and then

```
M[I][J]
```

denotes the component `J` (of type `T`) of component `I` of `M`.

For multidimensional arrays, it is customary to make these convenient abbreviations:

```
var M: array [A..B,C..D] of T;
```

and

```
M[I,J]
```

We may regard `M` as a matrix and say that `M[I,J]` is component `J` (in column `J`) of component `I` of `M` (of row `I` of `M`).

Arrays are not limited to two dimensions, for `T` can again be a structured type. In general, the (abbreviated) form is:

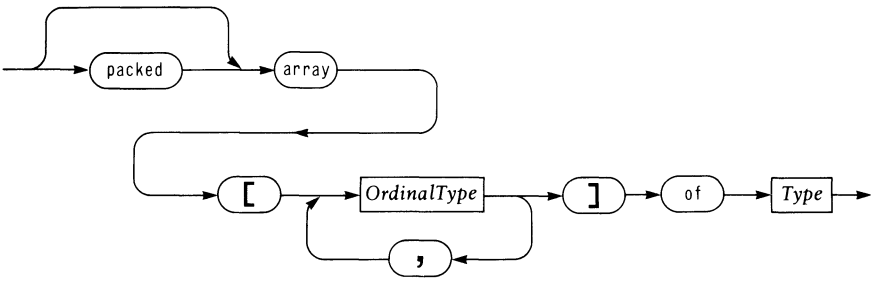

Figure 6.d Syntax diagram for *ArrayType*

If `n` index types are specified, the array is said to be n–dimensional, and a component is denoted by `n` index expressions.

If `A` and `B` are array variables of the same type, then the assignment statement

```
A := B
```

is allowed if the arrays are component–wise assignable:

```
A[i] := B[i]
```

(for each `i` that is a value of the index type), and is an abbreviation for the assignment of each corresponding component.