Nested Sketches
(Preliminary Version)

Horst Reichel*
TU Dresden

1 Introduction

Since the fundamental work of Lawvere in 1963 [7] it is common to understand a theory as category with additional structure, to understand a model of the theory as a functor preserving the additional structure, and to represent homomorphisms by natural transformations. The resulting model category becomes a suitable subcategory of a functor category. Many different classes of mathematical structures have been described and investigated in this way. The aim of this paper is, to find a functorial model theory for those classes of algebras that appear naturally as semantics of algebraic specifications of parameterized data types, using initial respectively more general free functor semantics, and to extend the functorial model theory to specifications that use as well *inductively defined data types* as *coinductively defined patterns of behavior* and their systematic combinations.

The final result is a categorical model theory of discrete mathematical structures whose *basic operations may have arbitrarily structured domains and codomains.* Such kind of structures have been first systematically investigated by T. Hagino in his PhD thesis , [5]. The basic idea to achieve this generalized categorical model theory is the use of combined left and right Kan extensions in order to constrain iteratively functor categories.

The resulting categorical model theory generalizes algebraic and essentially algebric theories, since algebras are structures whose basic operations have a structured domain, being a product or finite limit, and the codoamin is one of the basic types (usually called sorts in algebra). The approache also generalizes coalgebras, for which dually the codomains of the basic operations are structured and the domain is one of the basic types.

As we will the, we are also able to represent model categories that have not jet

*This work has been partially supported by EPSRC grant GR/M36694

been described by different kinds of sketches. To justify the last statement we consider the description of lists and trees as described in the electronic supplement of Barr and Wells book [2].

In the Section 1.1.2 *The sketch for lists* there is used a finite discrete sketch with objects $L, D, L^+, 1$ and morphisms $head : L^+ \to D, tail : L^+ \to L$ an two other unnamed ones that are used to express by a sum sketch that $L^+ = 1 + L^+$. Additional product sketches imply that 1 has to be mapped by a model M to a final object of a category, i.e., to a singleton set in the category of sets, and that for a model $M(L^+) = M(D) \times M(L)$ holds.

It is easy to check that the intended interpretation:

$$M(L) = set\ of\ all\ finite\ lists\ of\ elements\ in\ M(D)$$

with $M(L^+) = set\ of\ all\ finite\ nonempty\ lists\ in\ M(D)$ is a model.

But the intended interpretation is not the only model. The discrete sketch does only imply that for each model M the set $M(L)$ is a fixed point (up to isomorphism) of the recursive type equation

$$M(L) \equiv 1 + (M(D) \times M(L).$$

The intended interpretation is given by the least fixed point, and there are several other ones not isomorphic to the intended model. Thus, the used formalization by discrete sketches is not able to constrain the functor category to the intended class of models.

In the next section be will introduce the notion of nested sketches and we will show, that this kind of constraining functor categories can represent exactly the intended class of functors, such that up to isomorphisms, a functor

$$M : \mathbb{C} \to Set$$

is a model of a corresponding set of nested sketches, if and only if $M(L) = M(D)^*$, where $M(D)^*$ denotes the set of finite list with elements in $M(D)$.

2 Nested Sketches

In the following we assume that $\mathbb{C}$ denotes a finitely generated category, represented by a finite directed graph $G(\mathbb{C})$ and a finite set $Rel(\mathbb{C})$ of defining relations, where a defining relation $\langle w_1, w_2 \rangle \in Rel(\mathbb{C})$ is an ordered pair of finite paths over $G(\mathbb{C})$ with the same beginning and end node.

If $F : \mathbb{A} \to \mathbb{B}$ and $G : \mathbb{B} \to \mathbb{C}$ are functors, there composition will be denoted by $F; K : \mathbb{A} \to \mathbb{C}$.

Definition 2.1. A *projective sketch* (respectively *injective sketch*) in a category $\mathbb{C}$ is given by three functors

$$J : \mathbb{C}_0 \to \mathbb{C}_1,\ K : \mathbb{C}_1 \to \mathbb{C},\ F : \mathbb{C}_0 \to \mathbb{C}$$

and a natural transformation

$$\pi : J; K \Rightarrow F$$

respectively by a natural transformation

$$\eta : F \Rightarrow J; K.$$

A functor $M : \mathbb{C} \to Sem$ is a *model* of a projective sketch if

$$(K; M : \mathbb{C}_1 \to Sem,\ (\pi; M) : J; K; M \Rightarrow F; M)$$

is a right Kan extension of $F; M : \mathbb{C}_0 \to Sem$ along $J : \mathbb{C}_0 \to \mathbb{C}_1$.

Correspondingly $M : \mathbb{C} \to Sem$ is a *model* of an injective sketch if

$$(K; M : \mathbb{C}_1 \to Sem,\ (\eta; M) : F; M \Rightarrow J; K; M)$$

is a left Kan extension of of $F; M : \mathbb{C}_0 \to Sem$ along $J : \mathbb{C}_0 \to \mathbb{C}_1$.

A *sketch* in $\mathbb{C}$ is either an injective or a projective sketch in $\mathbb{C}$. A functor $M : \mathbb{C} \to Sem$ is a model of a finite set of sketches if it is a model of each of sketch in the given set. □

Sketches as defined here do not improve the expressiveness with respect to the kind of sketches used by Barr and Wells [2]. They have been considered by Ross Street in the 60th, according to a personal communication, but have not been published. One advantage of the introduced sketches is the possibility to represent countable limit and colimit sketches in a finite manner. This can be illustrated by a specification of natural numbers.

Example 1: We take the category $\mathbb{C}$ which is defined by the generating graph $G(\mathbb{C})$

$$B \xrightarrow{z} N \xrightarrow{s} N$$

and with the empty set of defining relations. We take $\mathbb{C}_0 = \mathbf{1}$, i.e., the category with exactly one object 1, $\mathbb{C}_1 = \mathbb{C}$, $K = Id_{\mathbb{C}}$, $J(1) = B, F(1) = B$ and the natural transformation $\eta : F \Rightarrow J; K$ is the identity of $F (= J; K)$. This defines an injective sketch in $\mathbb{C}$.

One can easily check that a functor $M : \mathbb{C} \to Set$ is a model of that injective sketch, if (up to isomorphisms)

$$M(N) = I\!N \times M(B) = \sum_{i \in I\!N} M(B)_i$$

Proving that one has to show that for each functor $X : \mathbb{C}_1 \to Set$ and each natural transformation $\alpha : F \Rightarrow J; X$ there is exactly one natural transformation $\alpha^* : K; M \Rightarrow X$ with $\alpha = (\eta; M) \circ (J; \alpha^*)$. This natural transformation is defined by $\alpha^*_B = \alpha_B$ and the component

$$\alpha^*_N : I\!N \times M(B) \to X(N)$$

is given by

$$\begin{array}{lcl} \alpha^*_N(0, x) & = & (X(z))(\alpha_B(x)) \\ \alpha^*_N(i+1, x) & = & X(s)^{i+1}((X(z))(\alpha_B(x))) \end{array}$$

If one wants to specify exactly the natural numbers (up to isomorphisms), one has to add a product sketch which forces a model to map the object B to the terminal object in the category of sets. □

However, trying to extend this idea to a specification of finite lists will fail.

Example 2: For the example of the parametric data type of finite lists we take a category $\mathbb{C}^2$ with $obj(\mathbb{C}^2) = \{B, C, L, P\}$ and with the following generating morphisms

$$\begin{array}{l} nil : B \to L \\ cons : P \to L \\ p_1 : P \to C \\ p_2 : P \to L \end{array}$$

which constitute the generating graph $G(\mathbb{C}^2)$. Also in that case the set of defining relations is empty.

Beside two product sketches, where one forces a model to map B to the terminal object and the other to map the object p to the product of the images of L and C, one takes an injective sketch defined as follows. $\mathbb{C}_1 = \mathbb{C}^2$, $\mathbb{C}_0 = 1 + 1$ having exactly the two objects 1 and 2. $K = Id_{\mathbb{C}}$, $F(1) = B = J(1)$, $F(2) = C = J(2)$ and $\eta : F \Rightarrow J; K$ is the identity again.

But, now a functor $M : \mathbb{C}^2 \to Set$ is a model of that injective sketch, if $M(L) = M(P) = \emptyset$. □

Why does the conjunction of that injective sketch with the two product sketches do not work as expected? That's because the intended interpretation $M : \mathbb{C}^2 \to Set$ with $M(L) = M(C)^*$ does not have the required universal property of a left Kan extension within the whole functor category $Set^{\mathbb{C}^2}$, but only within the subcategory of those functors being a model of the two product sketches.

How one can relate the universal property of a Kan extension to a subcategory of the functor category? This possibility will be achieved by the notion of *nested sketches* in a category $\mathbb{C}$ defined as follows.

Definition 2.2

1. For each category $\mathbb{C}$ there is a *trivial* nested sketch denoted by $\top(\mathbb{C})$.

2. Any finite set of nested sketches in an category $\mathbb{C}$ is again a nested sketch in $\mathbb{C}$, written
$$\Delta = \{\Delta_1, \ldots, \Delta_n\}$$

3. Let be
$$(J : \mathbb{C}_0 \to \mathbb{C}_1, F : \mathbb{C}_0 \to \mathbb{C}, K : \mathbb{C}_1 \to \mathbb{C},\ \eta : F \Rightarrow J; K)$$
be an injective sketch in $\mathbb{C}$, let Δ_1 be a nested sketch in $\mathbb{C}_1$, and Δ_0 a nested sketch in $\mathbb{C}_0$. Then
$$\Delta = \langle(\Delta_1, \mathbb{C}_1), (\Delta_0, \mathbb{C}_0), J, F, K,\ \eta : F \Rightarrow J; K\rangle$$
is a nested sketch in $\mathbb{C}$.

4. Let be
$$(J : \mathbb{C}_0 \to \mathbb{C}_1, F : \mathbb{C}_0 \to \mathbb{C}, K : \mathbb{C}_1 \to \mathbb{C},\ \pi : J; K \Rightarrow F)$$
be a projective sketch in $\mathbb{C}$, let Δ_1 be a nested sketch in $\mathbb{C}_1$, and Δ_0 a nested sketch in $\mathbb{C}_0$. Then
$$\Delta = \langle(\Delta_1, \mathbb{C}_1), (\Delta_0, \mathbb{C}_0), J, F, K,\ \pi : J; K \Rightarrow F\rangle$$
is a nested sketch in $\mathbb{C}$.

