

First Order Logic

Matilde Marcolli

Ma6c: Logic, Caltech, Spring 2026

Structure

- collection $\mathcal{X} = \langle X; \Phi = \{\phi_i\}; \mathcal{R} = \{R_j\} \rangle$
- X is a *nonempty set* (finite or infinite)
- $\Phi = \{\phi_i\}$ is a collection of **functions** $\phi_i : X^{n_i} \rightarrow X$ (n_i -ary)
- $\mathcal{R} = \{R_j\}$ is a collection of **relations** $R_j \subseteq X^{m_j}$ (m_j -ary)
write $R_j(x_1, \dots, x_{m_j})$ for $(x_1, \dots, x_{m_j}) \in R_j$
- Note that functions are a special kind of relation (viewing a function as its graph $\Gamma(\phi_i) \subset X^{n_i} \times X$ with $\Gamma(\phi_i) = \{(x_1, \dots, x_{n_i}, x) \mid x = \phi_i(x_1, \dots, x_{n_i})\}$).
- sets of functions and relations can be finite or infinite (or empty) and arity is also arbitrary
- **Example: structure of natural numbers** $\langle \mathbb{N}; 0, S, +, \cdot; < \rangle$ with $S =$ successor function $S(n) = n + 1$, arities $(0, 1, 2, 2; 2)$
- **Example: power set structure** X a set and $\langle \mathcal{P}(X); \emptyset, \cap, \cup; \subseteq \rangle$ with arities $(0, 2, 2; 2)$
- **Example: algebraic structures** for instance group $\langle G; e, \cdot; \rangle$ (no rels) **satisfying axioms**

First Order Language (classical logic)

- **logical symbols** (same for all first order languages):
 - 1 variables: $x_1, x_2, \dots, x_n, \dots$
 - 2 propositional connectives: \neg, \wedge, \vee , and $\Rightarrow, \Leftrightarrow (\rightarrow, \leftrightarrow)$
 - 3 quantifiers: \exists, \forall
 - 4 brackets: (and)
 - 5 equality: =
- **non-logical symbols** (language specific):
 - 1 functions: $\Phi_n = \{\phi_i\}$ n -ary function symbols, for each $n \geq 0$
 - 2 relations: $\mathcal{R}_m = \{R_j\}$ m -ary relation symbols, for each $m \geq 1$
- write language (not mentioning explicitly logical symbols) as $\mathcal{L} = \Phi \cup \mathcal{R}$ i.e.

$$\mathcal{L} = \bigcup_n \Phi_n \cup \bigcup_m \mathcal{R}_m$$

- Example **language of arithmetic**: $\mathcal{L}_{\text{arithm}} = \{0, S, +, \cdot, <\}$

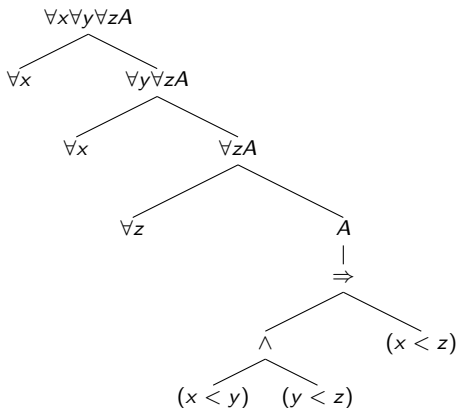
Terms of $\mathcal{L} = \Phi \cup \mathcal{R}$

- every variable x_i is a term
- if t_1, \dots, t_n are terms and $\phi \in \Phi_n$ then $\phi(t_1, \dots, t_n)$ is a term; any $\phi \in \Phi_0$ (constant functions) is also a term

Well Formed Formulae (WFF) of $\mathcal{L} = \Phi \cup \mathcal{R}$

- **atomic formulae** are WFF: $R(t_1, \dots, t_m)$ with $R \in \mathcal{R}_m$ and t_1, \dots, t_m terms
- if A, B are WFF then $\neg A, \neg B, A \wedge B, A \vee B, A \Rightarrow B, A \Leftrightarrow B$ are also WFF
- if A is WFF and x_i a variable then $\exists x_i A$ and $\forall x_i A$ are WFF
- deriving a WFF via iterations of these operations gives a **parse tree** for the formula

- **Example:** $\forall x \forall y \forall z A$ with $A = ((x < y) \wedge (y < z) \Rightarrow (x < z))$



- **subformula:** any formula occurring in the parse tree of a WFF

Quantifiers and scope

$$\exists y \forall x \underbrace{[\exists y R(x, y)]}_{\text{scope}} \Rightarrow Q(x, y)$$

$\underbrace{\hspace{10em}}_{\text{scope}}$

$\underbrace{\hspace{15em}}_{\text{scope}}$

A a WFF with x a variable and Ω a quantifier (either \forall or \exists), if the string Ωx occurs in A then can write $A = S_1 \Omega x B S_2$ for strings S_1, S_2 in the language and a uniquely determined WFF B called the **scope** of this occurrence of Ωx (induction on the parse tree of A); uniqueness of B since proper prefix of WFF is not WFF

bound and free variables in a WFF A of language \mathcal{L}

- **bound**: variable x occurring in a substring of the form $\Omega x B$ for a quantifier Ω and scope B
- **free**: any other occurrence
- example: $\forall x R(x, y)$ has x bound and y free variable
- **sentence** in \mathcal{L} : a WFF with no free variables

Semantics: structures provide **interpretations** for first order language $\mathcal{L} = \Phi \cup \mathcal{R}$

$$\mathcal{X} = \langle X, \Phi_{\mathcal{X}}, \mathcal{R}_{\mathcal{X}} \rangle$$

- $\Phi \rightarrow \Phi_{\mathcal{X}}$ with $\phi \in \Phi_n$ mapped to $\phi_{\mathcal{X}} : X^n \rightarrow X$ in $\Phi_{\mathcal{X}}$
- for $c \in \Phi_0$ (constants) this maps to points $c_{\mathcal{X}} \in X$
- $\mathcal{R} \rightarrow \mathcal{R}_{\mathcal{X}}$ with $R \in \mathcal{R}_m$ mapped to $R_{\mathcal{X}} \subseteq X^m$ in $\mathcal{R}_{\mathcal{X}}$
- set $X =$ the **universe** of \mathcal{X}

Structures and Models

- \mathcal{L} first order language, \mathcal{X} structure interpreting \mathcal{L}
- assignment $\alpha : x_i \mapsto a_i$ of points $a_i \in X$
- a WFF A in \mathcal{L} is **true in** (\mathcal{X}, α)

$$(\mathcal{X}, \alpha) \models A$$

if have

- 1 if A **atomic** $A = R(t_1, \dots, t_k)$ with $t_i = t_i(x_1, \dots, x_n)$ then

$$(\mathcal{X}, \alpha) \models A \quad \text{iff} \quad R_{\mathcal{X}}(t_{1,\mathcal{X}}(a_1, \dots, a_n), \dots, t_{k,\mathcal{X}}(a_1, \dots, a_n))$$

- 2 for **logical connectives**

$$(\mathcal{X}, \alpha) \models \neg A \quad \text{iff} \quad (\mathcal{X}, \alpha) \not\models A$$

$$(\mathcal{X}, \alpha) \models (A \wedge B) \quad \text{iff} \quad (\mathcal{X}, \alpha) \models A \quad \text{and} \quad (\mathcal{X}, \alpha) \models B$$

$$(\mathcal{X}, \alpha) \models (A \vee B) \quad \text{iff} \quad (\mathcal{X}, \alpha) \models A \quad \text{or} \quad (\mathcal{X}, \alpha) \models B$$

$$(\mathcal{X}, \alpha) \models (A \Rightarrow B) \quad \text{iff} \quad (\mathcal{X}, \alpha) \not\models A \quad \text{or} \quad (\mathcal{X}, \alpha) \models B$$

3 for **quantifiers**

$(\mathcal{X}, \alpha) \models \exists x_i A$ iff for some $b \in X$ $(\mathcal{X}, \alpha|_{a_i \mapsto b}) \models A$

$(\mathcal{X}, \alpha) \models \forall x_i A$ iff for any $b \in X$ $(\mathcal{X}, \alpha|_{a_i \mapsto b}) \models A$

- equivalent notation for $(\mathcal{X}, \alpha) \models A$

$\mathcal{X} \models A[a_1, \dots, a_n]$ or for no free variables $\mathcal{X} \models A$

- S set of WFF in \mathcal{L}

$\mathcal{X} \models S$ if $\mathcal{X} \models A$ for all $A \in S$

(and for all assignments α if A with free variables)

- S set of WFF in \mathcal{L} and A WFF in \mathcal{L}

$S \models A$ if for all \mathcal{X} with $\mathcal{X} \models S$ also $\mathcal{X} \models A$

- A **valid** if $\models A$ (meaning $\emptyset \models A$, every \mathcal{X} for \mathcal{L} models A)
- $A(x_1, \dots, x_n)$ valid iff its *universal closure* $\forall x_1 \dots \forall x_n A$ valid

S **satisfiable** if there is a \mathcal{X} and an assignment $\alpha : x_i \mapsto a_i \in X$ such that $(\mathcal{X}, \alpha) \models A$ for all $A \in S$ (\mathcal{X} satisfies all formulae of S)

- **tautologies in first order logic** (all valid): tautologies in propositional logic where all the propositional variables p_i are replaced by WFFs A_i of \mathcal{L}
- **substitutions**: x variable t term, A WFF, $x/t :=$ substitution of every *free* occurrence of x in A with t
- **notational warning**: also find t/x used for substitution of x with t (check notation!)

equivalence: A, B WFF in \mathcal{L}

$$A \equiv B \quad \text{if} \quad \models (A \Leftrightarrow B)$$

complete set \neg, \wedge, \exists (or \neg, \wedge, \forall)

