Calderwood, B., G. A. Klein, and B. W. Crandall. (1988). Time pressure, skill, and move quality in chess. *American Journal of Psychology* 101: 481–493.

Charness, N. (1989). Expertise in chess and bridge. In D. Klahr and K. Kotovsky, Eds., *Complex Information Processing: The Impact of Herbert A. Simon.* Hillsdale, NJ: Erlbaum, pp. 183–208.

Chase, W. G., and H. A. Simon. (1973). Perception in chess. *Cognitive Psychology* 4: 55–81.

Frey, P. W., and P. Adesman. (1976). Recall memory for visually presented chess positions. *Memory and Cognition* 4: 541–547.

Freyhoff, H., H. Gruber, and A. Ziegler. (1992). Expertise and hierarchical knowledge representation in chess. *Psychological Research* 54: 32–37.

Fürnkranz, J. (1996). Machine learning in computer chess: the next generation. *International Computer Chess Association Journal* 19: 147–161.

Gobet, F., and H. A. Simon. (1996a). The roles of recognition processes and look-ahead search in time-constrained expert problem solving: evidence from grandmaster level chess. *Psychological Science* 7: 52–55.

Gobet, F., and H. A. Simon. (1996b). Recall of rapidly presented random chess positions is a function of skill. *Psychonomic Bulletin and Review* 3: 159–163.

Goldin, S. E. (1978). Effects of orienting tasks on recognition of chess positions. *American Journal of Psychology* 91: 659–671.

Hartston, W. R., and P. C. Wason. (1983). *The Psychology of Chess.* London: Batsford.

Newell, A., and H. A. Simon. (1972). *Human Problem Solving.* Englewood Cliffs, NJ: Prentice-Hall.

Pitrat, J. (1977). A chess combinations program which uses plans. *Artificial Intelligence* 8: 275–321.

Robbins, T. W., E. Anderson, D. R. Barker, A. C. Bradley, C. Fearnyhough, R. Henson, S. R. Hudson, and A. D. Baddeley. (1995). Working memory in chess. *Memory and Cognition* 24: 83–93.

Simon, H. A. (1979). *Models of Thought, vol. 1.* New Haven: Yale University Press.

Simon, H. A., and M. Barenfeld. (1969). Information processing analysis of perceptual processes in problem solving. *Psychological Review* 76: 473–483.

55

Chinese Room Argument

The Chinese room argument is a refutation of strong artificial intelligence. "Strong AI" is defined as the view that an appropriately programmed digital computer with the right inputs and outputs, one that satisfies the Turing test, would necessarily have a mind. The idea of Strong AI is that the implemented program by itself is constitutive of having a mind. "Weak AI" is defined as the view that the computer plays the same role in studying cognition as it does in any other discipline. It is a useful device for simulating and therefore studying mental processes, but the programmed computer does not automatically guarantee the presence of mental states in the computer. Weak AI is not criticized by the Chinese room argument.

The argument proceeds by the following thought experiment. Imagine a native English speaker, let's say a man, who knows no Chinese locked in a room full of boxes of Chinese symbols (a data base) together with a book of instructions for manipulating the symbols (the program). Imagine that people outside the room send in other Chinese symbols which, unknown to the person in the room, are questions in Chinese (the input). And imagine that by following the instructions in the program the man in the room is able to pass out Chinese symbols that are correct answers to the questions (the output). The program enables the person in the room to pass the Turing test for understanding Chinese, but he does not understand a word of Chinese.

The point of the argument is this: if the man in the room does not understand Chinese on the basis of implementing the appropriate program for understanding Chinese, then neither does any other digital computer solely on that basis because no computer, qua computer, has anything the man does not have.

The larger structure of the argument can be stated as a derivation from three premises.

1. Implemented programs are by definition purely formal or syntactical. (An implemented program, as carried out by the man in the Chinese room, for example, is defined purely in terms of formal or syntactical symbol manipulations. The notion "same implemented program" specifies an equivalence class defined purely in terms of syntactical manipulations, independent of the physics of their implementation.)
2. Minds have mental or semantic contents. (For example, in order to think or understand a language you have to have more than just the syntax, you have to associate some meaning, some thought content, with the words or signs.)
3. Syntax is not by itself sufficient for, nor constitutive of, semantics. (The purely formal, syntactically defined symbol manipulations don't by themselves guarantee the presence of any thought content going along with them.)

Conclusion: Implemented programs are not constitutive of minds. Strong AI is false.

Why does the man in the Chinese room not understand Chinese even though he can pass the Turing test for understanding Chinese? The answer is that he has only the formal syntax of the program and not the actual mental content or semantic content that is associated with the words of a language when a speaker understands that language. You can see this by contrasting the man in the Chinese room with the same man answering questions put to him in his native English. In both cases he passes the Turing test, but from his point of view there is a big difference. He understands the English and not the Chinese. In the Chinese case he is acting as a digital computer. In the English case he is acting as a normal competent speaker of English. This shows that the Turing test fails to distinguish real mental capacities from simulations of those capacities. Simulation is not duplication, but the Turing test cannot detect the difference.

