Logic and Computability Notes

Autumn Mapes

Fall 2025

1 Mathematical Logic Foundations

Def: Binary Relation

and when it is said to contain another relation

Given some set X, $X \times X$ is said to be a relation on X. It contains another relation Y if for every $(a, b) \in Y$, $(a, b) \in X$.

It's an equivalence relation when it is reflexive, symmetric, and transitive.

Note that any intersection of equivalence relations always is an equivalence relation.

Card ID: 1775594407749

Def: Strict Order Axioms

1. Irreflexivity: $\forall x,\ x \not< x$
2. Asymmetry: $\forall a, b,\ a < b \implies b \not< a$
3. Transitivity: $\forall a, b, c,\ (a < b) \wedge (b < c) \implies a < c$

Card ID: 1777460526347

Def: Axiom of Extensionality

Two sets are equal if they have the same elements.

Card ID: 1780146565762

Def: Axiom of Separation

We can define sets as we normally do:

$$A = \{x \in S : \varphi(x)\}$$

Card ID: 1780146565764

Def: Axiom of Pairing

If you have two sets, there exists a set that contains those two sets:

$$\{A, B\}$$

Card ID: 1780146565766

Def: Axiom of Choice

Suppose you have a set X of nonempty sets. Then there exists a function f that maps each $x \in X$ to one of its elements.

Card ID: 1780146565768

Def: Statement of Diaconescu's Theorem

We can deduce the Law of Excluded Middle ($\forall \varphi,\ \varphi \vee \neg\varphi$) from the Axion of Choice in a minimal intuitionistic setting.

Card ID: 1780146565770

Prop

Diaconescu's Theorem

Want to show $\forall \varphi,\ \varphi \vee \neg\varphi$; fix φ.
Use the axiom of seperation to get:

$$A = \{x \in \{0,1\} : \varphi \vee (x = 0)\}$$

$$B = \{x \in \{0,1\} : \varphi \vee (x = 1)\}$$

Use the Axiom of Pairing to get $\{A, B\}$. Because $0 \in A$ and $1 \in B$, A and B are inhabited, so we have a choice function

$$f : \{A, B\} \to A \cup B$$

with $f(A) \in A$ and $f(B) \in B$.
So we have a proof of

$$(f(A) = 0 \vee \varphi) \wedge (f(B) = 1 \vee \varphi)$$

By intuistic logic, we can split into four cases:

$$f(A) = 0,\ f(B) = 0$$

$f(B) = 0$ implies φ above.

$$f(A) = 0,\ f(B) = 1$$

Hard case, see below.

$$f(A) = 1, f(B) = 0$$

$f(A) = 1$ implies φ above.

$$f(A) = 1, f(B) = 1$$

$f(A) = 1$ implies φ above.

Hard case: $f(A) = 0,\ f(B) = 1$. Prove $\neg\varphi = \varphi \to \bot$ by the following:
Assume φ.
By the Axiom of Extensionality, $A = B$
But $f(A)$ and $f(B)$ don't have the same value!

$$0 = f(A) = f(B) = 1$$

Contradiction, we have $\neg\varphi$

Card ID: 1780146565773

2 Foundations of Intuitionistic Logic

What do we mean by $\Gamma \vdash \varphi$?

In the context Γ, we have a proof of ρ.

Card ID: 1769802659702

What do we mean by $\Gamma_1 A \vdash B$?

Adding A to the context Γ, we have a proof of B.

Card ID: 1769802659708

($\wedge$ - I)

$$\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \wedge B}$$

Card ID: 1769802659710

($\vee$ - I)

$$\frac{\Gamma \vdash A}{\Gamma \vdash A \vee B} \text{ and } \frac{\Gamma \vdash B}{\Gamma \vdash A \vee B}$$

Card ID: 1769802659712

($\wedge$ - E)

$$\frac{\Gamma \vdash A \wedge B}{\Gamma \vdash A} \text{ and } \frac{\Gamma \vdash A \wedge B}{\Gamma \vdash B}$$

Card ID: 1769802659714

($\vee$ - E)

$$\frac{\Gamma \vdash A \vee B \quad \Gamma_1 A \vdash C \quad \Gamma_1 B \vdash C}{\Gamma \vdash C}$$

Card ID: 1769802659716

(⊥ - E)

For all A,

$$\frac{\Gamma \vdash \bot}{\Gamma \vdash A}$$

Card ID: 1769802659718

(→ - I)

$$\frac{\Gamma_1 A \vdash B}{\Gamma \vdash A \to B}$$

Card ID: 1769802659720

(→ - E)

$$\frac{\Gamma \vdash A \to B \quad \Gamma \vdash A}{\Gamma \vdash B}$$

Card ID: 1769802659722

(?x)

For all A,

$$\frac{.}{\Gamma_1 A \vdash A}$$

Card ID: 1769802659724

(LEM)

For all A,

$$\frac{.}{\Gamma \vdash A \vee (A \to \bot)}$$

Card ID: 1769802659725

Def: IQC

First-order intuitionistic logic. Includes the axioms for quantifiers (that's what the Q stands for)

Card ID: 1769802659727

What do we mean by $\varphi[x := t]$ for some term t?

Whatever φ is, with all instances of x replaced with t

Card ID: 1769802659729

($\exists$-I)

For all terms t,

$$\frac{\Gamma \vdash \varphi[x := t]}{\Gamma \vdash \exists x.\varphi(x)}$$

Card ID: 1769802659731

($\exists$-E)

$$\frac{\Gamma \vdash \exists x.\varphi \quad \Gamma_1 \varphi \vdash \psi}{\Gamma \vdash \psi}$$

Card ID: 1769802659732

($\forall$-I)

For all x not free in Γ,

$$\frac{\Gamma \vdash \varphi}{\Gamma \vdash \forall x.\varphi}$$

Card ID: 1769802659734

($\forall$-E)

For all terms t,

$$\frac{\Gamma \vdash \forall x.\varphi}{\Gamma \vdash \varphi[x := t]}$$

Card ID: 1769802659736

If we know $\Gamma \vdash \varphi$, what do we know about $\Gamma_1\psi$ and $\Gamma[p := \psi]$?

By Lemma 1.1.4, we must have $\Gamma_1\psi \vdash \varphi$

Furthermore, if we have a primitive proposition p and any proposition ψ, then $\Gamma[p := \psi] \vdash \varphi[p := \psi]$

Card ID: 1769802659739

3 The Simply-Typed λ-Calculus

3.1 Basic Grammar

What does $\mathcal{U}$ and V denote in the simply-typed λ-calculus?

$\mathcal{U}$ is a countable set of primitive types.

V is an infinite set of variables.