- any WFF A in \mathcal{L} can be written using only a complete set of connectives for classical propositional logic (such as $\{\neg, \wedge\}$) and either \exists or \forall
- because $\neg\forall xA \equiv \exists x\neg A$ and $\forall x\neg A \equiv \neg\exists xA$

Definability (in a structure \mathcal{X})

- **graph** $\Gamma_{A,\mathcal{X}} \subseteq X^n$ of a WFF $A(x_1, \dots, x_n)$ of \mathcal{L}

$$\Gamma_{A,\mathcal{X}}(a_1, \dots, a_n) \quad \text{iff} \quad \mathcal{X} \models A[a_1, \dots, a_n]$$

which means

$$\Gamma_{A,\mathcal{X}} = \{(a_1, \dots, a_n) \in X^n : (\mathcal{X}, \alpha) \models A \text{ with } \alpha : x_i \mapsto a_i\}$$

- **definable relation** $R \subseteq X^n$ is *first-order definable* in $(\mathcal{L}, \mathcal{X})$ if there is a WFF A of \mathcal{L} such that $R = \Gamma_{A,\mathcal{X}}$:

$$R(a_1, \dots, a_n) \quad \text{iff} \quad \mathcal{X} \models A[a_1, \dots, a_n]$$

- **definable function** $f : X^n \rightarrow X$ if $\text{graph } \Gamma(f) \subseteq X^{n+1}$ is definable, i.e. there is a WFF $A(x_1, \dots, x_{n+1})$ such that

$$(a_1, \dots, a_{n+1}) \in \Gamma(f) \quad \text{iff} \quad \mathcal{X} \models A[a_1, \dots, a_{n+1}]$$

with $(a_1, \dots, a_{n+1}) \in \Gamma(f)$ iff $a_{n+1} = f(a_1, \dots, a_n)$

- elements $a \in X$ (case $n = 0$) definable if there is a formula $A(x)$ with $\mathcal{X} \models A[a]$

- first-order language \mathcal{L} and structure \mathcal{X}
- set of n -ary **definable relations** is a **Boolean algebra** (with set-theoretic complement $\neg R = X^n \setminus R$, union and intersection)
- **projection**: if $R \subseteq X^n$ definable by $A(x_1, \dots, x_n)$ then

$$\text{Proj}(R) = \{(a_1, \dots, a_{n-1}) : \exists a_n \in X, (a_1, \dots, a_n) \in R\}$$

definable by $\exists x_n A(x_1, \dots, x_n)$

Example: every integer $n \in \mathbb{Z}$ is definable

- $\mathcal{L} = \{+, \cdot, <\}$ and $\mathcal{X} = \langle \mathbb{R}; +, \cdot, < \rangle$
- $n = 0$ definable by $A_0(x) = (x + x = x)$
- $n = 1$ definable by $A_1(x) = (x \cdot x = x)$
- inductively suppose $n > 0$ defined by a formula $A_n(x)$, then $n + 1$ defined by

$$A_{n+1}(x) = \exists y \exists z (x = (y + z) \wedge A_n(y) \wedge A_1(z))$$

- also $-n$ definable by

$$A_{-n}(x) = \exists y \exists z ((x + y = z) \wedge A_0(z) \wedge A_n(y))$$

- also every rational number $a \in \mathbb{Q}$ is definable and every algebraic number $a \in \bar{\mathbb{Q}}$ (root of a polynomial with \mathbb{Z} coefficients) is definable
- finite unions of intervals with algebraic endpoints are definable (in fact all the definable unary relations $R \subset \mathbb{R}$)

Example: $+$ is definable in structure $(\mathbb{N}, 0, S, \cdot)$

$$a + b = c \text{ iff } [(a + 1)(c + 1) + 1][b(c + 1) + 1] = (c + 1)^2[(a + 1)b + 1] + 1$$

i.e. $S(S(a) \cdot S(c)) \cdot S(b \cdot S(c)) = S((S(c) \cdot S(c)) \cdot S(S(a) \cdot b))$

definable with parameters:

- \mathcal{L} first-order language, $\mathcal{X} = (X, \Phi, \mathcal{R})$ structure
- $R \subseteq X^n$ **definable with parameters** if there is a WFF $A(x_1, \dots, x_n, x_{n+1}, \dots, x_{n+m})$ for some $m \geq 0$ and points $p_1, \dots, p_m \in X$ (parameters) such that

$$R(a_1, \dots, a_n) \text{ iff } \mathcal{X} \models A[a_1, \dots, a_n, p_1, \dots, p_m]$$

example: $\mathcal{L} = \{+, \cdot, <\}$ and $\mathcal{X} = (\mathbb{R}, +, \cdot, <)$, interval $(u, v) \subset \mathbb{R}$ is definable with parameters

$$A(x, y, z) = (y < x) \wedge (x < z) \quad \text{then } a \in (u, v) \text{ iff } \mathcal{X} \models A[a, u, v]$$

Definable sets (relations)

Two views of Structures:

- 1 “bottom-up approach”: $\mathcal{X} = (X, \Phi, \mathcal{R})$ as above
 - 2 “top-down approach”: a structure \mathcal{M} on a set M is a collection $\{\mathcal{M}_n\}_{n \geq 1}$ with
 - \mathcal{M}_n collection of subsets of M^n , closed under complements and finite unions
 - \mathcal{M}_n contains the diagonals $\Delta_{i,j} = \{(x_1, \dots, x_n) : x_i = x_j\}$
 - if $A \in \mathcal{M}_n$ then $M \times A$ and $A \times M$ are in \mathcal{M}_{n+1}
 - if $A \in \mathcal{M}_{n+1}$ then projection of A on first n coordinates in \mathcal{M}_n
- set $A \subset M^n$ **definable in \mathcal{M}** if $A \in \mathcal{M}_n$
 - **partial order** $\mathcal{M} \subseteq \mathcal{M}'$ (same set M) if for all $n \in \mathbb{N}$ have $\mathcal{M}_n \subseteq \mathcal{M}'_n$ (\mathcal{M}' is an **expansion** of \mathcal{M})
 - given $S \subset M$, set A **definable in \mathcal{M} with parameters in S** if A is in the expansion $(\mathcal{M}, (c)_{c \in S})$ of \mathcal{M}

top-down from bottom-up Structures

- start with structure $\mathcal{X} = (X, \Phi = \cup_n \Phi_n, \mathcal{R} = \cup_r \mathcal{R}_r), M = X$
- for each $n \in \mathbb{N}$ consider \mathcal{T}_n = smallest set of functions $M^n \rightarrow M$ with
 - 1 \mathcal{T}_n contains coordinate projections $(x_1, \dots, x_n) \mapsto x_i$
 - 2 $\phi \circ (f_1, \dots, f_m) \in \mathcal{T}_n$ for all $\phi \in \Phi_m$, any m , and for all $f_1, \dots, f_m \in \mathcal{T}_n$
- let $\mathcal{M}_{n,0}$ = Boolean algebra of subsets of M^n generated by all sets of the form
 - 1 $\{x \in M^n : f(x) = g(x), f, g \in \mathcal{T}_n\}$
 - 2 $\{x \in M^n : R(f_1(x), \dots, f_m(x)), f_i \in \mathcal{T}_n, R \in \mathcal{R}_m, m \in \mathbb{N}\}$
- for $k \geq 0$, assume $\mathcal{M}_{n,k}$ constructed, then $\mathcal{M}_{n,k+1}$ is Boolean algebra generated by $\mathcal{M}_{n,k}$ and the collection of all projections on first n coordinates of sets in $\mathcal{M}_{n+1,k}$
- $\mathcal{M}_n = \cup_k \mathcal{M}_{n,k}$ is structure in top-down sense
- smallest such in which all $R \in \mathcal{R}$ and all $\phi \in \Phi$ (all $\Gamma(\phi)$ graphs) are definable sets

$$\mathcal{M} = \mathcal{M}(X, \Phi, \mathcal{R}) \quad \text{generated by } \mathcal{X}$$

$\phi \in \Phi$ and $R \in \mathcal{R}$ are the **primitives** of \mathcal{M}

quantifier elimination

- sets in the $\mathcal{M}_{n,0}$ part of $\mathcal{M} = \mathcal{M}(X, \Phi, \mathcal{R})$ are the **quantifier-free definables** of \mathcal{M}
- e.g. sets in $\mathcal{M}_{n,1}$ (projections of sets in $\mathcal{M}_{n+1,0}$)

$$\Pi(A) = \{(x_1, \dots, x_n) : \exists x_{n+1} \in M, (x_1, \dots, x_{n+1}) \in A\}$$

get \forall using \exists (projection) and complement

- the structure $\mathcal{X} = (X, \Phi, \mathcal{R})$ has **quantifier elimination** (QE) if every definable set in $\mathcal{M} = \mathcal{M}(\mathcal{X})$ is quantifier-free definable (hence projections of quantifier-free definable sets are quantifier-free definable)
- the structure $\mathcal{X} = (X, \Phi, \mathcal{R})$ is **model complete** if every definable set is a coordinate projection of a quantifier-free definable set
- quantifier elimination implies model completeness

quantifier elimination vs model completeness

- **quantifier elimination** can be rephrased as the property that for any given WFF $A(x_1, \dots, x_n)$ there is an equivalent quantifier-free formula $B(x_1, \dots, x_n)$ (same definable set)
- **model completeness** for any given WFF $A(x_1, \dots, x_n)$ there is an equivalent *existential formula* of the form $\exists x_1 \cdots \exists x_k B(x_1, \dots, x_m)$ with $B(x_1, \dots, x_m)$ quantifier-free (equivalent to projection of a quantifier free formula)

examples of quantifier elimination

- let $\phi(a, b, c, d)$ be the WFF

$$\exists x \exists y \exists u \exists v (xa + yc = 1 \wedge xb + yd = 0 \wedge ua + vc = 0 \wedge ub + vd = 1).$$

The formula $\phi(a, b, c, d)$ asserts that the matrix

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

is invertible. By the determinant test,

$$F \models \phi(a, b, c, d) \leftrightarrow ad - bc \neq 0$$

for any field F .