In the following we define under which conditions a functor $M : \mathbb{C} \to \mathit{Sem}$ is a *model of a nested sketch* in $\mathbb{C}$:

1. Each functor $M : \mathbb{C} \to \mathit{Sem}$ is a model of the trivial nested sketch $\top(\mathbb{C})$.

2. $M : \mathbb{C} \to \mathit{Sem}$ is a model of $\Delta = \{\Delta_1, \ldots, \Delta_n\}$ if it is a model for each Δ_i, $i = 1, \ldots, n$.

3. $M : \mathbb{C} \to \mathit{Sem}$ is a model of
$$\Delta = \langle(\Delta_1, \mathbb{C}_1), (\Delta_0, \mathbb{C}_0), J, F, K,\ \eta : F \Rightarrow J; K\rangle$$
if
   (a) $K; M : \mathbb{C}_1 \to \mathit{Sem}$ is a model of Δ_1.
   (b) $F; M : \mathbb{C}_0 \to \mathit{Sem}$ is a model of Δ_0.
   (c) For each functor $X : \mathbb{C}_1 \to \mathit{Sem}$ such that X is a model of Δ_1 and $J; X$ is a model of Δ_0, and for each natural transformation $\alpha : F \Rightarrow J; X$ there is exactly one natural transformation $\alpha^* : K; M \Rightarrow X$ with $\alpha = (\eta; M) \circ (J; \alpha^*)$.

4. $M : \mathbb{C} \to \mathit{Sem}$ is a model of

$$\Delta = \langle (\Delta_1, \mathbb{C}_1), (\Delta_0, \mathbb{C}_0), J, F, K,\ \pi : J; K \Rightarrow F \rangle$$

if

   (a) $K; M : \mathbb{C}_1 \to \mathit{Sem}$ is a model of Δ_1.
   (b) $F; M : \mathbb{C}_0 \to \mathit{Sem}$ is a model of Δ_0.
   (c) For each functor $X : \mathbb{C}_1 \to \mathit{Sem}$ such that X is a model of Δ_1 and $J; X$ is a model of Δ_0, and for each natural transformation $\alpha : J; X \Rightarrow F$ there is exactly one natural transformation $\alpha^* : X \Rightarrow K; M$ with $\alpha = (J; \alpha^*) \circ (\pi; M)$.

A pair $(\mathbb{C}, \Delta)$ consisting of a finitely generated category and a nested sketch in $\mathbb{C}$ will be called a *theory of nested sketches* NS-theory for short. □

Using the trivial nested sketch for Δ_1 and Δ_0 in definitions (3) and (4) one gets the injective and projective sketches as special cases of nested sketches. Therefore we will in the following no more distinguish between nested and flat sketches, and will uniquely use the more general notion of nested sketches.

With this more general notion we are now able to specify the parametric data type of finite lists.

Example 3: We take the injective sketch of Example 2 and build a nested sketch according part (3) of the preceding definition, by taking Δ_1 to consists of the two product sketches (seen as nested sketches in $\mathbb{C}_1$), and taking for Δ_0 the trivial nested sketch $\top(\mathbb{C}_0)$.

Let Δ_{tree} denote the resulting nested sketch in $\mathbb{C}$. Now the class of models $M : \mathbb{C} \to \mathit{Set}$ of Δ_{tree} coincides with those interpretation where $M(B)$ is a singleton set $\{*\}$, $M(C)$ can be an arbitrary set, $M(L) = M(C)^*$ is the set of all finite list with elements in $M(C)$, and $M(P) = M(C) \times M(L)$. $M(p_1), M(p_2)$ are the projections, $M(\mathit{nil}) : \{*\} \to M(L)$ maps $*$ to the empty list and $M(\mathit{cons}) : M(C) \times M(L) \to M(L)$ maps an element $x_0 \in M(C)$ and a list $l = [x_1, \ldots, x_n] \in M(L)$ to the list $[x_0, x_1, \ldots, x_n]$. □

What about the existence of models? Can an NS-theory be unsatisfiable? This general questions can easily be answered, since for any NS-theory $(\mathbb{C}, \Delta)$ the unique functor $!_{\mathbb{C}} : \mathbb{C} \to \mathbf{1}$ is a model. Evidently the category $\mathbf{1}$ is not an interesting domain to construct models. The interesting question is, if in a specific category like *Set* models exist. In general the answer is no. To see that we take $\mathbb{C} = \mathbf{1}$ and let Δ consists of two discrete sketches which force a model $M : \mathbb{C} \to \mathit{Sem}$ to map the unique object as well to the initial as to the terminal object of *Sem*. Since in *Set* the initial object is not terminal and vice versa, there is no model in *Set* for that NS-theory. But, this NS-theory has for instance a model in the category of commutative groups.

Another interesting question concerns the existence of *generic models*, where a model $M_\Delta : \mathbb{C} \to \mathbb{L}_\Delta$ of an NS-theory $(\mathbb{C}, \Delta)$ is called generic, if for each model $M : \mathbb{C} \to Sem$ of $(\mathbb{C}, \Delta)$ there is a unique functor $M' : \mathbb{L} \to Sem$ with $M = M_\Delta; M'$.

Theorem 2.3 For each NS-theory $(\mathbb{C}, \Delta)$ there exists (up to isomorphisms) a unique generic model $M : \mathbb{C} \to \mathbb{L}$.

Proof: From the definition of generic models immediately follows that a generic model, if it exists, is unique up to isomorphisms.

So it remains to show that there is a generic model. This will be proved by induction on the depth of nested sketches, which will be defined next.

According to the inductive definition of nested sketches, the depth $i(\Delta)$ of a nested sketch can be defined as follows:

$$\begin{array}{lcl} i(\top(\mathbb{C})) & = & 0 \\ i(\{\Delta_1, \ldots, \Delta_n\}) & = & max\{i(\Delta_1), \ldots, i(\Delta_n)\} \\ i(\langle(\Delta_1, \mathbb{C}_1), (\Delta_0, \mathbb{C}_0), J, F, K,\ \pi : J; K \Rightarrow F\rangle) & = & 1 + max\{i(\Delta_1), i(\Delta_0)\} \\ i(\langle(\Delta_1, \mathbb{C}_1), (\Delta_0, \mathbb{C}_0), J, F, K,\ \eta : F \Rightarrow J; K\rangle) & = & 1 + max\{i(\Delta_1), i(\Delta_0)\} \end{array}$$

Let be Δ a nested sketch in $\mathbb{C}$ and $H : \mathbb{C} \to \mathbb{C}'$ any functor. Then we can define the nested sketch

$$\Delta; H \quad \text{in } \mathbb{C}' \qquad (*)$$

by using the inductive nature of nested sketches.
First we set
$\top_\mathbb{C}; H = \langle(\top_\mathbb{C}, \mathbb{C}), (\top_\mathbb{C}, \mathbb{C}), Id_\mathbb{C}, H, H, Id_H : H \Rightarrow H\rangle$
which can be seen as well as an instance of a nested sketch according (3) as well of (4) in Definition 2.2. The nested sketch $\top_\mathbb{C}; H$ is semantically equivalent to $\top_{\mathbb{C}'}$ since each functor $G : \mathbb{C}' \to \mathbb{C}''$ is a model of $\top_\mathbb{C}; H$.
Next we set $\Delta; H = \{\Delta_1; H, \ldots, \Delta_n; H\}$ if $\Delta = \{\Delta_1, \ldots, \Delta_n\}$, and
$\Delta; H = \langle(\Delta_1, \mathbb{C}_1), (\Delta_0, \mathbb{C}_0), J, F; H, K; H,\ \eta; H : F; H \Rightarrow J; (K; H)\rangle$
if $\Delta = \langle(\Delta_1, \mathbb{C}_1), (\Delta_0, \mathbb{C}_0), J, F, K,\ \eta : F \Rightarrow J; K\rangle$, and finally
$\Delta; H = \langle(\Delta_1, \mathbb{C}_1), (\Delta_0, \mathbb{C}_0), J, F; H, K; H,\ \pi; H : J; (K; H) \Rightarrow F; H\rangle$
if $\Delta = \langle(\Delta_1, \mathbb{C}_1), (\Delta_0, \mathbb{C}_0), J, F, K,\ \pi : J; K \Rightarrow F\rangle$.

Equipped with this notions we can start the inductive construction of a generic model.

The starting point is trivial, since for a nested sketch with Δ in $\mathbb{C}$ with $i(\Delta) = 0$ the identity of $\mathbb{C}$ is a generic model.

Now we assume, that for any nested sketch Δ with $i(\Delta) \leq n+1$ a generic model is given by $M_\Delta : \mathbb{C} \to \mathbb{L}_\Delta$.

Let $\Delta' = \langle(\Delta_1, \mathbb{C}_1), (\Delta_0, \mathbb{C}_0), J, F, K,\ \eta : F \Rightarrow J; K\rangle$ be a nested sketch with $i(\Delta') = n + 1$ which implies $max\{i(\Delta)_0), i(\Delta)_1)\} = n$.

Let be $\Delta = \{\Delta_1; K, \Delta_0; F\}$. Then $i(\Delta) = n$ and the induction hypothesis can be applied to Δ.

The generic model $M_{\Delta'} : \mathbb{C} \to \mathbb{L}_{\Delta'}$ will be constructed as injective limit of a chain

$$Q_i : \mathbb{L}_i \to \mathbb{L}_{i+1}\,,\ i = 0, 1, \ldots$$

with $\mathbb{L}_0 = \mathbb{L}_\Delta$, and each $\mathbb{L}_i$ will be given by a directed graph $G(\mathbb{L}_i)$, representing the generating morphisms, and by a congruence relation ρ_i in the category freely generated by $G(\mathbb{L}_i)$, i.e., in the category of finite paths over $G(\mathbb{L}_i)$. The chain of pairs $\langle G(\mathbb{L}_i), \rho_i\rangle$ will be inductively defined, and it will turn out that it is an increasing chain, so that the the injective limit will be given by the infinite unions, separately constructed in each component.

The construction of the chain starts with the pair $\langle G(\mathbb{L}_0), \rho_0\rangle$ with results from the induction hypothesis. Now let be given $\langle G(\mathbb{L}_i), \rho_i\rangle$.

Let $H_i : \mathbb{C} \to \mathbb{L}_i$ be the composition $H_i = M_\Delta; Q_0; \ldots O_{i-1}$.