Card ID: 1769802659741

What does Π represent in the simply-typed λ-calculus? Define its grammar

---

Π is the set of types in λ-calculus, denoted $\lambda_\Pi(\to)$:

$$\begin{aligned}\Pi := {} & \mathcal{U} \\ & \mid \Pi \to \Pi\end{aligned}$$

(also sometimes called the implicational fragment of λ_Π)

Card ID: 1769802659743

What does λ_Π represent in the simply-typed λ-calculus? Define its grammar and label it

---

λ_Π is the set of terms in simply-typed λ-calculus:

$$\begin{aligned}\Pi := {} & V \\ & \mid \lambda V : \Pi.\lambda_\Pi && (\lambda - \text{abstraction}) \\ & \mid \lambda_\Pi \lambda_\Pi && (\lambda - \text{application})\end{aligned}$$

Card ID: 1769802659745

Def: Context (in the λ-calculus)

A set of pairs of the form $x : \sigma$
(where $x \in V$ is a variable and $\sigma \in \Pi$ is a type)

Note that we write $\Gamma_1 x : \sigma$ for $\Gamma \cup \{x : \sigma\}$

Card ID: 1770998663267

Def: Range of Γ

Denoted $|\Gamma|$, the set of all types that appear in Γ.

Card ID: 1770998663270

3.2 Typing and Evaluation

Def: Typability Relation

basic definition

$$\Vdash \subseteq C \times \lambda_\Pi \times \Pi$$

where C is the set of all contexts. Links every term in a context to a type.
Must fulfill three properties (recursing on any term in λ_Π).
This type system is denoted $\lambda_\Pi(\rightarrow)$

Card ID: 1770998663272

Def: Typing Rule for Variables

1. Type of standalone variables:

$$\Gamma_1 x : \tau \Vdash x : \tau$$

Card ID: 1771285300805

Def: Typing Rule for λ-abstractions

2. Type of λ-abstractions:

$$\Gamma_1 x : \sigma \Vdash M : \tau \implies \Gamma \Vdash (\lambda x : \sigma . M) : \sigma \rightarrow \tau$$

(assuming x does not occur in Γ, $M \in \lambda_\Pi$)

Card ID: 1771285300807

Def: Typing Rule for λ-applications

3. Type of λ-applications:

$$(\Gamma \Vdash M : \sigma \to \tau) \land (\Gamma \Vdash N : \sigma) \implies \Gamma \Vdash (MN) : \tau$$

$(M, N \in \lambda_{\Pi})$

Card ID: 1771285300809

Def: α-equivalency

Two λ-terms are α-equivalent if they differ only in the name of the bound variables.

e.g. $\lambda x : \mathbb{Z}.x^2 \equiv_{\alpha} \lambda y : \mathbb{Z}.y^2$

Card ID: 1770998663273

What does it mean for a λ-term to be closed?

It has no free variables.

Card ID: 1770998663274

Def: Substitution in λ_Π

Inductive definition

Given λ-terms M, N and a variable x, we define the substitution $M[x := N]$ by pattern matching inductively:

Case 1: M is a variable equal to x:

$$x[x := N] := N$$

Case 2: M is a variable not equal to x:

$$y[x := N] := y$$

Case 3: M is a λ-abstraction:

$$(\lambda a : \sigma.A)[x := N] := (\lambda a : \sigma.A[x := N])$$

(assuming $x \neq y$ and y not free in N)

Case 4: M is a λ-application:

$$(PQ)[x := N] := (P[x := N])(Q[x := N])$$

Card ID: 1771333210701

Def: β-Reduction

Name the two sides of the reduction

$$(\lambda x : \sigma.P)Q \to_\beta P[x := Q]$$

The left side is called the β-redex and the right side is called the β-contraction.

We also write $M \twoheadrightarrow_\beta N$ if M β-reduces to N after a finite number of reductions.

Note that this is preserved under λ-abstraction and λ-application: if $P \to_\beta P'$,

$$\lambda x : \sigma.P \to_\beta \lambda x : \sigma.P'$$

$$PZ \to_\beta P'Z$$

$$ZP \to_\beta ZP'$$

(we can choose to evaluate in any order; we'll end up in the same place either way)

Card ID: 1771333210702

Def: β-Equivalence

The smallest equivalence relation containing $\to_\beta$, denoted $\equiv_\beta$.

Card ID: 1771333210703

Def: η-reduction

Simplify pointless λ-abstractions:

$$\lambda x : \sigma.(Px) \to_\eta P$$

Card ID: 1771333210705

Def: β-Normal Form

M is in β-NF if there is no N such that $M \to_\beta N$.

Card ID: 1771333210706

When we write a λ-term like ABC, what do we actually mean?

$$(AB)C$$

Card ID: 1771333210708

Def: Free Variables Lemma

Given $\Gamma \Vdash M : \sigma$,

---

then we must have:

1. For all $\Gamma \subseteq \Gamma'$, $\Gamma' \Vdash M : \sigma$.

2. The free variables of M occur in Γ

3. There exists some $\Gamma' \subseteq \Gamma$ consisting of exactly the free variables of M, preserving $\Gamma' \Vdash M : \sigma$.

Card ID: 1771333210709

Def: Generation Lemma

Just read each type inference rule in reverse:

1. If $\Gamma \Vdash x : \sigma$,

$$x : \sigma \in \Gamma$$

2. If $\Gamma \Vdash (\lambda x : M) : \sigma$, then there exists types $\tau, \rho \in \Pi$ such that

$$\Gamma_1 x : \tau \Vdash M : \rho \quad \wedge \quad \sigma = (\tau \to \rho)$$

3. If $\Gamma \Vdash (MN) : \sigma$, then there exists types $\tau \in \Pi$ such that

$$\Gamma \Vdash M : \tau \to \sigma \quad \wedge \quad \Gamma \Vdash N : \tau$$

Card ID: 1771333210710

Def: Substitution Lemma

1. If $\Gamma \Vdash M : \sigma$ and α is some type variable, then:

$$\Gamma[\alpha := \tau] \Vdash M : \sigma[\alpha := \tau]$$

(how substitution behaves for types in a context)

2. If $\Gamma_1 x : \tau \Vdash M : \sigma$ and $\Gamma \Vdash N : \tau$, then

$$\Gamma \Vdash M[x := N] : \sigma$$

(how substitution behaves for terms in a context)

Card ID: 1771333210712

Prop

Subject Reduction

If $\Gamma \Vdash M : \sigma$ and $M \to_\beta N$, then

$$\Gamma \Vdash N : \sigma$$

(β-reduction preserves types)

Proven by induction with the generation and substitution lemmas.

