$\mathcal{I}$-polynomial data types: adjunctions, equations, and theories

Peter Padawitz

Technical University of Dortmund, Germany

Abstract. $\mathcal{I}$-polynomial data types extend the well-known polynomial types in order to cover all widely used constructor-based (*white-box*) or destructor-based (coalgebraic, state-oriented, *black-box*) data types and to integrate the respective reasoning methods. Due to this goal we developed concrete representations of canonical (free or co-free) models, which reveal the duality between constructive and destructive data types more clearly than previous notions of (co)terms have done. First results of this approach pertain to *iterative equations*: unique solutions are obtained in constructive term models in almost the same way as in destructive coterm models. *Parameterized* iterative equations, however, may adequately present recursive functions in term models, but fail to present corecursive functions in coterm models. Instead, we intend to solve the equations with *function* variables in suitable *algebraic theories* that can be derived from constructor models as well as from destructor models.

Keywords: polynomial type, (co)free model, (co)term, iterative equation, algebraic theory

We only give some basic notions. Polynomial types are usually defined as images of composed set functors (see, e.g., [4]) including constant, identity, finite sum, finite product and power (also called *reader*) functors. Results on the existence and uniqueness of least or greatest fixpoints of functors (see, e.g., [1, 3]) have shown that such fixpoints are also obtained for arbitrarily indexed sums or products. This gives rise to the following definition: Let S be a set and $\mathcal{I}$ be a set of sets I (of indices) with $|I| > 1$. The set $\mathcal{T}(S,\mathcal{I})$ of **$\mathcal{I}$-polynomial types over** S is inductively defined as follows:

- $S \cup \mathcal{I} \cup \{1\} \subseteq \mathcal{T}(S,\mathcal{I})$.
- For all $I \in \mathcal{I}$ und $\{e_i \mid i \in I\} \subseteq \mathcal{T}(S,\mathcal{I})$, $\coprod_{i\in I} e_i, \prod_{i\in I} e_i \in \mathcal{T}(S,\mathcal{I})$.

Finite sum, finite product, power and word types come as special cases:

$$\begin{aligned} e_1 \times \ldots \times e_n &=_{def} \textstyle\prod_{i\in\{1,\ldots,n\}} e_i, \\ e_1 + \ldots + e_n &=_{def} \textstyle\coprod_{i\in\{1,\ldots,n\}} e_i, \\ e^I &=_{def} \textstyle\prod_{i\in I} e, \\ e^n &=_{def} e^{\{1,\ldots,n\}}, \\ e^+ &=_{def} e + \textstyle\coprod_{n>1} e^n, \\ e^* &=_{def} 1 + e^+. \end{aligned}$$

When interpreting an $\mathcal{I}$-polynomial type, we do not require a particular representation of the involved index sets, sums or products, but only *type compatibility*: A $\mathcal{T}(S,\mathcal{I})$-sorted set A, i.e., an object of the product category $Set^{\mathcal{T}(S,\mathcal{I})}$, is **type compatible** if for all $I \in \mathcal{I} \cup \{1\}$, $A_I \cong I$ and for all $\{e_i \mid i \in I\} \subseteq \mathcal{T}(S,\mathcal{I})$ there are function tuples $\pi = (\pi_i^A : A_{\prod_{i\in I} e_i} \to A_{e_i})_{i\in I}$ and $\iota = (\iota_i^A : A_{e_i} \to A_{\coprod_{i\in I} e_i})_{i\in I}$ such that $(A_{\prod_{i\in I} e_i}, \pi)$ is a product and $(A_{\coprod_{i\in I} e_i}, \iota)$ is a sum of $(A_{e_i})_{i\in I}$. Besides S and $\mathcal{I}$, a **signature** $\Sigma = (S,\mathcal{I},F)$ has a finite set F of typed function symbols $f : e \to e'$ such that $e, e' \in \mathcal{T}(S,\mathcal{I})$. f is a **constructor** if $e' \in S$ and a **destructor** if $e \in S$. Σ is **constructive** (**destructive**) if F consists of constructors (destructors) and for all $s \in S$, $\mathcal{I}$ contains the set of symbols $f : e \to e' \in F$ with $e' = s$ ($e = s$). A Σ**-algebra** $\mathcal{A} = (A, Op)$ consists of a type compatible $\mathcal{T}(S,\mathcal{I})$-sorted set A and an F-sorted set $Op = (f^{\mathcal{A}} : A_e \to A_{e'})_{f:e\to e'\in F}$ of functions.

Let V be an S-sorted set of “variables” resp. “colors”. We distinguish between five sets of (co)terms over V each of which is a subset of the set

$$Tree_\Sigma(V) = \{t : X^* \multimap\!\!\to Y \mid def(t) \text{ is prefix closed}\}$$

of trees with edge labels from $X = \bigcup \mathcal{I} \cup \{sel\}$ and node labels from $Y = \bigcup \mathcal{I} \cup V \cup \{tup\}$. Roughly speaking, *sel* selects a summand, *tup* tuples factors.

For a *constructive* signature Σ, $CT_\Sigma(V)$ and $T_\Sigma(V)$ denote the sets of all resp. all well-founded Σ-terms (V labels only leaves); $T_\Sigma(V)$ is the carrier of a free Σ-algebra over V and provides the right-hand sides of iterative Σ-equations. Dually, for a *destructive* signature Σ, $DT_\Sigma(V)$ and $coT_\Sigma(V)$ denote the sets of all resp. all well-founded Σ-coterms (V labels only inner nodes); $DT_\Sigma(V)$ is the carrier of a cofree Σ-algebra over V. The fifth set is also denoted by $T_\Sigma(V)$ because it yields – as $T_\Sigma(V)$ does for *constructive* signatures – the right-hand sides of iterative Σ-equations – which have unique solutions in $DT_\Sigma(1)$.

Due to the fixpoint property of initial (final) Σ-algebras, each constructive (destructive) signature Σ induces a destructive (constructive) signature $co\Sigma$ such that $CT_\Sigma(\emptyset)$ ($coT_\Sigma(1)$) is a final (initial) $co\Sigma$-algebra. This fact leads directly to the unique solvability of constructive Σ-equations in $CT_\Sigma(\emptyset)$ [2].

A Σ-algebra A induces an algebraic theory [5], here being a category with object set $\mathcal{T}(S,\mathcal{I})$ and morphisms in the product and sum extension closure of F. Iterative equations with function variables relate morphisms to each other and may be solvable in the theory as “first-order” equations are solvable in A.

References

1. J. Adámek, S. Milius, L.S. Moss, *Initial algebras and terminal coalgebras: a survey*, TU Braunschweig 2011
2. J. Adamek, S. Milius, J. Velebil, *Free iterative theories: a coalgebraic view*, Math. Struct. in Comp. Science (2003) 259–320
3. P. Padawitz, *From grammars and automata to algebras and coalgebras*, Proc. 4th Int. Conf. on Algebraic Informatics, Springer LNCS 6742 (2011) 21-43
4. J.J.M.M. Rutten, *Universal coalgebra: a theory of systems*, Theoretical Computer Science 249 (2000) 3-80
5. E. Wagner, *Algebraic semantics*, in: Handbook of Logic in Computer Science, Vol. 3: Semantic Structures, Clarendon Press (1994) 323-393