5. Signed theories

As discussed in Section 4 (see Part 1), algebraic theories in the general sense of Lawvere [23] are abstract with respect to presentations: neither any given operations nor any particular equations are determined by the isomorphism class of a theory; in fact, not even the ranks or the total number of operations are determined. This degree of abstraction is not appropriate for the specification of programming concepts, because the user of (say) a database will in fact have available a particular set of basic operations, out of which any other operations have to be explicitly synthesized (as so-called derived operators). That is, a particular signature is actually given. It will do no harm if the names of sorts and operations are changed, as long as the relationships among them are preserved. Thus, we can permit determination up to signature isomorphism. We do, however, definitely want full abstraction with respect to the equations part of a presentation. For example, the user of an abstraction...
Definition 7. A signed theory is a pair \( \langle \Sigma, T : \mathbb{T}_\Sigma \rightarrow \mathbb{T} \rangle \), where \( \Sigma \) is a signature, \( \mathbb{T} \) is a theory, and \( T \) is a theory morphism.

If \( \langle \Sigma', T' : \mathbb{T}_{\Sigma'} \rightarrow \mathbb{T}' \rangle \) is another signed theory, then a signed theory morphism \( \langle \Sigma, T : \mathbb{T}_\Sigma \rightarrow \mathbb{T} \rangle \rightarrow \langle \Sigma', T' : \mathbb{T}_{\Sigma'} \rightarrow \mathbb{T}' \rangle \) is a pair \( \langle f, H \rangle \) such that \( f : \Sigma \rightarrow \Sigma' \) is a signature morphism and \( H : \mathbb{T} \rightarrow \mathbb{T}' \) is a theory morphism such that

commutes (where \( \mathbb{T} \) denotes the free functor \( F : \text{Sig} \rightarrow \text{Theo} \), so that \( \mathbb{T}_\Sigma = F(\Sigma) \)).

Given also a signed theory morphism \( \langle f', H' \rangle : \langle \Sigma', T' : \mathbb{T}_{\Sigma'} \rightarrow \mathbb{T}' \rangle \rightarrow \langle \Sigma'', T'' : \mathbb{T}_{\Sigma''} \rightarrow \mathbb{T}'' \rangle \), the composition \( \langle f, H \rangle \langle f', H' \rangle \) is defined to be \( \langle f'' , HH' \rangle \). The identity \( I_T \) on \( T \) is \( \langle I_\Sigma, I_\mathbb{T} \rangle \). Let \( \text{SigTheo} \) denote the category of signed theories.

For notational convenience, we may write just \( T \) for the pair \( \langle \Sigma, T : \mathbb{T}_\Sigma \rightarrow \mathbb{T} \rangle \), and in fact, it is possible to recover \( S \) and \( \Sigma \) from the functor \( T : \mathbb{T}_\Sigma \rightarrow \mathbb{T} \) by examining the concrete elements of its source, the free theory \( \mathbb{T}_\Sigma \). (But \( S \) and \( \Sigma \) cannot be recovered from the isomorphism classes of \( T \) or of \( \mathbb{T} \).) Recall that we have already introduced the similar convention of denoting \( \langle S, \Sigma \rangle \) by just \( \Sigma \).

It is easy to see that \( \text{SigTheo} \) actually is a category. In fact, \( \text{SigTheo} \) is (isomorphic to) the comma category \( (\text{Sig}/ \text{U}) \), as can be seen by comparing Definition 7 with Section 2. Because \( F \) is a left adjoint, it preserves colimits, and we are able to conclude the following from Corollary 4:

**Proposition 12.** \( \text{SigTheo} \) is cocomplete.

By Proposition 1, \( \text{SigTheo} \) is also isomorphic to the comma category \( (\text{Sig}/ \text{U}) \), where \( \text{U} : \text{Theo} \rightarrow \text{Sig} \) is the forgetful functor. Under this isomorphism, \( T : \mathbb{T}_\Sigma \rightarrow \mathbb{T} \) is represented by \( D : \Sigma \rightarrow \text{U}(\mathbb{T}) \) such that \( D^e - T \).
Because of the fact that theories are themselves composite objects (namely, functors) it may be worth expanding Definition 7 by substituting in the relevant definitions from Section 4. However, it should definitely be emphasized that the handle which the comma category concepts give us on this situation, makes it in general unnecessary to actually work with these concepts at the finer level of detail.

Let us assume, then, that $T$ is in more detail the functor $J: T_{S_0} \to T$; and of course, $T_{S}$ is the inclusion function $I: T_{S} \hookrightarrow T_{X}$ with quite possibly $S_0 \neq S$. (If the reader is confused by this, he should review Definition 3 and related material in Section 4.) Next, a theory morphism $I \to J$ is a pair $(\psi, T)$ with $\psi: S \to S_0$ a function and $T: T_{S} \to T$ a functor, such that

\[
\begin{array}{ccc}
T_{S} & \xrightarrow{I} & T_{S} \\
\downarrow & & \downarrow \\
T_{S_0} & \xleftarrow{J} & T
\end{array}
\]

commutes in Theo, where $\Psi$ is the functor induced by $\psi$ (i.e., $\Psi = T_{\psi}$). This discussion should make clear the convenience of writing just $T: T_{S} \to T$.

While we are at it, we will also expand the definition of a morphism of signed theories. Let $T': T_{S} \to T'$ be another signed theory, with $\Psi', \psi', I', J', S_0', S'$ corresponding to the undashed symbols above. Then the signed theory morphism looks as follows:

\[
\begin{array}{ccc}
T_{S} & \xrightarrow{T'} & T_{S'} \\
\downarrow & & \downarrow \\
T_{S_0} & \xleftarrow{J'} & T'
\end{array}
\]

Once again, the value of the comma category concepts becomes clear, in that it is unnecessary to deal with this complexity directly.

If we restrict to $S$-sorted theories, for a fixed $S$, things get simpler. A signed theory is a diagram
and thus may as well just be \( T: \mathcal{T}_\Sigma \rightarrow \mathcal{T} \). A morphism of signed theories is a diagram

\[
\begin{array}{ccc}
\mathcal{T}_\Sigma & \xrightarrow{T} & \mathcal{T} \\
\downarrow & & \downarrow \\
\mathcal{T}_S & \xrightarrow{T \circ \rho} & \mathcal{T}^{'(\rho, \theta)} \\
\end{array}
\]

and therefore might as well just be the square, without the left-hand triangle.

There is an additional point about signed theories which should be mentioned here, although it will be taken up in more detail later. The functor \( T: \mathcal{T}_\Sigma \rightarrow \mathcal{T} \) need not be surjective on either objects or morphisms. This permits there to be in \( \mathcal{T} \) both sorts and operations which are 'hidden' from the point of view of a user of the abstraction who only has access to operations in \( \Sigma \) (plus combinations of them, giving all of \( \mathcal{L} \)).

It is often very convenient to have such hidden sorts and operations for use in defining the semantics of those which are in \( \Sigma \); but it is also the case that one sometimes wants to get rid of this scaffolding after it has done its job. This leads to the desirability of taking the image of \( T \) in \( \mathcal{T} \) to get a surjective theory morphism, as discussed in Section 7.

We conclude this section with a discussion of CLEAR's enrich . . . by . . . enden construction, which adds further sorts and operations to a given signed theory: Let \( T: \mathcal{T}_\Sigma \rightarrow \mathcal{T} \) be a signed theory, where \( \Sigma \) has sort set \( S \), and let \( P = (S \cup S', \Sigma \cup \Sigma', \mathcal{E}) \) be a presentation with \( S, S' \) and \( \Sigma, \Sigma' \) disjoint. Define \( T'': \mathcal{T}_{\Sigma \cup \Sigma'} \rightarrow \mathcal{T}_P \) to be the usual quotient morphism. Then we say that \( (S', \Sigma', \mathcal{E}) \) enriches \( T: \mathcal{T}_\Sigma \rightarrow \mathcal{T} \) iff there is an injective theory morphism \( F: \mathcal{T} \rightarrow \mathcal{T}_P \) such that

\[
\begin{array}{ccc}
\mathcal{T}_\Sigma & \xrightarrow{T} & \mathcal{T} \\
\downarrow & & \downarrow \\
\mathcal{T}_{\Sigma \cup \Sigma'} & \xrightarrow{T''} & \mathcal{T}_P \\
\end{array}
\]

\(^7\) We call a theory morphism \( (\Phi, F) \) injective iff both \( \Phi \) and \( F \) are injective on both objects and morphisms, for \( \Phi \) to be injective, where \( \Phi = T\psi \), it suffices for \( \psi \) to be injective.
commutes, where \( T \rightarrow T \Lambda \) is the evident inclusion; and the expression "enrich \( T \) by \( P \) enden" is taken to denote the morphism \( T' \). (If \( S' \neq 0 \), it should be called an extension, and if \( F \) is not bijective an extrusion; but CLEAR syntax does not make these distinctions. Note that if \( F \) is not injective, then the construction is taken to be ill-formed.) Some examples of this construction are given in Section 9.

If one is willing to restrict attention to the case of theories without hidden sorts or operations, then it is possible to give a description of signed theories in terms of presentations: the full subcategory of \( \textbf{SigTheo} \) with surjective morphisms as objects is equivalent to the full subcategory of \( \textbf{Pres} \) with deductively closed families of equations. (Note that such presentations are not necessarily fully closed, because the signature has not been closed, just the equations.)

6. Derivers and Kleisli categories

The version of CLEAR in [8] has a feature for 'deriving' one theory from another, by defining operators in the new theory in terms of derived operators in the old theory; this can be viewed as a construction upon signed theories, but it takes us outside the category given in Definition 7. The purpose of this construction is not only to define new operations in terms of old ones, but also to 'abstract' out unused aspects of the old structure. The abstraction aspect is handled by the factorization introduced in Section 7. The derive construction is conveniently developed by starting from the notion of a 'derivor' as a more general kind of morphism between signatures.

Because the construction is a very general one, using only adjointness, and because it is well known in its general form as the Kleisli category (see [25, p. 143]), we develop it in the following abstract setting: Let \( A, B \) be categories with \( F: A \rightarrow B, U: B \rightarrow A \) functors, \( F \) left adjoint to \( U \) with unit components \( \eta_A: A \rightarrow U(F(A)) \).

**Definition 8.** The category \( \text{Der}(F \dashv U) \) of derivors for the adjunction \( F \dashv U \) is defined as follows: its objects are those of \( A \); its morphisms \( A \rightarrow A' \) are derivors, which are defined to be morphisms \( D: A \rightarrow U(F(A')) \) in \( A \); its composition \( A \rightarrow A' \rightarrow A'' \) is defined to be the composition \( A \rightarrow U(F(A')) \rightarrow U(F(A'')) \) in \( A \) and its identity \( I_A \) on \( A \) is \( \eta_A \).