- let $\phi(a, b, c)$ be the WFF

$$\exists x ax^2 + bx + c = 0.$$

By the quadratic formula,

$$\mathbb{R} \models \phi(a, b, c) \leftrightarrow [(a \neq 0 \wedge b^2 - 4ac \geq 0) \vee (a = 0 \wedge (b \neq 0 \vee c = 0))],$$

whereas in the complex numbers

$$\mathbb{C} \models \phi(a, b, c) \leftrightarrow (a \neq 0 \vee b \neq 0 \vee c = 0).$$

In either case, ϕ is equivalent to a quantifier-free formula. However, ϕ is not equivalent to a quantifier-free formula over the rational numbers \mathbb{Q} .

Semialgebraic sets

- $f_i(x)$, $i = 1, \dots, m$, $x = (x_1, \dots, x_n) \in \mathbb{R}^n$ **polynomials**
- **basic semialgebraic sets**

$$S = \{x \in \mathbb{R}^n : f_i(x) \geq 0, \forall i = 1, \dots, m\}$$

- **general semialgebraic sets** obtained from basic ones via iterations of Boolean operations (unions, intersection, complement) starting with basic ones
- for $f_i(x)$ and h_j polynomials
 - a finite union of sets of the form
$$\left\{ x \in \mathbb{R}^n \mid f_i(x) > 0, h_j(x) = 0 \text{ for all } i = 1, \dots, m, j = 1, \dots, p \right\}$$
 - in \mathbb{R} , a finite union of points and open intervals

Every *closed* semialgebraic set is a finite union of basic closed semialgebraic sets; i.e., sets of the form

$$\left\{ x \in \mathbb{R}^n \mid f_i(x) \geq 0 \text{ for all } i = 1, \dots, m \right\}$$

Tarski-Seidenberg and Quantifier Elimination

Tarski-Seidenberg theorem: if $S \subset \mathbb{R}^{n+p}$ is semialgebraic, then so are

- $\{ x \in \mathbb{R}^n \mid \exists y \in \mathbb{R}^p (x, y) \in S \}$ (closure under projection)
- $\{ x \in \mathbb{R}^n \mid \forall y \in \mathbb{R}^p (x, y) \in S \}$ (complements and projections)

i.e., quantifiers do not add any expressive power

other properties of semialgebraic sets

- Semialgebraic sets form a Boolean algebra
- also closed under interior and closure operations
- have finitely many connected components, each a semialgebraic set

isomorphism of structures

- language $\mathcal{L} = \Phi \cup \mathcal{R}$
- structures $\mathcal{X} = (X, \Phi_{\mathcal{X}}, \mathcal{R}_{\mathcal{X}}) \rightarrow \mathcal{Y} = (Y, \Phi_{\mathcal{Y}}, \mathcal{R}_{\mathcal{Y}})$
interpretations of \mathcal{L}
- $\eta : \mathcal{X} = (X, \Phi_{\mathcal{X}}, \mathcal{R}_{\mathcal{X}}) \rightarrow \mathcal{Y} = (Y, \Phi_{\mathcal{Y}}, \mathcal{R}_{\mathcal{Y}})$ **bijective map of sets** $\eta : X \rightarrow Y$
- **intertwines functions**: for $\phi \in \Phi_n$ and $\phi_{\mathcal{X}} \in \Phi_{\mathcal{X},n}$ and $\phi_{\mathcal{Y}} \in \Phi_{\mathcal{Y},n}$

$$\eta(\phi_{\mathcal{X}}(a_1, \dots, a_n)) = \phi_{\mathcal{Y}}(\eta(a_1), \dots, \eta(a_n))$$

for all $(a_1, \dots, a_n) \in X^n$

- **matches relations**: for $R \in \mathcal{R}_m$ with $R_{\mathcal{X}} \subset X^m$ and $R_{\mathcal{Y}} \subset Y^m$

$$R_{\mathcal{X}}(a_1, \dots, a_m) \quad \text{iff} \quad R_{\mathcal{Y}}(\eta(a_1), \dots, \eta(a_m))$$

for all $(a_1, \dots, a_m) \in X^m$

Example: $\mathcal{L} = (f, R)$ binary function (operation), binary relation

- $\mathcal{X} = (\mathbb{R}^+, \cdot, <)$ and $\mathcal{Y} = (\mathbb{R}, +, <)$
- $\eta : \mathbb{R}^+ \rightarrow \mathbb{R}$ with $\eta(x) = \log(x)$ is isomorphism

isomorphism and definability

- WFF $A(x_1, \dots, x_n)$ of language \mathcal{L}
- isomorphism $\eta : \mathcal{X} \rightarrow \mathcal{Y}$ of structures over \mathcal{L}
- then for any $(a_1, \dots, a_n) \in X^n$

$$\mathcal{X} \models A[a_1, \dots, a_n] \quad \text{iff} \quad \mathcal{Y} \models A[\eta(a_1), \dots, \eta(a_n)]$$

... this can serve as a test for **non-definability**

tests of non-definability

- \mathbb{N} is **not** definable in $\mathcal{X} = (\mathbb{R}, 0, 1, \cdot, <)$
- use automorphism $\eta : \mathcal{X} \rightarrow \mathcal{X}$ given by $\eta(a) = a^3$
- does not preserve \mathbb{N} (not invariant)

test not always applicable: some structures have no non-identity automorphisms: for example $(\mathbb{N}, 0, S)$

Theories of first-order logic

- given first-order language \mathcal{L}
- **sentence** A is WFF with no free variables
- S a set of sentences in \mathcal{L}
- **logical consequences** $\text{Con}(S)$

$$\text{Con}(S) = \{A \text{ sentence} : S \models A\}$$

- $\text{Con}(\text{Con}(S)) = \text{Con}(S)$ closed under logical consequences
- a **theory** T is a set of sentences in \mathcal{L} that is closed under logical consequences

$$\text{Con}(T) = T$$

- if a theory T is $T = \text{Con}(S)$ then S is a set of **axioms** for T

example: $\mathcal{L} = \{<\}$ (binary relation symbol)

theory of partial order: axioms

$$(1) \quad \forall x \forall y (x < y \Rightarrow \neg y < x) \quad (\text{antisymmetry})$$

$$(2) \quad \forall x \forall y \forall z (x < y \wedge y < z \Rightarrow x < z) \quad (\text{transitivity})$$

A model $\langle M, < \rangle$ of this theory is called a *partially ordered set* (or just *partial order*). If we add the axiom:

$$(3) \quad \forall x \forall y (x < y \vee x = y \vee y < x) \quad (\text{linearity})$$

we obtain the theory of *linear order*, whose models are called *linearly ordered sets* (or just *linear orders*).

The theory of *dense linear order* is obtained by adding two more axioms:

$$(4) \quad \exists x \exists y (x \neq y)$$

$$(5) \quad \forall x \forall y (x < y \Rightarrow \exists z (x < z \wedge z < y)) \quad (\text{density})$$

Examples of models of this theory (i.e., *dense linear orders*) are $\langle \mathbb{Q}, < \rangle$, $\langle \mathbb{R}, < \rangle$. Examples of linear orders which are not dense are $\langle \mathbb{Z}, < \rangle$ and $\langle \mathbb{N}, < \rangle$.

Finally the theory of *dense linear orders without endpoints* is obtained by adding the axioms:

$$(6) \quad \forall x \exists y (x < y)$$

$$(7) \quad \forall x \exists y (y < x).$$

Models of these are again $\langle \mathbb{R}, < \rangle$, $\langle \mathbb{Q}, < \rangle$, but not $\langle [0, 1], < \rangle$.

limits of first-order logic

- first-order logic only quantifies only variables
- it does not quantify over relations
- so it *cannot* express the condition of being a **well ordered set** (every non-empty subset has a least element, e.g. in \mathbb{N})
- **example**: the least upper bound property (e.g. in \mathbb{R})

$$\forall A \left(\left(\exists w (w \in A) \wedge \exists z \forall u (u \in A \rightarrow u \leq z) \right) \rightarrow \right. \\ \left. \exists x \left(\forall u (u \in A \rightarrow u \leq x) \wedge \forall y \forall u \left((u \in A \rightarrow u \leq y) \rightarrow x \leq y \right) \right) \right)$$

is not an axiom in first-order logic because it quantifies over relations (sets)

- **second-order logic** allows this
- **higher-order logics** (quantifying over sets of sets etc) and **type theory** take care of these issues

example: $\mathcal{L} = \{e, \cdot\}$ (constant symbol, binary relation)

theory of groups: axioms

- $\forall x \forall y \forall z (x \cdot (y \cdot z) = (x \cdot y) \cdot z)$ (associativity)
- $\forall x ((x \cdot e = x) \wedge (e \cdot x = x))$ (unit)
- $\forall x \exists y ((x \cdot y = e) \wedge (y \cdot x = e))$ (inverse)

a **model** $\mathcal{X} = \langle X, e_{\mathcal{X}}, \cdot_{\mathcal{X}} \rangle$ for this theory is a **group**

- $\langle \mathbb{Z}/n\mathbb{Z}, 0, + \rangle$
- $\langle \text{GL}_n(\mathbb{Z}), \text{Id}_n, \cdot \rangle$

example: $\mathcal{L} = \{0, 1, +, \cdot\}$ (constant symbols, binary relations
theory of rings with unit

