Lecture 16
Chaitin's work

Michael Beeson

Some notation

- For this lecture, natural numbers will be identified with strings of 0s and 1s (giving the binary representation)
- By the size of a Turing machine, we mean the number of bits in its description. Remember that a Turing machine is just a string of symbols, which we can regard as taking 8 bits per symbol.
- We use, in this section, the notation of complexity theory, in which $|x|$ is the number of bits (digits) in x, so $|e|$ is the size of Turing machine e.
- Thus $|128| = 7$ because it takes 7 bits to write 128 in binary. This has nothing to do with absolute value. I did not invent this notation!
- We say that Turing machine e computes x if $\varphi_e(0) = x$. Intuitively this means that e, run with no input, produces output x.

Information content of a string or natural number

Definition

Let $H(x)$, the "complexity of x", be the size of the smallest Turing machine e that computes x. Then e is **minimal** if for some x, $\varphi_e(0) = x$ and for all $d < e$, we do not have $\varphi_d(0) = x$.

Some examples

- Any number or string can be computed by a program that contains a copy of the string and a "print this string" command.
- In the case of Turing machines, this program will have size a constant times n where n is the size of the string to be printed.
- So the complexity of x is at most $c|x|$.
- But it is often much less. For example $2^{2^{100}}$ can be printed (in theory) by a very short program, much shorter than its size, which is 2^{100}.

An information-theoretic approach to the halting problem

- We are primarily interested in Chaitin's work on incompleteness.
- But we will "warm up" on the halting problem, using the concept of algorithmic complexity.

Chaitin's proof of the unsolvability of the halting problem

- Suppose the halting problem is computably solvable.
- Then the following function is computable:
  $f(n) =$ the largest natural number computed by any program (Turing machine in our case) of size at most n.
- To compute $f(n)$, we just enumerate the machines of size at most n, determine using the hypothetical solution of the halting problem whether they compute an integer, and if so, use the universal machine to simulate that computation.
- Let e be a machine to compute f, and let N be at least the size of e. Then there is a machine M that computes the integer $f(2N)$.
- The size of M is at most $3|N|$ plus the size of e, which is $|N| + N$. (Explained on next slide.)
- Since (increasing N if necessary) $4|N| < N$, we have $|M| \leq 2N$.
- But then the integer computed by M has to be less than $f(2N)$, which is equal to the integer computed by M, contradiction.

Details about the size of M

- e computes function f.
- N is an integer
- Program for M says, apply machine e to input N.
- So N is hard-coded into the program for M.
- To do that with Turing machines, you use this technique: Say you want to apply e to the string cat. You have instructions as follows:

```
0*1cR   // * is a wild card for any symbol
1*2aR
2*3tR
3*4_L
4*4*L  //  here * stands for anything but blank
4_5_R  //  so it stops after writing "cat" and
           then moving left to the "c".
```

and in addition you have all the instructions of machine e with 5 added to all state numbers.

Is that proof so different from Turing's?

- Perhaps it was new, because it is based on algorithmic complexity.
- However, it wasn't brand-new and original, in some sense.
- It's quite similar to something known as the "busy beaver", because machine M is working like a beaver imitating all smaller machines.
- The classical "busy beaver" use the number of states, rather than the number of bits in the description.

Two paradoxes

On the next slides, we consider two classic "paradoxes" from philosophy, which will be useful in considering incompleteness. Namely,

- Berry's paradox, and
- The liar paradox, or the paradox of Epimenides

Berry's Paradox

Find the smallest positive integer which to be specified requires more characters than there are in this sentence.

If the phrase "this sentence" bothers you, consider this form:

Find the smallest positive integer which to be specified requires more than 90 characters.

Does this sentence specify a positive integer?

Epimenides, or Liar, Paradox

This statement is false.

Is this statement true?