For each pair (X, α), where $X : \mathbb{C}_1 \to \mathbb{L}_i$ is a functor such that it is a model of Δ_1 and $J; X$ is a model of Δ_0, and $\alpha : F; H_i \Rightarrow J; X$ is a natural transformation, such that no natural transformation $\alpha^* : K; H_i \Rightarrow X$ with $\alpha = (\eta; H_i) \circ (J; \alpha^*)$, for each such pair (X, α) we add the set of generating morphisms

$$\{f_{(X,\alpha,c)} : H_i(K(c) \to X(c) \mid c \in obj(\mathbb{C}_1)\}$$

to the generating graph $G(\mathbb{L}_i)$, which leads to the generating graph $G(\mathbb{L}_{i+1})$. The congruence relation ρ_{i+1} is the smallest congruence relation in the category freely generated by $G(\mathbb{L}_{i+1})$ such that $\rho_i \subseteq \rho_{i+1}$ and such that for each (X, α) used in the construction of $G(\mathbb{L}_{i+1})$ the family $\alpha^* = \{f_{(X,\alpha,c)} : H_i(K(c) \to X(c) \mid c \in obj(\mathbb{C}_1)\}$ becomes the unique natural transformation $\alpha^* : K; H_i \Rightarrow X$ satisfying $\alpha = (\eta; H_i) \circ (J; \alpha^*)$. The association $f \mapsto [f]_{\rho_{i+1}}$ defines the functor $Q_i : \mathbb{L}_i \to \mathbb{L}_{i+1}$.

Let be $G(\mathbb{L}_{\Delta'})$ be the injective limit of the increasing chain of generating graphs $G(\mathbb{L}_i)$, $i = 0, 1, \ldots$, and let $\rho_{\Delta'}$ be the smallest congruence relation in the category freely generated by $G(\mathbb{L}_{\Delta'})$. This two constructions give us a representation of the category $\mathbb{L}_{\Delta'}$, being the injective limit of the chain $Q_i : \mathbb{L}_i \to \mathbb{L}_{i+1}\,,\ i = 0, 1, \ldots$.

Let $M_0 : \mathbb{L}_0 \to \mathbb{L}_{\Delta'}$ the injection into the injective limit, and let $M_{\Delta'} : \mathbb{C} \to \mathbb{L}_{\Delta'}$ be the composition of M_Δ with M_0. The proceeding construction makes it evident that $M_{\Delta'} : \mathbb{C} \to \mathbb{L}_{\Delta'}$ is a model of Δ'.

It remains to show that this model is a generic mode of Δ'.

For that reason let $M : \mathbb{C} \to Sem$ be an arbitrary model of Δ'. By definition is $K; M$ a model of Δ_1 and $F; M$ a model of Δ_0, which implies that M is a model of both $\Delta_1; K$ and $\Delta_0; F$, i.e. M is a model of $\Delta = \{Delta_1, \Delta_0; F\}$. By induction hypothesis is $M_\Delta : \mathbb{C} \to \mathbb{L}_\Delta$ a generic model of Δ. This guarantees the existence of a unique functor $M_0 = \mathbb{L}_\Delta \to \mathit{Sem}$ with $M = M_\Delta; M_0$.

Inductively we will show that for each $i \in I\!N$ there is a unique functor $M_i : \mathbb{L}_i \to \mathit{Sem}$ with $(Q_0; Q_1; \ldots; Q_{i-1}); M_i = M_0$ which implies the existence of the required unique $M' : \mathbb{L}_{\Delta'} \to \mathit{Sem}$, since $\mathbb{L}_{\Delta'}$ is the injective limit of that chain.

For $i = 0$ the existence of $M_0 : \mathbb{L}_\Delta \to \mathit{Sem}$ has just been proved. Let $M_i : \mathbb{L}_i \to \mathit{Sem}$ with the corresponding properties be given. For each pair (X, α), used to construct new generating morphisms in $G(\mathbb{L}_{i+1})$ we obtain a natural transformation $\alpha; M_i : F; H_i; M_i \Rightarrow J; X; M_i$, i.e., $\alpha; M_i : F; M \Rightarrow J; X; M_i$, because of $H_i; M_i = M$. Because M is a model of Δ' there exists a unique natural transformation $\alpha^* : K; M \Rightarrow X; M_i$ with $\alpha; M_i = (\eta; M) \circ (J; \alpha^*)$. Mapping each new generating morphism $f_{(X,\alpha,c)} : H_i(K(c)) \to X(c)$ in $G(\mathbb{L}_{i+1})$ to $\alpha^*(c) : M(K(c)) \to M_i(X(c))$ defines together with M_i a graph homomorphisms from $G(\mathbb{L}_{i+1})$ to the underlying graph of the category Sem. The resulting unique graph homomorphism $M''_g : G(\mathbb{L}_{\Delta'})$ to the underlying graph of the category Sem induces finally a functor from the category freely generated by $G(\mathbb{L}_{\Delta'})$ to Sem whose kernel contains each ρ_i. This implies the existence of the unique functor $M^* : \mathbb{L}_{\Delta'} \to \mathit{Sem}$ with $M = M_{\Delta'}; M^*$.

The construction of a generic model of a nested sketch according point (4) of Definition 2.2 can be done analogously.

If $\Delta = \{\Delta_1, \ldots, \Delta_n\}$ with $i(\Delta) = n + 1$ and more than one of the Δ_i's has depth n, then one has to consider all pairs (X, α) simultaneously for all Δ_i's with depth n, in the inductive construction of the category $\mathbb{L}_{\Delta'}$. □

Next we point out a property of nested sketches and their models which basically guarantees that nested sketches built an institution in the sense of Goguen and Burstall [3].

Proposition 2.4. Let Δ be a nested sketch in $\mathbb{C}$ and $H : \mathbb{C} \to \mathbb{B}$ any functor. A functor $M : \mathbb{B} \to \mathit{Sem}$ is a model of $\Delta; H$ if and only if $H; M$ is a model of Δ.

Proof: The equivalence can be proved again by induction on the structure of nested sketches.

The case $\Delta = \top_\mathbb{C}$ is trivial, since each functor $H : \mathbb{C} \to \mathbb{D}$ is a model of $\top_\mathbb{C}$, and each functor $H' : \top_\mathbb{B} \to \top_\mathbb{D}$ is a model of $\top_\mathbb{C}; H$.

With the induction hypothesis immediately follows that Proposition 2.4 holds if $\Delta = \{\Delta_1, \ldots, \Delta_n\}$.

A bit more interesting are the cases where the nested sketches are constructed according point (3) or (4) in definition 2.2.

Let $\Delta = \langle(\Delta_1, \mathbb{C}_1), (\Delta_0, \mathbb{C}_0), J, F, K,\ \pi : J; K \Rightarrow F\rangle$ and assume that M is a model of $\Delta; H$.

To prove that $H; M$ is a model of Δ we assume that there is a functor $X : \mathbb{C}_1 \to Sem$ and a natural transformation $\alpha : F; (H; M) \Rightarrow J; X$. By associativity of functor composition we have a natural transformation $\alpha : (F; H;)M \Rightarrow J; X$, and since M is a model of $\Delta; H$ there is a unique natural transformation $\alpha^* : (K; H); M \Rightarrow X$ with $\alpha = ((\eta; H); M) \circ (J; \alpha^*)$. Using again associativity we obtain a unique natural transformation $\alpha^* : K; (H; M) \Rightarrow X$ with $\alpha = (\eta; (H; M) \circ (J; \alpha^*)$, which implies that $H; M$ is a model of Δ.

The converse and the case that Δ is constructed according point (4) in Definition 2.2 can be proved identically. $\square$

To get an *institution of nested sketches* in the sense of Goguen and Burstall one takes the category Cat_f of finitely generated categories as a category of signatures, one takes nested sketches in a category $\mathbb{C}$ as sentences over the signature $\mathbb{C}$, and one takes for each functor $H : \mathbb{C} \to \mathbb{B}$ in Cat_f the mapping

$$\Delta \longmapsto \Delta; H$$

as translation from sentences over $\mathbb{C}$ to sentences over $\mathbb{B}$. With respect to that construction Proposition 2.4 just states the validity of the satisfaction condition of the institution of nested sketches.

3 Examples of NS-theories and generic models

If a NS-theory only contains product sketches, the corresponding generic model corresponds to the term model, with the exception that only finitely many variables are given, since finitely many product sketches do not imply that the generic model is closed under finite products.

The term model considered as a category can be understood as a *minimal logic* for talking about structures whose domains of the basic operations are defined by products, i.e., for talking about many sorted algebras. The minimal logic, resulting from a Kan theory containing only product sketches, corresponds to equational logic.

In the following we will see that the minimal logic, given by the generic model of a NS-theory can be surprisingly expressive, depending on the constructions used to define domains and codomains of basic operations.

To get a better understanding for the range of expressiveness of NS-theories we consider as next theories which contain projective sketches.

Example 4: We take the category $\mathbb{C}^4$, given by the generating graph $G(\mathbb{C}^4)$:

$$B \xleftarrow{head} S \xrightarrow{tail} S$$

and the empty set of defining relations. We take the categories and functors $\mathbb{C}^4_0 = \mathbf{1}$, $\mathbb{C}^4_1 = \mathbb{C}^4$, $K = Id_{\mathbb{C}^4}$, $J(1) = B = F(1)$, and the natural transformation $\pi : J;K \Rightarrow F$ is the identity of $F(= J;K)$. With the notation of Definition 2.2 we consider the nested sketch (of depth one)

$$\Delta^4 = \langle (\top_{\mathbb{C}^4}, \mathbb{C}^4), (\top_{\mathbf{1}}, \mathbf{1}), J, F, K, \pi : F \Rightarrow J;K \rangle$$

in $\mathbb{C}^4$.