The reader may wish to verify that derivors, as defined above, correspond for algebraic theories to derivors in the sense of ADJ [3], [8] or [16]. A number of examples of derivors are given in these papers; and this paper gives one in the next section.

**Proposition 13.** In the above situation,

1. there is a bijection between derivors \( A \rightarrow U(F(A')) \) and morphisms \( F(A) \rightarrow F(A') \) in \( B \), given by \( \# \) as in Section 2;
2. under this bijection, \( \eta_A: A \rightarrow U(F(A)) \) corresponds to \( 1_{F(A)}: F(A) \rightarrow F(A) \);
(3) \( \text{Der}(F \to U) \) is a category, in fact, the Kleisli category of the monad of \( (F \to U) \) in \( A \):

(4) defining \( E(A) = F(A) \) for \( A \in |A| \), and \( E(A \to U(F(A'))) = F(A) \underbrace{\to}_{D^\phi} F(A') \),
gives a functor \( E: \text{Der}(U \to F) \to B \); corestricting this functor to a full subcategory \( \text{Der} \) of \( B \) with objects of the form \( F(A) \) for \( A \in |A| \) yields an equivalence of categories; it is an isomorphism if \( F \) is injective on objects and morphisms;

(5) if \( F \) is faithful, then \( A \) is (or can be viewed as) a subcategory of \( \text{Der}(F \to U) \) by the functor \( I: A \to \text{Der}(F \to U) \) with \( I(A) = A \) for \( A \in |A| \), and with, for \( A \to A' \), \( I(a) \) the composite \( A \xrightarrow{a} A' \xrightarrow{\eta} U(F(A')) \); \( I \) is functorial even if \( F \) is unfaithful; and, finally, \( I \circ E = F \).

**Proof.** Both (1) and (2) are general factors about adjunctions; indeed, the bijection is exactly the function \( \# \) of Section 2. The relevant pictures are

\[
\begin{array}{ccc}
A \xrightarrow{\eta} U(F(A)) & \text{and} & A \xrightarrow{\eta} U(F(A)) \\
F(A) \xrightarrow{\phi} F(A') & \text{and} & F(A) \xrightarrow{\phi} F(A')
\end{array}
\]

respectively.

(3) To show \( D = \text{Der}(F \to U) \) is a category, we must show that composition is associative, and that \( \eta_A \) is really an identity.

Let \( A \to A' \to A'' \) be morphisms in \( D \), let \( * \) denote composition in \( D \), and let \( \circ \) denote composition in both \( A \) and \( B \). First, note the correspondence

\[
A' \xrightarrow{D'} I(F(A'')) \xrightarrow{U(D')^\phi} U(F(A''))
\]

\[
F(A') \xrightarrow{D^\phi} F(A') \xrightarrow{D^\phi} F(A'')
\]

a special case of rule (RNAT). Then

\[
D \circ (D' \circ D'') = D \circ ((D' \circ D'(D''D^\phi)) \circ U(D''D^\phi)) = D \circ (U(D'(D''D^\phi)) \circ U(D''D^\phi)) = (D \circ U(D'') \circ U(D') \circ U(D''))
\]

as desired.

Next,

\[
(\eta_A \circ D)^\phi = (\eta_A \circ U(D) \circ D^\phi) = (1_A \circ D^\phi) = D^\phi
\]

so \( \eta_A \circ D = D \).

\[
\begin{array}{ccc}
A \xrightarrow{\eta} U(F(A)) & \text{and} & A \xrightarrow{\eta} U(F(A)) \\
F(A) \xrightarrow{\phi} F(A') & \text{and} & F(A) \xrightarrow{\phi} F(A')
\end{array}
\]

Similarly, \( D \circ \eta_A = D \).

We now show that \( D \) is the Kleisli category in \( A \) for the monad \( T = (T, \eta, \mu) \) of the adjunction \( (F \to U) \), where \( T = FU \), and \( \mu = FEU: F(UF)U \to FU \), where \( \eta \) and
\(e\) are the unit and co-unit of the adjunction (see [25, p. 137]). The Kleisli category objects are those of \(A\), and the Kleisli morphisms are as defined above. It remains to check that the two compositions agree. For \(A \overset{D}{\to} A' \overset{D'}{\to} A''\), the Kleisli composition \(\circ\) is given by (see [25, p. 143])

\[
D \circ D' = D \circ U(F(D')) \circ \mu_{A''}
\]

\[
= D \circ U(F(D')) \circ \varepsilon_{F(A''')}
\]

\[
= D \circ U(D''')
\]

\[
= D \ast D',
\]

using \(\varepsilon_B = (1_{U(B)})^*\) and naturality, as desired.

(4) To show that \(E\) is a functor, we must show that it preserves identities and compositions. Preservation of identities is shown by (2) above, and preservation of composition is the following calculation, for \(A \overset{D}{\to} A'\) and \(A' \overset{D'}{\to} A''\) in \(D\),

\[
E(D \ast D') = E(D \circ U(D'''))
\]

\[
= (D \circ U(D'''))^*
\]

\[
= D^* \circ D' \quad \text{(by (RNA\(\Gamma\))})
\]

\[
= E(D) \circ E(D').
\]

To show that \(E\) is an equivalence of \(A\) with the full subcategory \(\overline{\text{Der}}\) of \(B\) with objects \(F(A)\), it suffices to show (see [25, p. 91]) that \(E\) is full and faithful, and that each object in its target is isomorphic to an object in the image. \(E\) is full and faithful because in fact each \(E_{A, A'}: D(A, A') \to B(F(A), F(A'))\) is an isomorphism, namely \(\#\): and the third condition is satisfied because an object \(F(A)\) is in the target category \(\overline{\text{Der}}\) iff it is in the image of \(E\), by definition of \(\overline{\text{Der}}\) as the full subcategory of \(B\) with objects of the form \(F(A)\). If \(F\) is injective, then so is \(E\); then \(E\) is an isomorphism.

We omit the proof of (5), except to note that \(I \circ E = F\) follows from rule (LUN) of Section 2. \(\Box\)

In the case of particular interest, \(A = \text{Sig}, B = \text{Theo}\), and we shall write \(\text{Der}\) for the category \(\text{Der}(F \to U)\) of derivors.

Now if \(\Sigma \overset{D}{\to} U(T)\) is a derivor and \(T \overset{T}{\to} \mathbb{T}\) is a signed theory, then we can form the composition \(\Sigma \overset{D}{\to} U(T) \overset{U(T)}{\to} U(\mathbb{T})\), which gives a new signed theory \(T \overset{T}{\to} \mathbb{T}\) with signature \(\Sigma\), namely \((D \circ U(T))^*\). The derive construction of Burstall and Goguen [8] gives the image (using factorization—see Section 7) of this theory morphism.

**Definition 9.** A derived signed theory morphism, from signed theory \(T: T \overset{T}{\to} \mathbb{T}\) to \(T': T' \overset{T'}{\to} \mathbb{T}'\), is a pair \((\Phi, H)\), where \(\Phi: T \overset{T}{\to} T\) and \(H: \mathbb{T} \to \mathbb{T}'\) are morphisms in
Theo such that

![Diagram]

commutes. Defining composition the obvious way, let DSigTheo denote the category with signed theories as objects and derived signed theory morphisms.

The reason for the name is that any morphism $\Phi: T_\Sigma \to T_\Sigma'$ comes from a unique derivor $D: \Sigma \to U(T_\Sigma')$, namely $D = \eta_{\Sigma'} U(\Phi)$. Furthermore, we have the following.

**Proposition 14.** In the above situation:

1. DSigTheo is (isomorphic to) the comma category $(E/Theo)$, where $E$ is the extension of $F$ from source Sig to source Der given in (4) of Proposition 13.
2. Define $J: Der \to DSigTheo$ to send the object $C$ to the identity morphism $U_L \to T_X$ (an object in DSigTheo) and to send the derivor $\Sigma \to U(T_\Sigma)$ to the morphism

   ![Diagram]

   in which the horizontal (unlabelled) arrows are identities. Then $J$ is a full and faithful functor, and is injective if $F$ is injective on objects.

3. The category SigTheo of signed theories (Definition 7) is the subcategory of DSigTheo whose morphisms $(\Phi, H)$ have $\Phi = F(\varphi)$, with $\varphi: \Sigma \to \Sigma'$ in Sig (more accurately, there is an injective functor SigTheo $\to$ DSigTheo whose object part sends $T: T_\Sigma \to T$ to itself, and whose morphism part sends $(\varphi, H)$ to $(F(\varphi), H)$, where $F$ is the free theory functor Sig $\to$ Theo).

We now discuss colimits in Der, starting with the following result, whose proof we omit.

**Lemma 15.** If $D \subseteq A$ is a full subcategory, and $M: G \to D$ is a diagram in $D$ with colimit $\alpha: M \Rightarrow A$ in $A$ consisting of morphisms $\alpha(n): M(n) \to A$ actually in $D$, then $\alpha$ is also a colimit of $M$ in $D$.

Note that because $F: Sig \to Theo$ is a left adjoint, it preserves colimits; that Sig has finite colimits; and that $F$ factors through Der. Letting $F': Sig \to Der$ denote
this factorization, and letting \( \overline{E}: \overline{\text{Der}} \to \text{Der} \) denote the equivalence, we have the following diagram:

\[
\begin{array}{c}
\text{Sig} \rightarrow \text{Theo} \\
\downarrow F' \quad \downarrow F \\
\text{Der} \rightarrow \text{Der} \\
\end{array}
\]

in which \( H = F' \overline{E} \). Then we have the following.

**Proposition 16.** If \( M: G \to \text{Der} \) is a diagram of the form \( M' H \), and \( M': G \to \text{Sig} \) has colimit \( \alpha: M' \Rightarrow \Sigma \), then \( M \) has colimit \( \alpha H: M \Rightarrow \Sigma \).

**Proof.** \( M' F' \) has colimit \( \alpha F' \) in \( \text{Theo} \), and \( \alpha F' \) actually lies in \( \overline{\text{Der}} \); thus, \( M' F' \) has colimit \( \alpha F' \), by Lemma 15. Since \( \overline{E} \) is an equivalence, it also preserves colimits; thus \( M' F' \overline{E} \) has colimit \( \alpha F' \overline{E} \). \( \square \)

It follows that \( \text{Der} \) has colimits of finite diagrams coming from \( \text{Sig} \); in general, it has few others. However, note that finite coproducts in \( \text{Der} \) are included among those which do exist.

