Notre Dame Journal of Formal Logic
Volume XVI, Number 3, July 1975
NDJFAM

NOTE ON THE USE OF SEQUENCES IN 'LOGICS AND LANGUAGES'

M. J. CRESSWELL

The purpose of this note is to remedy a defect in the set-theoretical basis of my book *Logics and Languages* [1]. In most accounts of formal languages, expressions are regarded as strings of symbols. Thus in the propositional calculus

(1) $((p \supset q) \supset r)$

is a string whose first and second members are (, whose third member is p and so on. Two things are not usually made clear. First, what a string is; and second, what (and p and so on are. In [1] I tried to tackle these problems by giving a purely set-theoretical analysis of the expressions of a formal language. It has, however, been drawn to my attention that there are certain infelicities, if not actual errors, in my definitions, and it is the purpose of this note to remedy the major one of these and, hopefully, arrive at an adequate statement of a set-theoretical representation of expressions.

Beginning with the standard notions of set theory we define the *ordered pair* (x, y) as $\{\{x\}, \{x, y\}\}$. We now define a *relation* as a class of ordered pairs and a *function*, f, as a relation such that, if $(x, y) \epsilon f$ and $(x, z) \epsilon f$, then $y = z$. We let $f(x)$ denote the unique y such that $(x, y) \epsilon f$. The *domain* of f is the set of all x such that for some y, $(x, y) \epsilon f$, and the *range* of f is the set of all y such that for some x, $(x, y) \epsilon f$.

A *sequence* is a function whose domain is an initial segment of the natural numbers. We shall take the natural numbers to be the set $\{1, 2, 3, \ldots\}$ and not to include 0. Sequences can be finite or infinite. If i is a natural number and x is a sequence[1] then $x(i)$ is usually written x_i. If x is an infinite sequence then its domain is the whole of the natural numbers.

1. I am using the letters 'x', 'y', etc. to range quite generally over classes. Thus they range *inter alia* over functions. The letters 'f', 'g', etc. are usually reserved for functions only, but the important point is that functions are merely classes of a special kind. When x is a function, x_i denotes its value for the argument i.

Received July 26, 1974

If x is a finite sequence then there is some natural number n such that i is in the domain of x iff $1 \leqslant i \leqslant n$. In the second case we can call x an *n-membered* sequence or an *n-tuple.* This latter name is not quite standard, for reasons I shall explain, but it is the sense in which the word 'n-tuple', as it occurs in [1], is to be understood. Where x is an n-tuple we frequently write it as

(2) $\langle x_1, \ldots, x_n \rangle$

$x_1, \ldots, x_n$ are not strictly members (in the set-theoretical sense) of the n-tuple; rather they are members of the range of the n-tuple. However, as was done in [1], p. 242 it is very convenient to speak of them as, in a sense, members. Numerical subscripting is not necessary in representing n-tuples. We can have

(3) $\langle x, y, z \rangle$

(3) is to be understood as the function f such that $f_1 = x$, $f_2 = y$ and $f_3 = z$. (Remember that f_i abbreviates $f(i)$.) In fact the full set-theoretical analysis of (3) would be

(4) $\{(1, x), (2, y), (3, z)\}$

which in turn is

(5) $\{\{\{1\}, \{1, x\}\}, \{\{2\}, \{2, y\}\}, \{\{3\}, \{3, z\}\}\}$

Notice that (4) is the same set as, e.g.,

(6) $\{(2, y), (3, z), (1, x)\}$

The fact that x comes first has nothing to do with the metalinguistic representation of the set in (4). It still comes first in (6), because it is still associated with the first natural number.

The only feature of the present account which is not standard elementary set theory is the use of the name 'n-tuple' and the angle-bracket notation for sequences. The notation $\langle x, y \rangle$ is frequently used for ordered pairs. In the present note $\langle x, y \rangle$ is a 2-tuple, *viz.* a function f such that $f(1) = x$ and $f(2) = y$. A 2-tuple must be distinguished from an ordered pair, and where there is need to have a notation for both of them there is a certain arbitrariness in selecting '$\langle x, y \rangle$' as the notation for a 2-tuple and '(x, y)' as the notation for an ordered pair. The infelicity in [1] was simply that the importance of the distinction was not mentioned (because it was not appreciated), and the distinction itself was only alluded to in a footnote (p. 242n). Although it was not made clear in [1], the angle-bracket notation used there must be interpreted as indicating finite sequences. This means that the occurrences of '$\langle x, y \rangle$' and the like on pp. 242-244 should be replaced by '(x, y)', and the definition of an n-tuple on p. 242 should be ignored. In addition the notion of an n-place function will need abandoning. In this case it is probably best to treat an n-place operation as a function whose domain consists of n-tuples.

The point of all this complication is to ensure a distinction between, e.g.,

(7) $\langle x, y, z\rangle$
(8) $\langle\langle x, y\rangle, z\rangle$

and

(9) $\langle x, \langle y, z\rangle\rangle$

(7) if $\{(1, x), (2, y), (3, z)\}$ (where of course $(1, x)$ is $\{\{1\}, \{1, x\}\}$ and so on). (8) is $\{(1, \{(1, x), (2, y)\}), (2, z)\}$, and (9) is $\{(1, x), (2, \{(1, y), (2, z)\})\}$. All these are distinct. This is why parentheses are not needed as separate symbols of the formal languages introduced in [1].

The idea behind most formal languages seems to be that their expressions are simply sequences of symbols. Thus the two formulae:

(10) $((p \supset q) \supset r)$

and

(11) $(p \supset (q \supset r))$

would be the sequences

(12) $\langle (, (, p, \supset, q,), \supset, r\rangle$

and

(13) $\langle (, p, \supset, (, q, \supset, r,),),\rangle$

(12) and (13) are both sequences of symbols and the parentheses are also symbols, though it is possible to dispense with the parentheses if we always put the functor before its arguments, as in the Łukasiewicz notation. However, in [1] expressions are not sequences of symbols. E.g., (10) would be

(14) $\langle\langle p, \supset, q\rangle, \supset, r\rangle$

and (11) would be

(15) $\langle p, \supset, \langle q, \supset, r\rangle\rangle$

It is important to see how (14) and (15) differ from (12) and (13). (12) is an eight-membered sequence of symbols. (14), on the other hand, is a three-membered sequence, whose first member is itself a three-membered sequence of symbols, and whose second and third members are symbols. An analogous distinction can be drawn between (13) and (15).

It is perhaps a question that could be debated whether the (12)/(13) procedure is preferable to the (14)/(15) procedure. I am myself inclined to defned the practice adopted in [1] of the (14)/(15) type of analysis (mainly on the ground that the role of the parantheses in formal languages seems to be to indicate the proper grammatical analysis rather than to function as symbols of the language); but the important point is first to be alive to the difference, and second, to realize that both use the set-theoretical notion of a sequence. As far as the nature of the symbols is concerned it was stressed in [1] that they could be anything at all. It should by now be clear what sorts of things the complex expressions would then become, given of course appropriate formation rules (as, e.g., for the languages in [1]). It should also be clear why no 'concatenation' operation is needed. I trust that this brief note makes clear what may have appeared a somewhat puzzling practice in the metalinguistic representation in [1] of expressions of formal languages.

REFERENCE

[1] Cresswell, M. J., *Logics and Languages*, Methuen, London (1973).

Victoria University of Wellington
Wellington, New Zealand