Re-Using Specifications

Re-using code is an essential part of modern programming. If our program needs to sort an array, we don't write our own sorting procedure. Instead, we re-use a procedure written by someone else—probably as part of a standard library. It therefore seems obvious that we should re-use specifications.

However, what is obviously true for programming is not necessarily true for writing specifications. A mathematical description of something is much shorter and simpler than a program to compute it efficiently. If a library procedure doesn't do exactly what a programmer wants, she will probably modify her program to use the existing procedure rather than trying to modify the existing procedure. The opposite is generally the case with specifications.

As an example, consider graphs. There are many different kinds of graphs. A graph may be directed or undirected, with or without edges that go from a node to itself, with or without nodes having no incoming or outgoing edge, and so on. A programmer will use any graph package that handles a class of graphs sufficiently general enough to include the ones used by her program. However, we usually obtain the simplest specification by defining the exact class of graphs that we need. The power of mathematics almost always makes this easy to do.

For specifications, the best form of re-use is most often *copy*, *paste*, and *modify*.

The definitions in many of the standard modules are also short and simple. However, the TLC and TLAPS tools treat some of them specially. In particular, it would be impossible to write TLA$^+$ definitions that TLC could execute for many of the operators defined in the standard modules. In fact, for efficiency, the standard modules *Naturals*, *Integers*, and *Reals* used by the tools are not the ones defined in *Specifying Systems*.

While it's easy to modify specifications, it's not so easy to modify proofs. If, in addition to defining graphs, we have also written TLAPS-checked proofs of their properties, then we want to avoid changing our definitions and re-proving those properties. Specifications are worth re-using if they contain proofs.