Some Misc Notes on Chapter 3

In order to perform specification validation, building a functional interpreter is the first step. Such an interpreter is useful in its own right, i.e., apart for the specific idea of validating preconditions and postconditions.

For testing the logic of a "no junk, no confusion" rule, change the logical ’and’ to an ’or’ to see what happens.

Questions of the nature "is this strong enough", or "does this cover the case" may be answerable with validation invocations. E.g., "Does the no-junk,-no-confusion logic address the issue of duplicates, as in does it make any difference how many copies of a record there are?" This can be tested by supplying an input with duplicates to a validation call.

An important point to make about validation calls is that their purpose is to help answer specific questions about the logic of a specification, not necessarily test it exhaustively. In many cases, the strength of a specification can be invalidated with a simple counter example. The result of such a simple, incremental test case is to inform the specifier of a weakness or flaw in the logic, that then be addressed. An important contribution of a validation test case is to help focus the specifier on problems that may be difficult to see by purely human inspection of the specification logic. As well, an incremental validation invocation can be substantially quicker than waiting for the results of a complete model check or other form of correctness proof.