Arithmetization of syntax:

- Completeness theorem says: there is as-good-as-possible correspondence between truth-in-models (semantic notion) and proofs (syntactic notion)

- Since formulas and proofs are syntactic, would like to define what it means for sets of formulas/pfs to be recursive, r.e., etc.
  $\rightsquigarrow$ a key step toward proving incompleteness theorems
- need to code fmlas/pfs as natural numbers! (effectively)

- though we'll be interested primarily in lang. of arithmetic, we develop coding wrt a general countable language (i.e. ctbly many R/f/c symbols)

- fix $L = \{R_i\} \cup \{f_j\} \cup \{c_k\}$

- we'll write $\langle \cdots \rangle$ to mean "code for"
- say $\langle \cdots \rangle$ is Gödel number of ...

- recall our lang. also includes logical symbols
- for convenience (given our deductive system) let's assume these are $\neg, \Rightarrow, \forall, (,),$ "comma", $=$, and the variables $x_0, x_1, \ldots$

Coding symbols:

- we code non-variable logical symbols as binary codes w/ a leading 0.

$\langle \neg \rangle := \langle 0, 0 \rangle$ — abuse of $\langle , \rangle$'s here

$\langle \Rightarrow \rangle := \langle 0, 1 \rangle$ — by $\langle a, b \rangle$ we mean

$\langle \forall \rangle := \langle 0, 2 \rangle$ in old sense: $2^{a+1} 3^{b+1}$

$\vdots$

$\langle = \rangle = \langle 0, 6 \rangle$

- code x_n, R_i, f_j, c_k as follows:

$\langle R_n \rangle := \langle 0, 7+4n \rangle$

$\langle f_n \rangle := \langle 0, 7+4n+1 \rangle$

$\langle c_n \rangle := \langle 0, 7+4n+2 \rangle$

$\langle x_n \rangle := \langle 0, 7+4n+3 \rangle$

Coding terms: Inductively, if $t = f(t_1, \ldots, t_m)$ and t_i are already encoded terms define

$$\langle t \rangle = \langle 1, \langle f \rangle, \langle t_1 \rangle, \ldots, \langle t_m \rangle \rangle$$

Coding formulas:

Inductively define:

$\langle R(t_1, \ldots, t_m) \rangle := \langle 2, \langle R \rangle, \langle t_1 \rangle, \ldots, \langle t_m \rangle \rangle$

$\langle t = s \rangle := \langle 2, \langle = \rangle, \langle t \rangle, \langle s \rangle \rangle$

then:

$\langle \neg \varphi \rangle := \langle 2, \langle \neg \rangle, \langle \varphi \rangle \rangle$

etc. for $\langle \varphi \Rightarrow \psi \rangle$, $\langle \forall x_i \varphi \rangle$

Coding sequences of fmlas:

If $\varphi_0, \ldots, \varphi_n$ is a sequence of fmlas define

$\langle \varphi_0, \ldots, \varphi_n \rangle := \langle \langle \varphi_0 \rangle, \ldots, \langle \varphi_n \rangle \rangle$

no ambiguity w/ prev. codes cause $\langle \varphi_0 \rangle$ always > 2.

Now we define the relations:

Var(n)	iff	n =	Gödel number of a	variable
Term(n)	"	"	"	term
Form(n)	"	"	"	formula
Sent(n)	"	"	"	sentence
LogicAx(n)	"	"	"	logical axiom

MP(k, l, m) iff Form(k) $\wedge$ Form(l) $\wedge$ Form(m) $\wedge$ formula coded by m follows from fmlas coded by k, l by MP

Proof(n) iff n is Gödel # of a proof (in our deductive system)

- These sets depend on the particular language we use.
- But for a given L they are "clearly" recursive (and in fact primitive recursive)
- Hence in particular they are definable in $\mathcal{N} := \langle \mathbb{N}, <, S, +, \cdot, 0 \rangle$

For example, Proof(n) holds iff

$$\mathrm{Seq}(n) \wedge \forall i < \mathrm{length}(n) \, [\mathrm{Form}((n)_i) \wedge (\mathrm{LogicAx}((n)_i) \vee \exists j,k < i (\mathrm{MP}((n)_j, (n)_k, (n)_i)))]$$

which is p.r.... assuming Form, LogicAx, MP are p.r.
... even for Term need a nontrivial prim recursion.