- same as group axioms for $(0, +)$ with additional *commutativity* (abelian group)

$$\forall x \forall y (x + y = y + x)$$

- associativity, unit for $(1, \cdot)$
- additional *distributivity* axiom relating the $+, \cdot$ relations

$$\forall x \forall y \forall z (x \cdot (y + z) = x \cdot y + x \cdot z)$$

a **model** $\mathcal{X} = \langle X, 0_{\mathcal{X}}, 1_{\mathcal{X}}, +_{\mathcal{X}}, \cdot_{\mathcal{X}} \rangle$ for this theory is a **commutative ring with unit**

- $(\mathbb{Z}, 0, 1, +, \cdot)$ also commutativity for \cdot (commutative ring)
- $(M_n(\mathbb{Z}), 0_n, \text{Id}_n, +, \cdot)$ noncommutative ring of matrices

sometimes **infinitely many axioms** are *needed* to specify a theory

example: $\mathcal{L} = \{0, 1, +, \cdot\}$, **theory of fields of characteristic zero**

- same axioms as for rings, and additional axioms $0 \neq 1$ and $\forall x((x \neq 0) \Rightarrow \exists y(x \cdot y = 1))$ (existence of inverses)
- above axioms give **theory of fields**
- for **characteristic zero** need infinite set of axioms:

$$1 + 1 \neq 0, \quad 1 + 1 + 1 \neq 0, \quad \dots \quad \underbrace{1 + \dots + 1}_{n\text{-times}} \neq 0, \quad \dots$$

theories from structures

- \mathcal{L} first-order language and $\mathcal{X} = \langle X, \Phi, \mathcal{R} \rangle$ structure for \mathcal{L}
- the **theory of the structure** \mathcal{X}

$$T(\mathcal{X}) = \{A \in \text{WFF} : A \text{ sentence and } \mathcal{X} \models A\}$$

- this *does not specify axioms for the theory* $T(\mathcal{X})$
- can take all $A \in T(\mathcal{X})$ as axioms but that is not useful
- general question: identify a minimal “reasonable” set of axioms S of such a theory
- “reasonable” = there is an algorithm to decide if a sentence belongs to S
- is it always possible to obtain a sufficient set of axioms? **No**

first-order Peano arithmetic

- $\mathcal{L}_{\text{ar}} = \{0, S, +, \cdot, <\}$
- standard structure $\mathcal{N} = \langle \mathbb{N}, 0, S, +, \cdot, <\rangle$
- axioms of first-order Peano arithmetic

$$(1) \quad \forall x(S(x) \neq 0)$$

$$(2) \quad \forall x \forall y(x \neq y \Rightarrow S(x) \neq S(y))$$

$$(3) \quad \forall x(x + 0 = x)$$

$$(4) \quad \forall x \forall y(x + S(y) = S(x + y))$$

$$(5) \quad \forall x(x \cdot 0 = 0)$$

$$(6) \quad \forall x \forall y(x \cdot S(y) = x \cdot y + x)$$

$$(7)_A \quad \forall y_1 \cdots \forall y_n [(A(0, y_1, \dots, y_n) \wedge \forall x(A(x, y_1, \dots, y_n) \Rightarrow A(S(x), y_1 \dots y_n)))] \Rightarrow \forall x A(x, y_1, \dots, y_n)$$

for any formula $A(x, y_1, \dots, y_n)$. (So this is an infinite set of axioms.)

BUT... these axioms are **not** sufficient to give $T(\mathcal{N})$, in fact there are sentences A true in \mathcal{N} that cannot be derived from this (or any other reasonable) set of axioms: *Gödel incompleteness theorem*

limits of first-order logic

- **induction principle** in first-order Peano arithmetic is stated as a schema for generating an infinite set of axioms
- in **second-order logic** can formulate as a single axiom

$$\forall P(((P(0) \wedge \forall n(P(n) \Rightarrow P(n+1)))) \Rightarrow \forall nP(n))$$

- but here P is a predicate variable (meaning that it involves relations and functions not just propositional variables)
- with the induction axiom can derive $+$, \cdot , $<$ and need only $\{0, S\}$

- with the induction axiom for Peano need only $\{0, S\}$:
 - 1 addition: $a + 0 = a$ and $a + S(b) = S(a + b)$
 - 2 multiplication: $a \cdot 0 = 0$ and $a \cdot S(b) = a + (a \cdot b)$
 - 3 this gives $a \cdot S(0) = a$ and also get $S(0) \cdot a = a$ from $S(0) \cdot 0 = 0$ and induction as: if $S(0) \cdot a = a$ then $S(0) \cdot S(a) = S(0) + S(0) \cdot a = S(0) + a = a + S(0) = S(a + 0) = S(a)$
 - 4 also distributivity, associativity, commutativity follow:
 $(\mathbb{N}, +, \cdot, 0, S(0))$ **commutative semiring**
 - 5 then order from $n \leq m$ iff $\exists k(n + k = m)$
- model for Peano arithmetic $\mathcal{X} = \langle \mathbb{N}, 0, S \rangle$
- any two such $\mathcal{X}_1, \mathcal{X}_2$ have unique isomorphism $f : \mathbb{N}_1 \rightarrow \mathbb{N}_2$ with $f(0_1) = 0_2$ and $f(S(n)) = S'(f(n))$

von Neumann construction: set-theoretic model of \mathbb{N}

- define 0 as the empty set \emptyset
- define S as

$$S(X) = X \cup \{X\}$$

- the set \mathbb{N} is the smallest set (intersection of all sets) closed under the action of S
- this $\langle \mathbb{N}, 0, S \rangle$ satisfies Peano axioms
- Note: this requires a theory of sets (eg Zermelo-Fraenkel)

$$0 = \emptyset$$

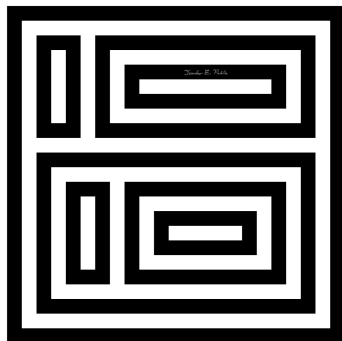
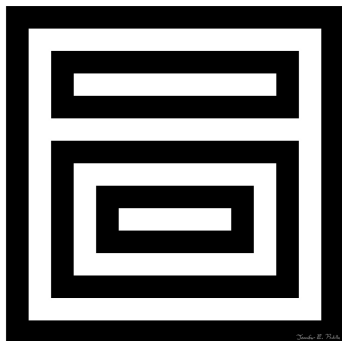
$$1 = s(0) = s(\emptyset) = \emptyset \cup \{\emptyset\} = \{\emptyset\} = \{0\}$$

$$2 = s(1) = s(\{0\}) = \{0\} \cup \{\{0\}\} = \{0, \{0\}\} = \{0, 1\}$$

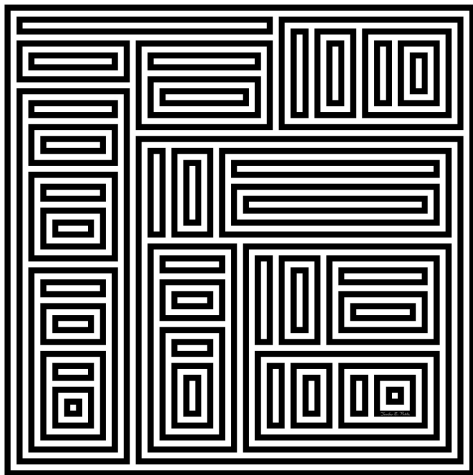
$$3 = s(2) = s(\{0, 1\}) = \{0, 1\} \cup \{\{0, 1\}\} = \{0, 1, \{0, 1\}\} = \{0, 1, 2\}$$

$$\begin{aligned}
 0 &= \{\} && = \emptyset, \\
 1 &= \{0\} && = \{\emptyset\}, \\
 2 &= \{0, 1\} && = \{\emptyset, \{\emptyset\}\}, \\
 3 &= \{0, 1, 2\} && = \{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}\}
 \end{aligned}$$

an artist's impression...



Jennifer E. Padilla, von Neumann ordinals, $N = 2$ and $N = 3$



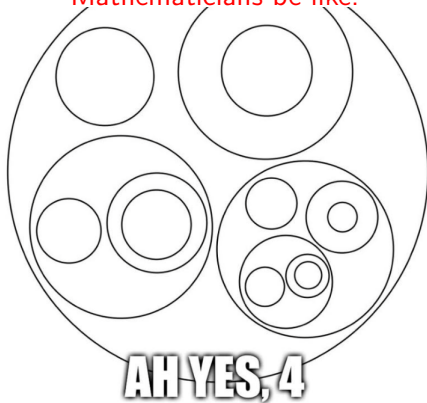
Jennifer E. Padilla, von Neumann ordinals, $N = 6$

or in meme-land...