A functor $M : \mathbb{C}^4 \to Set$ is a model of Δ^4 iff $M(S) = M(B)^{I\!N}$, with $(M(h))(f) = f(0)$ and $(M(t))(f) = \lambda\, x.f(x+1)$ for each $(f : I\!N \to M(B)) \in M(S)$, i.e., the NS-theory $(\mathbb{C}^4, \Delta^4)$ specifies the *parametric Type of infinite streams.* $\Box$

Example 5: We extend the preceding example by taking the category $\mathbb{C}^5$, defined by the generating graph

$$B \xleftarrow{head} S \xrightarrow{tail} S \xleftarrow{head'} S' \xrightarrow{tail'} S'$$

and the empty set of defining relations. Now we consider two nested sketches (both of depth one) given by the following projective sketches:

$$\Delta^5_1 : \begin{array}{ccc} \mathbb{C}^1_1 & \xrightarrow{K^1} & \mathbb{C}^5 \\ {\scriptstyle J^1}\uparrow & \nearrow{\scriptstyle F^1} & \\ \mathbf{1} & & \end{array} \qquad\qquad \Delta^5_2 : \begin{array}{ccc} \mathbb{C}^2_1 = \mathbb{C}^5 & \xrightarrow{Id} & \mathbb{C}^5 \\ {\scriptstyle J^2}\uparrow & \nearrow{\scriptstyle F^2(=K^1)} & \\ \mathbb{C}^2_0 = \mathbb{C}^1_1 & & \end{array}$$

Where $\mathbb{C}^1_1 = \mathbb{C}^2_1 = \mathbb{C}^4$, $F^1(1) = b = J^1(1)$ and K^1, J^2 are the inclusion functors. Since both diagrams commute we can take for π_1, π_2 the corresponding identities to define projective sketches.

A functor $M : \mathbb{C}^5 \to Set$ is a model of the NS-theory $(\mathbb{C}^5, \{\Delta^5_1, \Delta^5_2\})$ iff $M(S) = M(B)^{I\!N}$ and $M(S') = (M(B)^{I\!N})^{I\!N}$. Thus, the object S' represents the type of infinite streams of infinite streams with elements in $M(B)$. This type is a simple example of a higher order process type, since the outputs of a process are again processes which themselves output elements of base type B. $\Box$

In the following we generalize the Example 4 in such a way that we allow the processes to terminate after an arbitrary finite number of steps. The specification of this kind of processes requires a nested sketch of depth two.

Example 6: We take the category $\mathbb{C}^6$ given by the defining graph

$$\begin{tikzcd} B & S \arrow[l, "head"'] \arrow[r, bend left, "tail"] \arrow[r, bend right, "inl"'] & S+T & T \arrow[l, "inr"'] \end{tikzcd}$$

and the empty set of defining relations. The nested sketch Δ^6 is depicted by

$$\Delta^6 : \qquad \begin{tikzcd} \Delta^6_1 : \mathbb{C}^6 \arrow[r, "Id"] & \mathbb{C}^6 \\ \top_{\mathbf{1}} : \mathbf{1} \arrow[u, "J"] \arrow[ur, "F"'] & \end{tikzcd}$$

the natural transformation $\pi : J;Id \Rightarrow F(= id_F)$, where $J(1) = B = F(1)$ and where Δ^6_1 consists of one product sketch, constraining the object T to the terminal object, and one sum sketch, making the object $S+T$ to the sum of S and T with the two injections $inl : S \to S+T$, $inr : T \to S+T$. □

Similarly to the specification of the parametric data type of finite lists one has to nest sketches in Example 6, since for each fixed interpretation $M(B)$ of the sort B one has to characterize the set $M(S)$ of finite and infinite sequences of elements in $M(B)$ as a final coalgebra, or equivalently as a most abstract, deterministic, partial automata with a singleton set as input alphabet, the set $M(B)$ as output alphabet and $M(S)$ as state set, where $M(tail) : M(S) \to M(S) + \{*\}$ represents the state transition function.

The preceding examples have shown that nested sketches with an injective top sketch are strongly related to inductively defined data types, and that nested sketches with a projective top sketch are related to coinductively defined process types. More advanced examples can be found in the literature on algebraic specifications of data types and on coalgebraic approaches to systems. We will not go further in that direction.

In the following we will investigate in which way induction and coinduction is present in the corresponding generic models. i.e., if induction and coinduction is present in the canonically related term model of the structures specified by an NS-theory.

Let as first look if induction is available in a generic model of an NS-theory with injective sketches. For that reason we consider first the following preparative example.

Example 7: $\mathbb{C}^7$ is the category given by the defining graph

$$\begin{tikzcd} T \arrow[r, bend left, "T\_inl"] \arrow[r, bend right, "T\_inr"'] & T+T & (T+T)\times B \arrow[l, "p\_1"'] \arrow[r, "p\_2"] & B \arrow[r, bend left, "B\_inl"] \arrow[r, bend right, "B\_inr"'] & B+B \end{tikzcd}$$

and the empty set of defining relations. We are interested in the generic model of the NS-theory

$$(\mathbb{C}^7, \Delta^7 = \{\Delta^7_1, \Delta^7_2, \Delta^7_3, \Delta^7_4\})$$

where Δ^7_1 constrains the object $T+T$ to the sum of T and T with the two injections $T\_inl : T \to T+T$, $T\_inr : T \to T+T$, where Δ^7_2 analogously constrains the object $B+B$ to become the twofold sum of B with the injections $B\_inl : B \to B+B$, $B\_inr : B \to B+B$, Δ^7_3 constrains the object T to the terminal object, and Δ^7_4 finally constrains the object $(T+T)\times B$ to become the product of $T+T$ and B with the projections $p\_1 : (T+T)\times B \to T+T$, $p\_2 : (T+T)\times B \to B$.

In the generic model $\mathbb{L}_{\Delta^7}$ the projective sketch Δ^7_3 generates for each object X a unique morphism $!_X : X \to T$ with $!_T = id_T$ and the two sum sketches generate morphisms $cd_T : T+T \to T$, $cd_B : B+B \to B$ with $T\_inl; cd_T = id_T = T\_inr; cd_T$ and $B\_inl; cd_B = id_B = B\_inr; cd_B$. Whoever constructing the congruence ρ_1 according the proof of Theorem 2.2, the two generated morphisms cd_T and $!_{T+T}$ will be identified. In the next generation step the two morphisms $!_B; T\_inl, !_B; T\_inr : B \to B+B$ generate the morphism $!_B + !_b B : B+B \to T+T$. This morphism together with the morphism $cd_B : B+B \to B$ cause the product sketch Δ^7_4 to generate a morphism

$$\langle !_B + !_B, cd_B \rangle : B+B \to (T+T)\times B$$

with corresponding equations. We have pointed out that morphism, because each model $M : \mathbb{C}^7 \to Set$ of Δ^7 interprets that morphism as a bijection, i.e. as an isomorphism in Set, since the disjoint sum of a set $M(B)$ with itself can be constructed by a product $\{1,2\}\times M(B)$. As it is well known, this relation between products and sums does not hold in each category and so also not in the generic model. □

We have developed this simple example so carefully, since induction is usually expressed as parametric induction, i.e., induction applied to a component of a Cartesian product. Primitive recursion is a typical induction scheme of that kind. This kind of induction cannot be present in generic models in general. In which way the induction is present in generic models will be illustrated, by defining the operation of adding two natural numbers.

Example 8: If we want to specify the addition of natural numbers as a morphism in the generic model, then there has to be an object representing the domain of that operation. The argumentation above makes clear that a product would not work. Coming back to Example 1 we see that a functor $M : \mathbb{C} \to Set$ is a model of the injective sketch introduced there if

$$M(N) = I\!N \times M(B) = \sum_{i\in I\!N} M(B)_i$$

If one takes $M(B) = I\!N$ then we obtain $I\!N \times I\!N = \sum_{i\in I\!N} I\!N_i$. This instantiation of the parameter can be achieved by the following NS-theory $(\mathbb{C}^8, \Delta^8 = \{\Delta^8_1, \Delta^8_2, \Delta^8_3\})$, where $\mathbb{C}^8$ is given by the generating graph

$$T \xrightarrow{z_1} N_1 \xleftarrow{s_1} N_1 \xrightarrow{z_2} N_2 \xleftarrow{s_2} N_2$$

and the empty set of generating relations. Δ_1^8 constrains the object T to the terminal object, and the two injective sketches Δ_2^8, Δ_3^8 of depth one are given by the diagrams

$$\Delta_2^8: \begin{tikzcd} \mathbb{C}_1^1 \arrow[r, "K^1"] & \mathbb{C}^8 \\ \mathbf{1} \arrow[u, "J^1"] \arrow[ur, "F^1"'] & \end{tikzcd} \qquad \Delta_3^8: \begin{tikzcd} \mathbb{C}_1^2 = \mathbb{C}_1^1 \arrow[r, "K^2"] & \mathbb{C}^8 \\ \mathbf{1} \arrow[u, "J^2"] \arrow[ur, "F^2"'] & \end{tikzcd}$$

Where $\mathbb{C}_1^2 = \mathbb{C}_1^1$ is the category given by the defining morphisms

$$T \xrightarrow{z} N \xleftarrow{s} N$$

and the empty set of defining relations. The corresponding functors are defined by $J^1(1) = T = F^1(1), J^2(1) = T, F^2(1) = N_1, K^1(z) = z_1, K^2(z) = z_2, K^1(s) = s_1$ and $K^2(s) = s_2$. The corresponding natural transformations $\eta_2^8 : F^1 \Rightarrow J^1; K^1$ and $\eta_3^8 : F^2 \Rightarrow J^2; K^2$ are the identities.

Pairs of natural numbers $(n, m) \in I\!N \times I\!N$ can be represented by morphisms

$$z_1; \underbrace{s_1; \ldots; s_1}_{n-times}; z_2; \underbrace{s_2; \ldots; s_2}_{m-times} : T \to N_2.$$

The sketch Δ_1^8 generates for each object X uniquely morphisms $!_X : X \to T$.

In the next step we construct morphisms $p_1, p_2 : N_2 \to N_1$ in the generic model which satisfy

$$\begin{aligned} z_1; \underbrace{s_1; \ldots; s_1}_{n-times}; z_2; \underbrace{s_2; \ldots; s_2}_{m-times}; p_1 &= z_1; \underbrace{s_1; \ldots; s_1}_{n-times} \\ z_1; \underbrace{s_1; \ldots; s_1}_{n-times}; z_2; \underbrace{s_2; \ldots; s_2}_{m-times}; p_2 &= z_1; \underbrace{s_1; \ldots; s_1}_{m-times} \end{aligned}$$

and represent projections (not satisfying universal properties).