Card ID: 1771333210715

Thm: Church-Rosser Theorem for λ_Π

Supposing $\Gamma \Vdash M : \sigma$, if $M \twoheadrightarrow_\beta N_1$ and $M \twoheadrightarrow_\beta N_2$, then there exists some $L \in \lambda_\Pi$ such that

$$N_1 \twoheadrightarrow_\beta L \quad \wedge \quad N_2 \twoheadrightarrow_\beta L$$

with $\Gamma \Vdash L.\sigma$.

Also called the confluence property or the diamond property. Card ID: 1771333210718

Def: Corollary to the Church-Rosser Theorem for λ_Π

If $M \in \lambda_\Pi$ admits a β-NF, then it is unique.

Card ID: 1771333210719

Give an example of a λ term that is impossible to type

$$\lambda x.xx$$

Card ID: 1771333210721

Def: The height function

A recursively defined map on types $h : \Pi \to \mathbb{N}$ mapping:

1. $v \in V$ to 0
2. $\sigma \to \tau$ to $1 + \max(h(\sigma), h(\tau))$

Card ID: 1771333210723

Thm: The Weak Normalization Theorem for $\lambda_\Pi(\to)$

Let $\Gamma \Vdash M : \sigma$. There exists a finite reduction path

$$M := M_0 \to_\beta M_1 \to_\beta \dots \to_\beta M_n$$

where M_n is a β-NF. Card ID: 1771333210725

Def: β-redex

A term we can simplify into a β-contractum:

$$(\lambda x : \sigma.p)Q \to_\beta P[x := Q]$$

Read this as 'function application candidate'

Card ID: 1780146565818

Def: Weak Normalization Theorem m function

$$m(M) := (0, 0) \text{ if } M \text{ is in beta-normal form}$$

$$m(M) := (h(M), \text{redex}(M))$$

where $h(M)$ is the greatest height of a redex in M, and $\text{redex}(M)$ is the number of occurances of redexes in M of maximal height.

(the height of a redex is the height of the type of the function being applied)

Card ID: 1780146565820

TODO skipping the 'taming the hydra' proof.

Thm: The Strong Normalization Theorem for $\lambda_\Pi(\to)$

Let $\Gamma \Vdash M : \sigma$. There exists no infinite reduction chain

$$M := M_0 \to_\beta M_1 \to_\beta \ldots$$

Card ID: 1771333210728

4 The Curry-Howard Correspondence

Prop

Curry-Howard for IPC($\to$)

Let Γ be a context and φ be a proposition (type). Then:

1. If $\Gamma \Vdash M : \varphi$, then

$$|\Gamma| = \{\tau \in \Pi : (x : \tau) \in \Gamma \text{ for some variable } x\} \vdash_{IPC} \varphi$$

2. If $\Gamma \vdash_{IPC} \varphi$, then there exists some $M \in \lambda_\Pi(\to)$ such that

$$\{(x_\psi : \psi) \mid \psi \in \Gamma\} \Vdash M : \varphi$$

Card ID: 1771333210730

TODO proof omitted

Grammar of the Full Simply-Typed Lambda Calculus Types

$$\begin{aligned}
\Pi := \mathcal{U} & \qquad \text{Primitive Types} \\
\mid \Pi \to \Pi & \qquad \text{Function Types} \\
\mid \Pi \times \Pi & \qquad \text{Product Types} \\
\mid \Pi + \Pi & \qquad \text{Coproduct Types} \\
\mid 0 & \qquad \text{Initial Type Variable} \\
\mid 1 & \qquad \text{Terminal Type Variable}
\end{aligned}$$

Card ID: 1780146565824

Grammar of the Full Simply-Typed Lambda Calculus Types

$$
\begin{aligned}
\lambda_\Pi : &\ V && \text{Variable} \\
&| \lambda V : \Pi.\lambda_\Pi && \lambda\text{-Abstraction} \\
&| \lambda_\Pi \lambda_\Pi && \lambda\text{-Application} \\
&| \langle \Pi, \Pi \rangle && \text{Product Constructor} \\
&| \Pi_1(\lambda_\Pi) \,|\, \Pi_2(\lambda_\Pi) && \text{Projection Types / Product Eliminators} \\
&| \iota_1(\lambda_\Pi) \,|\, \iota_2(\lambda_\Pi) && \text{Left and Right Coproduct Constructors} \\
&| \mathrm{Case}(\lambda_\Pi; V.\lambda_\Pi; V.\lambda_\Pi) && \text{Coproduct Eliminator} \\
&| * && \text{Terminal Type Constructor} \\
&| !_\Pi \lambda_\Pi && \text{Explosion Type}
\end{aligned}
$$

Card ID: 1780146565826

todo a bunch of obvious type constructors

Def: Explosion Typing Rule

$$\frac{\Gamma \Vdash M : 0}{\Gamma \Vdash !_\varphi M : \varphi}$$

for every type $\varphi \in \Pi$

Card ID: 1780146565828

Def: Case Beta-Reduction

$$\mathrm{Case}(\iota_1(M); x.K; y.L)) \to_\beta K[x := M]$$
$$\mathrm{Case}(\iota_2(M); x.K; y.L)) \to_\beta L[y := M]$$

Card ID: 1780146565830

Extra BHK interpretation η-reductions

$$\langle \pi_1 M, \pi_2 M \rangle \to_\eta M$$

Also if $\Gamma \Vdash M : 1$ we have $M \to_\eta *$

Card ID: 1780146565832

5 Semantics for IPC

Def: The Three Laws of Lattices

A lattice L is a set equipped with a a binary operators $\wedge$ and $\vee$ such that both operations obey:

- commutativity, and
- associativity

Plus together they must follow the absorption laws.
Note together these imply $a \vee a = a$ and $a \wedge a = a$ (this is called being idempotent)

Card ID: 1775594407776

Def: Absorption Laws of Lattices

$$a \vee (a \wedge b) = a,$$
$$a \wedge (a \vee b) = a$$

Card ID: 1776873012844

Def: Distributive Lattice

A lattice that fulfills distributivity:

$$a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$$

Note this is true iff

$$a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$$

Card ID: 1775594407777

Def: Bounded Lattice

A lattice where there exists $\bot, \top \in L$ such that:

$$a \vee \bot = a,$$

$$a \wedge \top = a$$

Note that these imply:

$$a \wedge \bot = \bot$$

$$a \vee \top = \top$$

Card ID: 1775594407779

Def: Complemented Lattice

A bounded lattice such that, for all $a \in L$, there exists some $a^* \in L$ such that:

$$a \wedge a^* = \bot,$$

$$a \vee a^* = \top$$

Card ID: 1775594407781

Def: Boolean Lattice

A lattice that is distributive, bounded, and complemented.