$\{\{\}, \{\{\}\}, \{\{\}, \{\{\}\}\}, \{\{\}, \{\{\}, \{\{\}, \{\{\}\}\}\}\}$

Set Theory

Mathematicians be like:



note: a finite set is really just its cardinality: if there's anything else going on it means it's not a set but some other richer structure

write $\underline{\mathbb{N}}$ for the theory of \mathcal{L}_{ar} with the Peano axioms

- $\underline{\mathbf{N1}} \quad \forall x (Sx \neq 0)$
- $\underline{\mathbf{N2}} \quad \forall x \forall y (Sx = Sy \rightarrow x = y)$
- $\underline{\mathbf{N3}} \quad \forall x (x + 0 = x)$
- $\underline{\mathbf{N4}} \quad \forall x \forall y (x + Sy = S(x + y))$
- $\underline{\mathbf{N5}} \quad \forall x (x \cdot 0 = 0)$
- $\underline{\mathbf{N6}} \quad \forall x \forall y (x \cdot Sy = x \cdot y + x)$
- $\underline{\mathbf{N7}} \quad \forall x (x \neq 0)$
- $\underline{\mathbf{N8}} \quad \forall x \forall y (x < Sy \leftrightarrow x < y \vee x = y)$
- $\underline{\mathbf{N9}} \quad \forall x \forall y (x < y \vee x = y \vee y < x)$

Models of $\underline{\mathbb{N}}$

- (1) We usually refer to \mathfrak{N} as the *standard model* of $\underline{\mathbb{N}}$.
- (2) Another model of $\underline{\mathbb{N}}$ is $\mathfrak{N}[x] := (\mathbf{N}[x]; \dots)$, where $0, S, +, \cdot$ are interpreted as the zero polynomial, as the unary operation of adding 1 to a polynomial, and as addition and multiplication of polynomials in $\mathbf{N}[x]$, and where $<$ is interpreted as follows: $f(x) < g(x)$ iff $f(n) < g(n)$ for all large enough n .
- (3) A more bizarre model of $\underline{\mathbb{N}}$: $(\mathbf{R}^{\geq 0}; \dots)$ with the usual interpretations of $0, S, +, \cdot$, in particular $S(r) := r + 1$, and with $<$ interpreted as the binary relation $<_{\mathbf{N}}$ on $\mathbf{R}^{\geq 0}$: $r <_{\mathbf{N}} s \Leftrightarrow (r, s \in \mathbf{N} \text{ and } r < s) \text{ or } s \notin \mathbf{N}$.

all contain the standard \mathcal{N}

Hilbert-type Proof System for First Order Logic

- goal: **Gödel completeness theorem**: $S \vdash S$ iff $S \models A$
- basic symbols: $\neg, \Rightarrow, (,), x_1, x_2, \dots, x_n, \dots, \forall$
and view $(A \wedge B)$ as an abbreviation of $\neg(A \Rightarrow \neg B)$, $(A \vee B)$ as an abbreviation of $(\neg A \Rightarrow B)$, $(A \Leftrightarrow B)$ as an abbreviation of $\neg((A \Rightarrow B) \Rightarrow \neg(B \Rightarrow A))$ and
 $\exists xA$ as an abbreviation of $\neg\forall x\neg A$.
- *generalization* of a formula A :

$$\forall y_1 \forall y_2 \cdots \forall y_n A$$

- fix a first-order language $\mathcal{L} = \{\Phi, \mathcal{R}\}$

logical axioms for Hilbert-type Proof System

- (a) (i) $A \Rightarrow (B \Rightarrow A)$
 - (ii) $((A \Rightarrow (B \Rightarrow C)) \Rightarrow ((A \Rightarrow B) \Rightarrow (A \Rightarrow C)))$
 - (iii) $((\neg B \Rightarrow \neg A) \Rightarrow ((\neg B \Rightarrow A) \Rightarrow B))$
 - (b) $(\forall x(A \Rightarrow B) \Rightarrow (\forall xA \Rightarrow \forall xB))$
 - (c) $(A \Rightarrow \forall xA)$, provided x is not free in A
 - (d) $\forall xA \Rightarrow A[x/t]$, provided t is a term *substitutable* for x in A , where this proviso will be explained shortly.
 - (e) (i) $x = x$
 - (ii) $(x = y \wedge y = z) \Rightarrow (x = z)$
 - (iii) $x = y \Rightarrow y = x$
 - (iv) $(y_1 = z_1 \wedge \dots \wedge y_n = z_n) \Rightarrow (f(y_1, \dots, y_n) = f(z_1, \dots, z_n))$
 - (v) $(y_1 = z_1 \wedge \dots \wedge y_m = z_m) \Rightarrow (R(y_1, \dots, y_m) \Leftrightarrow R(z_1, \dots, z_m))$
- for arbitrary variables $x, y, z, x_i, y_i, z_i \dots$, formulae A, B, C, \dots , functions $f \in \Phi_n$ and relations $R \in \mathcal{R}_m$

- term t **substitutable** for variable x in formula A if no *free* occurrence of x in A is in the scope of a quantifier $\forall z$ or $\exists z$ with z a variable in t
- example: $f(x)$ *not* substitutable for z in formula $\forall x(R(x, z) \Rightarrow \forall yQ(y))$, but substitutable for z in $\forall yP(z, y)$

inference rule: modus ponens $\{A, A \Rightarrow B\} \models B$

$$\frac{A, A \Rightarrow B}{B}$$

formal proof from S $S \vdash A$

- set of formulae S (set of strings in the formal language)
- string of formulae $A_1, A_2, \dots, A_n = A$
- each A_i is either a logical axiom, or an element of S , or the result of applying modus ponens (for some $j < i$)

$$\frac{A_j, (A_j \Rightarrow A_i)}{A_i}$$

formally provable $S \vdash A$ equivalently: A is a **formal theorem** of S ($S = \emptyset$, then A is a formal theorem, $\vdash A$)

- same as in propositional logic:

- 1 **deduction:**

$$S \cup \{A\} \vdash B \quad \text{iff} \quad S \vdash (A \Rightarrow B)$$

- 2 **S formally inconsistent:** if $S \vdash A$ and $S \vdash \neg A$ for some A

- 3 **proof by contradiction:** $S \cup \{A\}$ formally inconsistent then $S \vdash \neg A$

- 4 **proof by contrapositive:** if $S \cup \{A\} \vdash \neg B$ then $S \cup \{B\} \vdash \neg A$

- quantification:

- 1 **generalization:** if $S \vdash A$ and x *not* a free variable in any formula of S then $S \vdash \forall xA$

generalization: by induction

- induction on proofs of A from S

Basis.

- (i) A is a logical axiom. Then $\forall xA$ is also a logical axiom, so clearly $S \vdash \forall xA$.
- (ii) A is in S . Then, by hypothesis, x is not free in A . So $A \Rightarrow \forall xA$ is a logical axiom. By MP then, $S \vdash \forall xA$.

Induction Step. Assume that $S \vdash \forall xA$, $S \vdash \forall x(A \Rightarrow B)$ in order to show that $S \vdash \forall xB$. Since $\forall x(A \Rightarrow B) \Rightarrow (\forall xA \Rightarrow \forall xB)$ is a logical axiom, this follows by MP applied twice. \dashv

generalization on constants

- \mathcal{L} first-order language, S set of formulae of \mathcal{L}
- $A(x)$ a formula and c a constant symbol *not* in \mathcal{L}
- if $S \vdash_{\mathcal{L} \cup \{c\}} A[x/c]$ then $S \vdash_{\mathcal{L}} \forall x A(x)$

Steps:

- since $S \vdash A[x/c]$ take $A_1, \dots, A_k = A[x/c]$ a proof in $\mathcal{L} \cup \{c\}$ (uses a finite subset S_0)
- substitute each c in A_i with a variable y not occurring anywhere in S_0 so $S_0 \vdash A[x/y]$
- generalization: $S_0 \vdash \forall y A[x/y]$
- x is substitutable for y with $A[x/y][y/x] = A(x)$
- so $\forall y A[x/y] \Rightarrow A(x)$ is axiom
- so get $\forall y A[x/y] \vdash A(x)$
- generalization: $\forall y A[x/y] \vdash \forall x A(x)$
- modus ponens: $S_0 \vdash \forall x A(x)$

Formal Theory

- **formal theory** T : set of sentences in \mathcal{L} **closed under formal provability**:

$$\text{if } T \vdash A \text{ then } A \in T$$

- **complete** set of sentences S if for any sentence A either $S \vdash A$ or $S \vdash \neg A$
- **formally inconsistent** if for some sentence A have $S \vdash A$ and $S \vdash \neg A$ and **formally consistent** otherwise
- a formal theory T that is **both formally consistent and complete**: for any sentence A , one of A or $\neg A$ belongs to T

closed terms

- **closed term**: a term with no free variables (substitutable for any x)
- take set of closed terms

$$C = \{t : t \text{ closed term in } \mathcal{L}\}$$

- equivalence relation on set of closed terms: $s \sim t$ iff the formula $s = t$ is a sentence in T

(i) $s \sim s$.

This means that $s = s \in T'$. But $\forall x(x = x) \Rightarrow s = s$ is a logical axiom (d) and $\forall x(x = x)$ is a logical axiom (e), so, by MP, $\vdash s = s$, so $T' \vdash s = s$, thus $s = s \in T'$.

(ii) $s \sim t$ implies $t \sim s$.

We have that $s = t \in T'$. Now $\forall x \forall y(x = y \Rightarrow y = x)$ is a logical axiom (e) and $\vdash \forall x \forall y(x = y \Rightarrow y = x) \Rightarrow (s = t \Rightarrow t = s)$, so by MP, $\vdash (s = t) \Rightarrow (t = s)$ so as $s = t \in T'$, $T' \vdash t = s$, thus $t = s \in T'$ (since T' is a formal theory), i.e., $t \sim s$.

(iii) $s \sim t$ and $t \sim u$ imply $s \sim u$.

Again $\forall x \forall y \forall z((x = y \wedge y = z) \Rightarrow x = z)$ is a logical axiom (e) and $\vdash \forall x \forall y \forall z((x = y \wedge y = z) \Rightarrow x = z) \Rightarrow ((s = t \wedge t = u) \Rightarrow s = u)$ by using three times logical axiom (d) and MP, so, by MP, $\vdash (s = t \wedge t = u) \Rightarrow s = u$, so $T \vdash (s = t \wedge t = u) \Rightarrow s = u$ and since $s = t, t = u \in T'$ it follows that $T' \vdash s = u$, so $s = u \in T'$.

