\Phi^@)^\# k"'] & TDX \arrow[r, "\Phi^2 k"] \arrow[ul, "\mu_X"'] \arrow[d, "T\Phi k"] & DT^2X \arrow[ur, "D\mu_X"] \arrow[d, "DT\Phi k"] \arrow[dr, phantom, "="] & \\ TDX \arrow[r, "TD\eta_X"'] & TDTX \arrow[r, "\Phi\delta_{TX}"'] & DTDTX \arrow[r, "DT\varepsilon_{TX}"'] & DT^2X \arrow[uu, "D\mu_X"'] \end{tikzcd}$$

Indeed, the value of the unit of the adjunction (9) at a coalgebra $\langle X, k\rangle$ is its structure $k : X \to DX$, thus, in particular, its value at $\langle TDX, \Phi\delta_X\rangle$ is $\Phi\delta_X : TDX \to DTDX$, hence:

$$\Phi^{@}h = (h \circ T\varepsilon_X)^{\flat} = D(h \circ T\varepsilon_X) \circ \Phi\delta_X = Dh \circ DT\varepsilon_X \circ \Phi\delta_X$$

Dually:

$$\Psi^{\#}k = \Psi\mu_X \circ TD\eta_X \circ Tk$$

Therefore:

$$\begin{array}{rcl} (\Phi^{@})^{\#}k & = & \Phi^{@}\mu_X \circ TD\eta_X \circ Tk \\ & = & D\mu_X \circ DT\varepsilon_{TX} \circ \Phi\delta_{TX} \circ TD\eta_X \circ Tk \end{array}$$

This proves the commutativity of the lower subdiagram in the above diagram. The other non-immediate fact is the commutativity of the subdiagram in the middle, but this follows from the fact that it is the image under the functor Φ of one of the two D-coalgebra laws for the structure $\Phi k : TX \to DTX$. That is,

$$\begin{tikzcd} TX \arrow[r, "\Phi k"] \arrow[d, "\Phi k"'] & DTX \arrow[d, "\delta_{TX}"] \\ DTX \arrow[r, "D\Phi k"'] & D^2TX \end{tikzcd} \quad \overset{\Phi}{\longmapsto} \quad \begin{tikzcd} T^2X \arrow[r, "T\Phi k"] \arrow[d, "\Phi^2 k"'] & TDTX \arrow[d, "\Phi\delta_{TX}"] \\ DT^2X \arrow[r, "DT\Phi k"'] & DTDTX \end{tikzcd}$$

This proves that $\Phi k = (\Phi^{@})^{\#}k$ and, by duality, $\Psi h = (\Psi^{\#})^{@}h$.

Φ-Algebras are $\Phi^{@}$-coalgebras

The algebras of an operational monad Φ and the coalgebras of its coinduced denotational comonad $\Phi^{@}$ are respectively of the form

$$\begin{tikzcd} TX \arrow[r, "\Phi k"] \arrow[d, "h"'] & DTX \arrow[d, "Dh"] \\ X \arrow[r, "k"'] & DX \end{tikzcd} \qquad \begin{tikzcd} TX \arrow[r, "Tk"] \arrow[d, "h"'] & TDX \arrow[d, "\Phi^{@}h"] \\ X \arrow[r, "k"'] & DX \end{tikzcd}$$

where $h : TX \to X$ is a T-algebra and $k : X \to DX$ is a D-coalgebra. For both, the arrows are those between their carriers (hence in $\mathbf{C}$) which are simultaneously T-algebra (hence in $\mathbf{C}^T$) and D-coalgebra (hence in $\mathbf{C}_D$) arrows.

The claim is that $Dh \circ \Phi k$ is equal to $\Phi^{@}h \circ Tk$. Indeed, everything in sight in the following diagram commutes.

$$\begin{tikzcd} TX \arrow[rr, "\Phi k"] \arrow[dd, "Tk"'] & & DTX \arrow[dl, "DTk"'] \arrow[d, equal] \\ & DTDX \arrow[r, "DT\varepsilon_X"'] & DTX \arrow[d, "Dh"] \\ TDX \arrow[ur, "\Phi\delta_X"'] \arrow[rr, "\Phi^{@}h"'] & & DX \end{tikzcd}$$