(Note that the structure this creates makes the complement function $*$ equivalent to $\neg$ since we must have $a \vee \neg a = \top$ and $a \wedge \neg a = \bot$. This is also where $\wedge$ and $\vee$ come from! $\vee$ prefers $\top$ and $\wedge$ prefers $\bot$)

Card ID: 1775594407782

Def: Induced Order from a Lattice

We say $a \leq b$ when $a \wedge b = a$ and equivalently $a \vee b = b$.

Card ID: 1775594407784

Prop

Induced Order of Bounded Lattices

For any bounded lattice, for the induced order, we have:

$$a \wedge b = \inf\{a, b\},$$

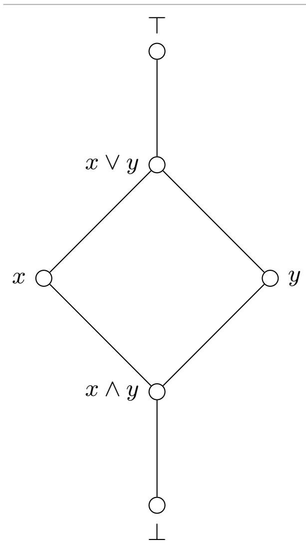
$$a \vee b = \sup\{a, b\}$$

Conversely, for any partial order, finite meets and joins form a lattice as above.

Note then that for any total order, we have $\wedge \equiv \min$ and $\vee \equiv \max$.

Card ID: 1775594407785

Draw the free distributive lattice on {x, y}



Card ID: 1775594407787

Def: Valuation

A function $v : L \to \{0, 1\}$ assigning truth values to statements in a consistent manner (respecting logical connectives)

Card ID: 1775594407789

Def: Heyting Algebra

A bounded lattice H equipped with an extra binary relation $\Rightarrow: H \times H \to H$ such that

$$a \wedge b \leq c \iff a \leq (b \Rightarrow c)$$

(So $a \Rightarrow b$ can be thought of the weakest thing you can assume that, when combined with b, gets you to c)

Card ID: 1775594407790

Def: Heyting Homomorphism

A function that preserves $\wedge$, $\vee$, and $\Rightarrow$.

Card ID: 1775594407791

Show that every Boolean algebra is a Heyting algebra

We can construct our own $\Rightarrow$ function:

$$a \Rightarrow b := (\neg a) \vee b$$

An example of this are the power sets of a set.

Card ID: 1775594407793

Show that every topology on a set X is a Heyting algebra

We construct a $\Rightarrow$ function:

$$U \Rightarrow V := \text{Int}(U^c \cup V)$$

Card ID: 1775594407794

Def: H-Valuation

Given some language L with a set of primitive propositions P, an H-valuation $v : P \to H$ is extended to L recursively by:

1. $v(\bot) = \bot$
2. $v(A \land B) = v(A) \land v(B)$
3. $v(A \lor B) = v(A) \lor v(B)$
4. $v(A \to B) = v(A) \Rightarrow v(B)$

Card ID: 1775594407795

Def: H-Valid

A proposition $A \in L$ is H-valid if, for all H-valuations v, $v(A) = \top$.

Card ID: 1775594407797

Def: H-Consequence

A proposition $A \in L$ is an H-consequence of a finite set of propositions $\Gamma \subseteq L$ if, for all valuations v, $v(\bigwedge \Gamma) \leq v(A)$.

We denote this by $\Gamma \vDash_H A$

Card ID: 1775594407799

Lemma: Soundness of Heyting Semantics

Card ID: 1775594407800

If H is a Heyting algebra and v is an H-valuation, we always have:

$$\Gamma \vdash_{IPC} A \implies \Gamma_{H,v} \vDash A$$

Proven over the structure of all IPC rules.

TODO proof of the soundness of Heyting semantics

Show that the LEM is not intuitionistically valid

Card ID: 1775594407802

By the soundness of Heyting semantics, we need only to find an example of a Heyting algebra with a valuation where $v(p \vee \neg p) \neq \top$ for some $p \in L$.

Consider the topology $\{\emptyset, \{1\}, \{1,2\}\}$ (called the Sierpinski space). Take some primitive proposition p, and define our valuation so that $v(p) = \{1\}$. Then we have

$$v(\neg p) = v(p) \Rightarrow \emptyset = \text{Int}(\{1,2\} \backslash \{1\}) = \emptyset$$

and thus

$$v(p \vee \neg p) = v(p) \vee v(\neg p) = \{1\} \neq \top = \{1,2\}$$

Def: Pierce's Law

$$((p \to q) \to p) \to p$$

This is a tautology in classical logic, but not intuitionistically valid.

Card ID: 1775594407803

Show that Pierce's Law is not intuitionistically valid.

Card ID: 1775594407804

By the soundness of Heyting semantics, we need only to find an example of a Heyting algebra with a valuation where $v(p \vee \neg p) \neq \top$ for some $p \in L$.

Take the usual topology on $\mathbb{R}^2$ with the valuation:

$$p \longmapsto \mathbb{R}^2 \backslash \{0,0\}$$
$$q \longmapsto \emptyset$$

Plugging in, you get $((p \to q) \to p) \to p = \mathbb{R}^2 \backslash \{0,0\} \neq \top = \mathbb{R}^2$.

Def: Lindenbaum-Tarski Algebra

A Lindenbaum-Tarski Algebra $F^Q(T)$ over an L-theory T is an algebra over equivalence classes of propositions where $\varphi \sim \psi$

$$T, \varphi \vdash_Q \psi \text{ and } T, \psi \vdash_Q \varphi$$

(equivalence class over $\varphi \iff \psi$)

If $\bowtie$ is any logical connective in T, we set $[\varphi] \bowtie [\psi]$ where $[\varphi \bowtie \psi]$.

Card ID: 1775594407805

Prop

Show that the LT Algebra of any theory relative to IPC $\setminus \to$ (IPC without implication) is a distributive lattice.

Commutativity and associativity are trivial, just show the two absorption laws and a distributivity law by IPC rules:

$$([A] \vee [B]) \wedge [A] = [(A \vee B) \wedge A] = [A]$$

$$([A] \wedge [B]) \vee [A] = [(A \wedge B) \vee A] = [A]$$

$$[A] \wedge ([B] \vee [C]) = [A \wedge (B \vee C)] = [(A \wedge B) \vee (A \wedge C)] = ([A] \wedge [B]) \vee ([A] \wedge [C])$$

Card ID: 1775594407807

Lemma: Show that the LT algebra of any theory relative to IPC is a Heyting Algebra

Card ID: 1775594407808

We showed that IPC $\setminus \to$ is a distributive lattice, so we just need to show that IPC has $\Rightarrow$ and is bounded. Use $\to$ as $\Rightarrow$, $\bot$ as $\bot$, and $\bot \to \bot$ as $\top$. So just show that:

Assuming $[A] \wedge [B] \leq [C]$, we have $[A] \leq ([B] \Rightarrow [C])$

Assuming $[A] \leq ([B] \Rightarrow [C])$, we have $[A] \wedge [B] \leq C$

Both directions needed to make $\bot \leq A$ for all A

Both directions needed to make $A \leq \top$ for all A.

