Functions versus Operators

When you define an operator Op by writing something like

$$Op(a) \quad \triangleq \quad a + 42$$

this defines $Op(e)$ to equal $e+42$ for any value e. For example, it defines $Op(1/2)$ to equal $(1/2) + 42$, and it defines $Op(\text{“}\mathsf{abc}\text{”})$ to equal “abc” $+ 42$, which is a nonsensical expression whose value we know nothing about.

Defining a function specifies its value only for elements of its domain. For example, either of the two equivalent definitions

$$fcn \quad \triangleq \quad [a \in Int \mapsto a + 42]$$

$$fcn[a \in Int] \quad \triangleq \quad a + 42$$

defines $fcn[e]$ to equal $e + 42$ only if e is an element of the domain of fcn, which is the set Int of integers. It tells us nothing about the value of $f[1/2]$ or $f[\text{“}\mathsf{abc}\text{”}]$; both of these are nonsensical expressions.

The function fcn by itself is a legal expression. The expression $fcn + 1$ is a syntactically legal (but nonsensical) expression. On the other hand, $Op + 1$ is not syntactically legal and produces a parsing error.

See Section 6.4 of *Specifying Systems* for a more extensive discussion of the difference between functions an operators.

CLOSE