Recall From Last Time

Definition: Γ is *consistent* iff $\Gamma \not\vdash$ **false**.

Completeness Theorem: If Γ is consistent then Γ is satisfiable (that is, there exists a model $\mathcal{A}$ such that $A \models \Gamma$).

Proof Outline:

- Add witnessing constants $c_{P(x)}$ for every formula $P(x)$ with one free variable.
- Extend Γ to an existentially complete theory Δ that satisfies the **Henkin Property**: If $\Delta \vdash \exists x : P(x)$, then $\Delta \vdash P(c_{P(x)})$.
- Build a model of **T** by taking the objects to be **equivalence classes** of the constants (including the witnessing constants), where two constants c_1 and c_2 are equivalent if $\Delta \vdash c_1 = c_2$.
- Check that this model is well-defined.

Versions of the Proof: There are two versions of Lecture 15, the one I wrote and the one that Prof. Immerman delivered. Both are now up on the course web site. The principal difference is that Prof. Immerman built Δ to be *formally complete* as well, which simplifies the proof somewhat. My version follows [BE] more closely.

Corollary to Completeness:

$$\Gamma \models \varphi \qquad \Leftrightarrow \qquad \Gamma \vdash \varphi$$

$$\models \varphi \qquad \Leftrightarrow \qquad \vdash \varphi$$

$$\textbf{FO-VALID} \qquad = \qquad \textbf{FO-THEOREMS}$$

Note that **FO-VALID** and **FO-THEOREMS** are statements that are true in any model of the given vocabulary. While we *constructed* a special model where *every* statement was either provably true or provably false, this is not true in general. An **FO-VALID** statement about graphs would be true for all graphs, but most interesting statements are true for some graphs and not for others.

Compactness Theorem

Just as in propositional logic, we can apply completeness to get another useful property:

Theorem 16.1 (Compactness Theorem) *Let Γ be a set of sentences. Suppose every finite subset of Γ has a model. Then Γ has a model.*

Proof: If Γ is inconsistent (meaning that $\perp$ can be proved from Γ in Fitch), then some finite subset of Γ is inconsistent because Fitch proofs are finite.

But no finite subset of Γ can be inconsistent because that set has a model and Fitch is sound.

So Γ is consistent.

By completeness, then, Γ has a model.

♠

Compactness Applications

The Compactness Theorem has surprising consequences for number theory:

$$\text{Theory}(\mathbf{N}) \;=\; \{\varphi \in \mathcal{L}(\Sigma_N) \mid \mathbf{N} \models \varphi\}$$

$$\Gamma \;=\; \text{Theory}(\mathbf{N}) \;\cup\; \{c > 0, c > 1, c > 2, c > 3, \ldots\}$$

Corollary 16.2 *This theory* Γ *has a model.*

There is a countable model of $\text{Theory}(\mathbf{N})$ *that is not isomorphic to* $\mathbf{N}$.

$\mathcal{L}(\Sigma_N)$ *cannot uniquely characterize* $\mathbf{N}$.

Proof: Every finite subset of Γ is satisfiable by $(\mathbf{N}, i)$ for i sufficiently large.

By Compactness, Γ is satisfiable. ♠

Corollary 16.3 *"Connectedness" is not expressible in the first-order language of graphs,* $\mathcal{L}(\Sigma_g)$

Proof:

Suppose that $\chi \equiv$ "I am connected."

$$\Gamma \quad = \quad \{\chi\} \ \cup \ \{\mathrm{DIST}(s,t) > 1, \mathrm{DIST}(s,t) > 2, \ldots\}$$

$$\mathrm{DIST}(x_0, x_n) > n \quad \equiv$$

$$(\forall x_1 \cdots x_{n-1}) \bigvee_{i=0}^{n-1} (x_i \neq x_{i+1} \wedge \neg E(x_i, x_{i+1}))$$

Every finite subset of Γ is satisfiable.

By Compactness, Γ is satisfiable.

This is a contradiction.

Thus "Connectedness" is not expressible in the first-order language of graphs. ♠

The Downward Lowenheim-Skolem Theorem:

Theorem 16.4 *If a set of first-order sentences* $\mathbf{T}_0$ *has any model at all, it has a* **countable** *model.*

Proof: Suppose that $\mathbf{T}_0$ has a model. By the Soundness Theorem, $\mathbf{T}_0$ must be consistent. Our proof of the Completeness Theorem thus constructs a model for $\mathbf{T}_0$ where the objects are the equivalence classes made from the witnessing constants $\{c_1, c_2, \ldots\}$.