Thm: Completeness of Heyting semantics

A formula is provable in IPC if it is H-valid for every Heyting algebra H (and only if by soundness) Card ID: 1775594407810

TODO PROVE THIS

6 The Kripke Model

Def: Principal Up-set

of some poset S

$$a\uparrow := \{s \in S : a \leq s\}$$

Card ID: 1775594407811

Def: Terminal Segment

of a poset S

A subset $U \subseteq S$ where $a\uparrow \subseteq U$ for every $a \in U$

Note that this isn't the same as a filter: for any lattice with incomparable elements, we can escape the terminal segment with finite meets!

Card ID: 1775594407812

What is meant when we write $T(S)$ where S is a poset?

The set of terminal segments of S:

$$\{U \subseteq S : U \text{ is a terminal segment}\}$$

Card ID: 1777585767994

Prop

If S is a poset, $T(S) := \{U \subseteq S : U \text{ is a terminal segment}\}$ is a Heyting algebra

---

Order by set inclusion, the fact that this is a bounded lattice and closed is taken as trivial. The key step is using

$$U \Rightarrow V := \{s \in S : (s\uparrow) \cap U \subseteq V\}$$

Must show that this is a terminal segment and both directions of the Heyting Algebra definition.

Card ID: 1775594407814

Def: Kripke Model

Let P be some set of primitive propositions. A triple $(S, \leq, \Vdash)$ where $(S, \leq)$ is a poset, with S called the set of worlds and $\leq$ the accessibility relation; and $\Vdash \subseteq S \times P$ is the forcing relation.
It must fulfill the Persistence Property.
Note that for $X \subseteq S$ we say $X \Vdash \varphi$ if all worlds in $x \in X$ force φ.

Card ID: 1775594407815

Def: The Persistence Property for Kripke Models

$$(s \Vdash p) \wedge (s \leq s') \implies s' \Vdash p$$

Worlds are ordered by the knowledge they believe.

Card ID: 1778592619300

What is a simple way to make a Kripke model?

Take a mapping ('valuation') v from the set of primitive propositions P to the set of terminal segments $T(S)$ (known to be a Heyting algebra).

This induces a Kripke model by setting

$$s \Vdash p \iff s \in v(p)$$

for $s \in S$, $p \in P$.
It's clear to see that this fulfills the persistence property:

$$s \Vdash p \implies s \in v(p) \implies (s\uparrow) \subseteq v(p) \implies s' \in v(p) \implies s' \Vdash p$$

Card ID: 1777585767996

Def: Extended Forcing Relation of the Kripke Model

When p is not a primitive proposition, we define the forcing relation inductively as follows:

1. There is no world that believes in $\bot$: $\nexists s,\ s \Vdash \bot$

2. We write $s \Vdash \varphi \wedge \psi$ iff $s \Vdash \varphi$ and $s \Vdash \psi$

3. We write $s \Vdash \varphi \vee \psi$ iff $s \Vdash \varphi$ or $s \Vdash \psi$

4. We write $s \Vdash (\varphi \to \psi)$ iff, for every more knowledgeable world $s \leq s'$, $s' \Vdash \varphi$ implies $s' \Vdash \psi$
(worlds are humble: they believe only in things that they believe more knowledgeable worlds believe in)

Card ID: 1775594407817

What are the conditions for negation in a Kripke model?

$$s \Vdash \neg\varphi \iff \forall s' \geq s,\ s' \nVdash \varphi$$

$\neg\varphi$ holds iff no more knowledgeable worlds believe in it
Card ID: 1775594407820

What are the conditions for double negation in a Kripke model?

$$s \Vdash \neg\neg\varphi \iff \forall s' \geq s,\ \exists s'' \geq s',\ s'' \Vdash \varphi$$

In all more knowledgeable worlds, there exists some even more knowledgeable world that believes in φ.

Card ID: 1775594407821

Def: Filter

A filter F on a lattice L is a subset $F \subseteq L$ with the following properties:

1. $F \neq \emptyset$
2. F is a terminal segment of L: ie if $x \leq y$ and $x \in F$, then $y \in F$.
3. F is closed under finite meets.

Card ID: 1775594407822

Proper Filter

A filter is called 'proper' when it isn't just the whole set X.

Card ID: 1777585767998

Def: Prime Filter

A filter is prime if $x \vee y \in F \implies x \in F$ or $y \in F$.

Card ID: 1775594407824

Extension of Filters

If F is a proper filter and $x \notin F$, then there exists a prime filter extending F that still avoids containing x (by Zorn's Lemma)

Card ID: 1777585768000

TODO completeness of Kripke semantics

7 Recursive Functions

Core idea: we can build any function out of a few atomic pieces.

Def: Partial Recursive Function

The smallest class of functions $\mathbb{N}^k \to \mathbb{N}$ that contains the three atomic functions, and is closed under composition, primitive recursion, and minimization.

For the class of primitive recursive functions, ignore minimization.

Card ID: 1777214223318

Def: The three atomic partial recursive function building blocks

1. Projections: Pick one number out of a list (

$$\Pi_i^k : \mathbb{N}^k \to \mathbb{N}$$

$$\Pi_i^k : (n_1, ..., n_k) \mapsto n_i$$

2. Successor: Add one

$$S^+ : \mathbb{N} \to \mathbb{N}$$

$$S^+ : n \to n + 1$$

3. Zero: Any number to zero

$$Z : \mathbb{N} \to \mathbb{N}$$

$$Z : n \mapsto 0$$

Card ID: 1777214223321

Def: Partial Recursive Function Closure under Composition

Given $g : \mathbb{N}^k \to \mathbb{N}$, and for each $i \in [k]$ we have a function $h_i : \mathbb{N}^m \to \mathbb{N}$, then there exists a function $f : \mathbb{N}^m \to \mathbb{N}$ given by

$$f(\bar{n}) := g(h_1(\bar{n}), ..., h_k(\bar{n}))$$

Note that $\bar{n}$ refers to a tuple of natural numbers.
Sort of like a weird inverse map-reduce, except we're mapping over a list of functions by applying $\bar{n}$ to all of them.