The only non-trivial sub-diagram is the one corresponding to the upper left corner but this is the image under the functor Φ of one of the two D-coalgebra laws for the structure $k : X \to DX$. That is,

$$\begin{tikzcd} X \arrow[r, "k"] \arrow[d, "k"'] & DX \arrow[d, "\delta_X"] \\ DX \arrow[r, "Dk"'] & D^2X \end{tikzcd} \quad \overset{\Phi}{\longmapsto} \quad \begin{tikzcd} TX \arrow[r, "Tk"] \arrow[d, "\Phi k"'] & TDX \arrow[d, "\Phi\delta_X"] \\ DTX \arrow[r, "DTk"'] & DTDX \end{tikzcd}$$

Thus, up to the permutation $\langle X, k, h\rangle \mapsto \langle X, h, k\rangle$, for any monad Φ lifting a monad T to the coalgebras of a comonad D, the two categories of Φ-algebras and $\Phi^{@}$-coalgebras are the same:

$$\mathbf{C}_D{}^{\Phi} = \mathbf{C}^T{}_{\Phi^{@}}$$

Dually,

$$\mathbf{C}^T{}_{\Psi} = \mathbf{C}_D{}^{\Psi^{\#}}$$

that is, Ψ**-coalgebras are** $\Psi^{\#}$**-algebras**.

Adequacy

If Φ is an operational monad, then the category $\mathbf{C}_D{}^{\Phi} = \mathbf{C}^T{}_{\Phi^{@}}$ can be seen as the category of **models of** Φ:

$$\Phi\text{-}\mathbf{Mod} = \mathbf{C}_D{}^{\Phi} = \mathbf{C}^T{}_{\Phi^{@}}$$

This category has both an initial and a final object which are 'lifted' from the initial T-algebra and the final D-coalgebra, respectively.

Lemma. The forgetful functor $\widetilde{U_D} : \mathbf{C}_D{}^{\Phi} \to \mathbf{C}^T$ with action $(TX \to X \to DX) \mapsto (TX \to X)$ has a right adjoint, namely

$$\begin{tikzcd} TX \arrow[d, "h"'] \\ X \end{tikzcd} \quad \overset{\widetilde{G_D}}{\longmapsto} \quad \begin{tikzcd} TDX \arrow[r, "\Phi\delta_X"] \arrow[d, "\Phi^{@}h"] & DTDX \arrow[d, "D\Phi^{@}h"] \\ DX \arrow[r, "\delta_X"'] & D^2X \end{tikzcd}$$

Dually, $\widetilde{U^T} : \mathbf{C}^T{}_{\Psi} \to \mathbf{C}_D$ has a left adjoint

$$(X \xrightarrow{k} DX) \overset{\widetilde{F^T}}{\longmapsto} (T^2X \xrightarrow{\mu_X} TX \xrightarrow{\Psi^{\#}k} DTX)$$

Proof. The counit of the adjunction is simply the counit ε of D, ie it is lifted from the adjunction (9). $\square$

Thus there are two adjunctions for the category of Φ-models, namely

$$\begin{tikzcd} \mathbf{C}^T \arrow[r, shift right, "\widetilde{G_D}"'] \arrow[r, phantom, "\perp"] & {\mathbf{C}_D}^{\Phi} = \Phi\text{-}\mathbf{Mod} = {\mathbf{C}^T}_{\Phi^{@}} \arrow[l, shift right, "\widetilde{U_D}"'] \arrow[r, shift right, "\widetilde{U^T}"'] \arrow[r, phantom, "\perp"] & \mathbf{C}_D \arrow[l, shift right, "\widetilde{F^T}"'] \end{tikzcd}$$

Hence, $\widetilde{F^T}$ maps the (trivial) initial D-coalgebra to the *initial* Φ-*model*:

$$(0 \xrightarrow{0} D0) \xmapsto{\widetilde{F^T}} (T^2 0 \xrightarrow{\mu_0} T0 \xrightarrow{\Phi 0} DT0)$$

Dually, $\widetilde{G_D}$ maps the (trivial) final T-algebra to the *final* Φ-*model*:

$$(T1 \xrightarrow{1} 1) \xmapsto{\widetilde{G_D}} (TD1 \xrightarrow{\Phi^{@}1} D1 \xrightarrow{\delta_1} D^2 1)$$

Then, by definition of Φ-algebra (alias Φ-model) arrow, the following holds.