Since there were only countably many constants to put into equivalence classes, there can be only countably many classes and thus this new model has a countable domain. ♠

The set of real numbers $\mathbf{R}$ and the set of countable binary sequences are both **uncountable**. But if we define a first-order vocabulary to talk about $\mathbf{R}$, for example, we get a first-order theory $\mathrm{Th}(\mathbf{R})$, the set of first-order sentences that are true about the actual real numbers $\mathbf{R}$. This theory has a countable model!

Thus there is a countable set, with "addition" and "multiplication" operations and a "zero" element, that satisfies exactly the same first-order sentences as does $\mathbf{R}$. Something *must* be wrong with this model, since it's not the "real" **R**, but we can't state whatever it is in first-order logic.

In first-order set theory even stranger things happen. You can prove that there are sets that are uncountable, bigger than the reals, and even bigger than that. So there is a countable model, $\mathcal{M}$, that has sets that $\mathcal{M}$ *thinks* are uncountable.

The reason this is possible is that the definition of "C is countable" is that there exists a one-to-one function from C to **N**. Thus $\mathcal{M}$ might think that a set C, that is actually countable, is uncountable because $\mathcal{M}$ doesn't have any of the functions that demonstrate that C is countable.

Formal Number Theory

$$
\begin{aligned}
\mathrm{NT}_1 &\equiv (\forall x)(\sigma(x) \neq 0) \\
\mathrm{NT}_2 &\equiv (\forall xy)(\sigma(x) = \sigma(y) \to x = y) \\
\mathrm{NT}_3 &\equiv (\forall x)(x = 0 \ \vee \ (\exists y)(\sigma(y) = x)) \\
\mathrm{NT}_4 &\equiv (\forall x)(x + 0 = x) \\
\mathrm{NT}_5 &\equiv (\forall xy)(x + \sigma(y) = \sigma(x + y)) \\
\mathrm{NT}_6 &\equiv (\forall x)(x \times 0 = 0) \\
\mathrm{NT}_7 &\equiv (\forall xy)(x \times \sigma(y) = (x \times y) + x) \\
\mathrm{NT}_8 &\equiv (\forall x)(x \uparrow 0 = 1) \\
\mathrm{NT}_9 &\equiv (\forall xy)(x \uparrow \sigma(y) = (x \uparrow y) \times x) \\
\mathrm{NT}_{10} &\equiv (\forall x)(x < \sigma(x)) \\
\mathrm{NT}_{11} &\equiv (\forall xy)(x < y \to \sigma(x) \leq y) \\
\mathrm{NT}_{12} &\equiv (\forall xy)(\neg(x < y) \leftrightarrow y \leq x) \\
\mathrm{NT}_{13} &\equiv (\forall xyz)((x < y \wedge y < z) \to x < z) \\
\mathrm{NT}_{14} &\equiv (\forall xy)(\neg(x < 0) \ \wedge \ ((\sigma(x) < \sigma(y)) \to x < y))
\end{aligned}
$$

$$\mathbf{N} \models \mathrm{NT} \quad = \quad \bigwedge_{i=1}^{14} \mathrm{NT}_i$$

NT is a set of statements in the vocabulary of number theory. As you can easily see, all of them are true for the structure **N** consisting of the actual natural numbers.

NT consists of the **first four** of the Peano axioms, together with the two parts of the inductive **definitions** of addition, multiplication, and exponentiation, and order.

But we're missing the **fifth Peano axiom**, the rule of mathematical induction. This means that there's plenty we **can't** prove from NT, including such simple things as the commutativity of addition and multiplication! We call the sentences provable from NT a **fragment** of **true number theory**.

Why don't we include the rule of induction? For one thing, in the world of first-order logic it is an **infinite set of axioms**. For every formula $P(x)$ with one free variable, we need a **separate axiom** to say that $\forall x : P(x)$ follows from $P(0)$ and $\forall x : P(x) \rightarrow P(\sigma(x))$.

There is a larger fragment of number theory called **Peano Arithmetic** that has these induction axioms. But we'll see that there are things true of **N** that can't be proved in Peano Arithmetic either. And NT will be powerful enough to allow us to prove the Incompleteness Theorem, which is our main goal here.

First let's see that NT is actually good for *something*. This theorem is number 6.1 in [P], which is a good reference if you want to learn more.

Theorem 16.5 *Let φ have no variables. Then*

$$\mathbf{N} \models \varphi \qquad \Leftrightarrow \qquad \mathrm{NT} \vdash \varphi$$

Proof: φ is a boolean combination of $t < t'$, $t = t'$.