Card ID: 1777214223324

Def: Partial Recursion Function Closure under Primitive Recursion

Given a base function $g : \mathbb{N}^m \to \mathbb{N}$ and step function $h : \mathbb{N}^{m+2} \to \mathbb{N}$, there exists a function $f : \mathbb{N}^{m+1} \to \mathbb{N}$ defined by:

$$f(0, \bar{n}) := g(\bar{n})$$
$$f(k+1, \bar{n}) := h(f(k, \bar{n}), k, \bar{n})$$

The first parameter is the fuel, the second is the surrounding context (constant through the process)

Card ID: 1777214223327

Def: Partial Recursive Function Closure under Minimization

Given $g : \mathbb{N}^{m+1} \to \mathbb{N}$, there exists a function $f : \mathbb{N}^m \to \mathbb{N}$ that maps $\bar{n}$ to the smallest k such that $g(k, \bar{n}) = 0$.

Note that it may not exist!
This is sometimes denoted $f(\bar{n}) = \mu k.g(k, \bar{n}) = 0$

Card ID: 1777214223330

Def: Total Recursive Function

A partial recursive function for which every every input maps to some outputs.

This just means it always terminates: there's no input that will search forever. All primitive recursive functions are total.

Card ID: 1777214223333

8 The Untyped Lambda-Calculus

Def: Fixed Point Theorem

There exists a closed λ-term Y such that for all F,

$$F(YF) \equiv_\beta YF$$

Y is called a fixed-point function.

Card ID: 1779828181022

9 Decidability in Logic

Def: Godel Numbering

Let L be a language. A Godel Numbering is an injection $L \hookrightarrow \mathbb{N}$ with three properties:

1. It's computable: there exists an algorithm you can follow to find the Godel number of any expression in finite time

2. Its image is a recursive subset of $\mathbb{N}$ (membership is decidable).

3. Its inverse (where defined) is also computable.

Card ID: 1777214223334

Def: Recursively Enumerable Set

$X \subseteq \mathbb{N}$ is recursively enumerable (or recognizable, or semi-decidable) if any of the following are true:

1. X is the image of some partial recursive function $f : \mathbb{N} \to \mathbb{N}$
(dovetail computations and generate a big list)

2. X is the image of some total recursive function $f : \mathbb{N} \to \mathbb{N}$
(just run on all numbers sequentially)

3. X is the domain of some partial recursive function $f : \mathbb{N} \to \mathbb{N}$
(same as 1, but mark the input on the list, not the output)

Card ID: 1777214223335

Def: Recursive Set

A set $X \subseteq \mathbb{N}$ is recursive if both X and its complement $\mathbb{N} \setminus X$ are recursively enumerable.

Card ID: 1777214223336

Def: Recursive Language

There exists an algorithm to check whether a string of symbols is an L-formula

(this is generally trivial: just checking if a statement makes sense)

Card ID: 1777214223338

Def: Recursive L-Theory

An L-Theory T is recursive if membership in T is decidable (we can check if a sentence is assumed or not)

Card ID: 1777214223339

Def: Decidable L-Theory

An L-Theory T is decidable if there is an algorithm for deciding $T \Vdash \varphi$ for any φ.

Card ID: 1777214223341

Prop

Craig's Trick

Every first-order theory with a recursively-enumerable set of axioms admits a recursive axiomatization

(if your list of axioms is recognizable, it can be reconfigured into a decidable one)

Proof:
There exists a totally recursive f with $f(n) = \phi_i$ for $n \in \mathbb{N}$.
Map each ϕ_i to a logically equivalent $\bigwedge_{j=1}^{i} \phi_i$.
The algorithm to check if a statement A is an axiom is thus:
First, check if A is equivalent to $f(1)$ (can do this in finite time due to finite sentence length). If it is, return yes, otherwise continue.
Second, find any decompositions $A = \bigwedge_{j}^{i} A_j$. Then just check axiom i for equivalence.

Card ID: 1777214223344

Notation for the Godel numbering of an expression

$\lceil x \rceil$ maps x to its Godel numbering
$\lfloor x \rfloor$ maps a Godel numbering x to the statement in represents

Card ID: 1777214223346

Prop

The set of total recursive functions (or their respective Godel numberings) is not recursively enumerable

---

Proof:
By definition, there exists some function f such that, for every total function h, there exists some $n \in \mathbb{N}$ such that $h = \lfloor f(n) \rfloor$.
In particular, this includes $g(x) = \lfloor f(x) \rfloor (x) + 1$.
Show a contradiction from there.

Card ID: 1777214223349

Def: The Language of Arithmetic $\mathcal{L}_{PA}$

The first-order language with signature $(0, 1, +, \cdot, <)$ where $0, 1$ are constants, $+, \cdot$ are functions with arity two, and $<$ is a relation with arity two.

Card ID: 1777214223352

Def: The Base Theory of Arithmetic PA^-

An $\mathcal{L}_{PA}$-theory fulfilling:

1. $+, \cdot$ are commutative and associative, and have identity elements 0 and 1.
(+, · individually)

2. $\cdot$ distributes over $+$.
(distributivity)

3. $<$ is a linear ordering compatible with $+$ and $\cdot$.
($<$ is a compatible linear ordering)

4. $\forall x, y, (x < y \implies \exists z, x + z = y)$.
(Quantifier definition of $<$)

5. $0 < 1$ and $\forall x, (x > 0 \implies x \geq 1)$.
(Zero is less than 1, $\leq$ base definition)

6. $\forall x, x \geq 0$.
(Everything is greater than or equal to zero)

Card ID: 1777214223354

Def: The Theory of Peano Arithmetic (PA)

PA^- with the extra set of axioms, the scheme of induction:
For each $\mathcal{L}_{PA}$-formula $\varphi(x, \bar{y})$, we have the axiom

$$I\varphi := \forall \bar{y},\, (\varphi(0, \bar{y}) \wedge \forall x,\, (\varphi(x, \bar{y}) \to \varphi(x+1, \bar{y})) \to \forall x,\, \varphi(x, \bar{y})$$

(proving the base case $n = 0$ and proving the inductive case $n \implies n+1$ is enough to prove something for all $n \in \mathbb{N}$)

Card ID: 1777214223356

Def: Delta-0 Formula (Δ_0)

A formula whose quantifiers ($\forall$, $\exists$) are bounded: for some fixed $n \in \mathbb{N}$, either

$$\forall x,\, x < n,\, \varphi(x)$$

or

$$\exists x,\, x < n,\, \varphi(x)$$

for n not free in $\varphi(x)$

Card ID: 1777214223358

Def: Pi-1 Formula and Sigma-1 Formula (Π_1, Σ_1)

A formula $\varphi(\bar{x})$ is a Σ_1-formula if it is provably equivalent to a there-exists statement with a Δ_0-formula body, ie, there exists a Δ_0 $\psi(\bar{x}, \bar{y})$ such that

$$PA \vdash (\varphi(\bar{x}) \iff \exists \bar{y}.\, \psi(\bar{x}, \bar{y}))$$

Similarly $\varphi(\bar{x})$ is a Π_1-formula if it is provably equivalent to a for-all statement with a Δ_0-formula body, ie, there exists a Δ_0 $\psi(\bar{x}, \bar{y})$ such that

$$PA \vdash (\varphi(\bar{x}) \iff \forall \bar{y}.\, \psi(\bar{x}, \bar{y}))$$

Card ID: 1777460526478

State a total recursive bijection from $\mathbb{N}^2 \to \mathbb{N}$, denoted $\langle x, y \rangle$

$$\langle x, y \rangle := \frac{(x+y)(x+y+1)}{2} + y$$

This is called the Cantor Pairing Function.