Adequacy Theorem. The unique (both by initiality and finality) arrow from $\widetilde{F^T}(0)$ to $\widetilde{G_D}(1)$ is both the *initial algebra semantics* from the closed programs $T0$ to the domain $D1$ with denotations $\Phi^{@}1$, and the *final coalgebra semantics* from the transition system $\Phi 0$ on the closed programs to the set of most abstract observations $D1$. □

Since by 'pulling back' this final coalgebra semantics one obtains the greatest B-bisimulation, the fact that it is also an initial algebra semantics gives the following

Corollary. B-bisimulation is a congruence wrt Φ.

GSOS is Functorial

First a preliminary remark. Notice that, in the operational semantics $[\![\text{-}]\!]_\Psi$ given above for basic process algebra, the construct or behaves as the join $\cup$ of the semi-lattice $\mathcal{P}_{\text{fi}}(\mathsf{Act} \times T)$. Thus the above Ψ can also be seen as a lifting of B to the $\langle \Sigma, E \rangle$-algebras, where E are the semi-lattice laws for the binary operator or (ie or is required to be associative, commutative, and absorptive). For simplicity, let us keep the notation $T =< T, \eta, \mu >$ also for the monad corresponding to the $\langle \Sigma, E \rangle$-algebras.

(Thus TX is now the quotient wrt (the congruence relation generated by) E of the (previous) free algebra of terms over X; thus one cannot distinguish in this syntax between, for instance, the terms $t_1 \,\mathsf{or}\, t_2$ and $t_2 \,\mathsf{or}\, t_1$. Keeping this quotient in mind, one can still regard the elements of TX as terms, that is, one can use representatives rather than equivalence classes.)

One can then embed the behaviour $BX = \mathcal{P}_{\text{fi}}(\mathsf{Act} \times X)$ into this new syntax T by mapping $\emptyset$ to nil, $\{<a, x>\}$ to $a.x$, and $\cup$ to or. This defines a natural transformation

$$\gamma : B \Rightarrow T$$

injective in each component. It is a **retraction** for the above basic process algebra $\Psi^\#$, in the sense that the composite $\Psi^\# \circ \mu \circ \gamma_T : BT \Rightarrow BT$ is the identity natural transformation on BT.

This retraction γ is important because it permits to regard every set $\mathcal{R}$ of 'GSOS' rules containing basic process algebra as a natural transformation $\lceil \mathcal{R} \rceil : \Sigma B \Rightarrow BT$. A **GSOS rule** specifies one possible transition for terms of the form $\sigma(u_1, \ldots u_l)$, for σ a given program construct of arity l:

$$\frac{\{u_i \xrightarrow{a_{ij}} v_{ij}\}^{1 \le i \le l}_{1 \le j \le m_i} \quad \{u_i \stackrel{b_{ij}}{\not\longrightarrow}\}^{1 \le i \le l}_{1 \le j \le n_i}}{\sigma(u_1, \ldots, u_l) \xrightarrow{a} C[\overrightarrow{u}, \overrightarrow{v}]} \tag{12}$$

The a_{ij}'s and b_{ij}'s are actions in Act; the u_i's and v_{ij}'s are all distinct (meta) variables ranging over terms, the expression $C[\overrightarrow{u}, \overrightarrow{v}]$ is a term formed by the context $C[\overrightarrow{-}]$ and some (meta) variables contained in the set of u_i's and v_{ij}'s.

Clearly, the rules of basic process algebra are in GSOS. Let us assume that every set $\mathcal{R}$ of GSOS rules *conservatively extends* basic process algebra. Therefore, the corresponding syntactical monad T contains terms $\mathsf{nil}, a\,.\,t, t_1 \,\mathsf{or}\, t_2$, and or is a semi-lattice join, hence the above retraction $\gamma : B \Rightarrow T$ is a retraction also for (the operational semantics induced by the rules) $\mathcal{R}$.

