 Last edited Monday, February 26, 2001

Solving the Halting Problem

I. Preliminaries

A. Review

1. The basic picture of Turing machines that we are working with is given in figure 1.
2. One of the main things we talked about on Tuesday was the notion of a *reasonable encoding*, and what the constraints were on the interpretation function (ρ).
3. We tried to see whether we could constrain ρ in terms of effectiveness, on either a mechanical/physical or a mathematical reading of that notion (define $\phi \equiv$ effective).
4. That failed, for various reasons
   a. Defining ρ in terms of φ was circular (and on some readings, a category error)
   b. If one pushed through, and tried to use it anyway it still fails:
      i. A physical reading of φ ($\equiv \phi_p$) led to a version of ρ that was too strong;
      ii. A mathematical reading of φ ($\equiv \phi_m$) led to a version of ρ that was too weak.
5. On the other hand we saw that *something* (interesting) must constrain ρ
6. In fact, we can see that we have established a series of successively stronger results wrt ϕ_m
   a. If ρ were completely unconstrained, ϕ_m would be empty as well.
   b. Interpretation (ρ) must be at least as constrained as $\text{effective}_{\text{mathematical}}$ (ϕ_m)
      i. $\rho \subseteq \phi_m$ (where ρ and ϕ_m are taken extensionally, as sets of functions)
   c. Moreover, if interpretation were *only* as constrained as $\text{effective}_{\text{mathematical}}$ (i.e., if $\rho = \phi_m$), then all computable functions could be computed by the "do nothing" machine.[1]
   d. So ρ is strictly stronger than ϕ_m
   e. I.e., $\rho \subset \phi_m$
7. In fact—this is important—the only *work* that the machine needs to do (the only work involved in what we intuitively call "computing") is in the *difference* between ρ and ϕ_m:
   a. Informally, that is:

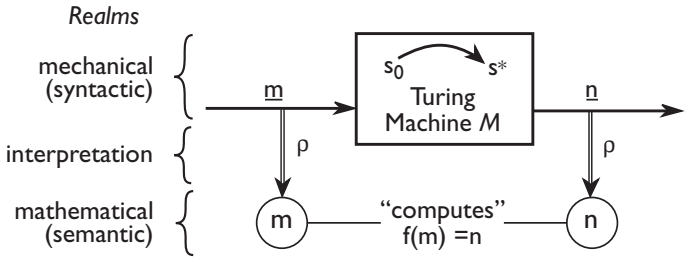

Figure 1 — Turing machines

- "Computing" ϵ $[\phi_m - \rho]$

[1] Figure 3, on page 8·8 of lecture notes 8a.

Q1. *At the (mechanical) level of marks*

- **a.** What are the representational conditions on the marks (**m** and **n**)?
- **b.** What is the horizontal effectiveness condition (fp) on their transformation?

Q2. *At the level of semantics or interpretation*

- **a.** What are the properties of and constraints on the vertical (semantical) interpretation function ρ?

Q3. *At the(mathematical) level of functions*

- **a.** What is the horizontal effectiveness condition ($\phi_{\mathbf{m}}$) on computable functions?
- **b.** More generally, can the discourse of recursion theory be carried on at the level of mathematical functions (*f*) defined over numbers (**m** and **n**)?

Figure 2 — Questions about Turing machines

b. We have to figure out what this (pseudo) equation really *means*[2]

B. Summary

1. Given these results, we summarised our investigation in terms of two basic questions, framed in terms of the dimension of the diagram:
   a. **Horizontal:** What's the origin / nature of the "effectiveness" ("calculable") constraint?
      i. Answer so far: We don't know. But in order to be coherent, it must be *horizontal*.
      ii. That is, effectiveness must be condition on one (or both?) of the following:
         α. Transformations over marks (i.e., upper syntactic or "mechanical" level in fig. 1)
         β. Functions over numbers, and perhaps other mathematical entities (i.e., at the lower "semantic" or "mathematical" level in figure 1).
   b. **Vertical:** What are the vertical (semantical) conditions on interpretation (ρ)?
      i. Answer so far: We don't know.
      ii. We do know, however, that the interpretation function ρ must be *more constrained* than being "effective" (if ρ could be any computable function, then any computable function could be computed by the null machine—the machine that does nothing).
      iii. I.e., $\rho \subset \phi_m$
2. Essentially the same issues can be pursued in terms of the three questions given in figure 2.