Intuition: it's just diagonalization. Count up the triangular number underneath the square we want the number of, then add how many squares into the next triangle we are.

Card ID: 1777460526479

Def: Godel's Beta Function

There exists a function $\beta(x, y)$ derivable in PA such that for any sequence x, there is a code u such that $\beta(u, i) = x_i$.

(Decodes the sequence and gives us the ith element. This is the projection function...)

Note that with Cantor's pairing function, we can reduce this to just one coded number.

Card ID: 1777460526482

Def: Chinese Remainder Theorem

Take integers a list of k integers $n_1, ..., n_k$, all greater than one. Assume they are pairwise coprime: no pair shares any divisor besides 1. no Let N be their product, $n_1...n_k$. Specify a list of k possible remainders for each n_k, $0 \leq a_k < n_k$.

Then there exists only one possible solution $0 \leq x < N$ such that the remainder of x with each n_i is a_i.

Another way to state it: with the same assumptions above, the system

$$\begin{aligned} x \equiv a_1 &\mod n_1 \\ &... \\ x \equiv a_k &\mod n_k \end{aligned}$$

has only one solution x mod N.

Card ID: 1778106432237

How do we encode numbers for use in Godel's Beta Function?

Suppose we want to code the numbers $x_0, ..., x_{n-1}$. Let $m = \max(0, x_0, ..., x_{n-1})!$. Define pairwise coprime $b_1 := m+1$, $b_2 := 2m+1, ..., b_n := nm+1$. Then our tuple becomes coded as the unique x guaranteed by the Chinese Remainder Theorem such that the remainder by b_i is x_i. Thus we have $u = \langle a, m \rangle$

Card ID: 1778106432240

How do we actually implement Godel's beta function given an encoded number?

$$\beta(u, i) := a\%(m(i+1)+1)$$

where (a, m) are the unique numbers such that $u = \langle a, m \rangle$.

Card ID: 1778592619317

Def: Godel's Lemma

Let $M \Vdash PA$, $n \in \mathbb{N}$, and $x_0, ..., x_{n-1} \in M$.
Then there exists a $u \in M$ such that $M \Vdash ((u)_i = x_i)$ for all $i < n$.

If M believes in PA, M can prove that Godel's beta function works as intended.

Card ID: 1777460526485

$PA \vdash \varphi$ versus $PA \vDash \varphi$ versus $\mathbb{N} \vDash \varphi$

$PA \vdash \varphi$ means that you can prove φ from axioms in PA,
$PA \vDash \varphi$ means that φ holds in every model of PA (even nonstandard ones),
$\mathbb{N} \vDash \varphi$ means that φ holds in the standard model specifically.

Card ID: 1779648617895

Def: Σ_1-definable Set

A set $A \subseteq \mathbb{R}^n$ is Σ_1-definable if there exists a Σ_1-formula φ such that

$$\bar{n} \in A \iff \mathbb{N} \vDash \varphi(\bar{n})$$

We need to construct a Σ_1 that is true if and only if $\bar{n} \in A$.
For instance, A is the set of perfect squares, then we could use $\varphi(n) = \exists x, (n = xx)$

Card ID: 1779648617896

Prop

Let $f : \mathbb{N}^k \to \mathbb{N}$ be a partial function.
f is (partial) recursive iff there is a Σ_1-formula $\theta(\bar{x}, y)$ such that

$$y = f(\bar{x}) \iff \mathbb{N} \Vdash \theta(\bar{x}, y)$$

(ie the graph of f is Σ_1-definable in $\mathbb{N}$)

First we show that if the graph of f is Σ_1-definable, then f must be partial recursive. We can just implement a minimization procedure then run the outer there-exists clause, checking every case exhaustively (computing results in $\mathbb{N}$ as normal).
Constructing a Σ_1 statement from a partial recursive f is more complicated. The base functions are simple. For the rest:
Composition: show this is equivalent to a there-exists over each subfunction output, with f applied to all of them together making the required full output.
Primitive Recursion: absolutely requires Godel's beta-function so that you can get an existence statement over the sequence length (we don't know how many times we'll need to run through the loop).
A similar argument holds for minimization.

Card ID: 1777460526487

todo corollary that recursively enumerable subsets are equivalent to formulas in $\mathbb{N}$ by applying the above

Def: Initial Segment

A subset of a poset that is downward closed (the reverse of a terminal segment)

Card ID: 1779648617899

Prop

$\mathbb{N}$ forms an initial segment of every model M of PA^-

---

Map $0 \in M$ to $0 \in \mathbb{N}$, $S(0) \in M$ to $1 \in \mathbb{N}$, and onwards.
It's an initial segment because

$$PA^- \vdash \forall x.(x \leq k \to x = 0 \lor x = 1 \lor \ldots \lor x = k)$$

for all standard $k \in \mathbb{N}$.

Card ID: 1779648617901

Def: Function Representation in a Theory

Let $f : \mathbb{N}^l \to \mathbb{N}$ be a total function with T being any L_{PA} theory extending PA^-. f is represented in T if there exists an L_{PA} formula $\theta(x_1, ..., x_l, y)$ such that, for all $\bar{n} \in \mathbb{N}^l$,
1)

$$T \vdash \exists! y.\theta(\hat{n}, y)$$

2)

$$k = f(\bar{n}) \implies T \vdash \theta(\bar{n}, k)$$

If θ can be made a Σ_1, we say f is Σ_1-represented in T.

Card ID: 1779648617903

Def: Subset Representation in a Theory

A subset $S \subseteq \mathbb{N}^l$ is represented in an L_{PA} theory extending PA^- if there exists an L_{PA} formula $\theta(x_1, ..., x_l)$ such that for all $\bar{n} \in \mathbb{N}^l$:
1)

$$\bar{n} \in S \implies T \vdash \theta(\bar{n})$$

2)

$$n \notin S \implies T \vdash \neg\theta(\bar{n})$$

If θ can be made a Σ_1, we say S is Σ_1-represented in T.