We now give a result on colimits in the category \( \text{DSigTheo} \) of signed theories with derivors as morphisms. Recall that there is a full injective functor \( J: \text{Der} \to \text{DSigTheo} \). Thus, we have the following diagram and result, using \( H \) as above:

\[
\begin{array}{c}
\text{Sig} \rightarrow \text{Der} \rightarrow \text{DSigTheo} \\
\end{array}
\]

**Proposition 17.** If \( M: G \to \text{DSigTheo} \) is a diagram of the form \( M' i i J \), where \( M': G \to \text{Sig} \) has colimit \( \alpha: M' \Rightarrow \Sigma \), then \( M \) has colimit \( \alpha H J: M \Rightarrow \Sigma J \).

**Proof.** Using that \( J \) is injective, apply Lemma 15 and Proposition 16. \( \square \)

7. Factorizations

As defined in [8], \( \text{CLEAR} \) has a construction called derive which permits one to define, from a signed theory \( T: \Sigma \Rightarrow T \), a new signed theory, say \( T': \Sigma \Rightarrow T' \), by
giving a derivor \( D: \Sigma' \to U(\Sigma) \). \( T' \) is actually obtained as a factorization of the composition \( D^* T \), as indicated in the diagram

\[
\begin{array}{ccc}
T' & \rightarrow & \Sigma' \\
\downarrow & & \downarrow \\
\Sigma & \rightarrow & T
\end{array}
\]

The idea is to pick out a subtheory \( T' \) of \( T \), and give it a signature \( \Sigma' \) whose operation symbols denote derived operators in \( \Sigma \), as defined by the derivor \( D \). The process of 'pruning' theories by discarding unused parts corresponds to what is called 'abstraction' in computer science, in that some details about how the theory was constructed are lost. It follows from results of Thatcher, Wagner and Wright [32] (following Majster [31]) that theories can be obtained in this way, which cannot be obtained purely from a finite equational presentation. The discarded parts of the theory \( T \) correspond to what Parnas has called 'hidden operators'.

The key to being able to do this, is to be able to factor an arbitrary theory morphism into a composition of an 'extremal' epimorphism (this is defined below) followed by an injective morphism. We shall first discuss factorization situations in general, since there are quite a number of different ways of factoring morphisms in arbitrary categories; the following is a convenient way of axiomatizing such situations, apparently due to Isbell [20] (see also Herrlich and Strecker [19]).

**Definition 10.** An image factorization system for a category \( C \) is a pair \( (E, M) \) of classes of morphisms of \( C \) such that

1. \( E \) and \( M \) are closed under composition,
2. all isomorphisms are in both \( E \) and \( M \),
3. (IFS3) morphisms in \( E \) are epic, and in \( M \) are monic,
4. every \( f \in C \) has an \( (E, M) \)-factorization \( \langle e, m \rangle \) (this means that \( f = e \circ m \)) and this factorization is unique up-to-isomorphism, in the sense that if \( \langle e', m' \rangle \) is another \( (E, M) \) factorization of \( f \), then there is a unique isomorphism \( c \in C \) such that the diagram

\[
\begin{array}{ccc}
A & \xrightarrow{e} & C \\
\downarrow & & \downarrow c \\
C' & \xleftarrow{e'} & B
\end{array}
\]

commutes in \( C \).

The reason for proving that this particular set of axioms is satisfied is, of course, that a number of pleasant consequences are known to follow (see [19] or [27]). First
of all, it follows from (IFS1) and (IFS2) that $E$ and $M$ are both subcategories of $C$. Furthermore, we have the following.

**Proposition 18.** If $(E, M)$ is an image factorization system for a category $C$, then

1. the 'diagonal fill-in property' holds: given $e \circ g = f \circ m$, with $e \in E$, $m \in M$, there is a (necessarily unique) $c \in C$, such that

   \[
   \begin{array}{ccc}
   A & \xrightarrow{e} & B \\
   f \downarrow & \xleftarrow{c} & \downarrow g \\
   C & \xrightarrow{m} & D
   \end{array}
   \]

   commutes in $C$,

2. conversely, if $m \in C$ has the property that whenever $e \circ g = f \circ m$ for $e \in E$, there is $c \in C$ with $e \circ c = f$, then we must have $m \in M$; dually, for $e \in C$ and $m \in M$, if $c \circ g = f \circ m$ implies there is $c \in C$ with $e \circ c = f$, then $e \in E$,

3. if $f \in E$ and $f \in M$, then $f$ is an isomorphism,

4. if $f \in E$, then $g \in E$,

5. if $f \circ g \in M$, then $f \in M$,

6. if $e_i \in E$ and $\prod_i m_i$ exists, then $\prod_i e_i \in E$,

7. if $m_i \in M$ and $\prod_i m_i$ exists, then $\prod_i m_i \in M$.

It follows from (1) and (2) that in an $(E, M)$-factorization situation, $E$ uniquely determines $M$, and also $M$ uniquely determines $E$. In particular, taking $M$ to be the monomorphisms of a category $C$ determines $E$ to be the extremal epimorphisms of $C$, those epimorphisms which satisfy the condition: if $e = fm$ and if $m$ is a monomorphism, then $m$ must be an isomorphism.

The following generalization of (6) (along with its dual, which generalizes (7)) may be of use in some situations.

**Proposition 19.** Let $(E, M)$ be an image factorization system for $C$, let $M$ be a non-empty diagram in $C$, and let $\alpha(n) : M(n) \to C$ be a colimit cone (where $n$ ranges over the nodes in the shape of $M$). Now let $\beta(n) : M(n) \to E$ also be a cone in $E$\(^3\). Then the resulting morphism $b : C \to E$ (such that $\alpha(n) \cdot b = \beta(n)$) is also in $E$.

**Method of proof.** Use (4) of Proposition 18. \(\square\)

The following in many cases provides a useful sufficient condition for a category to have (extremal epic, monic)-factorizations (for the terminology, and a proof, see [19; 17.16])

\(^3\) It will actually suffice that any single morphism $\beta(n_i)$ be in $E$. 
**Theorem 20.** If a category \( C \) is well-powered, has intersections (of subobjects) and equalizers, then \( C \) has (extremal epic, monic)-factorizations.

Unfortunately, this result does not tell us what these factorizations look like, and this is something which we really do want to know.

**Definition 11.** A theory morphism \( \langle \phi, F \rangle : T \to T' \) is injective iff \( \phi \) is injective and \( F \) is injective (meaning both its object and morphism parts are injective). We shall call \( \langle \phi, F \rangle \) an inclusion (and also call \( T \) a subtheory of \( T' \)) iff both \( \phi \) and \( F \) are inclusions. The diagram below recalls the elements of this situation:

\[
\begin{array}{ccc}
T_S & \xrightarrow{J} & T \\
\downarrow{\phi} & & \downarrow{F} \\
T_S' & \xrightarrow{J'} & T'
\end{array}
\]

where \( \Phi = T_S \).

The graph image of a functor \( F : T \to T' \) is the subgraph\(^4\) of \( U(T') \) with its nodes the objects \(|F(w)| \) for \( w \in |T| \), and with its edges the morphisms \( F(t) \) for \( t \in T \): it is denoted \( \text{im}(F) \).

A theory morphism \( \langle \phi, F \rangle : T \to T' \) is generous iff \( T' \) is the smallest sub-theory of \( T' \) containing the graph image of \( F \).

The generous image of a theory morphism \( \langle \phi, F \rangle : T \to T' \), denoted \( \text{Gim}(\phi, F) \), is the smallest sub-theory of \( T' \) containing the graph image of \( F \), with \( J : T_S \to \text{Gim}(\phi, F) \), where \( S = \text{im}(\phi) \) and \( J \) is the simultaneous restriction of \( J' \) to \( T_S \) and corestriction to \( \text{Gim}(\phi, F) \) (but some times 'generous image' refers just to the sub-category \( \text{Gim}(\phi, F) \)).

**Proposition 21.** Using the above notation:

1. the generous image \( \text{Gim}(\phi, F) \) always exists, and is in fact the least subgraph of \( T' \) containing \( \text{im}(F) \) and closed (in \( T' \)) under composition and tupling;
2. \( \langle \phi, F \rangle \) is generous iff \( T' = \text{Gim}(\phi, F) \);
3. \( \langle \phi, F \rangle \) is generous whenever \( \langle \phi, F \rangle = \langle \phi, H \rangle \langle \iota, I \rangle \) with \( \langle \iota, I \rangle \) injective, then \( \langle \iota, I \rangle \) is an isomorphism; moreover, if \( \langle \iota, I \rangle \) is an inclusion, then it is an equality;
4. every morphism \( t' \in \text{Gim}(\phi, F) \) can be written as an expression \( \epsilon'(F(t_1), \ldots, F(t_n)) \) in the operations of composition and tupling, with \( t_i \in T \); conversely, each such expression lies in \( \text{Gim}(\phi, F) \):
5. a generous theory morphism is epic;
6. a composite of generous theory morphisms is generous.

\(^4\) It is known that the graph image is not necessarily a subcategory; for a simple example, see ADJ [1, p. 55]. The problem is that the composite of two morphisms in the image need not be in the image.
Proof. Observe first (a) that when talking about subtheories $T$ of $T'$, we can ignore the theory morphism $J: T_S \to T$, because it is implied the corestriction of $J': T_S \to T'$ to $T$.

(1) Now, if we show (b) that a subgraph $T$ of a theory $T'$ gives a subtheory iff it is closed under composition and tupling (in $T'$); and (c) any intersection of subtheories is a subtheory; then it follows that (d) the intersection of all subtheories containing a subgraph $T$ is the least subtheory of $T'$ containing $T$; that (e) there is a least subtheory $Gim(\varphi, F)$ of $T'$ containing $im(F)$; and that (f) $Gim(\varphi, F)$ is (from (b)) therefore the least subgraph of $T'$ closed under composition and tupling (in $T'$) and containing $im(F)$.

(b) follows from (a) and the definitions; (c) follows from (b); then (d), (e), and (f) follow as indicated.

(2) follows directly from (1) and the definitions.

(3) Note that the graph image of an injection gives a subtheory (using (a)). Thus, if $\langle \varphi, F \rangle$ is generous and factors as indicated, with $\langle \iota, I \rangle$ not an isomorphism, then $\langle \varphi, F \rangle$ factors through a proper subtheory $T'$ containing $im(F)$, contrary to the definition of generous. Conversely, if $T'$ is not generous, let $T$ be a proper subtheory of $T'$ containing $im(F)$, and let $\langle \iota, I \rangle$ be its inclusion. Then $\langle \varphi, F \rangle = \langle \psi, H \rangle \langle \iota, I \rangle$ with $\langle \psi, H \rangle$ the corestriction of $\langle \varphi, F \rangle$ to $T$.

If $\langle \varphi, F \rangle$ is generous and $\langle \iota, I \rangle$ is an inclusion, we get $T = T'$.

