A note.

As we learned earlier, suppose that $\oplus$ is a binary operator which is associative and has an identity element, then the following expression

$$f.0 \ \oplus \ f.1 \ \oplus \ f.2 \ \oplus \ldots\ldots\ldots \oplus \ f.(N-1)$$

can be written as

$$\langle \oplus j : 0 \leq j < N : f.j \rangle$$

in this form it is said to be in quantified form. There are 2 exceptions to this which we now describe. Suppose we have an array f[0..100) of int and we want to state that all of the elements in f are even. Writing it the long way would give us

$$even.f.0 \ \wedge \ even.f.1 \ \wedge \ even.f.2 \ \wedge \ \ldots.. \wedge \ even.f.99$$

and this can be written as

$$\langle \wedge j : 0 \leq j < 100 : even.f.j \rangle$$

Similarly, if we wanted to state that at least one of the values in f was equal to 14 we would write

$$f.0 = 14 \ \vee \ f.1 = 14 \ \vee \ f..2 = 14 \ \vee \ \ldots\ldots \vee \ f.99 = 14$$

and this can be written as

$$\langle \vee j : 0 \leq j < 100 : f.j = 14 \rangle$$

However, large numbers of the Computer Science community really didn't want to move away from the old style notation and so for both of these examples many still wanted to write

$$\langle \wedge j : 0 \leq j < 100 : even.f.j \rangle$$

as

$$\langle \forall j : 0 \leq j < 100 : even.f.j \rangle$$

and to write

$$\langle \vee j : 0 \leq j < 100 : f.j = 14 \rangle$$

as

$$\langle \exists j : 0 \leq j < 100 : f.j = 14 \rangle$$

You can choose whichever you prefer.