Case 1: t and t' are numbers: $\sigma(\sigma \cdots \sigma(0) \cdots)$.

= use $\mathrm{NT}_1, \mathrm{NT}_2$

$<$ use NT_{10}, NT_{13}, NT_{14}

Case 2: t, t' use $+, \times, \uparrow$.

Use $\mathrm{NT}_4, \ldots, \mathrm{NT}_9$ to transform these to numbers. ♠

Definition 16.6 A formula $\varphi \in \mathcal{L}(\Sigma_N)$ is *bounded* iff it can be written with all quantifiers in front, and all universal quantifiers bounded. ♠

Example:

$$(\forall x < 9)(\exists y)(\forall z < 2 \uparrow (x \times y))(x \uparrow 3 + z \uparrow 3 \neq 17)$$

Remark: If $\varphi(v)$ is bounded and has only one free variable, v, then S_φ is r.e., where,

$$S_\varphi \quad = \quad \{n \in \mathbf{N} \mid \mathbf{N} \models \varphi(n)\} .$$

Note that the bounded sentences are not closed under negation!

They are sentences that you can check by (a) naming numbers and (b) doing sequences of tests that are guaranteed to finish.

This is *reminiscent* of Bloop but not the same thing, because there is no equivalent in Bloop for the unbounded $\exists$. A proof can name a number, but a Bloop program can't look for it without a limit on how far it may look.

NT is strong enough to deal with these sentences as well, as we see from the following ([P]'s Theorem 6.2):

Theorem 16.7 *Let φ be a bounded sentence (a bounded formula with no free variables). Then*

$$\mathbf{N} \models \varphi \qquad \Leftrightarrow \qquad \text{NT} \vdash \varphi \ .$$

Proof:

$\Leftarrow$: Since Fitch is sound and NT is true in **N**, everything we prove from NT is also true in **N**.

$\Rightarrow$: We use induction on the number of quantifiers in φ.

Assume: $\mathbf{N} \models \varphi$.

Base case: A sentence with no quantifiers has no variables, so we are done by the previous theorem.

Inductive step: $\varphi \equiv (\exists x)\psi$.

Thus, $\mathbf{N} \models \psi[x \leftarrow n]$, for some particular $n \in \mathbf{N}$.

Thus, NT $\vdash \psi[x \leftarrow n]$ by the inductive hypothesis

Thus, NT $\vdash \varphi$ by $\exists$-Intro.

Other Inductive step: $\varphi \equiv (\forall x < t)\psi$.

The term t must be **closed**, meaning that no variables occur in it. So by unrolling the operations, NT $\vdash t = n$ for some particular $n \in$ **N**.

$$\text{NT}_{10}, \text{NT}_{11}, \text{NT}_{14} \vdash (x < n \to x = 0 \vee x = 1 \vee \cdots \vee x = n-1)$$

$$\mathbf{N} \models \psi[x \leftarrow i], \quad i = 0, \ldots, n-1$$

$$\text{NT} \vdash \psi[x \leftarrow i], \quad i = 0, \ldots, n-1$$

Now we can do a big $\wedge$-elim, with $x \geq t$ as the other case, to get:

$$\text{NT} \vdash \varphi$$

Definition 16.8 Let $f : \mathbf{N}^k \to \mathbf{N}$.
Formula φ_f **represents** f iff forall $n_1, \ldots, n_k, m \in$ **N,**

$$f(n_1, \ldots, n_k) = m \qquad \Leftrightarrow \qquad \mathbf{N} \models \varphi_f(n_1, \ldots, n_k, m)$$

♠

Example: $f(n) = n^2 + 1$.
Let $\varphi_f(n, m) \equiv (n \times n) + 1 = m$.

$$\varphi_f(n, m) \quad \Leftrightarrow \quad (n \times n) + 1 = m \quad \Leftrightarrow \quad f(n) = m$$

What Does This Mean?

If $\varphi_f(n, m)$ is a bounded formula, then so is $\exists m : \varphi_f(n, m)$, which says that f is defined on input n. Remember that NT can prove any true bounded **sentence**. So while it may not be able to prove $\forall n : \exists m : f(n, m)$ ("f is well-defined"), it can prove that $f(n)$ exists for any particular n.