Chaitin versus Gödel

- Chaitin's approach to incompleteness is to Berry's Paradox as Gödel's approach is to the Liar Paradox.
- Gödel replaced 'true" by "provable" (in a specified theory)
  So "this sentence is not true" became
  "this sentence is unprovable."
- Chaitin replaced "specify" by "compute" (in a specified programming language)
  So "This specification cannot be made shorter" became
  "This computation cannot be made shorter"

Measuring the complexity of an axiom system

Chaitin's key new ideas are (in my opinion):

- the "algorithmic information content of", or "complexity of" an object is the length of the shortest program that computes it (without input).
- That definition applies to axiom systems. The information content of an axiom system is, intuitively, proportional to the number of pages it takes to write down the axioms. (One page in most cases.)
- The information contained in theorems comes from the information in the axioms, as deduction, though it takes time, adds no information.
- So theorems have low information content, but most truths have high information content.

We will flesh out these ideas on the next slides.

Information-theoretic proofs of incompleteness

Chaitin gave not one but *several* incompleteness theorems based on "algorithmic information theory." Here is one of the simplest:

Theorem

Consider an N-bit formal axiomatic system T. There is a Turing machine e of size N that does not halt (at input 0), but T cannot prove "$\varphi_e(0)$ does not halt."

Another Chaitin incompleteness theorem

An N-bit string or number is called algorithmically random if it is not computed by any program with fewer than N bits.

Theorem

Let T be a formal theory with information content N. Then there are only finitely many numbers k such that T can prove "$\bar{k}$ is algorithmically random." In fact any such k has no more that $N + c$ bits, where c is an absolute constant, not depending on the theory T.

A third Chaitin incompleteness theorem

- A minimal program is one that computes an algorithmically random output.
- More explicitly, p is minimal means that p computes an integer $\varphi_p(0)$, and no shorter program computes that same integer.
- For each N, the smallest program that computes N is minimal, so there are infinitely many minimal programs.
- But to prove a particular program is minimal is very hard.

Theorem

Let T be a true recursively axiomatizable extension of $\mathbf{PA}$. Then there are only finitely many programs provably minimal in T.

We give the proof of this theorem in complete detail in these lecture slides. Whether time permits going through it in class, we will see.

Comparing proofs of incompleteness by the form of the unprovable

- "Turing's proof" gives unprovable statements of the form "$\varphi_e(e)$ does not halt."
- So does Chaitin's first theorem quoted above.
- "Gödel's proof" gives unprovable statements of the form "T is consistent".
- But "p is minimal" is a third, different form.

Proof outline

Here is an outline of Chaitin's proof that a true recursively axiomatizable extension of $\mathbf{PA}$ can prove "$\bar{m}$ is minimal" only for finitely many m.

- if T could prove infinitely many programs to be minimal, then we can define a computable function f, such that given input k, it enumerates the proofs of T until it finds one that ends in "$\bar{m}$ is minimal" for some m with more than $k + 1000$ bits.
- Since T is a true theory, then m really is minimal, but I just gave you a short program to compute m.
- In particular the program I gave you has no more than $k + 1000$ bits,
- But that is fewer bits than m, contradicting the minimality of m. QED.
- All the details are on later slides.

The ubiquity of unprovability

- Pick an axiom system T that proves no false theorems.
- Let m be an index of the function f in the proof outline above.
- Then m is bounded by a universal constant c plus an index for computing whether a formula is an axiom of T or not.
- This is a constant times the number of lines it takes to write the axiom schemata for T on paper.
- All known interesting theories, even those with infinitely many axioms, have very low complexities in this sense.
- Then the only true theorems of the form $Min(\bar{k})$ provable in T are those with $k \leq m$.
- Hence if I pick a formula of the form $Min(\bar{k})$ at random, with finitely many exceptions it is false or unprovable.

The ubiquity of unprovability

- There has been extensive work, which we may or not have time to mention in more detail in this course, to connect formulae of $\mathbf{PA}$ of certain logical forms to simpler formula expressing facts about the solution of polynomial equations in integers.
- A polynomial equation, for which solutions are required to be integers, is called a Diophantine equation. You can express these problems in $\mathbf{PA}$ even though $\mathbf{PA}$ is about natural numbers, as we have already discussed.
- Using that work, Chaitin has shown that his version of the incompleteness theorem implies that there is a single "universal" Diophantine equation, such that *one coefficient* depends on the axioms of T, and for only finitely many values of the other coefficients, T proves the equation has infinitely many solutions, while indeed, there are infinitely many true theorems of the form "this equation has infinitely many solutions."

