LECTURE 10: GÖDEL'S THEOREMS

Problems taken from MIT's 24.118, "Paradox and Infinity". Explanation of Gödel's theorem taken from MIT's 6.004, "Computation Structures".

GÖDEL'S INCOMPLETENESS THEOREM

explained in words of one syllable

First of all, when I say "proved", what I will mean is "proved with the aid of the whole of math". Now then: two plus two is four, as you well know. And, of course, it can be proved that two plus two is four (proved, that is, with the aid of the whole of math, as I said, though in the case of two plus two, of course we do not need the whole of math to prove that it is four). And, as may not be quite so clear, it can be proved that it can be proved that two plus two is four, as well. And it can be proved that it can be proved that it can be proved that two plus two is four. And so on. In fact, if a claim can be proved, then it can be proved that the claim can be proved. And that too can be proved.

Now: two plus two is not five. And it can be proved that two plus two is not five. And it can be proved that it can be proved that two plus two is not five, and so on.

Thus: it can be proved that two plus two is not five. Can it be proved as well that two plus two is five? It would be a real blow to math, to say the least, if it could. If it could be proved that two plus two is five, then it could be proved that five is not five, and then there would be no claim that could not be proved, and math would be a lot of bunk.

So, we now want to ask, can it be proved that it can't be proved that two plus two is five? Here's the shock: no, it can't. Or to hedge a bit: if it can be proved that it can't be proved that two plus two is five, then it can be proved as well that two plus two is five, and math is a lot of bunk. In fact, if math is not a lot of bunk, then no claim of the form "claim X can't be proved" can be proved.

So, if math is not a lot of bunk, then, though it can't be proved that two plus two is five, it can't be proved that it can't be proved that two plus two is five.

By the way, in case you'd like to know: yes, it can be proved that if it can be proved that it can't be proved that two plus two is five, then it can be proved that two plus two is five.

RUSSELL'S PARADOX

Recall Russell's paradox: "Does the set of all sets that don't contain themselves... contain itself?"

PROBLEM 1

What is the analogue of the Russell's paradox set when it comes to:

- lists
- sentences
- portrait painters
- people who love other people

PROBLEM 2

An "autological word" is a word that describes itself. A "heterological word" is a word that does not describe itself.

Is the word "autological" autological?

PROBLEM 3

Is the word "heterological" heterological?

PROBLEM 4

List six heterological words.

PROBLEM 5

What is wrong with the following argument:

"For every object, there is some fact of the matter as to whether that object is a member of itself or not. Your computer: not a member of itself. The set of all natural numbers: not a member of itself. And so on. So *of course* there is a set of all such objects."

Say exactly which sentence is the one where the error creeps in, and why it is an error.

LOGICAL LANGUAGE

These next questions are to try to familiarize you with using a 'purely logical language'. The notation will be somewhat modernized from what existed in Russell's time, which was a logical system devised by Frege.

Frege's language contained analogues of the following symbols:

- logical connectives: '$\wedge$' (and), '$\vee$' (or), '$\neg$' (not), '$\Rightarrow$' (if-then, or implies), '$\Leftrightarrow$' (if-and-only-if)

- predicate letters: 'F', 'G', etc. P indicates a proposition that is true or false all on its own. $F(x)$ indicates that x has property F. xRy indicates that x and y are related by the relation R.
- object variables: 'x', 'y', etc.
- quantifiers: '$\forall$' (for all), '$\exists$' (there exists)
- identity for predicates: '$=$'
- brackets: '(', ')'

Here are some example sentences in Frege's language and their translations into pseudo-English:

- '$\exists x Fx$' There exists an object x that has property F.
- '$\forall x(Fx \Rightarrow \neg Gx)$' For all objects x, if x has property F then it does not have property G.
- '$\forall x \exists y(xRy)$' For all objects x there exists an object y such that x is related by relation R to y.
- '$\exists x \forall y$' There exists an object x such that for all objects y, x equals y. (That's a slightly weird thing to say because it is obviously false.)

