TECHNICAL CORRESPONDENCE

THE CONCEPT OF SUPERAUTOMATON

A recent review of my book *The Logic of Mind* in this journal refers to the key idea of the book, that of a superautomaton, as a "Moore machine". However, none of the central arguments of the book go through for Moore Machines. This note presents a sketch of the correct construction.

In his review of *The Logic of Mind* (Nelson 1982) in this journal (Vol. 11, no. 1), David Israel correctly identifies the idea of superautomaton as the key theoretical tool I use in attempting to explicate intentional terms of psychology such as **take, expectation,** and **belief.** However, his characterization of a superautomaton as a variety of Moore machine (Moore 1956) is very misleading. Inasmuch as the concept is central to the main argument of the book, I would like to describe it here in enough detail to cover the idea I really intended.

I specify, but do not offer a design or model of, an executive Turing machine T′ that (a) comprehends a finite number of finite automata connected in parallel, which it monitors; (b) has access to a stored encoded table representing the transition functions of each component automaton T; (c) includes means for deciding whether a given state of a component automaton can reach a final state. This complex device T′ is a "superautomaton".

The way it works is this. If an input string x to a component automaton T includes undefined (vague, degraded, or unclear) symbols u, then when T reach u it ceases processing. T′ decides whether there is a string y that could drive T to a final state. If not, it rejects x as not acceptable to T. If there is a string, T′ consults the table of T and determines by random choice a symbol s defined for T that drives T to a state for which there is a string leading to a final state. Then the undefined symbol u is *taken* to be s, and the computation of the string x continues.

Given the indicated resources T′ can *take* ill-defined, fuzzy input to be such as to satisfy *expectations* of the system. "Expectation" as well as other intentional concepts at the perceptual level are all analyzable in terms of ordinary logic operations, the indicated construction of T′, and standard mathematical machine theory.

(c) is equivalent to means for solving the halting problem; this entails that the component automata (which could be as complex as pushdown automata) must be less than full Turing machines, for which the halting problem is recursively unsolvable. It also entails that the executive part of T′ must be, in terms of competence, a two-way tape Turing machine, not a Moore machine. (In terms of *performance,* of course, one would be limited in the real world to Turing machines that are approximated by brains or digital computers, i.e., by finite sequential machines; but this is of little theoretical moment.)

Beyond the specification (a)–(c) and a program-like description of the function of T′ (Nelson 1976), I do not pretend to know what T′ would look like. By the recursion theorem of mathematical logic (Rogers 1967), some such thing must exist – i.e., there are self-describing Turing machines. There are also concrete analogous instances, i.e., generic codes.

I think this kind of idea is significantly relevant to computational theory and cognitive science, not just to the concerns of my book (which is meant to be a philosophical argument for the plausibility of computationalist theories of mind and cognition), but also to the very pervasive current employment of *self-reference* in cognitive science and artificial intelligence. My version, of course, is not strictly new as it is an adaptation of the insights of others (Lee 1963, von Neumann 1966), all of which stem from Goedel's work (1931) on the incompleteness of arithmetic.

R. J. Nelson
Department of Philosophy
Case Western Reserve University
Cleveland, OH 44106

REFERENCES

Goedel, Kurt 1931 Uber Formal Unentscheidbare Satze der Principia Mathematica und Verwandter Systeme I. *Monatsheft fur Mathematik und Physik* 38: 173-198.

Lee, C.Y. 1963 A Turing Machine which Prints its own Code Script. In Fox, Jerome, Ed., *Proceedings of the Symposium on Mathematical Theory of Automata.* Brooklyn Polytechnic Press, Brooklyn, New York: 155-164.

Moore, E.F. 1956 Gedanken Experiments on Sequential Machines. In Shannon, Claude E. and McCarthy, John, Eds., *Automata Studies.* Princeton Press, Princeton, New Jersey: 129-153.

Nelson, R.J. 1976 On Mechanical Recognition. *Philosophy of Science* 43(1): 24-52.

Nelson, R.J. 1982 *The Logic of Mind.* D. Reidel Publishing Co., Dordrecht, Holland.

Rogers, H. Jr. 1967 *Theory of Recursive Functions and Effective Computability.* McGraw-Hill Book Company, New York, New York.

von Neumann, J. 1966 *Theory of Self-Reproducing Automata.* (Burks, Arthurs W., Ed.) University of Illinois Press, Urbana, Illinois.