Gödel Completeness

- \mathcal{L} first-order language, S set of formulae, A formula

$$S \models A \quad \text{iff} \quad S \vdash A$$

- 1 **soundness** (easier part): if $S \vdash A$ then $S \models A$ (as in the case of propositional logic)
 - 2 **completeness**: if $S \models A$ then $S \vdash A$
- focus on proof of completeness

Henkin Theory

- a formal theory T for a first-order language \mathcal{L} with the property that, for any formula $A(x)$ there is a constant $c = c_A$ (Henkin witness) such that

$$\neg\forall xA(x) \Rightarrow \neg A[x/c]$$

is a formula in T

Model

- if \mathcal{L} is a first-order language and T a formally consistent complete Henkin theory for \mathcal{L} , then T has a model \mathcal{M}

Construction of Model

- **universe** of the model: take equivalence classes of closed terms

$$M = \{[t] : t \text{ is a closed term}\} = C / \sim$$

- function and relations symbols in \mathcal{L} : interpretation in the model

$$f_M([s_1], \dots, [s_n]) = [f(s_1, \dots, s_n)]$$

$$R_M([s_1], \dots, [s_m]) \quad \text{iff} \quad R(s_1, \dots, s_m) \in T$$

- both well defined because

$$\vdash (s_1 = t_1 \wedge \dots \wedge s_n = t_n) \Rightarrow (f(s_1, \dots, s_n) = f(t_1, \dots, t_n))$$

$$\vdash (s_1 = t_1 \wedge \dots \wedge s_m = t_m) \Rightarrow (R(s_1, \dots, s_m) \Leftrightarrow R(t_1, \dots, t_m))$$

Model property

- also have that $\mathcal{M} \models T$ \mathcal{M} is a model of T :

$$\mathcal{M} \models A \quad \text{iff} \quad A \in T$$

- induction on $N(A)$ = number of $\neg, \Rightarrow, \forall$ symbols in A
- if $N(A) = 0$ (A atomic): A is either $s = t$ or $R(s_1, \dots, s_n)$ for some closed terms s, t, s_i

$$\begin{aligned} \mathcal{M} \models s = t &\text{ iff } s^{\mathcal{M}} = t^{\mathcal{M}} \text{ iff } [s] = [t] \\ &\text{ iff } s \sim t \text{ iff } s = t \in T'. \end{aligned}$$

$$\begin{aligned} \mathcal{M} \models R(s_1, \dots, s_m) &\text{ iff } R^{\mathcal{M}}(s_1^{\mathcal{M}}, \dots, s_m^{\mathcal{M}}) \\ &\text{ iff } R^{\mathcal{M}}([s_1], \dots, [s_m]) \\ &\text{ iff } R(s_1, \dots, s_m) \in T'. \end{aligned}$$

- induction: if true for $N(B) \leq n$ and now take A with $N(A) = n + 1$, cases
 - 1 $\neg B$
 - 2 $B \Rightarrow C$
 - 3 $\forall x B(x)$

- **case $\neg B$** ($N(B) = n$ so $\mathcal{M} \models B$ iff $B \in T$): have $\mathcal{M} \models A$ iff $\mathcal{M} \not\models B$ iff $B \notin T$ iff $\neg B \in T$ (formally consistent & complete) iff $A \in T$
- **case $B \Rightarrow C$** (where $\mathcal{M} \models B$ iff $B \in T$ and $\mathcal{M} \models C$ iff $C \in T$): have

$$\begin{aligned}
 \mathcal{M} \models A &\text{ iff } \mathcal{M} \not\models B \text{ or } \mathcal{M} \models C \\
 &\text{ iff } B \notin T' \text{ or } C \in T' \\
 &\text{ iff } \neg B \in T' \text{ or } C \in T' \\
 &\text{ iff } B \Rightarrow C \in T' \\
 &\text{ iff } A \in T'
 \end{aligned}$$

- **case $\forall xB(x)$** (if $N(B) = n$ also $N(B[x/t]) = n$ so $\mathcal{M} \models B[x/t]$ iff $B[x/t] \in T$)
- need to show $\mathcal{M} \models \forall xB(x)$ iff $\forall xB(x) \in T$
- suppose $\mathcal{M} \models \forall xB(x)$ then since $\forall xB(x) \Rightarrow B[x/t]$ is valid have $\mathcal{M} \models B[x/t]$, so $B[x/t] \in T$ for any closed term t
- but if $\forall xB(x) \notin T$ have $\neg\forall xB(x) \in T$ and by Henkin property there is c with $\neg B[x/c] \in T$ contradiction
- then suppose $\forall xB(x) \in T$: logical axiom $\forall xB(x) \Rightarrow B[x/t]$ so $B[x/t] \in T$, so by induction $\mathcal{M} \models B[x/t]$ (and depends only on class $[t]$ of t)
- but classes $[t] = a$ are all the points of M of model \mathcal{M} so $\mathcal{M} \models B[x/a]$ gives $\mathcal{M} \models \forall xB(x)$

building a formally consistent complete Henkin theory

- start with \mathcal{L} first-order language and S formally consistent set of sentences in \mathcal{L}
- assume that sets Φ , \mathcal{R} of functions and relations of \mathcal{L} are countable, so can enumerate all formulae (hence all sentences) of \mathcal{L} (holds more generally)
- then need to show:

adding constant symbols to \mathcal{L} obtain a first-order language $\mathcal{L}' \supseteq \mathcal{L}$ and this has a formally consistent complete Henkin theory T' with $T' \supseteq S$

- **Gödel completeness** then follows from this:
 $S \models A$ and $S \subset T$ so $\mathcal{M} \models S$ and $\mathcal{M} \models A$, so $S \cup \{A\} \subset T$, formal consistence and completeness $\neg A \notin T$ so $S \cup \{\neg A\}$ inconsistent (otherwise get same way consistent complete $T' \supset T$ with $T' \ni \neg A$ but $T \cup \{\neg A\}$ inconsistent)

- define \mathcal{L}' as \mathcal{L} with added countable set of constant symbols (still countable)
 - 1 obtain S' a formally consistent Henkin theory for \mathcal{L}' with $S' \supseteq S$
 - 2 then need to find a formally consistent **and complete** theory $T' \supseteq S'$ for \mathcal{L}' (will continue to be Henkin)
- first step: need S' enlarging S by Henkin witnesses
- done recursively: \mathcal{L}_n and S_n with $\mathcal{L}_0 = \mathcal{L}$ and $\mathcal{L}_n \subseteq \mathcal{L}_{n+1}$ and $S_0 = S$ and $S_n \subseteq S_{n+1}$ where \mathcal{L}_{n+1} adds to \mathcal{L}_n one new constant symbol c_A for each sentence $\neg\forall xA(x)$ in \mathcal{L}_n (at most countably many constants c_A) and S_{n+1} adds to S_n all sentences

$$\neg\forall xA(x) \Rightarrow \neg A[x/c_A]$$

- second step proceeds exactly as in propositional logic
- then take $\mathcal{L}' = \bigcup_n \mathcal{L}_n$ and $S'' = \bigcup_n S_n$
- take formal theory generated by S''

$$S' = \{A \text{ sentence in } \mathcal{L}' : S'' \vdash A\}$$

- check that S'' still formally consistent (then S' also is)
- S is, then assume S_n is: if S_{n+1} inconsistent there are formulae $A_1(y_1), \dots, A_k(y_k)$ in \mathcal{L}_n and new constants c_1, \dots, c_k in \mathcal{L}_{n+1} with inconsistent

$$S_n \cup_{i=1}^k \{ \neg \forall y_i A_i(y_i) \Rightarrow \neg A_i[y_i/c_i] \}$$

- take $Q := S_n \cup_{i=1}^{k-1} \{ \neg \forall y_i A_i(y_i) \Rightarrow \neg A_i[y_i/c_i] \}$
- proof by contradiction

$$Q \vdash \neg(\neg \forall y_k A_k(y_k) \Rightarrow \neg A_k[y_k/c_k])$$

- equivalently $Q \vdash \neg \forall y_k A_k(y_k)$ and $Q \vdash A_k[y_k/c_k]$ since

$$\neg(\neg \forall y_k A_k(y_k) \Rightarrow \neg A_k[y_k/c_k]) \equiv \neg \forall y_k A_k(y_k) \wedge A_k[y_k/c_k]$$

from $X \vee Y \equiv \neg X \Rightarrow Y$

- by **generalization on constants** (c_k not anywhere in formulas in Q) $Q \vdash A_k[y_k/c_k]$ gives $Q \vdash \forall y_k A_k(y_k)$
- so Q inconsistent; same successively removing each c_i -formula until get S_n inconsistent **contradiction**

Gödel Compactness for first-order logic

- 1 \mathcal{L} first order language, S set of sentences of \mathcal{L} : if $S \models A$ then there is a **finite** subset $S_0 \subset S$ with $S_0 \models A$
- 2 if every **finite** subset $S_0 \subset S$ has a model then S has a model

compactness is a consequence of the completeness theorem as in the case of propositional logic

