Semantics of Algebraic Specifications

Given an ADT specification $(\Sigma, \mathcal{E})$ with signature Σ and equations $\mathcal{E}$, we first consider the collection of ground terms, $T(\Sigma,\varnothing)$. This collection itself is a data algebra for the operations of the signature Σ. For any k-ary operation f, and $t_1, t_2, \ldots, t_k \in T(\Sigma,\varnothing)$ whose sorts correspond to those required by f, we have that f applied to $t_1, t_2, \ldots, t_k$ yields the term $f(t_1, t_2, \ldots, t_k) \in T(\Sigma, \varnothing)$. This is called the **term (or free) algebra** over Σ.

Each constant is the name of some object of its sort. Each composite expression (or term) that involves no variable also denotes some object of its sort. Together these are the *ground* terms of the sort. Since a sort which contains no objects at all (and hence no usable operations) is effectively not involved in any behavior, it makes no sense to allow this in specifications, and we normally prohibit this.

Definition: an ADT is **sensible** if each sort has at least one ground term.

It is normally assumed that each ADT of interest is sensible, and the examples we present should exhibit this property. Not only do we wish to prohibit empty sorts, but we want to exclude the opposite extreme. The presence of potentially aberrant objects in the sort that have no role in any operation outcome and are not denoted by any ground term are also excluded.

Definition: The axiom of **no junk** for an ADT asserts that every object of every sort is equivalent to some ground term of the ADT.

We assume the axiom of no junk for our ADTs.

Definition: In an ADT, the collection of operations whose range is a certain sort S are called the **generators of S**. We sometimes identify a minimal subset of the generators called **constructors** with the property that every ground term of sort S is equivalent to some ground term involving only the constructors of sort S. That is, every object of type S is produced by some combination of the constructor operations.

By the “no junk” axiom, the ground terms involving the generators must yield all the objects of the sort. But it may be that even when we omit some of these operations, all the objects still obtain. For instance, in the Stack ADT, NEW, PUSH, and POP are generators, and NEW and PUSH serve as constructors — all stack objects can be produced using only these operations.

Initial algebra semantics

Two terms $t_1, t_2 \in T(\Sigma)$ are **equivalent with respect to $\mathcal{E}$**, $t_1 \equiv_{\mathcal{E}} t_2$, provided that $t_1 = t_2$ can be deduced from the equations $\mathcal{E}$ using the properties of an equivalence relation and substituting equals for equals. For $t \in T(\Sigma)$, let $[t]_{\mathcal{E}} = \{t' \in T(\Sigma) \mid t \equiv_{\mathcal{E}} t'\}$ denote its equivalence class. We omit the subscript indicating the equations $\mathcal{E}$ when this is understood from context.

The collection of equivalence classes of $T(\Sigma, \emptyset)$ under $\equiv_{\mathcal{E}}$ also forms a data algebra over Σ, called the **initial algebra** of $(\Sigma, \mathcal{E})$, and written $T(\Sigma, \emptyset)/\equiv_{\mathcal{E}}$. The application of an operation f to classes $[t_1], [t_2], \ldots, [t_k] \in T(\Sigma, \emptyset)/\equiv_{\mathcal{E}}$ yields the result class

$f([t_1], [t_2], \ldots, [t_k]) = [f(t_1, t_2, \ldots, t_k)]$.

Final (or terminal) algebra semantics

Two terms $t_1, t_2 \in T(\Sigma)$ are **distinguishable with respect to $\mathcal{E}$**, provided there is some other term t which involves a variable, say x, and whose sort is pre-defined (i.e., not the TOI) so that $t[x \rightarrow t_1] \not\equiv_{\mathcal{E}} t[x \rightarrow t_2]$, where $t[x \rightarrow t']$ denotes the result of substituting term t' in t in place of each occurrence of x. If two terms are not distinguishable, they are said to be **indistinguishable**, and we write $t_1 \approx_{\mathcal{E}} t_2$. Clearly $\approx_{\mathcal{E}}$ is an equivalence relation. For $t \in T(\Sigma)$, let $\ll t \gg_{\mathcal{E}} = \{t' \in T(\Sigma) \mid t \approx_{\mathcal{E}} t'\}$ denote the indistinguishability class of t. Again we omit the subscript indicating the equations $\mathcal{E}$ when this is understood from context.

The collection of indistinguishability classes of $T(\Sigma, \emptyset)$ under $\approx_{\mathcal{E}}$ also forms a data algebra over Σ, called the **final algebra** of $(\Sigma, \mathcal{E})$, and written $T(\Sigma, \emptyset)/\approx_{\mathcal{E}}$. The application of an operation f to classes $\ll t_1 \gg, \ll t_2 \gg, \ldots, \ll t_k \gg \in T(\Sigma, \emptyset)/\approx_{\mathcal{E}}$ yields the result class

$f(\ll t_1 \gg, \ll t_2 \gg, \ldots, \ll t_k \gg) = \ll f(t_1, t_2, \ldots, t_k) \gg$.

Loose semantics

A data algebra A is a **model** of a specification $(\Sigma, \mathcal{E})$ provided that

- for each sort s of Σ, there is a set A_s (the **carrier** of sort s),
- the operations of Σ correspond to the functions of A with respect to the sorts and carrier sets, so that all the equations of $\mathcal{E}$ are satisfied in A, and
- the carriers have no *junk*.

The **loose semantics** of $(\Sigma, \mathcal{E})$ is the collection of all its models.