Then, using the meta-variables r_i to range over the elements of $BX = \mathcal{P}_{\text{fi}}(\mathsf{Act} \times X)$, define $\lceil \mathcal{R} \rceil_X : \Sigma BX \to BTX$ by putting, for every rule (12) in $\mathcal{R}$,

$$< a, C[\overrightarrow{\gamma_X r}, \overrightarrow{x}] > \in \lceil \mathcal{R} \rceil_X(\sigma(r_1, \ldots, r_l))$$

if $\{< a_{ij}, x_{ij} > \in r_i\}^{1 \le i \le l}_{1 \le j \le m_i}$ and, for every $x \in X$, $\{< b_{ij}, x > \notin r_i\}^{1 \le i \le l}_{1 \le j \le n_i}$

The claim now is that this really defines a natural transformation

$$\lceil \mathcal{R} \rceil : \Sigma B \Rightarrow BT$$

ie for every 'variable renaming' $f : X \to Y$, $BTf \circ \lceil \mathcal{R} \rceil_X \;=\; \lceil \mathcal{R} \rceil_Y \circ \Sigma Bf$.

Consider the case of negative premises: if there is no pair $< b_{ij}, x >$ in $r_i \in BX$ for any $x \in X$, then there is also no pair $< b_{ij}, y >$ in $(Bf)(r_i) \in BY$ for arbitrary $y \in Y$. Assume thus only positive premises in the rule. Then the following lemma suffices to prove the claim.

Substitution Lemma. $(Tf)(C[\overrightarrow{\gamma_X r}, \overrightarrow{x}]) = C[\overrightarrow{\gamma_Y(Bf)(r)}, \overrightarrow{fx}]$

Proof. It is an immediate consequence of the naturality of the retraction γ from B to T and of the GSOS condition that the variables of $C[\overrightarrow{u}, \overrightarrow{v}]$ are contained in the set of u_i's and $v_{i,j}$'s (hence $(Tf)C[\ldots] = C[(Tf)\ldots]$). □

Next, this natural transformation $\lceil \mathcal{R} \rceil : \Sigma B \Rightarrow BT$ can be made into an action of Σ on BT:

$$\Sigma BT \stackrel{\lceil \mathcal{R} \rceil_T}{\Longrightarrow} BT^2 \stackrel{B\mu}{\Longrightarrow} BT$$

This family of Σ-algebras validates the semi-lattice laws, thus, using (6), it can also be seen as an action

$$\phi^{\mathcal{R}} : TBT \Rightarrow BT$$

of the syntactical monad T. Then, like in the basic property, one can obtain an operational monad Φ lifting the monad T to the B-coalgebras (instead of to the D-coalgebras) by putting $\Phi = (B\eta \circ \_)^\sharp$, ie

$$\begin{tikzcd} X \arrow[r, "\eta_X"] \arrow[d, "k"'] & TX \arrow[d, "\Phi k"] \\ BX \arrow[r, "B\eta_X"'] & BTX \end{tikzcd} \qquad \begin{tikzcd} TX \arrow[d, dashed, "\Phi k = (B\eta_X \circ k)^\sharp"'] & T^2X \arrow[l, "\mu_X"'] \arrow[d, "T\Phi k"] \\ BTX & TBTX \arrow[l, "\phi^{\mathcal{R}}_X"] \end{tikzcd}$$

(Notice the coalgebra $k : X \to BX$ can be seen, by (7), as a set of "δ-*rules*" in the sense of [BIM88].)

Theorem.

The operational semantics induced by $\mathcal{R}$ is *observationally equivalent* to Φ.

Proof. Consider, without loss of generality, the case of closed terms $T0$. Call $[\![\text{-}]\!]_{\mathcal{R}} : T0 \to BT0$ the coalgebra corresponding, via (7), to the transition system induced by $\mathcal{R}$ starting from the empty transition system (ie from the coalgebra $0 : 0 \to B0$). Similarly, put $[\![\text{-}]\!]_{\lceil\mathcal{R}\rceil} = \Phi 0 : T0 \to BT0$. The claim is that the final coalgebra semantics $[\![\text{-}]\!]^@_{\mathcal{R}} : (T0 \xrightarrow{[\![\text{-}]\!]_{\mathcal{R}}} BT0) \to (D1 \cong BD1)$ and $[\![\text{-}]\!]^@_{\lceil\mathcal{R}\rceil} : (T0 \xrightarrow{[\![\text{-}]\!]_{\lceil\mathcal{R}\rceil}} BT0) \to (D1 \cong BD1)$ are the same. The idea is that the final coalgebra semantics abstracts from the actual name of the states and just looks at the actions which can be performed. Then, since $< a, t' > \in [\![t]\!]_{\lceil\mathcal{R}\rceil}$ iff there exists a context $C[\overrightarrow{u}, \overrightarrow{v}]$ and terms u_i's and v_{ij}'s such that $< a, C[\overrightarrow{u}, \overrightarrow{v}] > \in [\![t]\!]_{\mathcal{R}}$ and $t' = C[\overrightarrow{\gamma_{T0}[\![u]\!]_{\lceil\mathcal{R}\rceil}}, \overrightarrow{v}]$, the theorem follows from the following