One can easily check that the two equations would hold if the two morphisms p_1, p_2 would satisfy the following equations:

$$\begin{aligned} z_2; p_1 &= !_{N_1}; z_1 & \qquad z_2; p_2 &= id_{N_1} \\ s_2; p_1 &= p_1; s_1 & \qquad s_2; p_2 &= p_2 \end{aligned}$$

These two projections can be constructed as

$$p_1 = f_{(H_{p_1}, id, N_2)}, \quad p_2 = f_{(H_{p_2}, id, N_2)}$$

where the functors $H_{p_i} : \mathbb{C}_1^2 \to \mathbb{C}^8$ for $i = 1, 2$ are defined by $H_{p_1}(z) = z_1, H_{p_1}(s) = s_1$ and $H_{p_2}(z) = id_{N_1}, H_{p_2}(s) = id_{N_1}$.

Even though one can define the two projections, they do not have the universal property necessary for a product. However, for each model $M : \mathbb{C}^8 \to Set$ of $(\mathbb{C}^8, \Delta^8)$ the mapping $M((p_1, p_2)) : M(N_2) \to M(N_1) \times M(N_1)$ is a bijection. This is one consequence of distributivity properties of the category of sets with respect to injective and projective limits.

Finally the morphism

$$add = f_{(H_+, id, N_2)} : N_2 \to N_1$$

using the functor $H_+ : \mathbb{C}_1^2 \to \mathbb{C}^8$ defined by $H_+(z) = id_{N_1}, H_+(s) = s_1$ defines the addition of natural numbers. The corresponding equations represent with the notation of pairs of natural numbers the following recursive definition:

$$\begin{array}{lcl} add(n, 0) & = & n \\ add(n, m+1) & = & add(n, m) + 1 \end{array}$$

This is not the only way to construct in the generic model a morphism representing the addition. □

This example does not only illustrate the expressive power of generic models, it shows in addition that the iterative construction of the generic model does not necessarily converge after finitely many steps, because for the given NS-theory the schema of primitive recursion can always be applied to that operation, that has been generated in the step before and delivers a new operation. Thus in the second step we would generate a morphism representing the multiplication, in the following a morphism representing the exponentiation and so on.

By the next example we illustrate that generic models also reflect *coinduction*.

Example 9: Let $\mathbb{C}^9$ be the category that results from $\mathbb{C}^4$ by adding a new object, denoted by $s \times s$ and two new generating morphisms $p_1, p_2 : s \times s \to s$. If Δ^4 denotes the projective sketch in $\mathbb{C}^4$ as defined in Example 4, and if $I : \mathbb{C}^4 \to \mathbb{C}^9$ denotes the inclusion, then let

$$(\mathbb{C}^9, \Delta^9 = \{\Delta^4; I, \Delta^4_{s \times s}\})$$

denote the NS-theory, where $\Delta^4_{s \times s}$ denotes a product sketch which makes the object $s \times s$ to the product with the projections $p_1, p_2 : s \times s \to s$. In the generic model the product sketch generates a morphism

$$\langle p_2, (p_1; tail)\rangle : s \times s \to s \times s$$

satisfying the equations $\langle p_2, (p_1; tail)\rangle; p_1 = p_1$ and $\langle p_2, (p_1; tail)\rangle; p_2 = p_1; tail$. By means of that morphism we can define a functor $H : \mathbb{C}_1^1 \to \mathbb{C}^9$ by $H(head) = p_1; head$ and $H(tail) = \langle p_2, (p_1; tail)\rangle$. If $\alpha : J; I \Rightarrow F; I$ denotes the identity transformation, so that in the next generation step a morphism

$$f_{(H, \alpha, s \times s)} : s \times s \to s$$

is generated which satisfies the equations

$$\begin{array}{lcl} f_{(H,\alpha,s\times s)}; head & = & p_1; head \\ f_{(H,\alpha,s\times s)}; tail & = & \langle p_2, (p_1; tail)\rangle; f_{(H,\alpha,s\times s)} \end{array}$$

One can easily recognize the coinductive definition of the merge operation. $\square$.

The examples show that in the generic models both induction and coinduction are present in a very natural and explicit way.

The next example represents the parameterized data type of the finitary power set construction.

Example 10: Let $\mathbb{C}^{10}$ be the category with the object set $\{t, b, p, b\times p, b\times b\times p\}$, with the defining morphism

$empty : t \to p$,
$add : b \times p \to p$,
$p_1^2 : b \times p \to b,\ p_2^2 : b \times p \to p$,
$p_1^3 : b \times b \times p \to b,\ p_2^3 : b \times b \times p \to b,\ p_3^3 : b \times b \times p \to p$,
$d : b \times p \to b \times b \times p$,
$\langle p_1^3, p_3^3\rangle : b \times b \times p \to b \times p$,
$\langle p_2^3, p_3^3\rangle : b \times b \times p \to b \times p$,
$\langle p_1^3, \langle p_2^3, p_3^3\rangle; add\rangle : b \times b \times p \to b \times p$,
$\langle p_2^3, \langle p_1^3, p_3^3\rangle; add\rangle : b \times b \times p \to b \times p$

and with the defining relations

$d; p_1^3 = d; p_2^3 = p_1^2,\ d; p_3^3 = p_2^2$,
$\langle p_1^3, p_3^3\rangle; p_1^2 = p_1^3,\ \langle p_1^3, p_3^3\rangle; p_2^2 = p_3^3$,
$\ldots$
$\langle p_2^3, \langle p_1^3, p_3^3\rangle; add\rangle; p_1^2 = p_2^3$,
$\langle p_2^3, \langle p_1^3, p_3^3\rangle; add\rangle; p_2^2 = \langle p_1^3, p_3^3\rangle; add$,
$d; \langle p_1^3, \langle p_2^3, p_3^3\rangle; add\rangle; add = add$,
$\langle p_1^3, \langle p_2^3, p_3^3\rangle; add\rangle; add = \langle p_2^3, \langle p_1^3, p_3^3\rangle; add\rangle; add.$

The last two equations express the idempocy and commutativity of adding an element to a finite set of elements. The other morphisms and equations represent manipulations with variables, which are not present explicitly.

Let $\Delta_t,\ \Delta_{b\times p},\ \Delta_{b\times b\times p}$ be product sketches in $\mathbb{C}^{10}$ that constraint the object t to the terminal one and the others to the indicated products and $\Delta = \{\Delta_t, \Delta_{b\times p}, \Delta_{b\times b\times p}\}$. Finally we define a nested Kan sketch Δ^{10} in $\mathbb{C}^{10}$

$$\begin{tikzcd} \Delta : \mathbb{C}^{10} \arrow[r, "Id"] & \mathbb{C}^{10} \\ \mathbf{1} \arrow[u, "J"] \arrow[ur, "F"'] & \end{tikzcd}$$

with $J(1) = F(1) = b$, where the natural transformation $\eta^{10} : F \Rightarrow J; (Id_{\mathbb{C}^{10}})$ is once more the identity.

A functor $M : \mathbb{C}^{10} \to Set$ is a model of the Kan theory $(\mathbb{C}^{10}, \Delta^{10})$ if $M(p)$ is the set of all finite subsets of $M(b)$. $\Box$

Example 10 demonstrates in addition to the preceding examples the use of defining equations for the constraint category.

But, the example also demonstrates that the notation used so far is not convenient for more complex examples. One needs some more readable textual representations and possibilities to built up complex NS-theories from smaller ones, defined before.

4 Structuring NS-theories

In the examples above most sketches have been commutative triangles, so that the corresponding natural transformation is the identity. In addition the functor $J : \mathbb{C}_0 \to \mathbb{C}_1$ has been the embedding of a subcategory. In the following we will show that natural transformations as parts of nested sketches ar not explicitly needed and that a restriction to this kind of sketches does not restrict the expressiveness of sketches.

Definition 4.1. An injective (respectively projective) sketch

$$(J : \mathbb{C}_0 \to \mathbb{C}_1, F : \mathbb{C}_0 \to \mathbb{C}, K : \mathbb{C}_1 \to \mathbb{C},\ \eta : F \Rightarrow J; K)$$

is called a *commutative triangle sketch, ct-sketch* for short, if $J : \mathbb{C}_0 \to \mathbb{C}_1$ is the embedding of a subcategory, if $J; K = F$ and if $\eta = id_F$ respectively $\pi : J; K \Rightarrow F = id_F$. A nested sketch is called a nested ct-sketch if all included sketches are ct-sketches or if it is the trivial nested sketch. $\Box$

Theorem 4.2. For each nested sketch Δ in a category $\mathbb{C}$ there exists a semantically equivalent nested ct-sketch.

Proof: We consider first nested sketches of depth one.

Let

$$\Delta \quad (J : \mathbb{C}_0 \to \mathbb{C}_1, F : \mathbb{C}_0 \to \mathbb{C}, K : \mathbb{C}_1 \to \mathbb{C},\ \eta : F \Rightarrow J; K)$$

be an injective sketch in $\mathbb{C}$. This sketch can equivalently be replaced by the following injective ct-sketch

$$\Delta^* \quad (J^* : \mathbb{C}_0 \to \mathbb{C}_1^*, F : \mathbb{C}_0 \to \mathbb{C}, K^* : \mathbb{C}_1^* \to \mathbb{C},\ id_F : F \Rightarrow J^*; K^*(= F))$$

where $J^*; K^* = F$ and $J^* : \mathbb{C}_0 \to \mathbb{C}_1^*$ is the embedding of a subcategory.

The category $\mathbb{C}_1^*$ can be constructed form $J : \mathbb{C}_0 \to \mathbb{C}_1$ as follows:

Take the sum $\mathbb{C}_0 + \mathbb{C}_1$, then adjoin for each object $X \in obj(\mathbb{C}_0)$ a morphism $h_X : X \to J(X)$ and adjoin an equation $f; h_Y = h_X; J(f)$ for each morphism $f : X \to Y$ in $\mathbb{C}_0$ to the union of the defining relations of $\mathbb{C}_0$ and $\mathbb{C}_1$. Let $J^* : \mathbb{C}_0 \to \mathbb{C}_1^*$ denote the embedding of $\mathbb{C}_0$ into $\mathbb{C}_1^*$

Let $J_1^* : \mathbb{C}_1 \to \mathbb{C}_1^*$ be the embedding of $\mathbb{C}_1$ and $\eta^* : J^* \Rightarrow J; J_1^*$ be the natural transformation with $\eta_X^* = h_X$ for each object $X \in \mathbb{C}_0$

The resulting pair $(\mathbb{C}_1^*, \eta^*)$ has the following universal property:
For each pair of functors $F : \mathbb{C}_0 \to \mathbb{C}, K : \mathbb{C}_1 \to \mathbb{C}$ and each natural transformation $\eta : F \Rightarrow J; K$ there exists a unique functor $K^* : \mathbb{C}_1^* \to \mathbb{C}$ with $J^*; K^* = F$ and $\eta = \eta^*; K^*$.