(4) Every $t' \in Gim(\varphi, F)$ is in the closure of $im(F)$ under composition and tupling in $T'$, by (1), and can therefore be given by an expression built up from composition, tupling, and the elements $F(t)$ of $im(F)$. The converse is obvious.

(5) Let $\langle \varphi, F \rangle$ be generous, and let

\[
\begin{array}{c}
\langle \varphi, F \rangle \\
\downarrow \\
T'
\end{array}
\quad \quad \quad
\begin{array}{c}
\langle \psi, H \rangle \\
\downarrow \\
T''
\end{array}
\quad \quad \quad
\begin{array}{c}
\langle \varphi, F \rangle \\
\downarrow \\
T'
\end{array}
\quad \quad \quad
\begin{array}{c}
\langle \psi, H' \rangle \\
\downarrow \\
T''
\end{array}
\]

 commute. By (2), $T' = Gim(\varphi, F)$. Let $t' \in T'$. Then by (4), $t' = e'(F(t_1), \ldots, F(t_n))$. Therefore, $H(t') = H(e'(F(t_1), \ldots, F(t_n))) = e'_\varphi(H(F(t_1)), \ldots, H(F(t_n)))$, using the fact that theory morphisms preserve both composition and tupling, but subscripting $e'_\varphi$ with $\varphi$ to indicate the sort of changes involved. Similarly, we have $H'(t') = e'_\varphi(H'(F(t_1)), \ldots, H'(F(t_n)))$. But $H(F(t_i)) = H'(F(t_i))$ (and $\psi(\varphi(s)) = \psi'(\varphi(s))$ for keeping track of sorts) implies $H(t') = H'(t')$. Thus, $H = H'$.

(6) Assume $\langle \varphi, F \rangle, \langle \varphi', F' \rangle$ generous in

\[
\begin{array}{c}
\langle \varphi, F \rangle \\
\downarrow \\
T'
\end{array}
\quad \quad \quad
\begin{array}{c}
\langle \varphi', F' \rangle \\
\downarrow \\
T''
\end{array}
\quad \quad \quad
\begin{array}{c}
\langle \varphi, F \rangle \\
\downarrow \\
T'
\end{array}
\quad \quad \quad
\begin{array}{c}
\langle \varphi', F' \rangle \\
\downarrow \\
T''
\end{array}
\]

Then $T' = Gim(\varphi, F)$ and $T'' = Gim(\varphi', F')$. Let $t'' \in T''$ be $e''(F'(t'_1), \ldots, F'(t'_n))$. Similarly, let $t'_i = e'_i(F(t_{i_1}), \ldots, F(t_{i_n}))$. Now substituting the expressions $e'_i$ in the $t_{i_j}$ for the $t_i$ into $e''$ gives an expression in $T''$ in the $F'(F(t_{i_j}))$. $\square$
Notice that (1) and (2) imply that every theory morphism can be factored as a generous morphism followed by an inclusion.

The explicit form of (4) turns out to be the most useful way of proving non-trivial facts about generosity. Condition (3) above can be paraphrased as "all generous theory morphisms are extremal". Here is one more collection of results before we summarize.

**Proposition 21 (continued).** Using the above notation still:

1. every theory morphism \((\varphi, F)\) factors uniquely as \((\psi, G)(\iota, I)\) with \((\psi, G)\) generous and \((\iota, I)\) an inclusion;
2. every theory morphism \((\varphi, F)\) factors as \((\psi, G)(\iota, I)\), with \((\psi, G)\) generous and \((\iota, I)\) an injection, uniquely up to isomorphism, in the sense that if \((\psi', G')(\iota', I')\) is another such factorization, then there is a unique theory isomorphism \((\rho, H)\) such that

\[
\begin{array}{ccc}
\psi, G & \Rightarrow & \iota, I \\
\cap & & \cap \\
\rho, H & \cong & \iota', I'
\end{array}
\]

commutes.

**Proof.** The notation of the above diagram will be used in the proofs of both (7) and (8) below.

We first prove uniqueness.

1. Let \(\tilde{t} \in \tilde{T}\). Then \(\tilde{t} = e(G(t_1), \ldots, G(t_n))\), for \(t_i \in T\). Now define \(\rho(\tilde{s})\) to be \(\psi'(s)\), where \(\tilde{s} = \psi(s)\), and define \(H(\tilde{t}) = e(G'(t_1), \ldots, G'(t_n))\). First, we show \(\rho\) is well defined: if \(\tilde{s} = \psi(s) = \psi(s')\), then \(\psi'(s) = \psi'(s')\), because \(\iota(\psi(s)) = \iota(\psi(s'))\) implies that \(\iota'(\psi(s)) = \iota'(\psi(s'))\). Then by definition, \(\rho(\psi(s)) = \psi'(s)\). Therefore, \(\iota'(\rho(\psi(s))) = \iota'(\psi'(s))\), and since \(\psi'(s) = \psi(s)\), \(\iota'(\rho(\psi(s))) = \iota(\psi(s))\), therefore \(\rho(\psi(s)) = \psi(s)\), and so \(\rho(\tilde{s}) = \tilde{s}\); that is, \(\rho\) is an inclusion. Furthermore, \(\rho\) is surjective, because \(\psi'\) is; therefore \(\rho\) is an identity.

Similar arguments apply as follows. To ensure that \(H\) is well defined, we first have to show that \(e(G'(t_1), \ldots, G'(t_n))\) makes sense in \(\tilde{T}\). Note that we do not need to write \(e\rho\) for \(e\), because \(\rho\) is the identity function: thus, it is only necessary to show that \(\partial_k G(t) = \partial_k G'(t)\), for all \(t \in T\) and \(k = 0, 1\). By calculation now, for \(k = 0, 1\),

\[
\partial_k G(t) = |G| \partial_k t = |F| \partial_k t
\]

and similarly

\[
\partial_k G'(t) = |G'| \partial_k t = |F| \partial_k t.
\]

We next have to show that the value of \(H(\tilde{t})\) is independent of the choice of an expression for \(\tilde{t}\): in fact, we show that \(H(\tilde{t}) = \tilde{t}\). By definition, \(H(G(t)) = G'(t)\).
Therefore, \( I'(H(G(t)) = I'(G'(t)) = I(G(t)) \), so that \( H(G(t)) = G(t) \). Therefore,
\[
H(e(G(t_1), \ldots, G(t_n))) = e'(G'(t_1), \ldots, G'(t_n)) = e(G(t_1), \ldots, G(t_n)) = i.
\]
Thus \( H \) is an inclusion. That \( H \) preserves identities, composition, and tupling, follows from this.

Finally, we show that \( H \) is surjective. Say \( t' \in \bar{T}' \). Then \( t' = e'(G'(t_1), \ldots, G'(t_n)) \), and if \( \bar{t} = e'(G(t_1), \ldots, G(t_n)) \), it follows that \( H(\bar{t}) = \bar{t}' \).

(8) easily follows from (7). First, note that any injection \( \langle t, I \rangle \) can be factored into an isomorphism followed by an inclusion \( \langle \bar{t}, \bar{I} \rangle \), because the graph image of an injection is the generous image, which is isomorphic to the source. Thus, we get the diagram

\[
\begin{array}{ccc}
\langle \psi, G \rangle & \equiv & \langle \bar{t}, \bar{I} \rangle \\
\bigcirc & & \bigcirc \\
\langle \psi', G' \rangle & \equiv & \langle \bar{t}', \bar{I}' \rangle \\
\end{array}
\]

Next, the composition of a generous morphism with an isomorphism is a generous morphism: so we get exactly the situation of (7), and therefore an identity \( \langle \bar{t}, \bar{H} \rangle \). But now, composing one isomorphism with the inverse of the other, gives the desired \( \langle \rho, H \rangle \), also an isomorphism. \( \Box \)

Now, the main result is the following theorem.

**Theorem 22.** In the category Theo of sorted algebraic theories, an image factorization system arises from letting \( E \) be the generous theory morphisms, and \( M \) the injective theory morphisms.

**Proof.** The only things not already proved are simple properties of injections. \( \Box \)

Our ‘generous’ morphisms are also the ‘\( G \)-generating’ morphisms of Herrlich and Strecker [19], with \( G \) the forgetful functor from theories to graphs. Thus, results in [19] can be applied to the present case, and would give fairly specific information about the generous image, but since this approach relies on \( G \) preserving limits (which follows from its being a right adjoint), we have followed a more concrete approach.

It is perhaps worth mentioning that a development very like the present one can be given for \( C \) the category \( \text{Cat} \) of categories. It is much simpler, because one doesn't have to deal with sorts or tupling; moreover, the results in [19] apply rather directly, because the forgetful functor from categories to graphs has a well-known left adjoint, the category of paths in a graph.
We conclude this section with a detailed discussion of an example. CLEAR as described in [13] requires us to factor a theory morphism arising from a derivor. Thus, we will also be illustrating material from Section 6. First, a presentation for the theory from which we start:

\[
\text{\textsc{natle}}
\]

\begin{verbatim}
\text{\textbf{sorts}}\quad\text{nat, bool}
\text{\textbf{opns}}\quad 0\quad \rightarrow\text{nat}
\quad \text{inc}\quad \rightarrow\text{nat\rightarrow nat}
\quad \leq\quad \text{nat, nat\rightarrow bool}
\quad \text{true, false}\quad \rightarrow\text{bool}
\quad \text{and}\quad \text{bool, bool\rightarrow bool}
\text{\textbf{eqns}}\quad 0\leq n\quad \rightarrow\text{true}
\quad \text{inc}(n)\leq 0\quad \rightarrow\text{false}
\quad \text{inc}(n)\leq \text{inc}(m)\quad n\leq m
\quad \text{true and p}\quad \rightarrow\text{p}
\quad \text{false and p}\quad \rightarrow\text{false}
\end{verbatim}

Next, the signature of the derived theory is given by

\[
\text{\textsc{even}}
\]

\begin{verbatim}
\text{\textbf{sorts}}\quad\text{even, bool}
\text{\textbf{opns}}\quad 0\quad \rightarrow\text{even}
\quad \text{succ}\quad \text{even\rightarrow even}
\quad \rightarrow\text{even, even\rightarrow bool}
\end{verbatim}

Let $\Sigma$ denote the signature of $\text{\textsc{natle}}$, and let $T: \Sigma \rightarrow T_{\text{\textsc{natle}}}$ be the quotient theory morphism.