There have been a number of attempts to answer this argument, all of them, in the view of this author, unsuccessful. Perhaps the most common is the systems reply: "While the man in the Chinese room does not understand Chinese, he is not the whole system. He is but the central processing unit, a simple cog in the large mechanism that includes room, books, etc. It is the whole room, the whole system, that understands Chinese, not the man."

The answer to the systems reply is that the man has no way to get from the SYNTAX to the SEMANTICS, but neither

does the whole room. The whole room also has no way of attaching any thought content or mental content to the formal symbols. You can see this by imagining that the man internalizes the whole room. He memorizes the rulebook and the data base, he does all the calculations in his head, and he works outdoors. All the same, neither the man nor any subsystem in him has any way of attaching any meaning to the formal symbols.

The Chinese room has been widely misunderstood as attempting to show a lot of things it does not show.

1. The Chinese room does not show that "machines can't think." On the contrary, the brain is a machine and brains can think.
2. The Chinese room does not show that "computers can't think." On the contrary, something can be a computer and can think. If a computer is any machine capable of carrying out a computation, then all normal human beings are computers and they think. The Chinese room shows that COMPUTATION, as defined by Alan TURING and others as formal symbol manipulation, is not by itself constitutive of thinking.
3. The Chinese room does not show that only brains can think. We know that thinking is caused by neurobiological processes in the brain, but there is no logical obstacle to building a machine that could duplicate the causal powers of the brain to produce thought processes. The point, however, is that any such machine would have to be able to duplicate the specific causal powers of the brain to produce the biological process of thinking. The mere shuffling of formal symbols is not sufficient to guarantee these causal powers, as the Chinese room shows.

See also COMPUTATIONAL THEORY OF MIND; FUNCTIONALISM; INTENTIONALITY; MENTAL REPRESENTATION

—*John R. Searle*

Further Readings

Searle, J. R. (1980). Minds, brains and programs. *Behavioral and Brain Sciences*, vol. 3 (together with 27 peer commentaries and author's reply).

Chunking

See CHESS, PSYCHOLOGY OF; EXPLANATION-BASED LEARNING; FRAME-BASED SYSTEMS; METAREASONING

Church-Turing Thesis

Alonzo Church proposed at a meeting of the American Mathematical Society in April 1935, "that the notion of an effectively calculable function of positive integers should be identified with that of a recursive function." This proposal of identifying an informal notion, *effectively calculable function,* with a mathematically precise one, *recursive function,* has been called Church's thesis since Stephen Cole Kleene used that name in 1952. Alan TURING independently made a related proposal in 1936, Turing's thesis, suggesting the identification of effectively calculable functions with functions whose values can be computed by a particular idealized computing device, a *Turing machine.* As the two mathematical notions are provably equivalent, the theses are "equivalent," and are jointly referred to as the Church-Turing thesis.

The reflective, partly philosophical and partly mathematical, work around and in support of the thesis concerns one of the fundamental notions of mathematical logic. Its proper understanding is crucial for making informed and reasoned judgments on the significance of limitative results—like GÖDEL'S THEOREMS or Church's theorem. The work is equally crucial for computer science, artificial intelligence, and cognitive psychology as it provides also for these subjects a basic theoretical notion. For example, the thesis is the cornerstone for Allen NEWELL's delimitation of the class of physical symbol systems, that is, universal machines with a particular architecture. Newell (1980) views this delimitation "as the most fundamental contribution of artificial intelligence and computer science to the joint enterprise of cognitive science." In a turn that had almost been taken by Turing (1948, 1950), Newell points to the basic role physical symbol systems have in the study of the human mind: "the hypothesis is that humans are instances of physical symbol systems, and, by virtue of this, mind enters into the physical universe . . . this hypothesis sets the terms on which we search for a scientific theory of mind." The restrictive "almost" in Turing's case is easily motivated: he viewed the precise mathematical notion as a crucial ingredient for the investigation of the mind (using computing machines to simulate aspects of the mind), but did not subscribe to a sweeping "mechanist" theory. It is precisely for an understanding of such—sometimes controversial—claims that the background for Church's and Turing's work has to be presented carefully. Detailed connections to investigations in cognitive science, programmatically indicated above, are at the heart of many contributions (cf. for example, COGNITIVE MODELING, COMPUTATIONAL LEARNING THEORY, and COMPUTATIONAL THEORY OF MIND).

The informal notion of an effectively calculable function, effective procedure, or algorithm had been used in nineteenth century mathematics and logic, when indicating that a class of problems is solvable in a "mechanical fashion" by following fixed elementary rules. Hilbert in 1904 already suggested taking formally presented theories as objects of mathematical study, and metamathematics has been pursued vigorously and systematically since the 1920s. In its pursuit concrete issues arose that required for their resolution a precise characterization of the class of effective procedures. Hilbert's *Entscheidungsproblem* (see Hilbert and Bernays 1939), the decision problem for first order logic, was one such issue. It was solved negatively—relative to the precise notion of recursiveness, respectively to Turing machine computability; though obtained independently by Church and Turing, this result is usually called Church's theorem. A second significant issue was the formulation of Gödel's Incompleteness theorems as applying to *all* formal theories (satisfying certain representability and derivability conditions). Gödel had established the theorems in his groundbreaking 1931 paper for specific formal systems like type theory of *Principia Mathematica* or Zermelo-Fraenkel set