It remains to prove that the two injective sketches Δ, Δ^* are semantically equivalent.

First let $M : \mathbb{C} \to Sem$ be any model of Δ. To prove that this model is aslo a model of Δ^* let $H : \mathbb{C}_1^* \to Sem$ by any functor and $\alpha : F; M \Rightarrow J_0^*; H$ be any natural transformation.

Then $\eta^*; H : J_0^* \Rightarrow J; (J_1^*; H)$ and since $M : \mathbb{C} \to Sem$ is a model of Δ for the functor $J_1^*; H : \mathbb{C}_1 \to Sem$ and the natural transformation $\alpha \circ (\eta^*; H) : F; M \Rightarrow J; (J_1^*; H)$ there exists exactly one natural transformation $\alpha^* : K; M \Rightarrow J_1^*; H$ satisfying

$$(*) \quad \alpha \circ (\eta^*; H) = (\eta; M) \circ (J_1^*; \alpha^*)$$

In the next step we lift the two natural transformations

$$\begin{aligned} &\alpha^* : K; M (= J_1^*; K^*; M) \Rightarrow J_1^*; H \\ &\alpha : F; M (= J_0^*; K^*; M) \Rightarrow J_0^*; H \end{aligned}$$

to a natural transformation

$$\alpha^{**} : (K^*; M)|_{\mathbb{C}_0 + \mathbb{C}_1} \Rightarrow H|_{\mathbb{C}_0 + \mathbb{C}_1}$$

Because of $obj(\mathbb{C}_1^*) = obj(\mathbb{C}_0 + \mathbb{C}_1)$ and since each morphism in $\mathbb{C}_1^*$ which is neither contained in $\mathbb{C}_0$ nor $\mathbb{C}_1$ can be represented as $X; \eta^* : X; J_0^* \Rightarrow X; J; J_1^*$ for some $X : \mathbf{1} \to \mathbb{C}_0$ the equation $(*)$ implies the commutativity of the following diagram:

$$\begin{tikzcd}
X; J_0^*; K^*; M (= X; F; M) \arrow[r, "X;\eta*;K^*;M(=X;(\eta;M))"] \arrow[d, "X;J_0^*;\alpha^{**}(=X;\alpha)"'] & X; J; J_1^*; K^*; M \arrow[d, "X;J;J_1^*;\alpha^{**}(=X;J;\alpha^*)"'] \\
X; J_0^*; H \arrow[r, "X;\eta^*;H"'] & X; J; J^*; H
\end{tikzcd}$$

which shows that the lifting defines a natural transformation

$$\alpha^{**} : K^*; M \Rightarrow H.$$

Evidently holds $\alpha = J_0^*; \alpha^{**}$. Since α determines α^* uniquely, and these two natural transformations determine α^{**} uniquely, it is proved that $M : \mathbb{C} \to Sem$ is a model of Δ^*.

For the second part we assume now that $M : \mathbb{C} \to Sem$ is a model of Δ^*, and that a functor $H : \mathbb{C}_1 \to Sem$ and a natural transformation $\alpha : F : M \Rightarrow J; H$ is given.

The universal property of $(\mathbb{C}_1^*, \eta^*)$ implies the existence of a functor $H^* : \mathbb{C}_1^* \to Sem$ satisfying $J_1^*; H^* = H$, $J_0^*; H^* = F; M$, and $\eta^*; H^* = \alpha$. Since $M : \mathbb{C} \to Sem$ is a model of Δ^* there is a unique natural transformation $\alpha^* : K^*; M \Rightarrow H^*$ with $J_0^*; \alpha^* = id_{F;M}$.

The sequential composition of the two natural transformations

$$\eta^*; \alpha^* : J_0^*; K^*; M \Rightarrow J; J_1^*; H^*$$

together with the associated equalities

$$\eta^*; \alpha^* = (J_0^*; \alpha^*) \circ (\eta^*; H^*) = (\eta^*; (K^*; M)) \circ ((J; J_1^*; \alpha^*))$$

deliver the equality

$$id_{F;M} \circ \alpha = (\eta; M) \circ (J; (J_1^*); \alpha^*)$$

that has to be proved.

If $\alpha' : K; M \Rightarrow H$ with $\eta; M) \circ (J; \alpha')$ would be given, then one can lift this natural transformation to $\alpha'' : K^*; M \Rightarrow H^*$ with $J_0^*; \alpha'' = id_{F;M}$ and $J_1^*; \alpha'' = \alpha'$. Because of $J_0^*; \alpha'' = id_{F;M}$ it follows $\alpha^* = \alpha''$, so that $J_1^*; \alpha^* = J_1^*; \alpha'' = \alpha'$. This completes the proof of the second part of the semantical equivalence of Δ and Δ^*.

The construction of a semantically equivalent projective ct-sketch for a given projective sketch can be done in a completely analogous way.

If Δ is a projective, respectively injective ct-sketch in $\mathbb{C}$ and $H : \mathbb{C} \to \mathbb{C}'$ any functor, then $\Delta; H$ is evidently a ct-sketch in $\mathbb{C}'$. This simple observation allows to extend the construction described above inductively to a construction of semantically equivalent nested ct-sketch of finite depth. □

This observation above makes it possible to describe complex NS-theories basically by the concept of *sub theories that are initially or finally constraint.* Initial constraints are expressed by injective ct-sketches and final constraints by means of projective ct-sketches.

The construction of complex NS-theories is furthermore based on the fact that the category of nested ct-sketches is closed under injective limits and that these limits can be computed in the category of categories.

Definition 4.3. Let $(\mathbb{C}^1, \Delta_1), (\mathbb{C}^2, \Delta_2)$ by NS-theories with nested ct-sketches. A functor $F : \mathbb{C}^1 \to \mathbb{C}^2$ is called a *theory morphism* if the translation $\Delta_1; F$ of Δ_1 along $F : \mathbb{C}^1 \to \mathbb{C}^2$ is a subset of Δ_2. $\mathbb{NST}$ denotes category of NS-theories with nested ct-sketches. □

Theorem 4.4. The category $\mathbb{NST}$ of NS-theories with nested ct-sketches is closed under injective limits and the forgetful functor $U : \mathbb{NST} \to \mathbb{CAT}$ into the category of categories preserves injective limits.

Proof: Since the proof is simple, we sketch only the idea by the example of pushouts. Let

$$\begin{array}{l} F : (\mathbb{C}^0, \Delta_0) \to (\mathbb{C}^1, \Delta_1), \\ G : (\mathbb{C}^0, \Delta_0) \to (\mathbb{C}^2, \Delta_2) \end{array}$$

be theory morphisms, and let

$$\begin{tikzcd} \mathbb{C}^0 \arrow[r, "F"] \arrow[d, "G"] & \mathbb{C}^1 \arrow[d, "Q_F"] \\ \mathbb{C}^2 \arrow[r, "Q_G"] & \mathbb{C} \end{tikzcd}$$

be a pushout diagram in $\mathbb{CAT}$. Then is

$$\begin{array}{l} Q_F : (\mathbb{C}^1, \Delta_1) \to (\mathbb{C}, \Delta_1; Q_F \cup \Delta_2; Q_G) \\ Q_G : (\mathbb{C}^2, \Delta_2) \to (\mathbb{C}, \Delta_1; Q_F \cup \Delta_2; Q_G) \end{array}$$

evidently a pushout in $\mathbb{NST}$. □

Injective limits can be used to define operations that glue together NS-theories. The main construction used in structuring NS-theories is the pushout. We will try to take over the structuring operations from the recently defined specification language CASL, [8].

A *specification* of an NS-theory $(\mathbb{C}, \Delta)$ consists of four parts: the first part, indicated by the key word **sort**, declares (the names of) the objects of $\mathbb{C}$, the second part, indicated by **op**, declares the generating morphisms of $\mathbb{C}$, the third part, starting with **equations**, represents the defining relations of $\mathbb{C}$, and the final part, indicated by **constraints**, lists the initial and terminal constraints contained in Δ.

The first operation for combining specifications is the *union*: If $SP_1, \ldots, SP_n$ are specifications then

$$SP_1 \textbf{ and } SP_2 \textbf{ and} \ldots, SP_n$$

is a specification which combines the specifications such that when any part is common to some of the combines specifications, its interpretation in a model has to be a common one too.

The next operation is the *translation* which renames the objects and morphisms in a specification by means of a functor, i.e., the translation represents the construction of translating an NS-theory along a functor. The translated specification is written:

$$SP \ \mathbf{with}\ \{X_1 \mapsto X_1', \ldots, f_n \mapsto f_n'\}$$

The key word **then** will be used to represent *extensions* of specifications. We distinguish three different kinds of extensions: *simple, free* and *cofree* extensions of a specification.

$$SP_1 \ \mathbf{then}\ SP'$$

adds conservatively new objects, new morphisms and new axioms to the given specification PS_1, i.e. the reduct of a model of the extended specification to the category described by SP_1 has to be a model of SP_1.

$$SP_1 \ \mathbf{then\ free}\ SP' \ [\mathbf{regarding}\ SP'']$$

requires that a model of the freely extended specification is an extension of a model of SP_1 that it satisfies all axioms stated in SP' and that it forms a left Kan extension of its reduct to SP_1 within the class of models satisfying SP''.

Accordingly

$$SP_1 \ \mathbf{then\ cofree}\ SP' \ [\mathbf{regarding}\ SP'']$$

requires that a model of the cofreely extended specification is an extension of a model of SP_1 that it satisfies all axioms stated in SP' and that it forms a right Kan extension of its reduct to SP_1 within the class of models satisfying SP''.

Thus, SP_1 represents $(\mathbb{C}_0, \Delta_0)$, SP' describes the extension to $\mathbb{C}_1$ and SP'' represents Δ_1. In this way we are able to represent a single ct-sketch with $\mathbb{C} = \mathbb{C}_1$ and $K = Id_{\mathbb{C}}$. In this way basic patterns of initial and terminal constraints can be defined. By translations this patterns can be adapted to other specifications

It is not the aim of this paper to define a new specification language. We only want to sketch in which way specifications could be composed by means of structuring operations on NS-theories, and we want to be able to represent more complex examples of NS-theories in a readable way.

In the following we give some simple specifications which will be used later to build more complex specifications.

