Definition: Γ is *consistent* iff $\Gamma \nvdash$ **false**

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

Corollary:

$$\Gamma \models \varphi \quad \Leftrightarrow \quad \Gamma \vdash \varphi$$

$$\models \varphi \quad \Leftrightarrow \quad \vdash \varphi$$

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

Note that **FO-VALID** and **FO-THEOREMS** are statements that are true in any model of the given structure. 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: If every finite subset of Γ has a model, then Γ has a model.

$$\mathrm{NT}_1 \equiv (\forall x)(\sigma(x) \neq 0)$$

$$\mathrm{NT}_2 \equiv (\forall xy)(\sigma(x) = \sigma(y) \rightarrow x = y)$$

$$\mathrm{NT}_3 \equiv (\forall x)(x = 0 \ \vee \ (\exists y)(\sigma(y) = x))$$

$$\mathrm{NT}_4 \equiv (\forall x)(x + 0 = x)$$

$$\mathrm{NT}_5 \equiv (\forall xy)(x + \sigma(y) = \sigma(x + y))$$

$$\mathrm{NT}_6 \equiv (\forall x)(x \times 0 = 0)$$

$$\mathrm{NT}_7 \equiv (\forall xy)(x \times \sigma(y) = (x \times y) + x)$$

$$\mathrm{NT}_8 \equiv (\forall x)(x \uparrow 0 = 1)$$

$$\mathrm{NT}_9 \equiv (\forall xy)(x \uparrow \sigma(y) = (x \uparrow y) \times x)$$

$$\mathrm{NT}_{10} \equiv (\forall x)(x < \sigma(x))$$

$$\mathrm{NT}_{11} \equiv (\forall xy)(x < y \rightarrow \sigma(x) \leq y)$$

$$\mathrm{NT}_{12} \equiv (\forall xy)(\neg(x < y) \leftrightarrow y \leq x)$$

$$\mathrm{NT}_{13} \equiv (\forall xyz)((x < y \wedge y < z) \rightarrow x < z)$$

$$\mathrm{NT}_{14} \equiv (\forall xy)(\neg(x < 0) \ \wedge \ ((\sigma(x) < \sigma(y)) \rightarrow x < y))$$

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

The statements of NT are all true, and can be used to prove a *fragment* of "true number theory".

Theorem 6.1 [Papa]: 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, t' 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 15.1 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,

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

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.

Theorem 6.2 [Papa]: Let φ be a bounded sentence, i.e., no free variables. Then, $\mathbf{N} \models \varphi \quad \Leftrightarrow \quad \text{NT} \vdash \varphi$.

Proof:

$\Leftarrow$: Soundness Theorem.

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

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

Base case: Theorem 6.1

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

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

Thus, $\text{NT} \vdash \psi[x \leftarrow n]$

Thus, $\text{NT} \vdash \varphi$.

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

t is a closed term, thus, $\mathrm{NT} \vdash t = n$, for some $n \in \mathbf{N}$.

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

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

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

$$\mathrm{NT} \vdash \varphi$$

Definition 15.2 Let $f : \mathbf{N}^k \to \mathbf{N}$.

Formula φ_f **represents** f iff forall $n_1, \ldots, n_k, m \in \mathbf{N}$,

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

♠

Example: $f(n) = n^2 + 1$.

Let $\varphi_f(n, m) \equiv (n \times n) + 1 = m$.

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

Lemma 15.3 *The following 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,

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

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

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

$$\text{length}(x,\ell) \equiv (\exists k,p,q)(\text{IsSeq}(x) \wedge k+1=\ell \wedge \text{PrimeF}(k,p) \wedge \text{PrimeF}(\ell,q) \wedge p|x \wedge q \not| x)$$

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

♠

Theorem 15.4 *Every primitive recursive function is representable by a bounded formula.*

Proof:

Base case: Obvious for the initial functions.

Inductive step: Composition.