Gödel Incompleteness: preview

- $\mathcal{L}_{ar} = \{0, S, +, \cdot, <\}$ first-order language of arithmetic
- $\mathcal{N} = \langle \mathbb{N}, 0, S, +, \cdot, <\rangle$ standard structure
- theory $T(\mathcal{N})$ sentences A with $\mathcal{N} \models A$ (true in \mathcal{N})
- $T_{\mathcal{P}} = \text{Con}(\mathcal{P})$ **Peano theory**: sentences A provable from Peano axioms
- difference between true in a given structure and provable from a given set of axioms
- **incompleteness and undecidability of arithmetic**: it is possible to construct a sentence A in \mathcal{L}_{ar} that $A \in T(\mathcal{N})$ (true in \mathcal{N}) but $A \notin T_{\mathcal{P}}$ (not provable from Peano axioms)
- **cannot exist an algorithm** that, given A sentence in \mathcal{L}_{ar} , decides if $A \in T(\mathcal{N})$ (hence no “reasonable” set S of axioms for $T(\mathcal{N})$, with “reasonable” meaning an algorithm can decide if a sentence is in S)

in comparison: **Tarski's theorem on the field of real numbers**

- every sentence in \mathcal{L}_{ar} true in the structure $\langle \mathbb{R}, 0, 1, +, \cdot, < \rangle$ is **provable** from the **axioms of ordered fields** with additional axioms
 - every positive element is a square
 - every odd degree polynomial has a zero
- there is an algorithm that decides for any given sentence in \mathcal{L}_{ar} if it is true in the structure $\langle \mathbb{R}, 0, 1, +, \cdot, < \rangle$

more precise statement for Gödel incompleteness

if S is a **computable** set of sentences in the language \mathcal{L}_{ar} that is true in \mathcal{N} (i.e. $S \subset T(\mathcal{N})$) then there exists sentence A of \mathcal{L}_{ar} with $\mathcal{N} \models A$ but $S \not\models A$

need to first discuss **computability**

Computable Functions

preliminary notation:

- $\mu x(A(x)) = \text{least } x \in \mathbb{N} \text{ such that } A(x) \text{ holds true (example: } \mu x(x^2 > 7) = 3)$
- $\mu_{<a} x(A(x)) = \text{same with least } x < a \in \mathbb{N}$
- relation $R \subseteq \mathbb{N}^m$: characteristic function $\chi_R(a) = 1$ if $a \in R$ and $\chi_R(a) = 0$ if $a \notin R$
- $\chi_{<}(m, n) = 1$ if $m < n$ and zero otherwise
- coordinate functions $l_i : \mathbb{N}^n \rightarrow \mathbb{N}$ with $l_i(a_1, \dots, a_n) = a_i$

Computable Functions (recursive functions)

Definition. The *computable functions* (or *recursive functions*) are the functions from \mathbf{N}^n to \mathbf{N} (for $n = 0, 1, 2, \dots$) obtained by inductively applying the following rules:

- (R1) $+$: $\mathbf{N}^2 \rightarrow \mathbf{N}$, \cdot : $\mathbf{N}^2 \rightarrow \mathbf{N}$, χ_{\leq} : $\mathbf{N}^2 \rightarrow \mathbf{N}$, and the coordinate functions I_i^n (for each n and $i = 1, \dots, n$) are computable.
- (R2) If $G : \mathbf{N}^m \rightarrow \mathbf{N}$ is computable and $H_1, \dots, H_m : \mathbf{N}^n \rightarrow \mathbf{N}$ are computable, then so is the function $F = G(H_1, \dots, H_m) : \mathbf{N}^n \rightarrow \mathbf{N}$ defined by

$$F(a) = G(H_1(a), \dots, H_m(a)).$$

- (R3) If $G : \mathbf{N}^{n+1} \rightarrow \mathbf{N}$ is computable, and for all $a \in \mathbf{N}^n$ there exists $x \in \mathbf{N}$ such that $G(a, x) = 0$, then the function $F : \mathbf{N}^n \rightarrow \mathbf{N}$ given by

$$F(a) = \mu x (G(a, x) = 0)$$

is computable.

A relation $R \subseteq \mathbf{N}^n$ is said to be *computable* (or *recursive*) if its characteristic function $\chi_R : \mathbf{N}^n \rightarrow \mathbf{N}$ is computable.

some direct consequences

- all compositions of computable are computable: e.g.
 $F(G(x_i, x_j), x_2, \dots, x_n)$ viewed as
 $F(G(I_i(x), I_j(x), I_2(x), \dots, I_n(x)))$
- functions $\chi_{\geq}(n, m) = \chi_{\leq}(m, n)$ and
 $\chi_{=}(n, m) = \chi_{\geq}(n, m) \cdot \chi_{\leq}(n, m)$ computable
- all constant functions computable
- compositions with relations $R(H_1(x), \dots, H_m(x))$ are
computable as $\chi_{R(H_1, \dots, H_m)} = \chi_{R(H_1, \dots, H_m)}$
- if P, Q computable then $\neq P, P \wedge Q, P \vee Q, P \rightarrow Q, P \leftrightarrow Q$
computable
- binary relations $=, \neq, <, >, \leq, \geq$ computable

some additional useful cases of **computable** functions:

- a function $F : \mathbb{N}^n \rightarrow \mathbb{N}$ computable iff its graph $R = \Gamma(F) \subset \mathbb{N}^{n+1}$ computable

$$F(a) = \mu x R(a, x)$$

- $R_1, \dots, R_k \subseteq \mathbb{N}^n$ computable relations where for each $a \in \mathbb{N}^n$ exactly one of $R_i(a)$ holds, $G_1, \dots, G_k : \mathbb{N}^n \rightarrow \mathbb{N}$ computable functions, $P_1, \dots, P_k \subset \mathbb{N}^n$ computable relations: then G, P defined “by cases” also computable:

$$G(a) = \begin{cases} G_1(a) & \text{if } R_1(a) \\ \vdots & \vdots \\ G_k(a) & \text{if } R_k(a) \end{cases} \quad P(a) \iff \begin{cases} P_1(a) & \text{if } R_1(a) \\ \vdots & \vdots \\ P_k(a) & \text{if } R_k(a) \end{cases}$$

$$G = G_1 \cdot \chi_{R_1} + \dots + G_k \cdot \chi_{R_k} \quad \text{and} \quad P = (P_1 \wedge R_1) \vee \dots \vee (P_k \wedge R_k)$$

- the **pairing function** is computable (and bijective)

$$\text{Pair} : \mathbb{N}^2 \rightarrow \mathbb{N}$$

$$\text{Pair}(x, y) := \frac{(x + y)(x + y + 1)}{2} + x$$

- associated functions (well defined because of bijectivity)

$$\text{Left}, \text{Right} : \mathbb{N} \rightarrow \mathbb{N}$$

$$\text{Pair}(x, y) = a \quad \text{iff} \quad \text{Left}(a) = x \quad \text{and} \quad \text{Right}(a) = y$$

also computable

$$\begin{aligned} \text{Left}(a) &= \mu x (\exists y_{<a+1} \text{Pair}(x, y) = a), \\ \text{Right}(a) &= \mu y (\exists x_{<a+1} \text{Pair}(x, y) = a). \end{aligned}$$

- ternary relation $a \equiv b \pmod{c}$ in \mathbb{N} is computable

$$(a \equiv b \pmod{c}) \Leftrightarrow (\exists x_{<a+1} a = x \cdot c + b) \vee (\exists x_{<b+1} b = x \cdot c + a)$$

- Gödel function $\beta : \mathbb{N}^2 \rightarrow \mathbb{N}$

$$\beta(a, i) := \mu x (x \equiv \text{Left}(a) \pmod{1 + (i + 1)\text{Right}(a)})$$

is computable, with $\beta(a, i) \leq \text{Left}(a) \leq a - 1$

- this function also satisfies: for any sequence a_0, a_1, \dots, a_{n-1} in \mathbb{N} there is an $a \in \mathbb{N}$ with

$$\beta(a, 0) = a_0, \dots, \beta(a, n - 1) = a_{n-1}$$

Let $a_0, \dots, a_{n-1} \in \mathbb{N}$. Take $N \in \mathbb{N}$ such that $a_i \leq N$ for all $i < n$ and N is a multiple of every prime number less than n . We claim that then $1 + N, 1 + 2N, \dots, 1 + nN$ are relatively prime. To see this, suppose p is a prime number such that $p \mid 1 + iN$ and $p \mid 1 + jN$ ($1 \leq i < j \leq n$); then p divides their difference $(j - i)N$, but $p \equiv 1 \pmod{N}$, so p does not divide N , hence $p \mid j - i < n$. But all prime numbers $< n$ divide N , and we have a contradiction.

By the Chinese Remainder Theorem there exists an $M \in \mathbb{N}$ such that

$$\begin{aligned} M &\equiv a_0 \pmod{1 + N} \\ M &\equiv a_1 \pmod{1 + 2N} \\ &\vdots \\ M &\equiv a_{n-1} \pmod{1 + nN}. \end{aligned}$$

Put $a := \text{Pair}(M, N)$; then $\text{Left}(a) = M$ and $\text{Right}(a) = N$, and thus $\beta(a, i) = a_i$ as required. \square

encoding sequences

- can use Gödel's β to **encode a sequence of numbers as a single number**
- if want to also keep track of *length* of sequence assign to a_1, \dots, a_n **sequence number**: least a with $\beta(a, 0) = n$, $\beta(a, i) = a_i$ for $i = 1, \dots, n$
- **injective** because can reconstruct a_i from a by $\beta(a, i) = a_i$
- length function $\ell(a) = \beta(a, 0)$ computable, each $(a)_i := \beta(a, i + 1)$ computable
- set $\text{Seq} \subseteq \mathbb{N}$ of sequence numbers is computable

$$(a \in \text{Seq}) \Leftrightarrow \forall x < a ((\ell(x) \neq \ell(a)) \vee (\exists i < \ell(a) (x)_i \neq a_i))$$