$\textsc{Triv} = \mathbf{sorts}: \quad Elem \quad \mathbf{end}$

$$
\begin{array}{lll}
\text{NAT1} = \text{TRIV} & \multicolumn{2}{l}{\textbf{then free}} \\
& \textbf{sort} & Nat \\
& \textbf{ops} & zero : Elem \to Nat \\
& & succ : Nat \to Nat \quad \textbf{end}
\end{array}
$$

This specification represents the NS-theory of Example 1.

Other basic patterns of constraints are the following:

$$
\begin{array}{lll}
\text{TERMINAL} = \emptyset & \multicolumn{2}{l}{\textbf{then cofree}} \\
& \textbf{sort} & 1 \quad \textbf{end}
\end{array}
$$

$$
\begin{array}{lll}
\text{SUM} = \textbf{sort } A, B & & \\
& \multicolumn{2}{l}{\textbf{then free}} \\
& \textbf{sort} & S \\
& \textbf{op} & in_A : A \to S \\
& & in_B : B \to S \quad \textbf{end}
\end{array}
$$

$$
\begin{array}{lll}
\text{PROD} = \textbf{sort } A, B & & \\
& \multicolumn{2}{l}{\textbf{then cofree}} \\
& \textbf{sort} & P \\
& \textbf{op} & p_A : P \to A \\
& & p_B : P \to B \quad \textbf{end}
\end{array}
$$

In the following specification some of this patterns are used within the **constraints** part of the specification.

$$
\begin{array}{lll}
\text{NAT2} = \textbf{sort } B, Nat1, Nat2 & & \\
& \textbf{op} & zero1 : B \to Nat1 \\
& & succ1 : Nat1 \to Nat1 \\
& & zero2 : Nat1 \to Nat2 \\
& & succ2 : Nat2 \to Nat2 \\
& \multicolumn{2}{l}{\textbf{constraints}} \\
& \multicolumn{2}{l}{\text{TERMINAL } \textbf{with } \{1 \mapsto B\}} \\
& \multicolumn{2}{l}{\text{NAT1 } \textbf{with } \{zero \mapsto zero1, succ \mapsto succ1\}} \\
& \multicolumn{2}{l}{\text{NAT1 } \textbf{with } \{zero \mapsto zero2, succ \mapsto succ2\}} \\
& \multicolumn{2}{l}{\textbf{end}}
\end{array}
$$

This specification describes Examples 8 and the following specification

$$
\begin{array}{lll}
\textsc{Streams} = \textsc{Triv}\ \textbf{then cofree} & & \\
\qquad \textbf{sort} & \mathit{Streams} & \\
\qquad \textbf{op} & \mathit{head} : \mathit{Streams} \to \mathit{Elem} & \\
 & \mathit{tail} : \mathit{Streams} \to \mathit{Streams} & \textbf{end}
\end{array}
$$

corresponds to Example 4.

Since product and sum sketches are so frequently used, we will use them as built in constructions, and will not explicitely use the injective or projective sketches described above. In addition we use for terms generated by products and sums the usual notations with variables in order to improve readability. Thus, if generating morphisms $f : A \times B \to C, h : B \to A, g : A \to B$ are given, then $f(h(x), g(y)) : B \times A \to C$, and $f(x, g(x)) : A \to C$, and if additionally $t : C \to (A + B)$ is given, then $[g(f(x)), f(x)] : C \to B$ and $[g(f(x)), h(f(x))] : C \to (B + A)$.

The specifications given so far correspond to ct-sketches of depth one. The next example represents a ct-sketch of depth two.

$$
\begin{array}{ll}
\textsc{Sequences} = \textsc{Triv}\ \textbf{then cofree} & \\
\qquad \textbf{sort} & \mathit{Sequences},\ 1 \\
\qquad \textbf{op} & \mathit{head} : \mathit{Sequences} \to \mathit{Elem} \\
 & \mathit{tail} : \mathit{Sequences} \to (\mathit{Sequences} + 1) \\
\qquad \textbf{regarding}\ 1,\ \mathit{Sequence} + 1 & \\
\qquad \textbf{end} &
\end{array}
$$

5 The expressive power of the generic model

By the following examples we will demonstrate, that the investigation of the generic model of an NS-theory can be very helpful for the understanding of the corrsponding structure. Since inductively defined data types are well understood we will predominantly deal with coinductively defined structures.

As made visible in Example 8 some very useful relations between injecitve and projective limites valid in the category of sets are not satisfied in generic models. Since we are basically intersted in model in *Set* we will enforce some of this properties. A first example in this direction is the specification Bool of *truth values.*

$$
\begin{array}{lll}
\textsc{Bool} = & \textbf{sort} & \mathit{Bool}, \mathit{Bool} + \mathit{Bool}, 1 \\
 & \textbf{op} & \mathit{true}, \mathit{false} : 1 \to \mathit{Bool} \\
 & & \mathit{inl}, \mathit{inr} : \mathit{Bool} \to \mathit{Bool} + \mathit{Bool}
\end{array}
$$

constraints
$\textsc{Terminal}$ **with** $\{1 \mapsto 1\}$
$\textsc{Sum}$ **with** $\{in_A \mapsto true, in_B \mapsto false\}$
$\textsc{Sum}$ **with** $\{in_A \mapsto inl, in_B \mapsto inr\}$
$\textsc{Prod}$ **with** $\{p_A \mapsto [()_{Bool+Bool}; true, ()_{Bool+Bool}; false],$
$\quad p_B \mapsto [id_{Bool}, id_{Bool}]\}$
end

Up to isomorphism the unique model $M : \textsc{Bool} \to Set$ is given by $M(Bool) = \{true, false\},\ M(inl)(x) = (true, x),\ M(inr)(x) = (false, x)$.

It is easy to see that all operations on truth values are represented in the generic model of $\textsc{Bool}$. For instance, the conjunction is represented by

$$[id_{Bool}, (()_{Bool}; false)] : Bool + Bool \to Bool$$

and the disjunction by

$$[(()_{Bool}; true), id_{Bool}] : Bool + Bool \to Bool.$$

As next we give a specification of the finitary power set construction, which will be needed later to represent the codomain of the state transition function of nondeterministic, image finite transition systems.

$\textsc{Sets} = \textsc{Triv}$ **and** $\textsc{Bool}$ **then free**
sort $Set, Elem \times Set, Elem \times Elem \times Set$
op $empty : 1 \to Set$
$\quad join : Elem \times Set \to Set$
equations $x, y : Elem, s : Set$
$join(x, join(x, s)) = join(x, s)$
$join(x, join(y, s)) = join(y, join(x, s))$
regarding $Elem \times Set, Elem \times Elem \times Set, \textsc{Bool}$
end

and let be

$\textsc{Set1} = \textsc{Set}$ **then**
op $h : Elem \to Bool$ **end**

In $\textsc{Set1}$ with exception of the sort $Elem$ and the morphism $h : Elem \to Bool$ all other objects and morphism are subject of a constraint, so that their interpretations are unique up to isomorphisms.

In the generic model of $\textsc{Set1}$ one can find morphisms

$$all_h : Set \to Bool, \quad ex_h : Set \to Bool$$

such that for each model $M : \textsc{Set1} \to Set$ of $\textsc{Set}$ and each finite subset $X \subseteq M(Elem)$ hold

- $M(all_h)(X) = true$ if and only if for all $x \in X, M(h)(x) = true$, and
- $M(ex_h)(X) = true$ if and only if there is at least one $x \in X$ such that $M(h)(x) = true$.

These morphisms in the generic model are induced by the free extension and respectively by the functors

$$\begin{array}{ll} F_{all}: & \{Set \mapsto Bool,\ empty \mapsto true, \\ & join \mapsto and(h(x), y) : Elem \times Bool \to Bool\} \\ F_{ex}: & \{Set \mapsto Bool,\ empty \mapsto false, \\ & join \mapsto or(h(x), y) : Elem \times Bool \to Bool\} \end{array}$$

To justify these constructions one has to proof that the given interpretations of the morphism $join$ satisfy the required equations. But, this is easy to see, since conjunction and disjunction are both idempotent and commutative.

The next example approaches *transition systems*. There are many ways to formalize transition systems. One way is, to understand a transition system as a relational structure with a ternary relation

$$next \subseteq States \times Actions \times States$$

which describes by $\langle s_1, a, s_2 \rangle \in next$ that s_2 is a possible successor state of s_1 if the system performs the action a.

Constraining a morphism $f : A \to B$ to a monomorphism or dualy to an epimorphism can be done by pullback or pushout sketches, which state that $id_A; f = id_a; f$ respectively $f; id_B = f; id_B$ are pullback, respectively pushout diagrams. To be more precise, let be

$\textsc{Pullback} =$

sort	A, B, C
op	$f : A \to C$
	$g : B \to C$
then cofree	
sort	P
op	$p_f : P \to A$
	$p_g : P \to B$

equations

$p_f; f = p_g; g$

end

PUSHOUT =

sort A, B, C

op $f : C \to A$

$g : C \to A$

then free

sort P

op $p_f : A \to P$

$p_g : B \to P$

equations

$f; p_f = g; p_g$

end

Transition systems can now be formalized as models of the following NS-theory:

TRSYSTEMS1 =

sort $Actions, States, R, States \times Actions \times States$

op $inc : R \to States \times Actions \times States$

constraints

$States \times Actions \times States$

PULLBACK **with**$\{f \mapsto inc, g \mapsto inc, p_f \mapsto id_R, p_g \mapsto id_R\}$

end

If we are interested in the *most abstract TR-system* for any given set of actions, we have to constrain the model class using a cofree extension.

TRSYSTEMS2 = TRIV **with** $\{Elem \mapsto Actions\}$ **then cofree**

sort $States, R, States \times Actions \times States$

op $inc : R \to States \times Actions \times States$

regarding $States \times Actions \times States$

PULLBACK **with**$\{f \mapsto inc, g \mapsto inc, p_f \mapsto id_R, p_g \mapsto id_R\}$

end

However, a functor M to Set is a model of TRSYSTEMS2 iff $M(States) = \{*\}$ and $M(R) = \{*\} \times M(Actions) \times \{*\}$, i.e., the state set of the most abstract model collapses. The collaps is caused by the lack of observations for the states. This fact may be seen as a hint not to formalize transition systems as such kind of structures.