C. Plan

1. Adopt the following strategy, in our search for a coherent notion of effectiveness:
   a. Try to analyse effectiveness solely at level of mathematical objects (numbers, functions)
   b. If that doesn't work, relax (move upwards, in terms of the figure), to include conditions on the interpretation relation ρ.
   c. If that doesn't work, relax (move upwards) a second time, to admit conditions on the marks and the (mechanical) transformations of them.
2. Another (more detailed) way to understand question Q3b (above, figure 2) is as follows:

---

[2]Note that this "equation" is so vague that it can be read on either the mathematical or physical versions.

◆ a. What is the fundamental origin of the computability constraints?
b. Are they (au fond) properties of:
i. Numbers & mathematical functions? (i.e., in the "mathematical realm")?
ii. Encodings and representation (i.e., in the "realm of interpretation")? or
iii. Marks and mechanical configurations (i.e., in the "syntactic realm)?
c. Traditionally, the answer is (i): they are *mathematical* constraints.
d. We will argue that the right answer is (iii): they are *mechanical* constraints.

II. The halting problem

A. Introduction

1. We will try to determine the answers to these questions by looking at the (promised) machine designed to solve the halting problem.
2. To do that, we need to know what the halting problem is.
3. Because of the strategy given above, we will start with a purely mathematical formulation:

> **H1** Given arbitrary Turing machine $\mathfrak{M}$, and input **m**, compute 0 or 1 depending on whether or not $\mathfrak{M}$ would halt on **m**.

4. But (as will soon be evident), we need to frame it in terms that refer to transformations of marks, etc., as well. So instead try (see figure 3):

> **H2** Given as input the marks **m** and **n**, representing the numbers **m** and **n**, respectively, produce as output the marks **0** or **1**, representing the numbers 0 or 1, respectively, depending on whether the Turing machine $\mathfrak{M}$ modelled by the set of quadruples μ coded by the number **m** would or would not halt if given as input the mark **n'** modelled by the number **n**.

B. Remarks (in passing)

1. There are three "semantic" relations (functions) in this formulation (see figure 3):
   a. The interpretation relation ρ relating marks **m** and **n** to the numbers **m** and **n**
   b. A "modelling" relation between the set of quadruples μ and the machine $\mathfrak{M}$
   c. A "coding" relation, from:
      i. The number **m** to the set of quadruples μ
      ii. The number **n** to the mark **n'**
   d. At the moment we are focusing on ρ, but if we were doing this com-

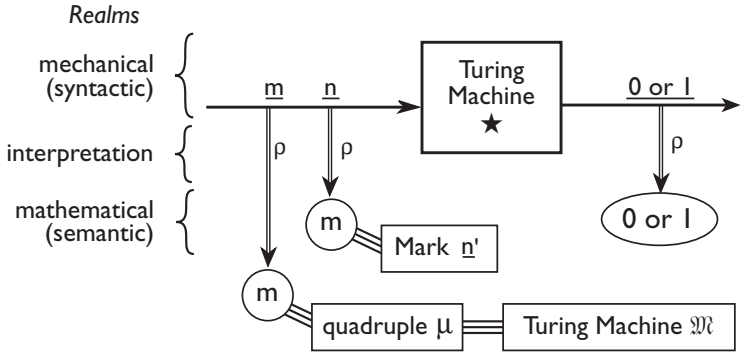

Figure 3 — The Halting Problem

pletely, all the others should be subjected to similar torture.

