Phil 320

Chapter 11: The Undecidability of First-order logic

Omit: 11.2

0. Introduction

Decision problem for halting: Is there an effective method that, applied to any Turing machine M and any input n, will in a finite amount of time tell us whether M halts on input n?

Answer: NO (assuming *Turing's thesis,* that *effective* means a Turing machine can do it)

Decision problem for logical implication: Is there an effective method that, applied to any finite set of sentences Γ and sentence D, will in finite time tell us whether or not Γ implies D?

Answer for sentential logic: YES (truth tables).
Answer for first-order logic: NO (again, assuming *Turing's thesis*).

Theorem 11.2 (Church's Theorem): The decision problem for logical implication is unsolvable.

To prove it, we "reduce" the decision problem for logic to the halting problem. We show that **if it is solvable, then the halting problem is solvable**. Actually, what we prove is: given any fixed machine M and input n, there are Γ (a set of sentences) and D (a single sentence) such that

M halts on input $n \leftrightarrow \Gamma$ implies D.

So, if there were an effective means of deciding in general whether Γ implies D, we'd have a way to solve the halting problem. But we don't!! **So the decision problem is unsolvable.**
(The technique is reminiscent of chapter 8: represent M and the tape, using Γ and D.)

1. The language and interpretation for the sentences Γ and D.

Start with the machine M and the input n; these remain fixed. The language and interpretation vary somewhat depending on M and n.

The language:

Constant: $\mathbf{0}$
One-place predicates: $\mathbf{Q_1,\ldots,Q_k}$ [M a k-state machine, not counting halted state]
Two-place predicates: **S**, **<**
Additional two-place predicates: **@, M**

The *standard interpretation*, $\mathcal{M}$:

(1) Domain $|\mathcal{M}|$: the integers {0, +1, -1, +2, -2, …}

- Steps or *times* in the computation will be numbered by non-negative integers, starting with 0
- Squares on the tape will be numbered by all the integers, with the pointer starting off at square 0

(2) Denotations
$\mathbf{0}^{\mathcal{M}} = 0$
$\mathbf{Q_i}^{\mathcal{M}}(t) \leftrightarrow$ M is in state i at time t.
The denotation of $\mathbf{Q_i}$ is the set of times at which the machine M is in state i.

[**Ex:** Initial state is state 1, so $\mathbf{Q_1}^M(0)$. If M goes into state 2 at once, then $\mathbf{Q_2}^M(1)$. If M eventually halts, each such denotation will be a finite set of times.]

S denotes the successor relation: $\mathbf{S}^M(m, n) \leftrightarrow n = m+1$

< denotes less-than

@ denotes all pairs (t, x) where $t \geq 0$ and M is at square x at time t. [In particular, $@^M(0,0)$.] Read this predicate as "at".

M denotes all pairs (t, x) where $t \geq 0$ and square x is 'marked' with a 1 (rather than 0) at time t. [In particular, initially we have only squares 0 through n marked.]

When talking about @ and **M**, we'll always use t for the first argument (to suggest time) and x for the second (for the square).

These denotations depend entirely on the "career" or "history" of machine M when given input n. But that career is entirely determined by M and n, so they are all well-defined.

2. The sentences in Γ (three kinds).

(a) Background about S and <.

(1) $\forall u \forall v \forall w(((Suv \,\&\, Suw) \rightarrow v=w) \,\&\, ((Svu \,\&\, Swu) \rightarrow v=w))$ [unique successor, predecessor]

(2) $\forall u \forall v(Suv \rightarrow u<v)$ [a number is less than its successor]

(3) $\forall u \forall v \forall w((u<v \,\&\, v<w) \rightarrow u<w)$ [< is transitive]

(4) $\forall u \forall v(u<v \rightarrow u \neq v)$ [no number is less than itself]

These sentences imply some other important ones. To state these, it is convenient to introduce the unofficial terms **1**, **-1**, **2**, **-2**, **3**, **-3**, etc. using **0** and **S**. For example:

$$
\begin{aligned}
x=\mathbf{1} &\quad \leftrightarrow S\mathbf{0}x \\
x=\mathbf{-1} &\quad \leftrightarrow Sx\mathbf{0} \\
x=\mathbf{2} &\quad \leftrightarrow \exists y(S0y \& Syx) \\
x=\mathbf{-2} &\quad \leftrightarrow \exists y(Sxy \& Sy\mathbf{0}), \qquad \text{etc.}
\end{aligned}
$$

Then write

$$
\begin{aligned}
\mathbf{Q_i 2} &\quad \text{for } \exists t(t=\mathbf{2} \& \mathbf{Q_i}t) \\
S\text{-}\mathbf{2}x &\quad \text{for } \exists y(y=\mathbf{-2} \& Syx)
\end{aligned}
$$

We can show that the following are a consequence of (1)-(4):

$$
\begin{aligned}
&\mathbf{p} \neq \mathbf{q} \quad \text{if } p \neq q \\
&\forall v(\mathbf{Sm}v \rightarrow v=\mathbf{k}) \qquad \text{where } k=m+1 \\
&\forall v(\mathbf{S}v\mathbf{m} \rightarrow v=\mathbf{k}) \qquad \text{where } k=m-1
\end{aligned}
$$

(b) Description of time 0 (depends on the input, *n*).