<u>Def'n</u> Sps Φ is a collection of fmlas. Say Φ is <u>recursive</u> if $\langle \Phi \rangle := \{ \langle \varphi \rangle : \varphi \in \Phi \}$ is recursive (as a subset of $\mathbb{N}$)

<u>Def'n</u> Given a set of fmlas Φ define $\mathrm{Proof}_\Phi(n)$ iff n is a code for a proof from the fmlas in Φ

i.e.

$$\mathrm{Proof}(n) \wedge \forall i < \mathrm{length}(n) \, [\mathrm{LogicAx}((n)_i) \vee (n)_i \in \langle \Phi \rangle \vee \exists j,k < i \, \mathrm{MP}((n)_j, (n)_k, (n)_i)]$$

<u>Observe</u>: Proof_Φ is recursive if Φ is recursive

<u>Def'n</u> Sps Σ is a collection of fmlas
Define: $\mathrm{Conseq}(\Sigma) := \{\varphi : \Sigma \vdash \varphi\}$

<u>Def'n</u> We say that a set of fmlas Σ is <u>complete</u> if for every φ we have $\Sigma \vdash \varphi$ or $\Sigma \vdash \neg\varphi$
(i.e. either $\varphi \in \mathrm{Conseq}(\Sigma)$ or $\neg\varphi \in \mathrm{Conseq}(\Sigma)$)

<u>Prop'n</u> Sps Σ is a collection of fmlas. If Σ is recursive then $\mathrm{Conseq}(\Sigma)$ is r.e.

Moreover, if Σ is complete then $\mathrm{Conseq}(\Sigma)$ is recursive.

Pf: - Observe $n \in \langle \mathrm{Conseq}(\Sigma) \rangle$ iff $\exists m\,(\mathrm{Proof}_\Sigma(m) \wedge (m)_{\mathrm{length}(m)-1} = n)$

- Hence $\langle \mathrm{Conseq}(\Sigma) \rangle$ is r.e. since Proof_Σ is recursive (as Σ is recursive) ✓

- Now sps further Σ is complete.

- If Σ proves both φ and $\neg\varphi$ for some φ we know $\mathrm{Conseq}(\Sigma) =$ all fmlas, which is recursive

- So sps Σ is consistent

- then by completeness we have $\varphi \in \mathrm{Conseq}(\Sigma)$ iff Σ cannot prove $\neg\varphi$

i.e.

$n \in \langle \mathrm{Conseq}(\Sigma) \rangle$ iff $\neg\exists m\,(\mathrm{Proof}_\Sigma(m) \wedge (m)_{\mathrm{length}(m)-1} = \mathrm{neg}(n))$

where neg(n) denotes code of negation of fmla coded by n (neg is p.r.)

$\Rightarrow \langle Conseq(\Sigma)\rangle$ is co-r.e. in this case
$\Rightarrow \langle Conseq(\Sigma)\rangle$ is recursive in this case (since by above, already r.e.) ✓

- recall $\mathcal{N} = \langle \mathbb{N}, <, S, +, \cdot, 0\rangle$

<u>Def'n</u>: $Th(\mathcal{N}) := \{\varphi : \mathcal{N} \models \varphi\}$

sentences

$Th(\mathcal{N})$ is the "theory of $\mathcal{N}$"

- the following is a famous theorem of Tarski that says $Th(\mathcal{N})$ is a (very) complicated set (... once encoded)
- Here we assume the language $L = \{<, S, +, \cdot, 0\}$ is the usual lang. of arithmetic, but the same pf works for any extension of L (over $\mathbb{N}$)

<u>Theorem (Tarski)</u> $\langle Th(\mathcal{N})\rangle$ is not an arithmetical set.

- Tarski's theorem sometimes marketed as "truth is undefinable in $\mathcal{N}$"

i.e. ~~xxxx~~ is equiv to: there is no formula $\Phi(x)$ (written in L) s.t. $\mathcal{N} \models \Phi(\langle\varphi\rangle)$ iff $\mathcal{N} \models \varphi$ (why?)

- another (softer) description: the collection of codes of true statements about $\mathcal{N}$ is "extremely far" from being recursive.

- to prove, need following:

<u>Lemma</u>: There is no universal arithmetical set, i.e. there is no binary rel'n $U \subseteq \mathbb{N}^2$ with $U \in \bigcup \Delta^0_n$ s.t. for every arithmetical new $B \subseteq \mathbb{N}$ there is e s.t. $m \in B$ iff $(e, m) \in U$.

used diagonal

<u>Pf</u>: since each Δ^0_n is closed under complements, so is $\bigcup \Delta^0_n$

- if we had such a U, then $W = \{m : (m,m) \in U\}$ is arithmetical (why?)
- hence so is W^c
- hence $\exists e$ s.t. $m \in W^c$ iff $(e,m) \in W$
- then ~~xxxx~~ $e \in W^c$ iff $(e,e) \in W$ iff $(e,e) \notin W$ contr. ✓