2. Instead of requiring that it produce **0** or **1**, we could formulate the requirement (that any machine solving the halting problem must meet) in terms of producing two particular marks—a single or a double *, say. But that would be too easy—and unsatisfying

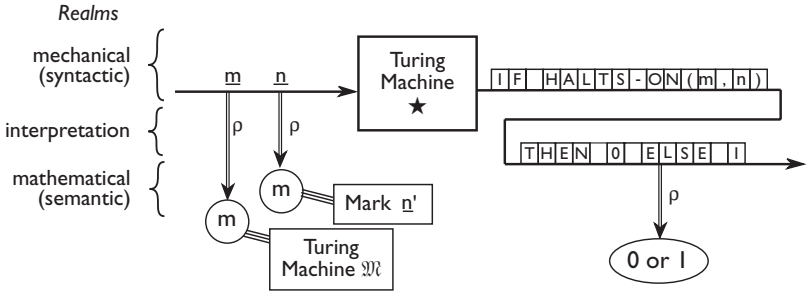

Figure 4 — Solving the halting problem

   a. As a general model of computing, most problems won't have that easy a form.
      i. Remember how widely we had to read the notion of "symbol" in order to avoid having the FSM construal fall off the "too narrow" cliff.
   b. Sure enough, having it produce one or two *s clearly (obviously? pretheoretically?) meets the condition of "being a reasonable encoding." That was never in doubt.
   c. What we want to know is: *what does that condition* of "being reasonable" come to?
   d. Simply adopting a particular instance won't tell us.
   e. So we need to stay with the more general (and for now mathematical) formulation.

III. The machine ★

A. Intro
   1. Condition **H2** is easy to meet
      a. Look at machine ★, given in figure 4.
         i. For inputs **m** and **n**, it simply produces as output the (composite) mark
            if halts-on(**m**, **n**) then 0 else 1
         ii. I.e., it simply inserts its inputs into the blanks in the following template
            if halts-on(__, __) then 0 else 1
      b. Which it could do by (as it were) running the following program:
      c. begin
            read (m-bar)
            read (n-bar)
            write ("if halts-on(")
            write (m-bar)
            write (",")
            write (n-bar)
            write (") then 0 else 1")
            end
   2. Diagnosis (initial)
      a. Clearly, ★ is a cheat.
      b. The question is *why*.
      c. Call the "answer" (i.e., output) that the machine produces α (actually there will be many different αs; call them $\alpha_{m,n}$)

d. There is some problem with α
e. Step through various suggestions

3. But note, first, that ★ does in fact meet **H2**.
   a. There is no (metaphysical) problem with the predicate "halts-on"
      i. It must be defined, for the halting problem to be coherent
      ii. I.e., there must be a metaphysical fact of the matter as to *whether* $\mathfrak{M}$ halts on **n'**
      iii. If that were not, there would be no justification in saying that there are any functions that cannot be computed
   b. There is no (semantical) problem with *designating* the halts-on predicate
      i. We did just that, for example, in stating the problem
      ii. And in the immediately-above discussion (III.A.3.a) of its metaphysical security
   c. So α is well defined
   d. Also, α satisfies all requirements (stated or implied) about what it is to be a **0** or **1**.
      i. For a mark **q** to count as a **0**, we only required that it represent 0.
         α. I.e. the requirement is that ρ(**q**) = 0
      ii. Similarly, for **q** to be a **1** is just to represent 1 (i.e., so that ρ(**q**) = 1)
      iii. Both these things α does.
      iv. If $\mathfrak{M}$ halts on **n'**, then $\alpha_{m,n}$ designates 0, as required
      v. If $\mathfrak{M}$ does not halt on **n'**, then $\alpha_{m,n}$ similarly designates 1.
      vi. So H2's requirements have all been met.

4. Remark #1 (important)
   a. It is evident that *our* human interpretation relation (i.e., the "ρ" leading from our heads to Plato's heaven) is sufficiently powerful to designate the "halts-on" predicate.
   b. This shows what should be evident: that *in general* (e.g., for human thought) semantic relations are *more* powerful (a larger class) than what is effectively computable (ϕ)
   c. But that raises an odd question. In trying to understand what constraints hold of ρ in the Turing machine case, we said that *that* ρ had to be strictly *weaker* (more constrained) than ϕ.
   d. So we have the following small—but nevertheless telling—result (let FAVs be functions, arguments, and values):