At start, M is in state 1, at square 0, and only squares 0,1,…,n are marked with "1". So here is a description of the configuration at time 0, using @ and **M**:

(*0) $\mathbf{Q}_1\mathbf{0} \,\&\, @\mathbf{00} \,\&\, \mathbf{M00} \,\&\, \mathbf{M01} \,\&\ldots\&\, \mathbf{M0n} \,\&$
$\forall x((x \neq 0 \,\&\, x \neq 1 \,\&\, \ldots \,\&\, x \neq n) \rightarrow {\sim}M\mathbf{0}x)$

(c) Description of M (one sentence for each nonhalting instruction).

Unofficial: Put $\mathbf{M_1}$ for **M** (marked with 1), $\mathbf{M_0}$ for **~M** (marked with 0).

Each instruction has the form:

If in state *i* scanning symbol *s*, then take {one of four actions} and go into new state *j*. {Here, *s* can be 0 or 1.}

The corresponding sentence has to be stated using *x* for the current square, *t* for the current time. Here it is:

$$\forall t \forall x((\mathbf{Q_i} t \ \& \ @tx \ \& \ \mathrm{M}_s tx) \rightarrow \exists u(\mathbf{S}tu \ \& \ \{\text{depends on action}\} \ \& \ \mathbf{Q_j}\, u \ \&$$
$$\forall y((y \neq x \ \& \ \mathbf{M_1}ty) \rightarrow \mathbf{M_1}uy) \ \& \ \forall y((y \neq x \ \& \ \mathbf{M_0}ty) \rightarrow \mathbf{M_0}uy))$$

[**IF** in state *i* scanning *s* at time *t*, **THEN** at the next moment *u* it is in state *j* and no other squares besides possibly *x* change their contents.]

- If the action is to print *s**, then we fill in the { } with

[s*] $@ux \ \& \ \mathbf{M_{s*}}ux$

- If the action is to move right, we fill in { } with

[R] $\mathbf{M_s}ux \ \& \ \exists y(\mathbf{S}xy \ \& \ @uy)$

- If the action is to move left, we fill in { } with

[L] $\mathbf{M_s}ux \ \& \ \exists y(\mathbf{S}yx \ \& \ @uy)$

Together, the sentences in Γ completely describe basic facts about < and S, the starting configuration, and the Turing Machine instructions.

3. The sentence D

Each immediate precursor instruction to halting looks like:

If in state *i* and scanning symbol *s*, then (do something) and go to (halted state).

Consider the sentence

$\exists t \ \exists x \ (\mathbf{Q}_i t \ \& \ @tx \ \& \ \mathbf{M_s}tx)$.

This sentence is true if and only if the machine at some point (some time and some square) is in state *i* and scanning symbol *s*, in which case it halts.

So let D be the disjunction of all such sentences (over all finitely many pre-halting instructions). Then M halts if and only if one of the disjuncts is true, i.e., if and only if D is true.

4. Γ implies D iff M halts on input *n*

First, suppose M does not halt on input *n*. Then D is false on the standard interpretation: none of its disjuncts is true (else M would halt). But every sentence of Γ is true. So there exists an interpretation that makes Γ true and D false; hence, Γ does not imply D.

Next, we want to show that if M ever does halt, then Γ implies D.

- First, we need a sentence that gives the **description of time *a*:**

If at time *a*, M is in state *i*, at square *p*, and the marked squares are $q_1,\ldots,q_m$, then the description sentence (analogous to (*0)) is (using **a** denotes a and **p** denotes p):

(**a*) $\mathbf{Q_i a} \ \& \ @\mathbf{ap} \ \& \ \mathbf{Maq_1} \ \& \ \ldots \ \& \ \mathbf{Maq_m} \ \&$
$\forall x((x \neq \mathbf{q_1} \ \& \ \ldots \ \& \ x \neq \mathbf{q_m}) \rightarrow {\sim}\mathbf{Ma}x)$

This sentence tells us for each square whether it is marked or not – either directly a conjunct of (**a*), or a consequence of (**a*) and Γ. (In particular, if the currently scanned square, *p*, is blank, then *p* will be distinct from each q_i and then $\mathbf{p} \neq \mathbf{q_i}$ and we have ${\sim}\mathbf{Map}$.)

- Next, suppose M halts at time $b = a+1$. Then at time a, instruction was one of the pre-halting instructions. So one disjunct of D is implied by (via existential generalization for t and x):

  $$\mathbf{Q_i}\mathrm{a}\ \&\ @\mathbf{ap}\ \&\ \mathbf{M_s ap}$$

  where **s** is the symbol in square p at time a. But this sentence is implied by (*a) and Γ, and hence D is implied by (*a) and Γ.

