Computation, Information, and Intelligence (COMS/ENGRI/INFO/COGST 172), Fall 2005
11/18/05: **Lecture 35 aid — Turing machines**

Agenda: Finish discussion of attentional structure; introduce Turing machines.

Announcements: Today, we are handing out two papers constituting the reading for the week of November 28th. We would like to have a heavily interactive discussion of these readings, so please be prepared to participate in class regarding your responses to these papers.

The first paper is Turing's "Computing Machinery and Intelligence", *Mind* 59 (not 49):433-460 (1950). It is a seminal essay in the philosophy of artificial intelligence, and also touches on a number of topics we have investigated in this course, including Turing machines, natural language processing, and machine learning. Focus on the definition of the "game" and the main arguments pro and con as to its utility. Feel free to ignore those passages that reflect the times in which the paper was written and/or the idiosyncracies of the author, especially those that would be inappropriate in a modern research publication.

The second paper is an "unedited, penultimate draft" of Searle's "Minds, Brains, and Programs", *The Behavioral and Brain Sciences* 3(3):417–457 (1980), an influential criticism of Turing-test like proposals. Focus on the third through fifth pages.

I. From last time: Example conversation

DS1
C1: ...I need to travel in May, leaving from Pittsburgh.
  DS2
  A1: And, what day in May did you want to travel?
  C2: OK uh I need to be there for a meeting that's from the 12th to the 15th.
  DS3
  A2: And you're flying into what city?
  C3: Seattle.
  DS4
  A3: And what time would you like to leave Pittsburgh?
    DS5
    C4: Uh hmm I don't think there's many options for non-stop.
    A4: Right. There's three non-stops today.
    C5: What are they?
    A5: The first one departs PGH at 10:00am arrives Seattle at 12:05 their time.
    The second flight departs PGH at 5:55pm, arrives Seattle at 8pm.
    And the last flight departs PGH at 8:15pm arrives Seattle at 10:28pm.
  C6: OK I'll take the 5ish flight on the night before on the 11th.
A6: On the 11th? OK. Departing at 5:55pm arrives Seattle at 8pm, U.S.Air flight 115.
C7: OK.

II. One proposal for the corresponding focus spaces

Slightly revised from last time. Implicitly invoked entities are italicized; questions marks indicate variables.

FS_1: DSP_1, *flight?, departure date and time?, arrival date and time?, etc.*, May, Pittsburgh
FS_2: DSP_2, departure day? → *May 11*, meeting, May 12th, May 15th
FS_3: DSP_3, arrival city → Seattle
FS_4: DSP_4, departure time → 5:55
FS_5: DSP_5, preference for non-stop, today, the three flights and details thereof

(OVER)

III. Sample one-tape Turing machine We specify the state set as "carry" and "no-carry"; the start state as "carry"; the allowable input symbols as 0, 1,...,9 (note that "blank" should not be an allowable input symbol); we'll ignore other details that would be required in a full specification; and specify the machine's behavior as follows.

> If reading a "0" and in state "carry", write "1", change to state "no-carry", stay put.
> If reading a "1" and in state "carry", write "2", change to state "no-carry", stay put.
> If reading a "2" and in state "carry", write "3", change to state "no-carry", stay put.
> ...
> If reading a "9" and in state "carry", write "0", stay in state "carry", move right.
> If reading a "blank" and in state "carry", write "1", change to state "no-carry", stay put.

(In a way, we are allowing a completely blank tape to represent the input number zero even though we have a symbol "0", but for simplicity we have elected not to deal with this issue.)

IV. Another example one-tape TM For brevity, we'll skip most of the initial specification that should be given. We'll assume there's a special marker "!" at the beginning of the tape. The start state is "no-carry".

end-of-tape rule:

> If reading a "!" and in state "no-carry", write "!", change to state "carry", move right.

carry rules:

> If reading a "0" and in state "carry", write "1", change to state "no-carry", move left.
> If reading a "1" and in state "carry", write "2", change to state "no-carry", move left.
> If reading a "2" and in state "carry", write "3", change to state "no-carry", move left.
> ...
> If reading a "9" and in state "carry", write "0", stay in state "carry", move right.
> If reading a "blank" and in state "carry", write "1", change to state "no-carry", move left.

return-to-tape-end rules:

> If reading a "0" and in state "no-carry", write "0", stay in state "no-carry", move left.
> If reading a "1" and in state "no-carry", write "1", stay in state "no-carry", move left.
> If reading a "2" and in state "no-carry", write "2", stay in state "no-carry", move left.
> ...
> If reading a "9" and in state "no-carry", write "9", stay in state "no-carry", move left.

V. Definition of TM function computation Let $f : D \to R$ be a function. A Turing machine M *computes* f if, for every $x \in D$, when M is initialized with input x, it eventually *halts* — ends up in a situation where no rule applies — with $f(x)$ on its (output) tape.

We will also allow for encodings of x and $f(x)$. For example, we might have a Turing machine that, given n "hash marks" as input, returns n^2 "hash marks"; we would then still say that the Turing machine computes $f(x) = x^2$, where x is a non-negative integer.

VI. Alteration to III above

Add the state "idle".

Add all rules of the following two forms, where s is a variable standing for any allowed input symbol:

> If reading an "s" and in state "no-carry", write "s", change to state "idle", stay put.
> If reading an "s" and in state "idle", write "s", change to state "no-carry", stay put.