More explanation of the ubiquity theorem

- To oversimplify, what if

$$ax^3 + by^3 + cz^3 = d$$

  is the equation; so a depends on the theory T, and once T is fixed, we ask of various b, c, d whether there are infinitely many or finitely many solutions (x, y, z) in integers. Then for infinitely many values of (b, c, d) there are indeed infinitely many solutions, but only for finitely many of those (b, c, d) is it provable in T.
- Of course, in the real theorem, the equation has dozens of variables and high degree, but it can be explicitly given, and Chaitin actually computed it!

Incompleteness and complexity

- When we go from $\mathbf{PA}$ to our most powerful set theories, we just increase the complexity of the axiom system (as a string of symbols) by a small amount.
- That is how much we increase the bound on our knowledge of which Diophantine equations have infinitely many solutions.
- The complexity of our axiom systems (in the information-theoretic sense) is certainly limited, since we need to understand these axiom systems, not just to write them down. Imagine if it required a long computation just to write down the axioms!
- Hence no matter what axioms we come up with, we're still only going to be able to settle finitely many of the questions whether that equation has (with given coefficients) infinitely many or finitely many solutions.

Randomness

- A real number is Chaitin random if the first n digits of its binary expansion have algorithmic complexity at least $n - cn$ for some constant c.
- This is shown to agree with an earlier definition *Martin-Löf random*, which does not involve algorithmic complexity. So we just call it "random."
- Chaitin defines the number

$$\Omega = \sum 2^{-|p_k|}$$

  where the sum is over all programs p_k that produce values. In other words, each halting program (without input) that is m bits long contributes 2^{-m} to the sum.
- Technically you can't use Turing machines but must have prefix-free programs, i.e. no initial segment of a legal program is also a program, or else the sum doesn't converge.
- Ω is Chaitin random.

Incompleteness and Randomness

Consider statements of the form

"$f(\mathbf{x}, n) = g(\mathbf{x}, n)$ has infinitely many natural-number solutions $\mathbf{x}$"

Fixing f and g determines a sequence of 0s and 1s. Namely,

$$a_n = \begin{cases} 1 \text{ if } f(\mathbf{x}, n) = g(\mathbf{x}, n) \text{ has infinitely many solutions} \\ 0 \text{ otherwise} \end{cases}$$

Theorem (Chaitin's best)

Given a true recursively axiomatizable extension T of $\mathbf{PA}$, for some f and g the sequence a_n is random, i.e. the real $\Sigma a_n/2^n$ is random.

In particular, for all but finitely many n, it will be independent of T whether the equation has infinitely many solutions or not. But what is more, the answers are as random as flipping a coin.

Chaitin's conclusion in his words

We see that proving whether particular Diophantine equations have finitely or infinitely many solutions is absolutely intractable. Such questions escape the power of mathematical reasoning. This is a region in which mathematical truth has no discernible structure or pattern and appears to be completely random. These questions are completely beyond the power of human reasoning. Mathematics cannot deal with them.

Now for the technical details

- In the rest of the lecture, we will prove one of Chaitin's incompleteness theorems.
- Here is the proof outline, which we saw in an earlier slide:
- if T could prove infinitely many programs to be minimal, then we can define a computable function f, such that given input k, it enumerates the proofs of T until it finds one that ends in "$\bar{m}$ is minimal" for some m with more than $k + 1000$ bits.
- Since T is a true theory, then m really is minimal, but I just gave you a short program to compute m.
- In particular the program I gave you has no more than $k + 1000$ bits,
- But that is fewer bits than m, contradicting the minimality of m. QED.
- The next slides fill in the details.

The complexity of minimal programs

- Recall $H(x)$ is the complexity of x, i.e. the minimal size program for computing x. The number stays fixed and the program varies.
- Let p be minimal. The complexity of p can't be much less than $|p|$, since if some smaller machine could compute p, then we could use that smaller machine to compute p, and then use p to compute $p(0)$.
- The following lemma states this more formally.

