The EXCEPT Notation

Don't try to make any sense of the EXCEPT and the !. They are meaningless pieces of syntax. Just remember that

$$[fcn \text{ EXCEPT } ![a] = d]$$

is the value that a function *fcn* has after executing the assignment statement

$$fcn[a] := d$$

(assuming that a is in the domain of *fcn*). Since assignments to arrays are common in algorithms, we need a simple way of writing this function. Mathematics doesn't provide it, so I had to invent one.

The notation has some useful generalizations—for example,

$$[fcn \text{ EXCEPT } ![a][b] = d]$$

is the value of *fcn* after executing the assignment

$$fcn[a][b] := d$$

and

$$[fcn \text{ EXCEPT } ![a] = d, ![b] = e]$$

is the value of *fcn* after executing the two assignment statements

$$fcn[a] := d;\ fcn[b] := e$$

You understand the notation if you understand that:

$$[fcn \text{ EXCEPT } ![a] = d, ![b] = e] = [\,[fcn \text{ EXCEPT } ![a] = d] \text{ EXCEPT } ![b] = e]$$

$$[fcn \text{ EXCEPT } ![a][b] = d] = [fcn \text{ EXCEPT } ![a] = [fcn[a] \text{ EXCEPT } ![b] = d]\,]$$

Records (also known as *structs* in C) are represented in TLA$^+$ as functions, and the value of record R after executing $R.d := e$ is

$$[R \text{ EXCEPT } !.d = e]$$

These notations can be combined, as in

$$[B \text{ EXCEPT } ![i].d[j] = e]$$

The EXCEPT notation looks weird, and no one likes it—including me. However, I've found no alternative that I like better. One can devise a more compact notation by replacing the "EXCEPT" with some punctuation, but I think that would make it even more obscure. In time, you'll get used to it.

CLOSE