- **Lemma:** If $a \geq 0$ and $b=a+1$ is a time at which the machine has not yet halted, then Γ plus (*a) implies (*b).

This Lemma completes the proof. For (*0) is part of Γ, and Γ plus (*0) imply (*1), so Γ implies (*1). Continuing in this way, Γ implies (*2), …, (*a) where $b=a+1$ is the time when M halts. And we just saw that D is implied by (*a) and Γ; hence D is implied by Γ. So if M halts, D is implied by Γ.

Proof of the Lemma:

Here is (*a):

$$\mathbf{Q_i a}\ \&\ @\mathbf{ap}\ \&\ \mathbf{Maq_1}\ \&\ \ldots\ \mathbf{Maq_m}\ \&$$
$$\forall x((x \neq \mathbf{q_1}\ \&\ x \neq \mathbf{q_2}\ \&\ \ldots\ \&\ x \neq \mathbf{q_m}) \rightarrow {\sim}\mathbf{Ma}x)$$

The instruction at time a is to go to state j and perform one of four overt actions: L, R, 0, 1.

- If the instruction is R, then here is (*b):

  $$\mathbf{Q_j b}\ \&\ @\mathbf{br}\ \&\ \mathbf{Mbq_1}\ \&\ \ldots\ \mathbf{Mbq_m}\ \&$$
  $$\forall x((x \neq \mathbf{q_1}\ \&\ x \neq \mathbf{q_2}\ \&\ \ldots\ \&\ x \neq \mathbf{q_m}) \rightarrow {\sim}\mathbf{Mb}x),$$

where $r = p+1$. Note that no marks on the tape are changed.

The sentence corresponding to the instruction, which is part of Γ, is

$$\forall t\forall x((\mathbf{Q_i}\boldsymbol{t}\ \&\ @tx\ \&\ \mathrm{M}_s tx) \rightarrow \exists u(\mathbf{S}tu\ \&\ \mathbf{M_s}ux\ \&\ \exists \mathrm{y}(\mathbf{S}xy\ \&\ @uy)\ \&\ \mathbf{Q_j}\,u\ \&$$
$$\forall y((y \neq x\ \&\ \mathbf{M_1}ty) \rightarrow \mathbf{M_1}uy)\ \&\ \forall y((y \neq x\ \&\ \mathbf{M_0}ty) \rightarrow \mathbf{M_0}uy))$$

Then (*b) follows from this sentence together with (*a) and Γ. To see this:

i) Put closed term **a** for t and **p** for x in the instruction sentence (follows from the instruction by instantiation):

$$((\mathbf{Q_i a}\ \&\ @\mathbf{ap}\ \&\ \mathrm{M}_s\mathbf{ap}) \rightarrow \exists u(\mathbf{Sa}u\ \&\ \mathbf{M_s}u\mathbf{p}\ \&\ \exists \mathrm{y}(\mathbf{Sp}y\ \&\ @uy)\ \&\ \mathbf{Q_j}\,u\ \&$$
$$\forall y((y \neq \mathbf{p}\ \&\ \mathbf{M_1a}y) \rightarrow \mathbf{M_1}uy)\ \&\ \forall y((y \neq \mathbf{p}\ \&\ \mathbf{M_0a}y) \rightarrow \mathbf{M_0}uy))$$

ii) (*a) and Γ imply the antecedent, so (still an implication) we get the consequent,

$$\exists u(\mathbf{Sa}u\ \&\ \mathbf{M_s}u\mathbf{p}\ \&\ \exists \mathrm{y}(\mathbf{Sp}y\ \&\ @uy)\ \&\ \mathbf{Q_j}\,u\ \&$$
$$\forall y((y \neq \mathbf{p}\ \&\ \mathbf{M_1a}y) \rightarrow \mathbf{M_1}uy)\ \&\ \forall y((y \neq \mathbf{p}\ \&\ \mathbf{M_0a}y) \rightarrow \mathbf{M_0}uy))$$

iii) From section 2(a), we get u=**b**, where $b=a+1$, and y=**r**, where $r=p+1$ so that we have

$$\mathbf{M_s bp}\ \&\ @\mathbf{br}\ \&\ \mathbf{Q_j\,b}\ \&$$
$$\forall y((y \neq \mathbf{p}\ \&\ \mathbf{M_1a}y) \rightarrow \mathbf{M_1b}y)\ \&\ \forall y((y \neq \mathbf{p}\ \&\ \mathbf{M_0a}y) \rightarrow \mathbf{M_0b}y))$$

iv) From first conjunct, the mark in square p is unchanged; from the last two, the marks in all other squares are unchanged; and the second and third conjuncts give the rest of (*b).

- If the instruction is L, similar argument; the book gives the case where instruction is to write 1, and similar argument is possible if instruction is to write 0.