Card ID: 1779648617905

Prop

Every total recursive function $f : \mathbb{N}^m \to \mathbb{N}$ is Σ_1-represented in PA^-

---

We apply the previous result for $\mathbb{N}$ to get a Σ_1 $\exists \bar{z}.\phi(\bar{x}, y, \bar{z})$ and equivalently $\exists z.\phi(\bar{x}, y, z)$ by coding.
By example sheet 4 question 2, any Σ_1-sentence true in $\mathbb{N}$ is true in any model of PA^-. So we have the second part of the representation definition, but we need the first part. That's tricky because we don't yet have uniqueness for the ϕ statement above (we could pick a z larger than the smallest minimizable).
So just write a predicate that forbids smaller solutions:

$$\psi(\bar{x}, y, z) := \phi(\bar{x}, y, z) \wedge \forall u, u \leq (y+z).\forall v, v \leq (y+z).(u+v < y+z \to \neg\phi(\hat{x}, u, v))$$

Corollary:
Every recursive set $A \subseteq \mathbb{N}^m$ is Σ_1-represented in PA^-.

Card ID: 1779648617907

What do we call an L_{PA} formula with one free variable?

An unary predicate

Card ID: 1779648617908

Def: State the Diagonalization Lemma

Let T be an L_{PA} theory in which every total recursive function is Σ_1-represented, and let $\theta(x)$ be an L_{PA} formula with one free variable. Then there exists some sentence G of L_{PA} such that

$$T \vdash (G \iff \theta(\ulcorner G \urcorner))$$

(So for any numeric property, there's a sentence G that provably states that its own Godel number has that property)

Card ID: 1779648617909

Def: Diag

diag(n) :
If $n = \ulcorner \sigma(x) \urcorner$, return $\ulcorner \forall y.(y = n \rightarrow \sigma(y)) \urcorner$
Else return 0.

We're just rewriting the sentence $\sigma(x)$ coded by n to include n itself.
diag is the coded function that plugs a number into the conditional it represents (if possible).

Card ID: 1779648617911

Def: G

$$G := \forall y.(y = n \rightarrow \psi(y))$$

where

$$\psi(x) := \forall z.(\delta(x, z) \rightarrow \theta(z))$$

($\delta(x, z)$ is the Σ_1-representation of diag)
and $n = \ulcorner \psi(x) \urcorner$

Card ID: 1780667436249

Def: Crude Incompleteness

Let T be a recursive set of (Godel numbers of) L_{PA} sentences that is consistent (never contains both ϕ and $\neg\phi$) and contains all the Σ_1 and Π_1 sentences provable in PA^-.
Then there exists a Π_1 sentence τ such that $\tau \notin T$ and $\neg\tau \notin T$ (T can't prove τ or its negation).

Card ID: 1780945172503

Def: Godel-Rosser Theorem

Let T be a consistent L_{PA} theory extending PA^- and admitting a recursively enumerable axiomatization.
Then T is Π_1 incomplete: there exists a Π_1 sentence τ for which $T \nvdash \tau$ and $T \nvdash \neg\tau$

Assume that T is complete to get a contradiction. Use Craig's trick to get a recursive axiomatization, do exhaustive proof search from axioms (paired with completeness) to show that S is recursive. But then we can apply Crude Incompleteness to get a contradiction.

Card ID: 1780945172504

Def: Recursive (and countable) L_{PA}-structures

A (countable) L_{PA}-structure $\mathcal{M}$ is recursive if and only if there exist recursive functions $\oplus : \mathbb{N}^2 \to \mathbb{N}$, $\otimes : \mathbb{N}^2 \to \mathbb{N}$, a binary recursive relation $\preceq \subseteq \mathbb{N}^2$, and natural numbers $n_0, n_1 \in \mathbb{N}$ such that

$$\mathcal{M} \cong (\mathbb{N}, \oplus, \otimes, \preceq, n_0, n_1)$$

Card ID: 1780945172505

Def: Prime Function

The Σ_1 formula $\mathrm{pr}(x, y)$ is true only when y is the xth prime number.

$$PA \vdash \forall x, \exists! y, \mathrm{pr}(x, y)$$

So if $\mathbb{N}$ believes y is the xth prime number, all models of PA do.
We denote the kth prime π_k.

Card ID: 1780945172507

Def: Overspill Lemma

Let $\mathcal{M}$ be a nonstandard model of PA, $\phi(x, \bar{y})$ be an L_{PA} formula, and $\bar{a} \in \mathcal{M}$.
If $\mathcal{M} \vDash \phi(n, \bar{a})$ for all standard natural numbers n, then there exists a nonstandard $e \in M$ such that $\mathcal{M} \vDash \forall x \leq e.\phi(x, \bar{a})$

Card ID: 1780945172510

Proof of Overspill

Suppose there exists some function $\mathrm{st}(x, \bar{a})$, that, given some helper data $\bar{a}$, is true only when x is a standard natural number.
But clearly we can apply 'st' inductively from zero:

$$M \vDash \mathrm{st}(0, \bar{a}) \wedge \forall x.\mathrm{st}(x, \bar{a}) \rightarrow \mathrm{st}(x + 1, \bar{a})$$

and remember by the inductive axiom of PA, we must then have $\forall x, \mathrm{st}(x, \bar{a})$!
But that defeats the whole point: every element of a nonstandard model can't be standard!

Card ID: 1780945172513

Def: Canonical Coding

A subset $S \subseteq \mathbb{N}$ is canonically coded in a model $\mathcal{M}$ of PA if there exists some element $c \in \mathcal{M}$ such that

$$S = \{n \in \mathbb{N} : \exists y.(\pi_{n^*} \times y) = c\}$$

where n^* is the standard natural number n in the model.

We proved in class that for any nonstandard model $\mathcal{M}$ of PA, there exists a non-recursive set S which is canonically coded in $\mathcal{M}$.

Card ID: 1780945172515

todo needs proof

Def: Recursive Inseparability

Subsets A, B of $\mathbb{N}$ are recursively inseparable if they are disjoint and there is no recursive subset C of $\mathbb{N}$ such that $B \cap C = \emptyset$ and $A \subseteq C$.

We proved in class that they must exist.

Card ID: 1780945172517

todo needs proof

Def: Tennenbaum's Theorem

Let $\mathcal{M} = (M, \oplus, \otimes, \preceq, n_0, n_1)$ be a countable nonstandard model of PA. Then $\oplus$ is not recursive.

Card ID: 1780945172519

todo needs proof