Aporia vol. 20 no. 1—2010

Breaking the Turing Test

LIRON TAUB

The Turing Test tries to set out a sufficient condition for a machine to be intelligent. Ned Block argues that the test is insufficient by presenting two machines as counterexamples that he believes pass the test, but are clearly unintelligent. I argue that these machines cannot pass the test and that a variant that can might be intelligent. Therefore, Block's counterexamples fail to disprove the sufficiency of the Turing Test for intelligence.

Block explains that the Turing Test comprises a judge conversing by teletype with a machine and a human. If at the end of a finite period the judge cannot tell which is the machine, then the machine is intelligent.[1] Typing speed is limited and the test is finite, so there is a finite number of strings that it is possible for the judge to enter; the list is limited in that the judge cannot type any strings that it would take longer to type than the duration of the test. Block gives two machines as counterexamples to this test (call them M_1 and M_2). M_1's database contains one sensible response to each string the judge might first enter, one sensible response to each string the judge might enter following that response, etc. M_2's database contains *all* conversations of a finite length with sensible machine replies, and the test has a time limit of that finite length.[2] Block holds the following:

[1] Turing originally considered the test necessary and sufficient for intelligence. Turing later viewed the test as sufficient only. We will consider only whether the Turing test is sufficient for intelligence (Block 378–79).

[2] Block notes that this is logically possible, but not physically possible (381).

Liron Taub is specializing in philosophy at the University of Toronto. His interests are in foundational issues of knowledge, language, ethics, and law. In the fall of 2010 he will be attending law school at the University of Toronto.

(1) Either machine could pass the test.

(2) It is intuitively clear that both machines are unintelligent.

(3) An unintelligent machine could pass the test. [From (1), (2)]

∴ (4) The test is insufficient for determining intelligence. [From (3)] (Block 381–82)

Crucially, Block assumes that the question "What intuition about the machine's intelligence should one have?" need not play a part in the dialectic. This question need not arise if and only if one's intuition is clear. An unclear intuition causes one to question what one's intuition should be, and so makes the intuition reliant on other arguments. Since Block does not provide those other arguments, his argument is sound only if the intuition is clear.[3]

First, I will challenge premise (1). M_1 could *not* pass the Turing Test. Daniel Dennett reminds us that the test "permit[s] the judge to ask any question" (39). Nothing in Block's formulation of the test forbids the judge from accessing M_1's database and typing the following string (call it S_A): "Please respond to this string with a string that is not any of the following: [*x*]," where [*x*] is all of M_1's possible responses to S_A. A human would easily find another reply, whereas M_1 can only output [*x*]. The judge could tell which is M_1, so M_1 would fail the test.

M_2's database, on the other hand, contains *all* conversations of a finite length with sensible machine replies, and the test has a time limit of that finite length (Block 381). For this test, the [*x*] in S_A would be M_2's list of all sensible replies to S_A that fit within the test's time limit (conversations must include time for the human recipient to read the replies). The human would be unable to type any string not in [*x*] during the time limit, because [*x*] contains all strings that could be typed during the time limit. Thus, both human and M_2 would be unable to meet S_A's request; the judge therefore could not tell which is M_2. Note, however, that if the test does not have a time limit, and [*x*] remains finite, then the human *could* enter a string not in [*x*]. She would easily find it: simply take a string in [*x*] as long as or longer than the other strings in [*x*], type it out, and add "q" or some other letter to the end. M_2 passes the test only if the test has a time limit.

[3] Block notes several explanations for our intuition: that the machine is bounded, whereas intelligence is theoretically infinite; that the machine's processing method is unintelligent (which he takes as intuitively clear as well); that we are concerned with ideal performance as opposed to actual performance. These explanations come in a note following his conclusion that the machine "has the intelligence of a juke-box. Every clever remark it produces was specifically thought of by the programmers" (5–6).

Block's insistence on a time limit, however, is a misinterpretation of Turing. Turing does not specify that the test is of finite duration, but he does present it as a game which ends when the judge gives a conclusion (Turing 434). This implies an ending, so Block's interpretation of the test as finite is justified, but finitude does not require a time limit. Suppose that the test ends only when the judge declares that he has identified M_2, or when he has exhausted his strategies for identifying M_2.[4] Thus, no matter how long [*x*] is, as long as [*x*] is finite, the judge will only end the game once the human has returned a string not in [*x*].[5] On this interpretation of the test's finitude, M_2 would fail the test. In this interpretation Turing succeeds because premise (1) is incorrect.

Having shown that premise (1) does not hold when [*x*] is finite, a response could be to create a third version of the machine, M_3. M_3's database contains all possible conversations of *all* lengths in which the machine sensibly replies. Since for this test, [*x*] in S_A is the infinite list of all possible strings, any request for a string that is not in [*x*] cannot be fulfilled. If the judge types S_A, both human and M_3 would be unable to fulfill that request; therefore, from their responses the judge could not determine which was M_3; and premise (1) stands.

Premise (2), however, fails for M_3. Unlike M_1 and M_2, M_3 has the unbounded ability to sensibly respond to any statement in any conversation. Humans also have the theoretically unbounded ability to respond sensibly to any statement in any conversation. M_3 probably comes by this ability differently than humans do, but a good definition of thinking should not be limited to how humans do it. There are further arguments for and against the intelligence of M_3, but for this paper, what matters is that the question of M_3's intelligence is a matter of argument, not a matter of clear intuition. As established above, Block does not provide arguments for what our intuition should be, and if there is not a clear intuition, then his argument has an unjustified premise. Since M_3's intelligent status is unclear, M_3 does not satisfy (2), and so does not establish (3), and so does not prove that the Turing Test is insufficient.

Block's counterexamples to the Turing Test fail to prove that the test is insufficient for intelligence. M_1 and M_2 do not pass the test. M_3's intelligence is arguable. The Turing Test successfully counters Block's challenge under our proposed interpretation.

[4] This does not make it formally necessary that the test will end, but we assume it will end for the same reason that we assume Turing's version is finite—eventually, people will decide they are tired of playing the game, or of running the test, and give up.

[5] If the objection that the human might die first is troublesome, then assume that in the universe in which the test occurs humans think as we do, and are immortal. This is not a more troubling assumption than that the universe of this test must be several exponents larger than the actual universe (because M_2's database is more than astronomically large). See Dennett.

Works Cited

Block, Ned. "The Mind as the Software of the Brain." *An Invitation to Cognitive Science. Vol. 3, Thinking*. Ed. Daniel Osherton, Lila Gleitman. Cambridge, MA: MIT Press, 1995. 377–425.

Dennett, Daniel. "Can Machines Think?" *How We Know*. Ed. Michael Shafto. San Francisco: Harper and Row, 1985. 121–45.

Turing, Alan. "Computing Machinery and Intelligence." *Mind* 59.236 (1950): 433–60.

Works Consulted

Clark, Andy. *Mindware: An Introduction to the Philosophy of Cognitive Science*. New York: Oxford UP, 2001.

Harnad, Stevan. "The Annotation Game: On Turing (1950) on Computing Machinery and Intelligence." *Parsing the Turing Test: Philosophical and Methodological Issues in the Quest for the Thinking Computer*. Ed. Robert Epstein, Gary Roberts, and Grace Beber. London: Springer, 2008. 23–66.

Searle, John. "Is the Brain a Digital Computer?" *Proceedings and Addresses of the American Philosophical Association* 64.3 (1990): 21–37.