Lemma (Chaitin's Lemma)

If p is minimal, then $H(p) \geq |p| - c$ for some constant c.

Proof of Chaitin's lemma

Lemma (Chaitin's Lemma)

If p is minimal, then $H(p) \geq |p| - c$ for some constant c.

Proof. Let $x = \varphi_p(0)$. Let q be the smallest Turing machine such that $\varphi_q(0) = p$, so $H(p) = |q|$. Then $\varphi_{\varphi_q(0)}(0) = \varphi_p(0) = x$, so $\Lambda z\, \varphi_{\varphi_q(0)}$ is a Turing machine that computes x. Then by definition of $H(x)$, the size of this Turing machine is less than $H(x) = p$. But the size of $\Lambda z\, \varphi_{\varphi_q(0)})$ is at most a constant (say c) plus the size of q. (Details next slide.) Hence

$$\begin{aligned} c + |q| &= |\Lambda z\, \varphi_{\varphi_q(0)}| \\ &\leq H(x) = p \end{aligned}$$

Since $H(p) = |q|$, we have

$$c + H(p) \leq p.$$

Subtracting c from both sides, we find the formula of the lemma. That completes the proof.

A detail from the proof of Chaitin's lemma

I claimed that the size of $\Lambda z\, \varphi_{\varphi_q(0)})$ is at most a constant (say c) plus the size of q.

I will describe a two-tape machine for $\Lambda z\, \varphi_{\varphi_q(0)})$.

- Copy the input onto the aux tape.
- Compute $\varphi_q(0)$.
- With that still on the main tape, skip a blank to the right and copy the contents of the aux tape to the main tape.
- Move all the way to the left and run the universal machine.

That machine has size $|q| + c$ for some constant c.

Converting the two-tape machine to a one-tape machine just increases the constant.

Chaitin's Incompleteness Theorem

Chaitin used his concept of the information content of an integer to give a new version of the First Incompleteness Theorem. The predicate $y = H(x)$ is definable in $\mathbf{PA}$ this way:

$$Min(x) := \exists e \left(\exists k \left(\mathbb{T}(e, 0, k) \wedge U(k) = x\right) \wedge\right.$$

$$\left.\forall d < e(\forall j(\mathbb{T}(e, 0, j) \supset U(j) \neq x))\right)$$

Theorem (Chaitin)

Let T be any true recursively axiomatizable extension of $\mathbf{PA}$. Then there cannot be infinitely many numbers x such that $T \vdash Min(\bar{x})$.

Proof of Chaitin's Theorem

Suppose, for proof by contradiction, that there are infinitely many numbers provably minimal in T. Then we define the computable function f as follows:

- Given input k (as a unary string), search for a proof in T ending in $Min(\bar{p})$ for some p with $|p| > 2k$, and define $f(k) = p$.
- Note that p is minimal, because T proves only true theorems.
- Then $2k < H(p) + c$ for some constant c, by the lemma.
- Now for each k, $\Lambda z\, f(k)$ is a Turing machine e such that $\varphi_e(0) = f(k)$. (It just erases its input, writes k on the tape, and then starts the machine for f.)
- so $H(f(k)) \leq |\Lambda z\, f(k)|$, or with $p = f(k)$,

$$H(p) \leq |\Lambda z\, f(k)|$$

Proof continued

On the previous slide we had $2k < H(p) + c$ and

$$H(p) \leq |\Lambda z\, f(k)|.$$

- Now how big is $\Lambda z\, f(k)$? It is obtained by substituting k into a fixed expression, so its size is bounded by $k + c_2$ for some constant c_2.
- (We could get $|k|$ if we let f compute in binary, because then we could write k in binary, but this doesn't matter now.)
- Hence $H(p) \leq k + c_2$.
- Combining this with $2k < H(p) + c$ we have

$$2k < k + c_2 + c$$

- Taking k large enough, we have a contradiction.

That completes the proof of the theorem.

Chaitin's theorem implies the First Incompleteness Theorem

Corollary (First incompleteness theorem)

Let T be any recursively axiomatizable true extension of $\mathbf{PA}$. Then T is incomplete.