Lemma. $[\![C[\overrightarrow{\gamma_{T0}[\![u]\!]_{\lceil\mathcal{R}\rceil}}, \overrightarrow{v}]]\!]^@_{\lceil\mathcal{R}\rceil} = [\![C[\overrightarrow{u}, \overrightarrow{v}]]\!]^@_{\lceil\mathcal{R}\rceil}$

Proof. From the second corollary of the main theorem, the final coalgebra semantics $[\![\text{-}]\!]^@_{\lceil\mathcal{R}\rceil} : T0 \to D1$ is also an initial algebra semantics (wrt the denotations $\Phi^@ 1$), hence $[\![\text{-}]\!]^@_{\lceil\mathcal{R}\rceil}$ is compositional and the lemma can be reduced to

$$[\![\gamma_{T0}[\![u_i]\!]_{\lceil\mathcal{R}\rceil}]\!]^@_{\lceil\mathcal{R}\rceil} = [\![u_i]\!]^@_{\lceil\mathcal{R}\rceil}$$

which is a consequence of the fact that γ is a retraction for (basic process algebra and hence, as one can check, for) $[\![\text{-}]\!]_{\lceil\mathcal{R}\rceil}$. $\square$

Notice that $[\![\text{-}]\!]^@_{\lceil\mathcal{R}\rceil} : T0 \to D1$ is the unique arrow from the initial to the final Φ-algebra.

GSOS models are Φ-models. Spelling out the definition of Φ-models (alias Φ-algebras) for the operational monad Φ corresponding to a set $\mathcal{R}$ of GSOS rules, one obtains those

$$TX \xrightarrow{h} X \xrightarrow{k} BX$$

such that $h : TX \to X$ validates the T-algebra laws and such that

$$<a, x'> \in (k \circ h)(\sigma(x_1, \ldots, x_l))$$

iff there exists a rule (12) in $\mathcal{R}$ such that $<a_{ij}, y_{ij}> \in k(x_i)$, $x' = h(C[\overrightarrow{x}, \overrightarrow{y}])$, and, for all $y \in X$, $<b_{ij}, y> \notin k(x_i)$. Up to the isomorphisms (5) and (7), this is the definition of **GSOS models** given in [Sim95].

Guarded Recursion, coalgebraically. Every set of terms (mutually) recursively defined by means of equations in some variables $x_i \in X$

$$x_1 = t_1[X],\ x_2 = t_2[X], \ldots$$

where $t_i[X]$ are elements of TX (hence might contain variables from X), can be seen as a T-coalgebra $k : X \to TX$ by putting $k(x_i) = t_i[X]$. (And vice versa.) In order to interpret the recursive terms $x_i = t_i[X]$ operationally, the usual requirement is that they are **guarded**, that is, every term t_i is of the form $(a_{i_1} . t_{i_1})$ $\mathsf{or} \ldots \mathsf{or}$ $(a_{i_n} . t_{i_n})$. Notice then, that if all terms in a recursive definition are guarded, the corresponding coalgebra $k : X \to TX$ always factorizes through a BT-coalgebra $g : X \to BTX = \mathcal{P}_{\mathrm{fi}}(\mathsf{Act} \times TX)$ as follows.

$$k = \mu_X \circ \gamma_{TX} \circ g : X \to TX$$

Clearly, $g(x_i) = \{<a_{i_1}, t_{i_1}>, \ldots, <a_{i_n}, t_{i_n}>\}$. Conversely, every BT-coalgebra can be seen as a set of mutually recursive definitions.