> **P1** It is intrinsic to the coherence of the halting problem—and to the notion of effective computability more generally—that if: (i) ρ is the (class of reasonable) interpretation functions from marks on tapes to FAVs; (ii) ϕ_m is the class of effectively computable functions; and (iii) ρ' is the *human* interpretation function, from thoughts to FAVs; then:
>
> $\rho \subset \phi_m \subset \rho'$

   e. This (tremendously important) result should be kept in mind throughout (◆)

5. Remark #2:
   a. What we have shown is that the "halts-on" function (i.e., specific values for specific arguments) can be *designated*. Unfortunately, no one doubted that.

b. What has not happened is for it to be *computed*
c. That in turn implies that *computing* cannot be something that happens at the level of FAVs.
d. That is, from the fact that ★ satisfies **H2**, it follows that the answer to **Q3b**[3] is *no*.

> **P2** The theory of effectively computable functions cannot be fully expressed at the (mathematical) level of functions, arguments, and values.

e. Note: we still don't know the answer to **Q3a**.

B. Attempted solution #1: different answers
1. It seems as if ★ always produces the same answer, *independent* of whether $\mathfrak{M}$ halts on **n'** .
2. So try: **0** and **1** should be *different* (whereas it seems as if the $\alpha_{m,n}$ are all the *same*).
3. But in fact (by design!) *all* the answers $\alpha_{m,n}$ are different (since no information is lost)

C. Attempted solution #2: different *types* of answer
1. Try again, but this time require that there be two *types* of answer: one that it produces when the input machine (i.e., the one modelled by the number **m** designated by **m**) halts, another when it does not.
2. More specifically, it seems as if there are three distinct criteria that should be met, two factual, one counterfactual (see figure 5, on the next page):

> **C1** *Different* inputs should lead to the *same* output, if they represent the *same* halting behaviour;
>
> **C2** *Different* inputs should lead to *different* outputs, if they represent *different* halting behaviour; and
>
> **C3** Counterfactually, any given input $\mathbf{m}_i$, $\mathbf{n}_j$ *should have led to a different output*, had the (metaphysical, ontological, conceptual, whatever) facts about whether $\mathfrak{M}_i$ halts on $\mathbf{n}_j$ been different.

3. These three criteria all deal with ranges of variation, requiring that the effective mapping be many-to-two:
a. **C1** and **C2** deal with ranges of actual inputs
i. Cf. the solid lines on the right side of figure 5.
b. **C3** deals with potential variation[4]

---

[3]See figure 2, above, on page 8·12.

[4]Some may object to the counterfactual case (**C3**) on the grounds that whether a machine $\mathfrak{M}$ halts on a given input **n'** is a mathematical fact, immune to revision, and thus identical in all possible worlds. To them, talk of a situation in which the output "would have been different" makes as much sense as saying that 2+2 might have equaled 5.

Over almost ten years of teaching this material, however, it has been my repeated experience that many students (typically, those with computational rather than mathematical backgrounds) find the counterfactual case to be the most compelling of the three. For some, in fact, it is the only case that matters. At this stage in the argu-

4. Formulate a new statement of the halting problem to ensure this (figure 5):

> **H3** Given as input marks **m** and **n**, representing numbers **m** and **n**, respectively, produce as output marks **0** or **1**, representing 0 or 1, respectively, depending on whether the Turing machine $\mathfrak{M}$ modelled by the set of quadruples μ coded by the number **m** would or would not halt, if given as input the mark **n'** modelled by the number **n**, such that all tokens of **0** lead to a single output state or path, and all tokens of **1** lead to a different, single output state or path.

5. **H3** fails
   a. ★ *already meets this formulation!*
   b. Problem is with the terms "same" and "different"
   c. Sameness (identity) and difference always *relative to an appropriate metric of equivalence*.
   d. Take the upper path in fig. 4 to be "represents 0", and the lower path to be "represents 1".