We now give a derivor $D: \text{\textsc{even}} \rightarrow U(\Sigma)$, defined to be $(\varphi, g)$, where $\varphi(\text{even}) = \text{nat}$, $\varphi(\text{bool}) = \text{bool}$, and $g_{(\text{even}, 0)} = 0$, $g_{\text{even, even}(\text{succ})(n)} = \text{inc}(\text{inc}(n))$, $g_{\text{even, even, bool}(==)}(x, y) = (x \leq y \text{ and } y \leq x)$. The situation is summarized by the following diagram:

in which $(T', I)$ is a (generous, monic)-factorization of $D^* I$. The purpose of this construction is to obtain the signed theory $T'$, and Proposition 21(4) tells us what the subtheory $T'$ of $T_{\text{\textsc{natle}}}$ looks like: $T'(\text{even, \lambda})$ contains $0$, $\text{inc}(\text{inc}(0))$, $\text{inc}(\text{inc}(\text{inc}(\text{inc}(0)))))$, etc.; $T'(\text{bool, \lambda})$ contains only the (equivalence classes of) true, false (these are the images of the terms $(0 == 0)$, $\text{succ}(0) == 0$), respectively); $T'(\text{even, even})$ contains the identity function $\pi_{\text{in}}$ plus the even constant
functions 0, inc(inc(0)), etc.; \(T'(\text{bool, bool})\) contains the identity function \(x_0\) plus the constant functions true, false; \(T'(\text{bool, even, even})\) contains the constants true, false, plus the terms \((x_0 \leq x_1 \text{ and } x_1 \leq x_0)\), \((x_0 \leq x_0 \text{ and } x_0 \leq x_0)\), \((x_1 \leq x_0 \text{ and } x_0 \leq x_1)\), \((x_1 = x_1 \text{ and } x_1 = x_1)\). In general, \(T'(\text{even, even}^n)\) contains the even constants, plus the projections \(x_0, \ldots, x_{n-1}\); \(T'(\text{bool, even}^n)\) contains the constants true, false, plus the terms \((x_i = x_j \text{ and } x_j = x_i)\) for all \(0 \leq i, j \leq n - 1\).

It is interesting to notice that it is not the case that \((x_0 \leq x_0 \text{ and } x_0 \leq x_0)\) is in the same equivalence class (in \(T'\)) as \((x_0 = x_0)\), or as true. It is only after the inductive closure is taken that this happens (see the next section for this).

It is also interesting to notice that, if we chose to use the notation of \textit{even} rather than of \textit{nat} we could get a theory isomorphic to \(T'\), but not a subtheory of \(T_{\text{nat}}\), by taking morphisms to be terms such as \(\text{succ}(0)\), \(x_0 = x_1\), and so on. The theory \(T'\) described above is the \textit{unique} one given by (generous, injective)-factorization of \(D^*T\).

8. Algebras and induction

This section introduces models for our algebraic theories; following Lawvere [23] we call them 'algebras', and indeed, they correspond to universal algebras in the more usual sense, satisfying the equations of the theory. We are particularly concerned with two constructions: the 'most typical' algebra of a given theory (its 'initial' algebra) and the theory of a given algebra. The latter represents a kind of 'induction', of properties from a given structure. If \(T\) is a theory and \(T^i\) is its initial algebra, then the theory of \(T^i\) is a sort of 'inductive completion' of \(T\); it can be thought of as adding to \(T\) all the particular properties of its most typical model \(T^i\).

In the application to the semantics of \textsc{clear}, a theory \(T\) is often used as a description of \(T^i\), and it is then very desirable to have the complete description of \(T^i\). The theory of the initial algebra of \(T\) is denoted \textit{induce}(\(T\)), and our main objective is to precisely describe and exemplify the \textit{induce} construction. A particular complication we will have to take account of is that we are working not just with theories, but with signed theories.

For \(T\) an algebraic theory, Lawvere defined a \(T\)-algebra to be a coproduct-preserving functor \(A: T \rightarrow \text{Set}^{\text{op}}\), but this is somewhat unsatisfactory because \(A(s_1, s_2)\) is then only isomorphic to \(A(s_1) \times A(s_2)\), rather than equal to it. Lawvere's definition is of course satisfactory for the abstract theory of algebra; but for our applications it is more convenient to use the \textit{particular} Cartesian coproduct structure on \(\text{Set}^{\text{op}}\) and strict coproduct preserving functors, which then gives \(A(s_1, s_2) = A(s_1) \times A(s_2)\). From here, it is not a large step to defining a \textit{theory} \(\text{Tes}^3\) which embodies the coproduct structure of \(\text{Set}^{\text{op}}\) in a particularly convenient way, so that a \(T\)-algebra can be defined to be simply a theory morphism \(T \rightarrow \text{Tes}\).
We construct \textbf{Tes} as follows: For its sort set, take the class \(S = \mid \text{Set}\mid\) of all (small) sets, we then get the free theory \(T_{\mid \text{Set}\mid}\), as in Section 4. Next, we define the category \(\text{Tes}\) to have as its objects, strings \(W \in \mid \text{Set}\mid^*,\) and as its morphisms \(W \to V,\) set functions \(\pi(V) \to \pi(W),\) where if \(W = S_1 \ldots S_m,\) then \(\pi(W) = S_1 \times \cdots \times S_m.\) Note that this category \(\text{Tes}\) is equivalent to \(\mid \text{Set}\mid^\text{op},\) but not equal to it. Now, define a functor \(J: T_{\mid \text{Set}\mid} \to \text{Tes}\) by sending \(W \in \mid \text{Set}\mid^*\) to \(W \in \mid \text{Tes}\mid,\) and \(f: W \to V \in T_{\mid \text{Set}\mid}(W, V)\) to \(\pi(f): \pi(V) \to \pi(W),\) the unique map \(F\) to \(\pi(W)\) such that \(F \circ p_i^W = p_i^V,\) for \(1 \leq i \leq n,\) where \(p_i^W: \pi(W) \to S_i\) are the projections (for \(1 \leq i \leq n\)), and similarly for \(p_i^V.\) It is not difficult to verify that \(J: T_{\mid \text{Set}\mid} \to \text{Tes}\) is a \(\mid \text{Set}\mid\)-sorted theory, which we shall hereafter denote by just simply \(\text{Tes}.\)

We can now give the following definition.

**Definition 12.** Let \(T\) be a theory. Then a \(T\)-algebra is a morphism \(A: T \to \text{Tes}\) in \(\text{Theo}.\) As a special case, for \(\Sigma\) a signature, a \(\Sigma\)-algebra is a morphism \(A: T_\Sigma \to \text{Tes}\) in \(\text{Theo}.\) A signed algebra is a \(\Sigma\)-algebra, for some signature \(\Sigma.\) The theory of a signed algebra \(A: T_\Sigma \to \text{Tes}\) is the extremal epic part of its (extremal epic, monic)-factorization \(T_\Sigma \Rightarrow A \Rightarrow \text{Tes}\) denoted \(h(A): T_\Sigma \to A;\) notice that it is a signed theory.

If \(T: T_\Sigma \to T\) is a signed theory, then a \(T\)-algebra is simply a \(T\)-algebra \(A: T \to \text{Tes}.\) But notice that it can always be seen as the \(\Sigma\)-algebra \(TA: T_\Sigma \to \text{Tes}.\) The theory of a \(T\)-algebra \(A: T \to \text{Tes}\) is the theory of its \(\Sigma\)-algebra, and thus is of the form \(h(A): T_\Sigma \to A,\) a \(\Sigma\)-signed theory.

The interested reader can check that these definitions agree with the usual ones, in concrete cases such as the theory of groups.

Because algebras are theory morphisms, which are in particular functors, it would make sense to assume that homomorphisms are something analogous to natural transformations between theory morphisms. This indeed does work, as we shall see. Moreover, it is not necessary to assume the additional property of such natural transformations, that they too 'preserve coproducts', because that follows automatically, by the following lemma.

**Lemma 23.** Let \(A\) and \(B\) be categories with given (finite) coproduct structures, let \(F, G: A \to B\) be (finite) strict coproduct preserving, and let \(\eta: F \Rightarrow G\) be a natural transformation. Suppose that \(A = \prod_{i \in I} A_i\) is a (finite) coproduct. Then \(\eta_A = \prod_{i \in I} \eta_{A_i}: \prod_{i \in I} F(A_i) \to \prod_{i \in I} G(A_i).\)

**Proof.** Noting that \(F(A) = \prod_{i \in I} F(A_i)\) and that \(G(A) = \prod_{i \in I} G(A_i),\) for each \(j \in I\) we have commutativity of the diagram

\[\begin{array}{ccc}
F(A) & \xrightarrow{\eta_A} & G(A) \\
\downarrow & & \downarrow \\
\prod_{i \in I} F(A_i) & \xrightarrow{\prod_{i \in I} \eta_{A_i}} & \prod_{i \in I} G(A_i)
\end{array}\]

Because \(\text{Set}\) is not small, this involves extending \(\text{Sig}\) to include classes, as well as sets, and then suitably extending the functor \(1: \text{Sig} \to \text{Theo}.\) This poses no foundational difficulties, see, for example, [25, pp. 21-24].
in which the vertical arrows are the injections. But the unique such morphism $F(A) \to G(A)$ is the coproduct of the $\eta_{A_i}$. □

Now if $T$ and $T'$ are theories and $(\varphi, F), (\psi, G): T \to T'$ are theory morphisms, we define a morphism $(\varphi, F) \Rightarrow (\psi, G)$ to be a natural transformation $\eta: F \Rightarrow G$. Suppose that $T, T'$ are respectively $S, S'$ sorted, by functors $J: T_S \to T, J': T_{S'} \to T'$. Then by Lemma 23, $\eta$ is completely determined by what it does on $S$: or more precisely, by $\eta_{J(s)}: F(J(s)) \to G(J(s))$ for $s \in S$, which is $\eta_s: \varphi(s) \to \psi(s)$ when $J$ is the identity on objects.

This can be used to define homomorphisms as follows.

**Definition 12 (continued).** Let $T$ be a theory, and let $A, B: T \to \mathbb{F}es$ be $T$-algebras. Then a $T$-homomorphism $A \to B$ is a natural transformation $\eta: A \Rightarrow B$. Similarly, if $T: T_S \to T$ is a signed theory, and $A, B: T \to \mathbb{F}es$ are $T$-algebras, then a $T$-homomorphism $A \to B$ is a natural transformation $\eta: A \Rightarrow B$.

This gives us categories $\mathbf{Alg}_T$ and $\mathbf{Alg}_S$ of $T$-algebras and $T$-algebras. Actually, there is a great deal more we can say: some relevant properties are contained in the following abstract notions, which are also used in Section 10.

