CS80—Senior Seminar
Discussion Questions—Hodges

Jim Rogers

jrogers@cs.earlham.edu

Fall 2000

These are a few questions for thought and discussion.

1. In what ways are computing machines different in type than more traditional sorts of machines? In what ways are they similar?

2. In distinguishing earlier computing machinery from "computers" in the modern sense, people often cite capabilities like programmability, the ability to modify program flow depending on the results of prior operations ("conditional branching"), and the ability to operate on programs (of the same type they execute) as ordinary data ("stored programs"). We can classify computing machinery on the basis of whether or not they have these capabilities.

   Is the halting problem solvable for any of these classes of machines? Which? Thinking, for a moment, of the human brain as a physical device, does it necessarily fall into any of these classes?

3. The existence of uncomputable problems is necessarily dependent on how we choose to model computation—we can only say that there are problems that are not computable by any machine of some particular type. The diagonalization argument sketched on pages 100–102, however, is quite general. It, in essence, says that there are functions that cannot be computed by any machine that can be described with finitely many symbols ("finitely presentable"). But it doesn't really tell us anything intuitively meaningful about what those functions might be. Turing, on the other hand, went on to argue that there was a particular problem that is undecidable, at least for TMs—the halting problem. This is actually also quite general, the argument works for any notion of computation for which there is a universal program (*Universality*).

   How does the argument depend on Universality?

4. Note that this is stronger than simply being able to operate on programs as data—any machine that is programmable can be presented with a program (of its own sort) as input which it will, inevitably, operate on. Universality comes closer to a kind of introspection: not only does the universal program operate on the input program as

data, but it must have the meaning of the program built into it in the sense that it can emulate the operation of the machine on that program.[1]

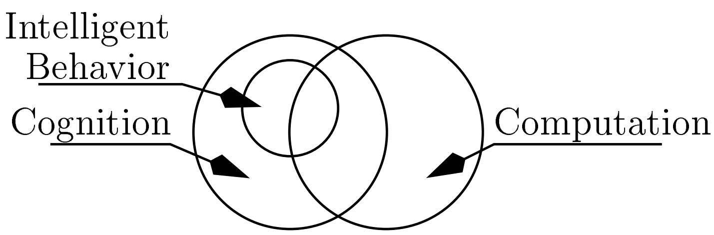
Is introspection necessary for intelligence? If it is, does this imply that there are problems that are unsolvable by human intelligence.

5. In Turing's initial proposal for ACE he notes (Pg. 332–333) that, while it might be capable of playing chess it would play badly since "chess requires intelligence", but then goes on to say that "it is possible to make the machine display intelligence at the risk of its making occasional serious mistakes" and that under these circumstances it might be possible to program ACE to play chess well. This is, apparently, a reference to his ideas about machine learning (see Pg. 358ff).

   Why did Turing's focus turn from programs ("rules-of-thumb") for playing chess to programs for learning to play chess? Is learning necessary for intelligence? Fallibility seems to be necessary for learning—else what is there to learn? Is fallibility necessary for intelligence? (See Turing's comments quoted on Pg. 378.) How does Deep Blue (the IBM system that beat Kasparov) meet with Turing's expectations about programs that play chess well?

   (As a digression) it is often stated that Turing claimed that within 100 years a computer would be able to play a creditable game of chess. But, at least in the claim reported in the *Times* (Pg. 349), he seems only to be claiming that the question of whether good chess requires judgment might be possible to settle within 100 years. Both claims are evidently redeemed by Deep Blue. How are the questions resolved?

6. At the outset we noted that Turing's claim that TMs correctly characterized the class of all computable problems depends on an argument that every computable problem could, in principle, be computed by human cognitive processes—that the area Computation − Cognition, in the figure, is empty. The Big Questions about artificial intelligence seem to be whether it is possible for a machine to exhibit intelligent behavior, whether it is possible for a machine to possess, in some sense, intelligence, and whether there is any aspect of human cognition that is, in principle, not computable. How do these questions differ from each other? What are they asking in terms of the figure? Which of the questions are addressed by Turing's Imitation Game thought experiment (the "Turing Test") (Pg. 415ff)? By chess playing computers? By programs that learn?



[1]There is actually a theoretical result that establishes that the class of machines that are equivalent in computing power to Turing Machines is characterized by Universality and the ability to specialize programs by binding arguments to fixed values. Remarkably, this latter capability is just as necessary as the first.

7. The structure of the Turing Test forces one to restrict arguments about intelligence to the Input/Output behavior of the agents involved, but arguments that dismiss, for instance, Deep Blue as an example of intelligence on the part of computers are based not on the external behavior of the system but rather on its internal structure. Is intelligence properly a question of external or internal behavior (or both)? To what extent does the notion of intelligence depend on (human) introspection? Is it even possible to develop concrete criteria for intelligence on such a foundation?

   In one of his arguments for the plausibility of the TM as a model of computation (and, in particular, as a model of human computation) Turing argued that at any given time a (human) computer might choose to write out a set of notes detailing their position in the computation so that they might lay it aside and pick up again where they left off later. (Thus justifying the finite configurations of the TM as analogs of the state of mind of the computer.) This seems to make an explicit connection between internal operation and (at least potential) I/O behavior. It also seems to be limited to those processes for which the human has sufficiently accurate introspection. What (if any) is the relationship between such accurate (or, perhaps, quantifiable) introspection and computability? Certainly, every computation is such a process. Is every behavior that is a result of such a process necessarily computable? Does intelligent behavior require ignorance of the process responsible for the behavior?