PROBLEM 6

(CHALLENGE PROBLEM for Week 3. Due Wednesday of Week 4.)

We're going to work towards writing down Hume's principle in this language. Hume's principle states that the number of objects in A is equal to the number of objects in B if and only if there is a one-to-one-correspondence (a *bijection*) between the objects in A and the objects in B.

1) Write down a logical sentence that is true if and only if exactly one thing has property F. (hint: you'll need to use '$=$')
2) Write down a logical sentence that is true if and only if everything that has property F is related by R to something that has property G.
3) Write down a logical sentence that is true if and only if everything that has property F is related by R to *exactly one thing* that has property G.
4) Write down a logical sentence that is true if and only if everything that has property G has exactly one thing with property F related to it by R.
5) Write down a logical sentence that is true if and only if everything that has property F is related by R to exactly one thing that has property G, and everything that has property G has exactly one thing with property F related to it by R.
6) If the last sentence you wrote down is true, what is the relationship between the number of things which have property F and the number of things which have property G?

7) Let '#F' stand for 'the number of things that have property F'. Write down Hume's principle using Frege-like logical notation plus '#'.

GÖDEL'S THEOREMS

Logic and Arithmetic

PROBLEM 7

Using the logical language described above, formalize the following claims. This is to give you some practice expressing arithmetical claims in logical language.

1) Find a logical sentence that expresses the claim that there are no numbers a, b, and c such that $a^3 + b^3 = c^3$. (Note that you are not being asked to prove Fermat's Last Theorem, only to state it.)
2) Find a logical sentence that expresses the claim that there are infinitely many prime numbers.

Problems about Gödel Numbering

The point of this next sequence of problems is to get you to verify one of the main building-blocks of Gödel's Theorem: the fact that a formal mathematical system is able to talk about finite sequences of natural numbers.

Our construction will be based on the observation that one can 'code' the sequence $[a_1, ...a_n]$ as the number

$p_1^{a_1+1} \times ... \times p_n^{a_n+1}$

where p_i is the i-th prime number.

(Look familiar? This is similar to how Gödel numbering of Turing machines works.)

PROBLEM 8

Using the coding system above, show that you can use the logical language to write a proposition $Pair(n, a, b)$ which is true only when n codes the two-item-long sequence $[a, b]$.

PROBLEM 9

Using the coding system above, show that you can use the logical language to write a proposition $Includes(n, m)$ which is true only when n codes for a sequence that includes m as one of its members.

PROBLEM 10

For any sequence $[a_1, ...a_k]$, let a *number-sequence* for $[a_1, ...a_k]$ be any re-ordering of the sequence $[[1, a_1], [2, a_2], ...[k, a_k]]$.

For instance, each of the following is a number-sequence for $[10, 100, 1000]$:

- $[[1, 10], [2, 100], [3, 1000]]$
- $[[2, 100], [3, 1000], [1, 10]]$

but none of the following is a number-sequence for $[10, 100, 1000]$:

- $[[1, 100], [2, 10], [3, 1000]]$
- $[[1, 10], [2, 100], [3, 1000], 17]$
- $[[1, 10], [1, 10], [2, 100], [3, 1000]]$

Use $Pair(n, a, b)$ and $Includes(n, m)$ to show that you can use the logical language to write a proposition $NSeq(n, k)$ which is true only when n codes for a k-membered *number-sequence* for any $[a_1, ...a_k]$.

PROBLEM 11

Use $NSeq(n, k)$ to show that you can use the logical language to write a proposition $Seq(n, k, i, m)$ which is true only when n is a k-membered *number-sequence* which includes the pair $[i, m]$.

PROBLEM 12

Bonus Problem (Junction cup points!) Give an informal description of how one might use the logical language to write a proposition $Halt(p, t, r, s)$ which is true only when a Turing machine with program p would halt after a finite number of steps if it started out in state s, tape t, and with the tape-reader in position r.