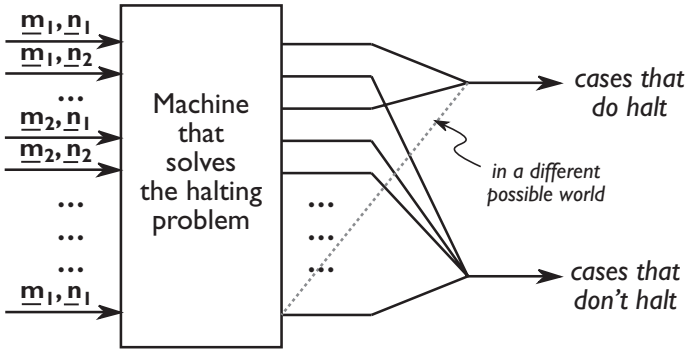

Figure 5 — Channeling answers onto different paths

   e. *If we can classify answers* (individuate "paths") *by interpretation*, **C1–C3** do no work!
   f. So **H3 = H2**.

D. Attempted solution #3: *effectively* different answers
   1. We have still not caught ★. But it is starting to be clear why.
   2. The intent of **H3** was to *shift attention upwards*, to predicates on the marks
   3. It has always been possible, through sufficiently devious means (and strong ρs) to *shift the attention back down* to functions and values
   4. We might try to modify **C1–C3** so that they prohibit reference to any semantical or interpretive properties, but that wouldn't work
      a. It would still be possible to identify other properties that, even if not semantical, are *sufficiently remote* that (intuitively) no one could *tell* whether or not the answer was the answer we want it to be

---

ment, I therefore recommend either (i) that **C3** be read under a notion of possibility strong enough—call it logical, metaphysical, conceptual, what you will—to "get under" whatever facts secure the answer's being the answer that it is, or (ii) if no such notion is available to you, that you set the third clause aside. As it happens, moreover, the whole issue will not matter much, in the end, for it turns out that the whole issue is something of a temporary red herring. On the reconstruction of Turing machines to be arrived at presently, the counterfactual reading will turn out to make sense after all (vindicating the students who wanted it), because the "fact" about whether a machine $\mathfrak{M}$ halts on a given input **n'** will by that point no longer be (understood to be) a mathematical one—but rather a physical one, mathematically modelled (and therefore requisitely contingent).

b. So the original intuition would not have been captured

5. Better: instead of *ruling out what is forbidden*, why not *rule in what is legitimate*?
6. The basic idea (I believe) is the following:

> **P3** One must be able to marshal all the inputs that represent situations where machines halt (i.e., that represent 0) onto *one effective path*, and similarly to marshal all the inputs that represent situations where machines do not halt (i.e., that represent 1) onto *a different effective path*.

7. I don't yet have an analysis of what a "single effective path" is (we'll get to that presently), but the rough idea is that all the answers of the "same effective type" should be able to *turn on a single switch*, or *end up physically indistinguishable* (where the "abstraction" over any variations among them can be "physically" ignored, somehow).
8. Just how this intuition should be *formulated* (carefully) is absolutely non-trivial. But (for now) I hope that the basic underlying intuition is relatively clear.

E. Effectively discriminable paths

1. This last remark leads to a new formulation of the problem.

> **H4** Given as input marks **m** and **n**, representing the numbers **m** and **n**, respectively, produce as output marks **0** or **1**, representing 0 or 1, respectively, depending on whether the Turing machine $\mathfrak{M}$ modelled by the set of quintuples μ coded by the number **m** would or would not halt, if given as input the mark **n'** modelled by the number **n**, such that (i) all tokens of **0** lead (immediately?) to a single effective state or path, (ii) all tokens of **1** lead (immediately) to a single effective state or path, and (iii) all tokens of **0** are (again, immediately) effectively discriminable from all tokens of **1**.

2. Sure enough, **H4** does seem to win the prize: it catches ★!
3. Are we done?
4. No! There is a very serious problem:

> ◆ **H4** (which seems to catch ★) is defined in terms of the notion "effective"—*which is exactly what we were supposed to be defining!* So it looks as if **H4** may be *true* but *circular*.

F. Next Tuesday, we will talk about how to fix **H4**—and thereby get to at least the beginnings of a tenable answer to our original question.

—— *end of file* ——