In a growing number of papers transition systems are formalized as coalgebras, see [9] and [6]. This would leed to the following specification of most abstract transition systems for given sets of actions:

$$
\begin{array}{ll}
\text{TR}\textsc{systems} = \text{T}\textsc{riv}\ \textbf{with}\{\mathit{Elem} \mapsto \mathit{Actions}\}\textbf{then cofree} & \\
\qquad \textbf{sort} & \mathit{States}, \mathit{Statesets}, \mathit{Actions} \times \mathit{States} \\
\qquad \textbf{op} & \mathit{next} : \mathit{Action} \times \mathit{States} \to \mathit{Statesets} \\
 & \mathit{empty} : 1 \to \mathit{Statesets} \\
 & \mathit{join} : \mathit{States} \times \mathit{Statesets} \to \mathit{Statesets} \\
\qquad \textbf{regarding} & \\
\qquad \text{B}\textsc{ool}, \mathit{Actions} \times \mathit{States} & \\
\qquad \text{S}\textsc{ets}\ \textbf{with}\{\mathit{Elem} \mapsto \mathit{States}, \mathit{Sets} \mapsto \mathit{Statesets}\} & \\
\qquad \textbf{end} &
\end{array}
$$

What about the collaps of the state set in that case? Even though there is no explicite observation operation in the specification TRsystems the state set does not collaps in the most abstract model. This is caused by the fact that in the most abstract model M, within the class of models in Set, two states $s_1, s_2 \in M(\mathit{States})$ are different, if there is a morphism $t : \mathit{States} \to \mathit{Bool}$ in the generic model such that $M(t)(s_1) \neq M(t)(s_2)$.

In the following we investigate what kind of observations or experiments $t : \mathit{States} \to \mathit{Bool}$ in the generic model $\mathbb{L}_{\text{TRsystems}}$ exist.

First we have the observation $()_{\mathit{States}}; \mathit{true} : \mathit{States} \to \mathit{Bool}$ which is induced by constraining the object 1 to the terminal one. Using the constraint Set in TRsystems, this observation induces the morphisms

$$\mathit{all}_{(()_{\mathit{States}};\mathit{true})} : \mathit{Statesets} \to \mathit{Bool},\ \mathit{ex}_{(()_{\mathit{States}};\mathit{true})} : \mathit{Statesets} \to \mathit{Bool}.$$

$M(\mathit{all}_{(()_{\mathit{States}};\mathit{true})}) : M(\mathit{Statesets}) \to \{\mathit{true}, \mathit{false}\}$ becomes the constant function true, but

$$M(\mathit{ex}_{(()_{\mathit{States}};\mathit{true})})(X) = \mathit{true} \text{ iff } X \neq \emptyset.$$

Therefor we obtain for any $(a, s) \in M(\mathit{Actions}) \times M(\mathit{States})$ that

$$M(\mathit{next}; \mathit{ex}_{(()_{\mathit{States}};\mathit{true})})(a, s) = \mathit{true}$$

if and only if there is at least one successor state of $s \in M(\mathit{States})$ performing the action $a \in M(\mathit{Actions})$, i.e.

$$M(\mathit{next}; \mathit{ex}_{(()_{\mathit{States}};\mathit{true})})(a, s) = \mathit{true} \text{ iff } M \models_s \langle a \rangle \mathit{true}.$$

This observation can now be used to construct other ones. To make this generation process precise we have to work in the following specification.

Let be Act be a fixed set of actions and let TRsystems[Act] be the specification

$$
\begin{array}{l}
\text{TR}\textsc{systems}[\mathit{Act}] = \text{TR}\textsc{systems}\ \textbf{then} \\
\qquad \textbf{constraints} \\
\qquad \text{I}\textsc{nf}\text{S}\textsc{um}\ \textbf{with}\ \mathit{Actions} \mapsto \sum_{a \in \mathit{Act}} 1
\end{array}
$$

$\textsc{InfSum}$ **with** $\mathit{Actions} \times \mathit{States} \mapsto \sum_{a \in \mathit{Act}} \mathit{States}$
end

The generic model of TRSYSTEMS[*Act*] contains as observations $t : \mathit{States} \to \mathit{Bool}$ all formulae of the modal logic with action set *Act*. Let for instance be $a_1, a_2 \in \mathit{Act}$ and $in_{a_1}, in_{a_2} : \mathit{States} \to \sum_{a \in \mathit{Act}} \mathit{States}$ the corresponding injections of the infinite sum, then

$$t_{\langle a_1 \rangle true} = in_{a_1}; next; ex_{(()_{States}; true)} : \mathit{States} \to \mathit{Bool}$$

represents the modal formula $\langle a_1 \rangle true$, and

$$in_{a_2}; next; all_{t_{\langle a_1 \rangle true}} : \mathit{States} \to \mathit{Bool}$$

represents the modal formula $[a_2]\langle a_1 \rangle true$.

The specifications TRSYSTEMS and TRSYSTEMS[*Act*] are related in such a way that the unique model of TRSYSTEMS[*Act*] in *Set* is isomorphic to that model M of TRSYSTEMS with $M(\mathit{Actions}) = \mathit{Act}$.

It is well known that for image finite transition systems two states are equivalent with respect to strong bisimulation if they cannot be distinguished by any formula of modal logic. Therefore, the unique model of TRSYSTEMS[*Act*] in *Set* is fully abstract with respect to strong bisimulation and can be constructed as the *canonical model* of the modal logic with action set *Act*, see [4]. A construction of the unique model of TRSYSTEMS[*Act*] as terminal coalgebra of the endofunctor,

$$P_{Act} : \mathit{Set} \to \mathit{Set} \quad \text{with} \quad P_{Act}(X) = (\mathcal{P}_{fin}(X))^{Act},$$

where $\mathcal{P}_{fin} : \mathit{Set} \to \mathit{Set}$ denotes the finitary power set functor, is given in [1] and [10].

It is this example which from our point of view justifies to understand the generic model of an NS-theory as a minimal logic representing essential properties of the specified structures. But this example demonstrates additionally, that one can not expect very interesting properties of the categories of models of an NS-theory in a given category *Sem*. It seems that there is some kind of tradeoff in the interest in the model class of an NS-theory and the generic model: If the model class has reach structure the generic model seems to be rather simple and conversely, if the model class has a simple structure (being an isomorphisms class for instance) all the more interest deserves the generic model.

An advantage of models of NS-theories over coalgebras is the possibility to deal with binary state transition functions. To demonstrate this we consider the following specification:

2-1-TREEFUN = TRIV **with cofree**

sort $St,\ St \times St$

op $head : St \to Elem$

$tail : St \to St$

$comp : St \times St \to St$

regarding $St \times St$

end

Without the product object $St \times St$ and the binary state transition operation $comp : St \times St \to St$ this specification would define the infinite streams over the freely interpretable sort $Elem$. The existence of the binary state transition operation does not cause any problems to qualify the models M_{cofree} : 2-1-TREEFUN $\to$ Set. One has just to apply the usual construction of a right Kan extension to see that

$$M_{cofree}(St) = \{f : T_{St \to Elem} \to M(Elem)\},$$

where $T_{St \to Elem}$ denotes the set of all terms with one variable of sort St that produce a value of sort $Elem$. These terms may be illustrated as trees with at most two subtrees, i.e., one or twofold branching trees.

To prove the required universal property of M_{cofree} : 2-1-TREEFUN $\to$ Set we consider the algebraic signature

2-1-TREE =

sort St

op $s_0 : 1 \to St$

$tail : St \to St$

$comp : St \times St \to St$

end

The set of terms $T_{St \to Elem}$ forms an initial algebra of 2-1-TREE, so that for each model M : 2-1-TREEFUN $\to$ Set and each $m_0 \in M(St)$ there is exactly one 2-1-TREE-homomorphism

$$f_{m_0} : T_{St \to Elem} \to M_{cofree}|_{\text{2-1-TREE}}$$

with $f_{m_0}(s_0) = m_0$ which defines a mapping

$$h_{m_0} : T_{St \to Elem} \to M_{cofree}(Elem)$$

by $h_{m_0}(t) = M(head)(f_{m_0}(t))$, i.e. by applying $M_{cofree}(head)$ to the unique value of t (als element of the initial 2-1-TREE-algebra) in the 2-1-TREE-algebra

$M_{cofree}|_{\text{2-1-TREE}}$ canonically derived from M_{cofree} by interpreting s_0 by m_0. This construction delivers the required unique homomorphism from M to M_{cofree}.

It is worth to mention, that in this and similar examples the *set of distinguishing observations* can canonically be constructed as an initial algebra. This makes it additionally possible, to use equations within the cofree constraint part of a specification, i.e. one can easily deal with *state valued equations* in specifications of processes.

References

[1] M. Barr. Terminal coalgebras in well–foundet set theory. *Theoretical Computer Science*, 114:299–315, 1993.

[2] M. Barr and C. Wells. *Category Theory for Computing Science (second edition)*. International series in computer science. Prentice Hall, 1996.

[3] J.A. Goguen and R. M. Burstall. Institutions: Abstract model theory for specification and programming. *Journal of the Association for Computing Machinery*, 39:95–146, 1992.

[4] Robert Goldblatt. *Logics of Time and Computation*. Number 7 in CSLI Lecture Notes. CSLI, Center for the Study of Language and Information, second edition, 1992.

[5] T. Hagino. *A categorical programming language*. PhD thesis, Edinburgh University, 1987.

[6] Bart Jacobs and Jan Rutten. A tutorial on (co)algebras and (co)induction. *Bulletin of the EATCS*, (62):222–259, June 1997.

[7] F.W. Lawvere. Functorial semantics of algebraic theories. *Proc. Nat. Acad. Sci. U.S.A.*, 50:869–873, 1963.

[8] CoFI Task Group on Langugage Design. Casl, the common algebraic specification language (summary). Technical report, CoFI: The Common Framework Initiative, 1998.

[9] J.J.M.M. Rutten. A calculus of transition systems (towards universal coalgebra). In A. Ponse, M. de Rijke, and Y. Venema, editors, *Modal Logic and Process Algebra, a bisimulation perspective*, volume 53 of *CSLI Lecture Notes*, pages 231–256, Stanford, 1995. CSLI Publications. FTP-available at `ftp.cwi.nl` as pub/CWIreports/AP/CS-R9503.ps.Z.

[10] J.J.M.M. Rutten. Universal coalgebra: a theory of systems. Technical Report CS-R9652, CWI, Amsterdam, 1996. to appear in Theoretical Computer Science.