$f(x_1, \ldots, x_k) \quad = \quad h(g_1(x_1, \ldots, x_k), \ldots, g_m(x_1, \ldots, x_k))$

By inductive assumption h and the g_i's are representable.

$\varphi_f(\bar{x}, z) \quad \equiv$

$(\exists y_1, \ldots, y_m)(\varphi_{g_1}(\bar{x}, y_1) \wedge \cdots \wedge \varphi_{g_m}(\bar{x}, y_m) \wedge \varphi_h(y_1, \ldots, y_m, z))$

Inductive step: Primitive Recursion.

$$\begin{aligned} f(0, y_1, \ldots y_k) &= g(y_1, \ldots, y_k) \\ f(n+1, y_1, \ldots y_k) &= h(f(n, y_1, \ldots, y_k), n, y_1, \ldots, y_k) \end{aligned}$$

By inductive assumption g and h are representable.

f is representable as follows,

$$\begin{aligned} \varphi_f(x, \bar{y}, z) \quad \equiv \quad & (\exists s)(\forall i < x)(\exists a, b < s)(\text{IsSeq}(s) \\ & \wedge \text{Item}(s, i, a) \ \wedge \ \text{Item}(s, i+1, b) \\ & \wedge (i = 0 \ \rightarrow \ \varphi_g(\bar{y}, a)) \\ & \wedge \varphi_h(a, i, \bar{y}, b) \ \wedge \ \text{Item}(s, x, z)) \end{aligned}$$

♠

Bloop, Floop, and Bounded Formulas

Note: In fact whenever we use an ∃ quantifier in this proof, with some more effort we could make it a *bounded* ∃ 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.)

A formula is Floop-computable iff it is represented by a ∀-bounded formula as we have defined it here.

I'm not going to prove these assertions – all we need to prove Gödel's theorem is that Bloop-computable (primitive recursive) *implies* representability by a bounded formula.

A Fact We Didn't Prove This Time:

The primitive recursive predicate, COMP(n, x, c, y), meaning that c is a halting computation of Turing machine M_n on input x and its output is y.

How to prove it? Write Bloop programs for PrimeF, IsSeq, Item, and so forth. Then write a Bloop program that interprets c as a sequence of configurations and tests that each configuration follows from the one before according to the rules of the machine M_n. Section 6.2 of [P] in effect does this – I'll just appeal to the intuition I hope you've formed about Bloop.

Corollary 15.5 *K is representable by a bounded formula.*

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

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

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

$$\text{NT} \quad = \quad \bigwedge_{i=1}^{14} \text{NT}_i$$

Definition: 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.

Theorem 6.2 [Papa]: Let φ be a bounded sentence. Then

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

Definition: φ_f *represents* f iff forall $n_1, \ldots, n_k, m \in$ **N**,

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

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

Corollary: K is representable by a bounded formula.

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

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

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

Definition 15.6 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 $\text{Theory}(\mathbf{N})$ is *true number theory*.

Theorem 15.7 (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 $S \subseteq \overline{K}$.

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.,

$$\mathbf{N} \models \neg\varphi_K(n) \quad \text{and} \quad \Gamma \nvdash \neg\varphi_K(n) \qquad \spadesuit$$

[P] states this result in the following form:

proves this in the form of his Theorem 6.3:

[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.

Recall that the sets $\{M\colon M$ outputs "yes" on $\epsilon\}$ and $\{M\colon M$ outputs "no" on $\epsilon\}$ are recursively inseparable.

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".

- 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 axiomitization 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 $G = \{n \mid \neg \text{Theorem}(R_n(n))\}$
- $G = \{n \mid R_q(n)\}$ for some q
- $R_q(q) \equiv \neg \text{Theorem}(R_q(q)) \equiv$ "I am not a theorem"
- If $R_q(q)$ then $\Gamma \nvdash R_q(q)$; If $\neg R_q(q)$ then $\Gamma \vdash R_q(q)$.

Theorem 15.8 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:

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

♠