Proof.

- For every x, there is a shortest Turing machine that computes x.
- Therefore there are infinitely many minimal numbers.
- But in $\mathbf{PA}$, or any true recursively axiomatizable extension of $\mathbf{PA}$, only finitely many of them are provably minimal.
- Hence there are infinitely many formulas of the form $Min(\bar{m})$ that are true but not provable in T.

That completes the proof.

Discussion

Chaitin's proof of the First Incompleteness Theorem is quite different from either Turing's or Gödel's proof. The evidence for this claim is

- There is no diagonal method in sight.
- The unprovable formulas $Min(\bar{m})$ are of more complex logical form than those produced by Turing or Gödel, requiring both $\exists$ and $\forall$. Just so you can check that point I repeat the definition of Min:

$$Min(x) := \exists e\,(\exists k\,(\mathbb{T}(e,0,k) \wedge U(k) = x) \wedge$$

$$\forall d < e(\forall j(\mathbb{T}(d,0,j) \supset U(j) \neq x))$$

The second incompleteness theorem implies Chaitin's theorem

- We showed that Chatin's theorem implies the first incompleteness theorem.
- On the other hand, Chaitin's incompleteness theorem can easily be derived from the Second Incompleteness theorem itself.
- The following proof yields (in principle) an explicit bound on the provably minimal programs of T in terms of the function that enumerates the axioms of T.
- Chaitin's proof also yields such a bound, but not quite so explicitly.
- The value of the following proof is that it gives a direct connection between Con_T and a bound on the provably minimal programs of T.
- I do not think this proof has appeared in the literature.

The Second Incompleteness theorem implies Chaitin's Theorem

Proof. Let T be a true recursively axiomatizable theory. Let M be an integer such that $T \vdash Min(\bar{M})$. Since T is a true theory, M is indeed a minimal program. Let t be the integer computed by M. Define

$$f(k) := \begin{cases} 1 & \text{if } \mathrm{Prf}_T(k, \overline{\ulcorner 0 = 1 \urcorner}) \\ \text{undefined} & \text{otherwise} \end{cases}$$

Let e be an index of f and let c_t be the constant function with value t. Define

$$g(x) := c_t(\mu k\, \mathbf{T}(e, x, k))$$

Then $g(x)$ is defined if and only if T is inconsistent, and if it is defined, it takes the value t. (The value of $g(x)$ does not depend on x.) Let m be an index of g. Then m computes t, if and only if T is inconsistent. Formalizing the argument so far, we find

$$T \vdash \neg \mathrm{Con}_T \equiv \exists k\, (\mathbb{T}(\bar{m}, 0, k) \wedge U(k) = t)$$

Proof continued

We proved

$$T \vdash \neg \mathrm{Con}_T \equiv \exists k \, (\mathbb{T}(\bar{m}, 0, k) \wedge U(k) = t) \tag{1}$$

Now assume, for proof by contradiction, that $m < M$. Then, since $T \vdash Min(\bar{M})$, T proves that m does not compute t. That is,

$$T \vdash \neg \exists k \, (\mathbb{T}(\bar{m}, 0, k) \wedge U(k) = t)$$

But then by (1), $T \vdash \mathrm{Con}_T$, contradicting the Second Incompleteness Theorem. That contradiction shows that actually $M \leq m$. In other words, m is a bound on the provably minimal programs of $\mathbf{T}$. That completes the proof.

Is Chaitin's proof really new, or just a variant of Gödel's?

I have heard it said that Chaitin's proof is not very different from Gödel's.

- Gödel's theorem produces unprovable formulas with one $\forall$ followed by bounded formulas.
- Chaitin's theorem produce formulas of the form $\forall\exists$ followed by bounded formulas.
- Gödel's theorem can be improved to show there are true unprovable formulae of the form "this diophantine equation has no solutions."
- Chaitin's theorem involving the randomness of whether Diophantine equations have infinitely many or finitely many solutions is not true with "no solutions" instead of "infinitely many solutions".
- For this reason, I submit that Chaitin's proof is really different.