# Implementing Algebraic Specifications

Since our principal goal is to provide a precise specification of the behavior of implementations. It is vital to have general agreement on what is expected of an implementation in order that it satisfies a specification. This description will necessarily treat implementations in a generic way since there are no assumptions made about what programming language and data structures are used. It might seem that this makes a discussion of the topic impossible, but in fact it simplifies it.

Indeed, the intention of a specification is that it should provide all we need to know to write client code using it. That being the case, we can write the programs for one ADT that *realize* the operations of another ADT. It turns out to be useful to express a formal definition in terms that regard both the specification *and* the implementation as abstract data types.

*Definition*: a **realization** of ADT $T_1$ by ADT $T_2$ is a pair of *representation functions*

$\rho$: TOI -objects($T_2$) $\rightarrow$ TOI-objects($T_1$)

$\pi$: programs ($T_2$) $\rightarrow$ operations($T_1$)

with the properties that $\rho$ is a total onto function, $\pi$ is a partial 1-1 function, and for the composition functions, $\rho \circ \pi = \pi \circ \rho$ (this is the isomorphism property used earlier).

Notice that the orientation of these functions means that the ADT to be realized is the image. This orientation admits the possibility that there may be multiple representations of an object to be realized the assumption being that any of the representations work equally well. The sense of the commuting property of the two representation functions is that any computation of $T_1$ can effectively be carried out in $T_2$ by using corresponding objects and operations and then translating back to $T_1$. For any representative x in $T_2$ of the object $\rho(x)$ in $T_1$ and program P associated with operation $f = \pi(P)$ in $T_1$ (or in fact any series of programs and their associated series of operations), $\rho(P(x)) = f(\rho(x))$ the term on the left denotes performing a computation (program) in $T_2$ and then accessing the representation of the result, and the term on the right denotes first accessing the representation to obtain an object in $T_1$ and then performing the corresponding operation there. That is, we can perform the associated program on the representation of an object and the outcome will always exactly represent the result of performing the desired operation on the original object.

With realization defined in this general way, we may envision the realization of one ADT by another ADT. We may also evaluate a concrete implementation in a similar way identify corresponding representations, and chek the programs for this communtation property in the ADT.