**Definition 13.** A 2-category $K$ has objects or 0-cells; arrows, morphisms, or 1-cells; and 2-cells. The objects and morphisms form an ordinary category, denoted $K_0$ (or possibly just $K$ again), called the underlying category of $K$. A given 2-cell $\alpha$ has both source and target objects, say $A$ and $B$, and source and target morphisms, say $f$ and $g$: it must be the case that $f$ and $g$ both have source $A$ and target $B$. The following diagram is helpful in visualizing this:

$$
\begin{array}{c}
\alpha \\
\downarrow \\
A \\
\bigcirc \\
\downarrow \quad \quad \downarrow \\
B \\
\end{array}
$$

For fixed objects $A, B$, the morphisms $A \to B$ as objects and the 2-cells between them form a category $K(A, B)$, with composition known as vertical composition, and with identities known as vertical identities. Given $\alpha: f \Rightarrow g$ and $\beta: g \Rightarrow h$ in $K(A, B)$, their
vertical composition is $\sigma \circ \beta : f \Rightarrow h$:

$$
\begin{array}{ccc}
A & \xrightarrow{g} & B \\
\downarrow^{\alpha} & \uparrow_{\beta} & \downarrow^{h} \\
\end{array}
$$

and the vertical identity on $f$ is $1_f : f \Rightarrow f$. There is also a horizontal composition of 2-cells, whereby from

$$
\begin{array}{ccc}
A & \xrightarrow{f} & B \\
\downarrow^{g} & \uparrow_{\alpha} & \downarrow^{h} \\
\end{array} \quad \text{and} \quad
\begin{array}{ccc}
B & \xrightarrow{u} & C \\
\downarrow^{v} & \uparrow_{\gamma} & \downarrow^{w} \\
\end{array}
$$

we get a 2-cell

$$
\begin{array}{ccc}
A & \xrightarrow{f} & B & \xrightarrow{u} & C \\
\downarrow^{g} & \uparrow_{\alpha} & \downarrow^{v} & \uparrow_{\gamma} & \downarrow^{w} \\
\end{array}
$$

Under this composition, the 2-cells are a category, with identity horizontal 2-cells of the form

$$
\begin{array}{ccc}
A & \xrightarrow{1_A} & A \\
\downarrow^{1_A} & \uparrow_{1_A} & \\
\end{array}
$$

Finally, we require, in the situation

$$
\begin{array}{ccc}
A & \xrightarrow{g} & B & \xrightarrow{v} & C \\
\downarrow^{h} & \uparrow_{\beta} & \downarrow^{\gamma} & \downarrow^{w} \\
\end{array}
$$

that the composites $(\alpha \circ \beta) \ast (\gamma \ast \delta)$ and $(\alpha \ast \gamma) \circ (\beta \ast \delta)$ shall be equal; and we require, in the situation

$$
\begin{array}{ccc}
A & \xrightarrow{1_f} & B & \xrightarrow{1_u} & C \\
\downarrow^{f} & \uparrow & \downarrow^{1_u} & \uparrow^{u} \\
\end{array}
$$

that $1_f \circ 1_u = 1_{fu}$.
As with ordinary categories, it is conventional and convenient on occasion, to denote an identity morphism or 2-cell, by its object or morphism: thus $A$ for $1_A$, $f$ for $1_f$, and even $A$ for $1_A$. It may also sometimes be convenient to write composition, whether vertical or horizontal (but more often horizontal) by just juxtaposition. For example, we may write the horizontal composite of

$$
\begin{array}{ccc}
A & \xrightarrow{f} & B \\
\downarrow{} & & \downarrow{y} \\
\downarrow{} & & \downarrow{v} \\
\end{array}
\quad
\begin{array}{ccc}
B & \xrightarrow{u} & C \\
\downarrow{} & & \downarrow{} \\
\downarrow{} & & \downarrow{} \\
\end{array}
\quad
\begin{array}{ccc}
C & \xrightarrow{g} & D \\
\downarrow{} & & \downarrow{} \\
\downarrow{} & & \downarrow{} \\
\end{array}

as simply $fuyg$.

The prototypical example of a 2-category is the category $\mathbf{Cat}$ of all (small) categories: its objects are categories; its morphisms are functors; and its 2-cells are natural transformations. See [30] or [22] for further information on 2-categories. We now have the following theorem.

**Theorem 24.** The category of theories, with morphisms as in Definition 12, and with 2-cells taken to be natural transformations, is a 2-category.

Moreover, once it has been verified that algebras and homomorphisms as we have defined them correspond to their usual meanings, the following is well known [3, 4, 5].

**Theorem 25.** For each theory $\mathbb{T}$, there is an initial $\mathbb{T}$-algebra $\mathbb{T}^i$, characterized (up to isomorphism) by the property that for any other $\mathbb{T}$-algebra $A$, there is a unique $\mathbb{T}$-homomorphism $\mathbb{T}^i \to A$.

From this, easily obtain the following.

**Proposition 26.** If $T: \mathbb{T}_\Sigma \to \mathbb{T}$ is a signed theory, then there is an initial $T$-algebra, $T^i$, characterized (up to isomorphism) by the property that for any other $T$-algebra $A$, there is a unique $T$-homomorphism $T^i \to A$. It follows that $T^i = T \mathbb{T}^i$.

We are now ready for the following definition.

**Definition 14.** Let $T: \mathbb{T}_\Sigma \to \mathbb{T}$ be a signed theory; let $T^i = TT^i$ be its signed initial algebra. Then $\text{induce}(T) = h(T^i)$ is the inductive closure of $T$; it is an extremal epimorphism in $\mathbf{Theo}$ with source $\mathbb{T}_\Sigma$ and is thus a signed theory.

If $T: \mathbb{T}_\Sigma \to \mathbb{T}$ is an extremal epimorphism—as is often the case—then there is an epimorphism $N_T: \mathbb{T} \to A$ arising from the factorization property for $h(T^i)$ (see (1).
of Proposition 18),

\[
\text{NatLE}
\]

\begin{align*}
\textbf{sorts} & \quad \text{nat, bool} \\
\textbf{opns} & \quad 0 : \to \text{nat} \\
& \quad \text{inc} : \text{nat} \to \text{nat} \\
& \quad < : \text{nat}, \text{nat} \to \text{bool} \\
& \quad \text{true} : \to \text{bool} \\
& \quad \text{false} : \to \text{bool} \\
& \quad \text{and} : \text{bool}, \text{bool} \to \text{bool} \\
\textbf{eqns} & \quad 0 \leq n \quad = \text{true} \\
& \quad \text{inc}(n) \leq 0 \quad = \text{false} \\
& \quad \text{inc}(n \leq \text{inc}(m)) = n \leq m \\
& \quad \text{true and } p \quad = p \\
& \quad \text{false and } p \quad = \text{false}
\end{align*}

For the remainder of this section, let \( \Sigma \) denote the signature of \text{NatLE}; let \( \mathcal{T} \) be \( \mathcal{T}_{\text{NatLE}} \), which is \( \mathcal{T}_\Sigma / Q \), where \( Q \) is the theory congruence generated in \( \mathcal{T}_\Sigma \) by the equations in \text{NatLE}; and let \( T : \mathcal{T}_\Sigma \to \mathcal{T} \) be the quotient morphism, noting that it is an extremal epimorphism (we can use (2) of Proposition 21). Then \( \mathcal{T}' \) is the initial algebra of \( \mathcal{T} \) and we let \( \mathcal{A} \) denote \( \mathcal{T}' \), the corresponding signed algebra: it is, as expected, (up to \( \Sigma \)-isomorphism) the nonnegative integers with booleans and the less-than-or-equal relation \( \leq \). There are quite a number of equations which are true of \( \mathcal{A} \), but are not in \( T \) (that is, to be precise, they do not appear as pairs in \( Q \)). For example, \( "x_0 \leq x_0" \) is not in the same equivalence class as \"true\" in \( T \); and \( "x_0 \text{ and } x_0" \) is not in the same equivalence class as \"x_0\". (Note that the symbol \"x_0\" is actually being used to denote two different variables, one of sort nat and one of sort bool. This is in accord with the notational conventions of Section 4, because the sorts can be recovered from the context.) However, both the equations

\[
x_0 \leq x_0 = \text{true}, \quad x_0 \text{ and } x_0 = x_0
\]

are satisfied in the initial algebra (this means that the left- and right-hand sides evaluate to the same truth value for any choice of a value for \( x_0 \) (of the right sort, of course). These equations therefore appear in the theory \( h(\mathcal{A}) = \text{induce}(T) \), and
they represent distinct arrows of $T$ which are mapped to the same arrow in $A$ by the functor $N_F$. The above equations are therefore ‘inductively true’ of the initial algebra $A$, but not (equationally) deductively true of the original theory $T$.

Let us look a little more carefully at the initial algebra $A$ of $T$. Its carriers are determined as follows: $A_{\text{nat}}$ is the set $T(\text{nat}, \lambda)$ and consists of the terms $0$, $\text{inc}(0)$, $\text{inc}(\text{inc}(0))$, etc.; $A_{\text{bool}}$ is the set $T(\text{bool}, \lambda)$, and consists of (the equivalence classes of) the terms true and false (for example, the class of true also contains true and true, true and (true and true)). Viewed as a theory morphism $T \to \text{Tes}$, we have for example, $A(x, x_0): A_{\text{nat}} \times A_{\text{nat}} \to A_{\text{bool}}$, which turns out to be the constant function with value true. Similarly, $A(\text{and}): A_{\text{bool}} \times A_{\text{bool}} \to A_{\text{bool}}$ is the usual boolean conjunction. These assertions permit verification of those made above about the two equations.

9. Applying theory-valued procedures

The specification language CLEAR has features for defining theory-valued procedures, and for applying them to theory actuals to get new theories. For example, the text

```
proc Table(Value: Triv*) =
  induce enrich Nat + Value by
  sorts table
  opns nilt : → table
    put : value, nat, table → table
    [ ] : table, nat → value
  eqns nilt[n] = *
    put(v, n, put(v', n', t)) = if n = n' then put(v, n, t) else clsc
    put(v', n', put(v, n, t)) fi
    put(v, n, t)[n'] = if n = n' then v else t[n'] fi
  enden end
```

defines a theory-valued procedure which, given a theory $V$ (containing a sort $v$) satisfying Triv* (we will define Triv* and ‘satisfaction’ a little later), produces a theory $\text{Table}(V)$ of tables of $v$'s indexed by natural numbers with operations for updating and reading values of sort $v$, nilt denotes the initial table, all of whose values are $*$, a constant of sort $v$ which must be supplied in $V$: $\ldots[\ldots]$ is a ‘mix-fix’ operator for reading the table; induce is as described in Section 8, $+$ denotes combine, and enrich . . . by . . . enden is as described at the end of Section 5.