What other functions can we represent? All our tools for talking about sequences, for starters:

Lemma 16.9 *The primitive recursive functions* Prime, PrimeF, IsSeq, Length, *and* Item *are each representable by bounded formulas.*

Proof:

PrimeF(n, p) asserts that p is prime number n , by asserting that there exists a number,

$$s \quad = \quad 2^0 \times 3^1 \times 5^2 \times 7^3 \times 11^4 \times \cdots \times p^n$$

$$\begin{aligned}
x|y \;&\equiv\; (\exists z < y+1)(x \times z = y) \\
\text{DE}(x, e, y) \;&\equiv\; x^e|y \;\wedge\; x^{e+1} \;\not| y \\
\text{Prime}(x) \;&\equiv\; x > 1 \;\wedge\; (\forall y, z < x)(y \times z \neq x) \\
\text{PrimeF}(n, p) \;&\equiv\; (\exists s)(\text{Prime}(p) \;\wedge\; 2 \;\not| s \;\wedge\; \text{DE}(p, n, s) \;\wedge \\
&\qquad (\forall q \leq p)(\forall q' < q)(\neg\text{Prime}(q) \\
&\qquad \vee \neg\text{Prime}(q') \\
&\qquad \vee (\exists q'' < q)(q' < q'' \wedge \text{Prime}(q'')) \\
&\qquad \vee (\exists e < q)(\text{DE}(q', e, s) \wedge \text{DE}(q, e+1, s))))
\end{aligned}$$

$$\begin{aligned}\text{IsSeq}(x) \;\equiv\; & (\exists z < x)(\forall i < x)(\exists p < 2 \uparrow x)(\text{PrimeF}(i,p) \;\wedge \\ & \quad ((i \leq z+1 \;\wedge\; p|x) \;\vee\; (i > z+1 \;\wedge\; \neg p|x)))\end{aligned}$$

$$\begin{aligned}\text{Length}(x,\ell) \;\equiv\; & (\exists k,p,q)(\text{IsSeq}(x) \;\wedge\; k+1=\ell \\ & \wedge \text{PrimeF}(k,p) \\ & \wedge \text{PrimeF}(\ell,q) \;\wedge\; p|x \;\wedge\; q \not| x)\end{aligned}$$

$$\begin{aligned}\text{Item}(x,i,e) \;\equiv\; & (\exists p)(\text{IsSeq}(x) \;\wedge\; \text{PrimeF}(i,p) \\ & \wedge \text{DE}(p,e+1,x))\end{aligned}$$

♠

It's not a coincidence that these formulas are all primitive recursive and all representable by bounded formulas:

Theorem 16.10 *Every primitive recursive function is representable by a bounded formula.*

Proof: I'd make this an exercise, but it's in Lecture 15 of the Spring 2003 notes. You might think we could have skipped the clever proofs for the individual functions above. But the proof of *this* theorem uses sequences to deal with primitive recursion. We have to do the work of coding sequences as numbers someplace. ♠

Bloop, Floop, and Bounded Formulas

Note: In fact whenever we use an $\exists$ quantifier in the proofs above, with some more effort we could have made it a *bounded* $\exists$ quantifier. A function is Bloop-computable iff it is represented by a formula where *both kinds* of quantifiers are bounded. (It should be pretty clear that a Bloop program can test the truth of such a "completely bounded" formula.)

On HW#5 I'll have you prove that a function is **Floop-computable** iff it is represented by a bounded formula as we have defined it here, i.e., a $\forall$-bounded formula. The intuition should be pretty clear – in Floop you can look for the number satisfying the unbounded $\exists$, and with the bounded formula you can use $\exists$ to express that the `while` loops will terminate.

But note that all we are going to need to prove Gödel's theorem is that being Bloop-computable (primitive recursive) *implies* representability by a bounded formula.

Now What is This Good For?

Remember the most complicated primitive recursive predicate we looked at earlier – COMP(n, x, c, y), meaning that c is a halting computation of the Turing machine M_n on input x, and that its output is y.

Theorem 16.11 *COMP(n, x, c, y) is represented by a bounded formula.*

Proof: This would follow from the general theorem about primitive recursive functions, of course. But even without that, you can go back to Lecture 9 where we proved that COMP is a primitive recursive predicate by expressing it in terms of the sequence operations. This argument essentially constructs the bounded formula directly – I'll have you follow up on the details in HW#5. ♠

Now things get interesting!

Corollary 16.12 *K is representable by a bounded formula.*

Proof:

$$\varphi_K(n) \equiv (\exists c)(\text{COMP}(n, n, c, 1))$$

♠

By our earlier results about NT:

$$K \;=\; \{n \mid \mathbf{N} \models \varphi_K(n)\}$$

$$K \;=\; \{n \mid \text{NT} \vdash \varphi_K(n)\}$$

Our standard unsolvable problem can be defined in terms of NT.

