CORRECTION TO 'NATURALIZING INTENTIONS'

In the last part of my (Nelson, 1984) I argue that probabilistic automata do not provide useful models of perceptual phenomena in which the subject takes a perfectly distinct, nondegraded sensory pattern of type Q to be of a perceptual type P in order to fulfill expectations. A pattern y can resemble a pattern z of type P more than x does, while the likelihood of x fulfilling the subject's expectations of being of type P might be greater than the likelihood of y being so.

The general line of argument including the interpretation of it on page 196 I believe to be right, although the technical details are faulty. In particular my assertion, page 195, that the longer an input string x the "lower the value $p(x)$ (roughly)" of the probability of it being accepted by an automaton P , is not correct, even roughly. There are cases where it might be higher. The trouble is that the value of $p(x)$ is a product of sums of probabilities, not of products absolutely.

The necessary changes are as follows. The definition of a *winner*, page 198, is supposed to capture the idea of a state q for which there is a string y and a 2-ary map from (q, y) to probabilities (see Appendix) that indicate acceptance. This requires extension of the transition map M from symbols S to strings on S . Then suppose that an automaton P goes from the initial state to q_i under input x during a computation at t ; i.e., that $M(q_0, x) = q_i$ (alternatively, $a^x = 1$). Then we have

q_i is a *winner* if and only if there is a y such that $p(y) > \lambda$, where $p(y) = \sum_{j \in J} a_{ij}^y$, and J is the set of indices on the final states Q' (see Appendix).

Next, delete the remark, page 196, which is superfluous.

Finally, replace the discussion in the two paragraphs beginning on page 195 "The next step..." and ending on page 196 "... the foregoing argument" by the following:

The next step is to formulate the new condition suggested by our intuitions about resemblance and expectation. We suppose a string z is

accepted by P while x and y are not. The subject is to take x to be z at some time t during the computation if x is sufficiently close to z . In particular if x is closer to z than y is, then put z for x , but not for y . In our model, in a computation at t , this rule is

$$\begin{aligned} &\text{If } C(x, z) > C(y, z) \text{ then } K_P(x, z, t) \\ &\text{If } C(y, z) > C(x, z) \text{ then } K_P(y, z, t) \end{aligned}$$

Now a condition for applying this rule must be that if z is put for x rather than y , the probability of x being accepted by P must be greater than or equal to that of y being so taken. This is condition (4):

$$(4) \quad \text{If } C(x, z) > C(y, z) \text{ then } p(x) \geq p(y) \text{ must hold.}$$

One can see from the definitions of the Appendix that (4) is not satisfiable by a probabilistic model of perception. For consider the automaton $P = \langle \{0, 1\}, \{q_0, q_1, q_2\}, q_0, M, \{q_0, q_1\} \rangle$ where the transition map M is given by the probability matrices

		0					1		
		q_0	q_1	q_2			q_0	q_1	q_2
M :	q_0	0.1	0.4	0.5	q_0	0	0.7	0.3	
	q_1	0.2	0.4	0.4	q_1	0.7	0	0.3	
	q_2	0.6	0.4	0	q_2	0.1	0.1	0.8	

Here, for example $M(q_0, 0) = (0.1, 0.4, 0.5)$, which means that in the initial state q_0 with input 0 the automaton transits to state q_0 with probability 0.1, to q_1 with probability 0.4, and to q_2 with probability 0.5. With input 1 it goes to the indicated states with probabilities 0, 0.7, and 0.3, respectively; and so forth. The set of final or accepting states is $\{q_0, q_1\}$. We specify that the acceptance point $\lambda = 0.75$.

Now consider the three input strings ("patterns") $x = 100$, $y = 011$, and $z = 010$. The A -matrices corresponding to these strings (see Appendix) are

$$\begin{aligned} &0.28 \quad 0.40 \quad 0.32 \\ A(x = 100) &= 0.32 \quad 0.40 \quad 0.29 \\ &0.19 \quad 0.40 \quad 0.42 \\ &0.39 \quad 0.40 \quad 0.21 \\ A(z = 010) &= 0.37 \quad 0.40 \quad 0.23 \\ &0.29 \quad 0.40 \quad 0.31 \end{aligned}$$

$$\begin{array}{rcccl}
 & 0.14 & 0.29 & 0.58 & \\
 A(y = 011) = & 0.18 & 0.27 & 0.55 & \\
 & 0.28 & 0.42 & 0.30 &
 \end{array}$$

(If we let the input symbols 0, 1 stand for names of their associated matrices in the M table, $A(x = 100)$ is just the matrix product $1 \times 0 \times 0$, and similarly for the others).

From the given input strings we readily calculate the closeness functions $C(x, z) = 1/3$, and $C(y, z) = 2/3$, so y is closer to z . We read-off from the matrices the following probabilities of the several strings: $p(x) = 0.28 + 0.40 = 0.68$ (by the Appendix, this is the sum of probabilities of string x taking the automaton from the initial state to either one of the final states q_0, q_1 ; since P starts in the initial state, the associated matrix elements are a_{00} and a_{01} , which have just the values shown in the calculation; $p(y) = 0.14 + 0.29 = 0.43$. We note that $p(z) = 0.79$, indicating it is an accepted string.

From these calculations we have that $C(y, z) > C(x, z)$ while $p(x) > p(y)$, contrary to condition 4. So there are cases such that a string is, by a reasonable measure, closer to a 'fit' than another but has a lower probability of acceptance.

Of course this holds only for our nearness function C .

A moral that might be drawn from this note is that computational and programming models, which are alluded to frequently in the literature of cognitive science and philosophy of mind, must be handled with great care. The issues are extremely complex even though the models are relatively elementary and finitistic.

REFERENCE

Nelson, R. J.: 1984, 'Naturalizing Intentions', *Synthese* **61**, 173–203.

Department of Philosophy
Case Western Reserve University
Cleveland, OH 44106
U.S.A.