The primary purpose of this section is to describe what it means to apply a theory-valued procedure like $\text{Table}$ to a theory actual, let us say $V$, to get a result theory, in this case $\text{Table}(V)$, tables of $v$'s. This kind of procedure application is perhaps the most characteristic and significant feature of CLEAR, and like so much
else discussed in this paper, it depends on colimits of theories. It is not the purpose of this section to give a precise meaning for every feature of the CLEAR text above; a semantics for CLEAR is given in [10].

First, let us define Triv*. It is the theory given by the presentation

\[
\begin{align*}
\text{sorts} & \quad \text{triv} \\
\text{opns} & \quad \ast : \to \text{triv}
\end{align*}
\]

which in CLEAR would be denoted by

\[
\begin{align*}
\text{const Triv*} = \\
\text{sorts triv} \\
\text{opns} & \quad \ast : \to \text{triv}
\end{align*}
\]

This is the theory of 'pointed sets'; that is, a Triv*-algebra \(A\) is (in its set-theoretic, rather than its functorial form) a set \(A_{triv}\) and an element \(*\in A_{triv}\) (i.e., a 'point' in \(A_{triv}\)).

We will use the following terminology.

**Definition 15.** Let \(T: T_2 \to T\) be a signed theory with \(T\) surjective, and let \(V\) be a theory. We say that \(V\) satisfies \(T\) iff there is a theory morphism \(F: T_2 \to V\) which factors through \(T\), i.e., such that there is a theory morphism \(F'\) such that

\[
\begin{align*}
T_2 & \xrightarrow{T} T \\
F & \xrightarrow{F'} V
\end{align*}
\]

commutes in Theo; when such an \(F'\) exists, \(F\) is called a fitting morphism for \(V\) to \(T\). (Because \(T\) is surjective, \(F'\) is unique if it exists.)

In our applications, \(V\) generally has a signature, i.e., there is a given theory morphism \(V: T_{it} \to V\); but we do not wish to export this information in the result of the procedure application. Consequently, we ignore the signatures of actuals, and colimits are taken in the category of theories, rather than signed theories.

We call Triv* the metasort of the procedure Table; it defines the conditions which a theory actual must satisfy, and an expression like "\(V::\text{Triv*}\)" is analogous to a sort declaration like "\(b::\text{bool}\)".

Notice that we are not allowed to write Table(Nat) because this contains no indication of which constant of sort nat should be chosen for *. But if we define a

This approach was prefigured in our original CLEAR paper [8] and spelled out in detail in [10]. It also appears in [13, 14], among other places.
new theory $\text{Nat}^*$ by
\[
\begin{align*}
\text{const } \text{Nat}^* &= \\
enrich \text{Nat} \text{ by} \\
opns*: \rightarrow \text{nat} \\
eqns* = 0 \text{ enden}
\end{align*}
\]
then it is obvious what Table($\text{Nat}^*$) should mean. However, strictly speaking, a fitting morphism should be provided. Fitting morphisms can be defined by derivors. In the case at hand, we could either define a theory morphism $D: \text{Triv}^* \rightarrow \text{Nat}$ by $D(*) = 0$; or we could define $D^*: \text{Triv}^* \rightarrow \text{Nat}^*$ with $D^*(*) = *$ (the sort mapping of these derivors is of course $\text{triv} \rightarrow \text{nat}$).

Let us now consider the general case. The ‘body’ of the procedure is a surjective signed theory $B: \Sigma \rightarrow B$, and there are $n$ parameter metasorts, $M_i: \Sigma \rightarrow T_i$ for $i = 1, \ldots, n$, each surjective, where each $M_i$ is a subtheory of $B$, i.e., $\Sigma_i \subseteq \Sigma$, $T_i \subseteq T$, and the diagrams

\[
\begin{array}{ccc}
\Sigma & \xrightarrow{B} & B \\
| & & | \\
\Sigma & \xrightarrow{M_i} & T_i
\end{array}
\]
all commute in Theo: we shall let $K_i: M_i \rightarrow B$ denote the inclusions in SigTheo, or (ambiguously) $K_i: T_i \rightarrow B$. Let us further assume, for the sake of generality, that there are a number of signed theories, $D_1, \ldots, D_m$, with targets $D_1, \ldots, D_m$, which are shared in various ways among the $M_i$ (for example, it is quite possible that the theory of boolean truthvalues is a subtheory of $B$ and several of the $M_i$); let \{$(K_i: D_i \rightarrow M_i, (j, i) \in I)$\} be a set of subtheory inclusions, for $I \subseteq \{1, \ldots, m\} \times \{1, \ldots, n\}$ an index set, such that if $(j, i)$ and $(j, i')$ are both in $I$, then the diagram

\[
\begin{array}{ccc}
K_i & \xrightarrow{B} & K_i \\
| & & | \\
M_i & \xleftarrow{D_i} & M_i
\end{array}
\]
of inclusions commutes in SigTheo.

Now let $A_i$ for $i = 1, \ldots, n$ be a collection of ‘actuals’ to be ‘substituted’ into the procedure body $B$ for $T_i$, and assume that $A_i$ satisfies $M_i$ by given fitting morphisms $F_i: \Sigma \rightarrow A_i$; further, assume that there are given inclusions \{$(K_i: D_i \rightarrow A_i, (j, i) \in I')$\} such that $I \subseteq I'$ and such that if $(j, i) \in I$, then $F_i$ ‘preserves’ $D_i$ in the sense that the
Finally, to account for the possibility that some $D_j$ may occur in $B$ and in some $A_i$, even though it does not occur in any $M_n$ we shall assume there are inclusions $\{K''_j : D_j \rightarrow B | j \in I'' \}$ such that if $j \in I''$ and $(j, i) \in I'$, then the diagrams commute if $(j, i) \in I$.

Thus, we have altogether a commutative diagram, let us call it $\mathcal{D}$,

in which the upward arrows are inclusions and the downward arrows are fitting morphisms. This diagram expresses the environment in which procedure application is to take place, including the shared subtheories which must be kept track of (Burstall and Goguen [10] handle shared subtheory environments in a more elegant way using ‘based theories’; but that would only distract from the main point in the present context).

The result of the procedure application in the above situation is simply the colimit $C$ of the diagram $\mathcal{D}$; moreover, the injection of the theories $D_j$ into $C$ provide the necessary information about shared subtheories in case $C$ is later combined with other theories.
For the example discussed at the beginning of this section, the diagram \( \mathcal{D} \) looks as follows:

\[
\begin{array}{c}
\text{Table} \\
\mid \downarrow \\
\text{Triv}^* \\
\mid \downarrow \\
F' \\
\mid \downarrow \\
\text{Nat} \\
\mid \downarrow \\
\text{Nat}
\end{array}
\]

and the theory which results from applying \textbf{Table} to \textbf{Nat} via \( F' \) is \textbf{induce} of the theory* with the following signature:

- **sorts** bool, nat, table
- **opns**
  - \texttt{true} : \( \to \) bool
  - \texttt{false} : \( \to \) bool
  - \texttt{and} : bool, bool \( \to \) bool
  - \texttt{0} : \( \to \) nat
  - \texttt{inc} : nat \( \to \) nat
  - \texttt{=} : nat, nat \( \to \) bool
  - if\_then\_else\_fi : bool, nat, nat \( \to \) nat
  - \texttt{nilt} : \( \to \) table
  - \texttt{put} : nat, nat, table \( \to \) table
  - \texttt{[]} : table, nat \( \to \) nat

where the theory \textbf{Nat} contains equality, conditional, and booleans—the dots indicate additional operations which may be present—and we have omitted all equations (note that 0 must be substituted for * in the \textbf{Table} equations).

Let us turn again to the general case. The following shows how to calculate the result of the application of a theory-valued procedure to theory actuals.

**Theorem 27.** Let \( B: T \to B, M_i: T \to A, K_i: M_i \to B, \text{ etc. as above; let } C \text{ be the colimit of the diagram } \mathcal{D} \text{ above; let } P = (S, \Sigma, \mathcal{E}) \text{ and } (S', \Sigma', \mathcal{E}') \text{ be presentations for } B \text{ and } A, \text{ respectively, with } F_i: T \to A, \text{ given by a derivor } d_i: \Sigma_i \to U(T_{\iota_i}); \text{ assume that } S_i \subseteq S, \Sigma_i \subseteq \Sigma, \text{ and that there are also given corresponding subpresentations } Q_i = (S_i', \Sigma_i', \mathcal{E}_{i}'') \text{ of } P \text{ for the } D_i. \text{ Then a presentation for } C \text{ is given}

* One subtlety we have to mention in treating this example, is that the initial \textbf{induce} in the body of \textbf{Table} is only applied after the actuals have been substituted for the parameters, not before.
by \( P = (S, \Sigma, \bar{S}) \), obtained as follows:

1. By renaming some sorts if necessary, we may assume that the following conditions are satisfied by \( S \) and the \( S_i \):
   (a) For \( s \in S \) and \( s_0 \in S_n, s \neq s_0 \) unless there are some \( j \) and \( s' \in S'_j \) such that
       \[ K'_j(s') = s \] and \( K''_j(s') = s_0. \]
   (b) For \( s \in S_i \) and \( s' \in S'_i, s \neq s' \) unless there are some \( j \) and \( s_0 \in S'_j \) such that
       \[ K'_j(s_0) = s \] and \( K''_j(s_0) = s'. \]

2. Let \( \bar{S} = S \cup \bigcup_i S_i \); let \( \bar{S} \) be \( \bar{S} \) with \( d_i(s) \) substituted for \( s \) wherever \( d_i(s) \) is defined (i.e., for each sort \( s \) in \( T_i \)).

3. By renaming some operations if necessary, we may assume that the following conditions are satisfied by \( \Sigma \) and the \( \Sigma_i \) (with their indices ranging over \( \bar{S} \)):
   (a) for \( \sigma \in \Sigma_{w,s} \) and \( \omega \in \Omega_{w,s}, \sigma \neq \omega \) unless there are some \( j \) and \( \sigma' \in \Sigma_{w,s} \) such that
       \[ K'_j(\sigma') = \omega. \]
   (b) for \( \omega = \xi_{w,s} \) and \( \omega' \in \Omega_{w,s}, \omega \neq \omega' \) unless there are some \( j \) and \( \sigma \in \Sigma_{w,s} \) such that
       \[ K'_j(\sigma) = \omega. \]

4. Let \( \bar{\Sigma} = (\Sigma \cup \bigcup_i \Omega_i) - \bigcup_i \Sigma_i. \)
5. Let \( \bar{\xi} \) be \( \xi \cup \bigcup_i \xi_i \), with \( d_i(\sigma) \) substituted for each occurrence of \( \sigma \) in each equation, for each \( \sigma \) in \( \Sigma_i \), for each \( i \).

Moreover, the injections \( A_i \to C \) and \( D_i \to C \) are inclusions.