Now, one can take the left adjunct wrt the adjunction (4) of every $g : X \to BTX$ using a given set of GSOS rules $\mathcal{R}$:

$$\begin{tikzcd}
X \arrow[r, "\eta_X"] \arrow[dr, "g"'] & TX \arrow[d, "{[\![\text{-}]\!]^g_{\lceil \mathcal{R} \rceil}}"] & & TX \arrow[d, dashed, "{[\![\text{-}]\!]^g_{\lceil \mathcal{R} \rceil} = g^\sharp}"'] & T^2X \arrow[l, "\mu_X"'] \arrow[d, "{T[\![\text{-}]\!]^g_{\lceil \mathcal{R} \rceil}}"] \\
& BTX & & BTX & TBTX \arrow[l, "\phi^{\mathcal{R}}_X"]
\end{tikzcd}$$

Then the final coalgebra semantics $([\![\text{-}]\!]^g_{\lceil \mathcal{R} \rceil})^@ : TX \to D1$ from the resulting coalgebra $[\![\text{-}]\!]^g_{\lceil \mathcal{R} \rceil} : TX \to BTX$ to the final coalgebra $D1 \cong BD1$ gives the desired interpretation of g as a recursive process. Notice that no variable binding operator (like, eg, 'fix' in [BIM88]) is (explicitly) needed here.

Example. Let $\mathcal{R}$ be basic process algebra together with the rules for (simple) *interleaving*

$$\frac{u \xrightarrow{a} u'}{u \parallel v \xrightarrow{a} u' \parallel v} \qquad \frac{v \xrightarrow{a} v'}{u \parallel v \xrightarrow{a} u \parallel v'}$$

and let g be the BT-coalgebra corresponding to the guarded recursive definition

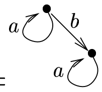
$$x = a\,.\,x \qquad y = (a\,.\,y)\ \mathsf{or}\ (b\,.\,x)$$

in $X = \{x, y\}$. (Notice that the x's and y's in the interleaving rules are *meta*-variables not to be confused with the actual variables x, y used in the recursive definition.) Writing, for simplicity,

$$[\![\text{-}]\!] = [\![\text{-}]\!]^g_{\lceil\mathcal{R}\rceil} : TX \to BTX$$

and letting $[\![\text{-}]\!]^@ : TX \to D1$ be the unique Φ-algebra arrow from $T^2X \xrightarrow{\mu_X} TX \xrightarrow{[\![\text{-}]\!]} BTX$ to the final Φ-algebra $TD1 \xrightarrow{\Phi^@1} D1 \xrightarrow{\sim} BD1$, one has, omitting the insertion-of-variables η_X and the final coalgebra isomorphism $D1 \cong BD1$,

$$[\![x]\!]^@ = \{<a, [\![x]\!]^@>\} =$$ 

$$[\![y]\!]^@ = \{<a, [\![y]\!]^@>, <b, [\![x]\!]^@>\} =$$ 

$$\begin{array}{lcl}
[\![a\,.\,t]\!]^@ & = & \{<a, [\![\gamma_{TX}[\![t]\!]]\!]^@>\} = \{<a, [\![t]\!]^@>\} \\
[\![t_1\ \mathsf{or}\ t_2]\!]^@ & = & \{<a, [\![t_1']\!]^@> \mid t_1 \xrightarrow{a} t_1'\} \cup \\
 & & \{<a, [\![t_2']\!]^@> \mid t_2 \xrightarrow{a} t_2'\} \\
 & = & [\![t_1]\!]^@ \cup [\![t_2]\!]^@ = \Phi^@1([\![t_1]\!]^@\ \mathsf{or}\ [\![t_2]\!]^@) \\
[\![t_1 \parallel t_2]\!]^@ & = & \{<a, [\![t_1' \parallel t_2]\!]^@> \mid t_1 \xrightarrow{a} t_1'\} \cup \\
 & & \{<a, [\![t_1 \parallel t_2']\!]^@> \mid t_2 \xrightarrow{a} t_2'\} \\
 & = & \Phi^@1([\![t_1]\!]^@ \parallel [\![t_2]\!]^@)
\end{array}$$

Final Remarks. The retraction $\gamma : B \Rightarrow T$ gives a general way of dealing with guarded recursion, but it is not clear whether its use and the assumption that the rules conservatively extend basic process algebra are really necessary to present GSOS functorially. At the moment, basic process algebra, with its natural *denotational* definition, seems to be *the* language for the behaviour $BX = \mathcal{P}_{\mathrm{fi}}(\mathsf{Act} \times X)$ (somewhat like the untyped lambda-calculus is *the* language for a suitable function space functor [Sco80]), while all other GSOS rules seem to be intrinsically *operational* and in a less direct correspondence with the behaviour, although denotationally well-behaved.