Definition 16.13 For a structure $\mathcal{A} \in \text{STRUC}[\Sigma]$,

$$\text{Theory}(\mathcal{A}) \quad = \quad \{\varphi \in \mathcal{L}(\Sigma) \mid \mathcal{A} \models \varphi\}$$

♠

$$\text{Theory}(\mathbf{N}) \quad = \quad \{\varphi \in \mathcal{L}(\Sigma_N) \mid \mathbf{N} \models \varphi\}$$

Thus Theory$(\mathbf{N})$ is *true number theory*.

Theorem 16.14 (Gödel's Incompleteness Theorem) *There is no r.e. set of sentences* Γ *such that*

1. $\mathbf{N} \models \Gamma$, *and*
2. $\Gamma \vdash \text{Theory}(\mathbf{N})$.

"There is no axiomatization of number theory, much less all of mathematics."

Proof: Let Γ be r.e. and $\mathbf{N} \models \Gamma$.

$$S \quad = \quad \{n \in \mathbf{N} \mid \Gamma \vdash \neg\varphi_K(n)\}$$

S is r.e. and $S \subseteq \overline{K}$. (Why the latter? Because if a number n were in K, we would have NT $\vdash \varphi_K(n)$, which is impossible because NT and Γ are both true in $\mathbf{N}$.)

Intuitively, $\quad S \quad = \quad \{n \in \mathbf{N} \mid \Gamma \vdash n \in \overline{K}\}$

S is an r.e. subset of the non-r.e. set $\overline{K}$. It can't be equal to $\overline{K}$, and in fact it has to miss infinitely many elements. (Since if it missed only finitely many, S plus those elements would still form an r.e. set.)

So there exist infinitely many $n \in \mathbf{N}$ s.t.,

$$\mathbf{N} \models \neg\varphi_K(n) \quad \text{and} \quad \Gamma \not\vdash \neg\varphi_K(n) \qquad \spadesuit$$

Actually [P] states this result in the following form:

[P] Theorem 6.3: *The set of unsatisfiable sentences and the set of sentences provable from* **NT** *are recursively inseparable.*

Thus a recursive set not only cannot separate true number theory from false number theory, but can't even include all the true bounded formulas without letting in something inconsistent.

[P] has proved that the sets $\{M\colon M$ outputs "yes" on $\epsilon\}$ and $\{M\colon M$ outputs "no" on $\epsilon\}$ are recursively inseparable. (Try to show this yourself!)

Look at the sentence "**NT** holds and there is an accepting computation of M on ϵ". If M says "yes", this is provable from **NT**. If M says no, it is inconsistent, because it says that the computation says "no" while **NT** can prove that it says "yes".

Sketch of Gödel's Original Proof

- Encode symbols as natural numbers.
- Encode formulas as finite sequences of natural numbers.
- Encode proofs as finite sequences of formulas.
- Let Γ be a primitive recursive axiomatization of some portion of mathematics including number theory. The following predicates are primitive recursive and thus first-order definable in $\mathcal{L}(\Sigma_N)$.
  - Formula(x): "x is the number of a formula"
  - Axiom(x): "x is the number of an axiom"
  - Proof(x): "x is the number of a proof"
  - Theorem(x): "x is the number of a theorem"
- Let $R_0, R_1, \ldots$ list all first-order formulas with one free variable, i.e., first-order definable sets.
- Let $G \quad = \quad \{n \mid \neg\,\text{Theorem}(R_n(n))\}$
- $G = \{n \mid R_q(n)\}$ for some q
- $R_q(q) \;\equiv\; \neg\,\text{Theorem}(R_q(q)) \;\equiv\;$ "I am not a theorem"
- If $R_q(q)$ then $\Gamma \not\vdash R_q(q)$; If $\neg R_q(q)$ then $\Gamma \vdash R_q(q)$.

Theorem 16.15 FO-THEOREMS *is* **r.e.** *complete.*

Proof: We have already seen that FO-THEOREMS is **r.e.**.

Recall that K is represented by a bounded formula φ_K.

$$n \in K \quad \Leftrightarrow \quad \mathbf{N} \models \varphi_K(n) \quad \Leftrightarrow \quad \text{NT} \vdash \varphi_K(n)$$

$$n \in K \qquad \Leftrightarrow \qquad \text{“NT} \rightarrow \varphi_K(n)\text{”} \in \text{FO-THEOREMS}$$

We have shown $K \leq$ FO-THEOREMS, by defining f so that:

$$f(n) = \text{“NT} \rightarrow \varphi_K(n)\text{”}$$

♠