Proof. We omit details, but the result follows from: the general result about calculating colimits in comma categories by calculating colimits of each component separately (Proposition 2 and Corollary 4); the comma category form of \( \text{Pres} \) (see the discussion after Theorem 9); and the relationship between theories and presentations (Theorems 10 and 10\text{th}). \( \square \)

Notice that this result actually gives an algorithm for computing \( C \).

The signature of the result of a theory-valued procedure application is defined to be the signature \( \Sigma \) of its body, with the signed theory being the theory morphism \( T_\Sigma \to C \) which is the composite \( T_\Sigma \xrightarrow{b} B \xrightarrow{b} C \) where \( b \) is the coproduct injection. In general, this will result in signed theories having hidden sorts and operators, those in the \( A_i \), not named by \( \Sigma_i \).

The reader may wish to check that all this is consistent with the example given in this section.

10. Abstract theories

It appears that many different kinds of theory are relevant to computation, not just the many-sorted equational (i.e., Lawvere) theories discussed previously in this paper. For example, the error theories of Goguen [16], the Horn theories of Keane [21], the continuous and rational algebraic theories of ADJ [2], the iterative theories of Elgot [15], and various combinations and elaborations of these. It is far from
clear that there is, or ever will be, some notion of theory which is the best for studying computation. However, whatever notion of theory one is using, one would certainly like to have as many nice properties as possible, and in particular, the kind of properties proved in this paper for many-sorted Lawvere theories. It would indeed be tedious to have to do this over and over again for each particular kind of theory which might arise: one would like to have a general theory of theories which would make available at least the results which are most important for specification. The purpose of this section is to sketch one way that this might be done.

Let us begin by summarizing the steps we have already taken:

1. generalize the notion of ‘theory’ to an arbitrary category with (finite) coproducts: thus, the objects need no longer be strings of sort names, and indeed, we can regard the opposite category of (small) sets as a theory $\text{Tes}$ with a particular coproduct structure;
2. the ‘algebras’ of a theory, $\mathcal{T}$ are now the theory morphisms (i.e., coproduct preserving functors) from $\mathcal{T}$ to the theory $\text{Tes}$;
3. the ‘homomorphisms’ between algebras $A: \mathcal{T} \to \text{Tes}$ and $A': \mathcal{T} \to \text{Tes}$ are the natural transformations between the functors $A$ and $A'$.

The generalization can be stated in terms of the four axioms already given in Section 1 of Part 1. However, they now carry more force because all the terms used in the have been precisely defined, and also exemplified in the case of many-sorted Lawvere theories.

**Assumption 1.** There is a ‘forgetful’ functor $U: \text{Th} \to \mathcal{S}$ with a left adjoint, $F: \mathcal{S} \to \text{Th}$, which is injective.

**Assumption 2.** Both the categories $\mathcal{S}$ of signatures and $\text{Th}$ of theories have finite colimits.

**Assumption 3.** Every morphism $f$ in $\text{Th}$ has a factorization $f = em$, where $e$ is an extremal epimorphism and $m$ is a monomorphism.

**Assumption 4.** The category $\text{Th}$ of theories and theory morphisms is the underlying category of a 2-category, also denoted $\text{Th}$. There is a distinguished ‘ground’ object $G$ in $\text{Th}$ such that for each object $\mathcal{T}$ in $\text{Th}$, the category $\text{Th}(\mathcal{T}, G)$ has an initial object, denoted $\mathcal{T}'$. The 2-cells in $\text{Th}(\mathcal{T}, G)$ are called $\mathcal{T}$-homomorphisms, and this category is also denoted $\text{Alg}_1$.

Much of what has gone before can be done in this setting. For example, the very basic notions of algebra, and theory of an algebra can be generalized directly from Definition 12, as follows.

**Definition 16.** Let $\mathcal{T} \in |\text{Th}|$ be a theory. Then a $\mathcal{T}$-algebra is a morphism $A: \mathcal{T} \to G$ in $\text{Th}$. As a special case, for $\Sigma \in |\mathcal{S}|$ a signature, a $\Sigma$-algebra is a morphism $A: \mathcal{T}_\Sigma \to G$ in $\text{Th}$. A signed algebra is a $\Sigma$-algebra, for some signature $\Sigma$. The theory of a signed
algebra $A: T \to G$ is the extremal epic part of its (extremal epic, monic)-factorization $T \to A \to G$, denoted $h(A): T \to A$; notice that it is a signed theory.

If $T: T \to T'$ is a signed theory, then a $T$-algebra is simply a $T$-algebra $A: T \to G$ (notice that it can always be seen as the $\Sigma$-algebra $TA: T \to G$). The theory of a $T$-algebra $A: T \to G$ is the theory of its $\Sigma$-algebra, and thus is of the form $h(A): T \to A$, a $\Sigma$ signed theory.

**Assumption 4.** The category $\text{Th}$ of theories and theory morphisms is the underlying category of a 2-category, also denoted $\text{Th}$. There is a distinguished object $G$ in $\text{Th}$ such that for each object $T$ in $\text{Th}$, the category $\text{Th}(T, G)$ has an initial object, denoted $T^i$. The 1-cells and 2-cells in $\text{Th}(T, G)$ are called $T$-algebras and $T$-homomorphisms, respectively, and this category is also denoted $\text{Alg}_T$.

Given a signed theory $T: T \to T'$ by Assumption 4', there is an initial $T$-algebra, $T^i: T \to G$. Let $T^i: T \to G$ be the composite $TT^i$; it is the signed algebra corresponding to $T^i$.

We now extend $^i$ to be a functor, from the category $\text{STh}$ of signed theories, to the category of signed algebras.

**Definition 17.** Let $A: T \to G$ and $A': T \to G$ be signed algebras. Then a morphism $A \to A'$ of signed algebras is a pair $(\varphi, \eta)$, where $\varphi: \Sigma \to \Sigma'$ and $\eta: A \Rightarrow T^iA'$. Let $\text{Salg}$ denote the category of signed algebras, with composition

$$
(\varphi, \eta)(\varphi', \eta') = (\varphi \varphi', \eta \circ (T^i \star \eta')) \quad \text{for} \quad (\varphi', \eta'): A' \to A''
$$

and identity $A \to A$ given by $(1_A, 1_A)$.

**Proposition 28.** $\text{Salg}$ as defined above is in fact a category. Moreover, we obtain a functor $^i: \text{STh} \to \text{Salg}$ by extending $^i$ from signed theories to their morphisms, as follows: let $T: T \to T$ and $T': T \to T'$ be signed theories, and let $(\varphi, F)$ be a morphism $T \to T'$; then $F T^i: T \to G$ is a $T$-algebra, so there is a unique 2-cell $\eta: T^i \Rightarrow FT^i$; define $(\varphi, F)^i$ to be $(\varphi, T \star \eta)$, a signed theory morphism $TT^i \to TT'$.

We omit the (straightforward) proof; the following diagram may help visualize the definitions involved:
We are now ready to generalize Definition 12.

**Definition 18.** Let $T: \mathbb{T}_\Sigma \rightarrow \mathbb{T}$ be a signed theory; let $T' = T^{\uparrow}$ be its signed initial algebra. Then $\text{induce}(T) = h(T')$ is the inductive closure of $T$; it is an extremal epimorphism in $\mathbb{Th}$ with source $\mathbb{T}_\Sigma$ and thus a signed theory.

Other remarks in Section 8 also generalize, such as the existence of a signed theory epimorphism $N_f: T \rightarrow h(T')$ when $T: \mathbb{T}_\Sigma \rightarrow \mathbb{T}$ is an epimorphism. In fact, all of the clear constructions described in this paper can be carried out in the general framework of the four axioms given above, and this suffices for the semantics of the original Burstall and Goguen [8] version of the language.

We have more recently developed other ways to approach the problems discussed above. Burstall and Goguen [10] suggest using 'institutions', and also abandon induce in favour of 'data constraints'. Burstall and Rydeheard [12] generalize theories in the form of so-called monads (also called triples; see [25]) in a way which can handle signatures and induce. This generalization is able to encompass continuous theories, and also the order sorted theories of Goguen [17], but unfortunately not the higher-order theories of Parsaye-Ghomi [31]. It appears that much work remains to be done in this area.

11. Concluding remarks

This paper might seem to raise as many questions as it answers. Three areas for further research that we regard as particularly important, are the following:

(1) The previous section mentioned many different kinds of theory, but the results of this paper suffice to establish the basic assumptions mentioned for only two relatively simple cases, many-sorted Lawvere theories having as their abstract signatures either signatures with co-arity, or else presentations. Similar results for other choices of theory and signature are very much to be desired. For example, Meseguer [28] has shown that categories of continuous theories are cocomplete; he also has studied their factorization situations.

(2) Our treatment of 'initial algebra induction' is very incomplete. Explicit syntactic proof rules are needed, in order that this can be implemented; for example, are
versions of the usual structural induction principle valid? How can presentations be used efficiently? Some related questions are explored in [18, 29]. The latest version of \textsc{clear} uses ‘data theories’ and similar questions should be raised in this connection.

(3) It is also desirable to explore the general 2-category framework of Section 10 further, and in particular, to relate it explicitly to more familiar approaches. We need to know what further assumptions are needed to do induction, and whether they are satisfied for appropriate special kinds of theories. (Perhaps, for example, some kind of Yoneda property for $G$.)

There are also some more specific questions:

(4) For a fixed sort set $S$, the category $\text{Theo}_S$ of $S$-sorted algebraic theories is actually the category of algebras of a theory, the theory of $S$-sorted theories! This gives a great deal of information about $\text{Theo}_S$, in particular, its colimits and factorizations. Unfortunately, there does not seem to be any way to do this trick with the category $\text{Theo}$ of all sorted algebraic theories. However, the relationship between $\text{Theo}$ and all the $\text{Theo}_S$'s is an instance of a general ‘indexed category’ construction, we believe that this fact can be exploited. (Notice that $\text{Sig}$ and $\text{ISet}$ are also indexed categories.)

(5) Are the functors $h$ and $\tau$ of Section 8 related by an adjoint situation?

(6) We believe that a kind of $\text{Lisp}$ extension can be gotten as a special case of \textsc{clear}, with objects in $\text{Th}$ being sets of equations used to describe $S$-expressions which quite possibly involve circularity, and sharing of subexpressions (see [24] and [7] for some relevant ideas). The extensions to $\text{Lisp}$ which result include adding a kind of type checking (e.g., for the parameters of functions) for structure which properly takes account of sharing, and the gathering together of sets of functions into groups, as in \textsc{simula} classes.

References


S. MacLane, Categories for the Working Mathematician, Graduate Texts in Mathematics (Springer, Berlin, 1971).