- function $(a_1, \dots, a_n) \mapsto \langle a_1, \dots, a_n \rangle = a$ sequence number is computable

operations on sequences and encoding

- sequences a_1, \dots, a_n and b_1, \dots, b_m with $a_i, b_j \in \mathbb{N}$,
computable functions $\text{In} : \mathbb{N}^2 \rightarrow \mathbb{N}$ and $\star : \mathbb{N}^2 \rightarrow \mathbb{N}$

$$\text{In}(\langle a_1, \dots, a_n \rangle, i) := \langle a_1, \dots, a_i \rangle$$

$$\langle a_1, \dots, a_n \rangle \star \langle b_1, \dots, b_m \rangle := \langle a_1, \dots, a_n, b_1, \dots, b_m \rangle$$

$$\text{In}(a, i) = \mu x (\text{lh}(x) = i \text{ and } \forall j < i (x)_j = (a)_j),$$

$$a \star b = \mu x (\text{lh}(x) = \text{lh}(a) + \text{lh}(b) \text{ and } \forall i < \text{lh}(a) (x)_i = (a)_i$$

$$\text{and } \forall j < \text{lh}(b) (x)_{\text{lh}(a)+j} = (b)_j)$$

Encoding by prime factorization

- way of encoding sequences $(a_1, \dots, a_n) \in \mathbb{N}^n$ by a single number $\langle a_1, \dots, a_n \rangle$ using uniqueness of prime factorization
- take sequence of prime numbers $2, 3, 5, 7, 11, 13, \dots, p_k, \dots$ by ordering in \mathbb{N} and map (a_1, \dots, a_n) to

$$\langle a_1, \dots, a_n \rangle = 2^{a_1} 3^{a_2} 5^{a_3} \dots p_n^{a_n}$$

- have injectivity $\langle a_1, \dots, a_n \rangle = \langle b_1, \dots, b_m \rangle$ iff $n = m$ and $a_i = b_i$ by uniqueness of prime decomposition
- simple encoding but decoding (identifying a_i from sequence number) depends on prime factorization of (possibly large) numbers
- both sequence number $\langle a_1, \dots, a_n \rangle$ via β function and by prime factorization used in Gödel's 1931 and 1932 papers on Incompleteness

problem with computable (recursive) functions

- third rule $F(a) = \mu x(G(a, x) = 0)$ in definition of computable function is **not** constructive: there is not necessarily an algorithm (an explicit procedure) for checking if for a computable function G there is some $a \in \mathbb{N}^n$ with $G(a, x) = 0$
- in fact it is **impossible** to have a constructive generative process for all computable functions
- **diagonal argument**: assume \exists sequence α_i of algorithm that define computable functions f_i and every computable function occurs in the sequence $f_0, f_1, \dots, f_n, \dots$; define

$$f_{\text{diag}}(n) = f_n(n) + 1$$

- $f_{\text{diag}}(n)$ is also computable, but $f_{\text{diag}} \neq f_n$ for all $n \in \mathbb{N}$

diagonal argument

	0	1	2	3	4	5	6	...
f_0	$f_0(0)$	$f_0(1)$	$f_0(2)$	$f_0(3)$	$f_0(4)$	$f_0(5)$	$f_0(6)$...
f_1	$f_1(0)$	$f_1(1)$	$f_1(2)$	$f_1(3)$	$f_1(4)$	$f_1(5)$	$f_1(6)$...
f_2	$f_2(0)$	$f_2(1)$	$f_2(2)$	$f_2(3)$	$f_2(4)$	$f_2(5)$	$f_2(6)$...
f_3	$f_3(0)$	$f_3(1)$	$f_3(2)$	$f_3(3)$	$f_3(4)$	$f_3(5)$	$f_3(6)$...
f_3	$f_3(0)$	$f_3(1)$	$f_3(2)$	$f_3(3)$	$f_3(4)$	$f_3(5)$	$f_4(6)$...
f_5	$f_5(0)$	$f_5(1)$	$f_5(2)$	$f_5(3)$	$f_5(4)$	$f_5(5)$	$f_5(6)$...
f_6	$f_6(0)$	$f_6(1)$	$f_6(2)$	$f_6(3)$	$f_6(4)$	$f_6(5)$	$f_6(6)$...
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	

primitive recursive functions

- a class of computable functions that can be defined constructively
- inductive construction by generators and operations
- generated by *basic functions*
 - 1 Successor $S : \mathbb{N} \rightarrow \mathbb{N}$, $S(x) = x + 1$;
 - 2 Constant $c^n : \mathbb{N}^n \rightarrow \mathbb{N}$, $c^n(x) = 0$ (for $n \geq 0$);
 - 3 Projection (coordinate functions) $\pi_i^n : \mathbb{N}^n \rightarrow \mathbb{N}$, $\pi_i^n(x) = x_i$ (for $n \geq 1$);
- with *elementary operations*
 - 1 Composition
 - 2 Bracketing
 - 3 Recursion (primitive recursion)

elementary operations:

- ① Composition $\mathfrak{c}_{(m,m,p)}$: for $f : \mathbb{N}^m \rightarrow \mathbb{N}^n$, $g : \mathbb{N}^n \rightarrow \mathbb{N}^p$,

$$g \circ f : \mathbb{N}^m \rightarrow \mathbb{N}^p, \quad \mathcal{D}(g \circ f) = f^{-1}(\mathcal{D}(g));$$

- ② Bracketing $\mathfrak{b}_{(k,m,n_i)}$: for $f_i : \mathbb{N}^m \rightarrow \mathbb{N}^{n_i}$, $i = 1, \dots, k$,

$$f = (f_1, \dots, f_k) : \mathbb{N}^m \rightarrow \mathbb{N}^{n_1 + \dots + n_k}, \quad \mathcal{D}(f) = \mathcal{D}(f_1) \cap \dots \cap \mathcal{D}(f_k);$$

- ③ Recursion \mathfrak{r}_n : for $f : \mathbb{N}^n \rightarrow \mathbb{N}$ and $g : \mathbb{N}^{n+2} \rightarrow \mathbb{N}$,

$$h(x_1, \dots, x_n, 1) := f(x_1, \dots, x_n),$$

$$h(x_1, \dots, x_n, k+1) := g(x_1, \dots, x_n, k, h(x_1, \dots, x_n, k)), \quad k \geq 1,$$

where recursively $(x_1, \dots, x_n, 1) \in \mathcal{D}(h)$ iff $(x_1, \dots, x_n) \in \mathcal{D}(f)$

and $(x_1, \dots, x_n, k+1) \in \mathcal{D}(h)$ iff

$$(x_1, \dots, x_n, k, h(x_1, \dots, x_n, k)) \in \mathcal{D}(g).$$

here $\mathcal{D}(f) \subseteq \mathbb{N}^n$ domain of f (relevant for computable case)

from primitive recursive to partial recursive functions

- enlarge from primitive recursive to partial recursive: same elementary operations \mathfrak{c} , \mathfrak{b} , \mathfrak{r} of composition, bracketing and recursion but additional μ operation
- μ operation: input function $f : \mathbb{N}^{n+1} \rightarrow \mathbb{N}$, output

$$h : \mathbb{N}^n \rightarrow \mathbb{N}, \quad h(x_1, \dots, x_n) = \min\{x_{n+1} \mid f(x_1, \dots, x_{n+1}) = 0\},$$

with domain $\mathcal{D}(h)$ those (x_1, \dots, x_n) such that $\exists x_{n+1} \geq 1$

$$f(x_1, \dots, x_{n+1}) = 0, \quad \text{with } (x_1, \dots, x_n, k) \in \mathcal{D}(f), \forall k \leq x_{n+1}$$

restricted minimization

- a relation $R \subseteq \mathbb{N}^n$ is primitive recursive iff characteristic function χ_R is primitive recursive
- $R \subseteq \mathbb{N}^{n+1}$ and $H : \mathbb{N}^n \rightarrow \mathbb{N}$ primitive recursive and $\forall a \in \mathbb{N}^n \exists x < H(a)$ with $R(a, x)$, then $F(a) = \mu x R(a, x)$ is primitive recursive
- applies to β by previous argument on encoding sequences so $\text{Seq} \subseteq \mathbb{N}$ is primitive recursive

note on terminology

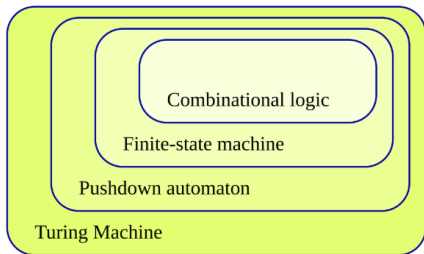
- sometimes what we called “computable functions” is also called “general recursive” or “partial recursive” or “ μ -recursive”
- what we called “primitive recursive” is sometimes just called “total computable”

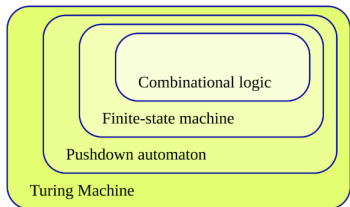
other properties

- primitive recursive functions are **total** (so no need to worry about domain $\mathcal{D}(f)$)
- partial recursive functions in general not (but some can also be total and not primitive)

Computability and Turing Machines

- mathematical models of computation: **automata**
- hierarchies of automata (e.g. Chomsky hierarchy of formal languages and their machine recognition)
- some steps of automata hierarchy:





- **combinatorial logic**: circuits computing Boolean functions through classical propositional logic
- **finite-state automata** (FSA): memoryless processes, directed graphs with states (vertices) and instructions (edges), *regular languages*
- **pushdown stack automata**: FSA plus last-in/first-out memory stack, *context-free languages*
- intermediate class of “linear-bounded Turing machines”, *context-sensitive languages*
- **Turing machines**: universal model for computation, *recursively enumerable languages*

Finite state automaton (FSA)

$$M = (Q, F, \mathcal{A}, \tau, q_0)$$

- Q finite set: set of possible states
- F subset of Q : the final states
- \mathcal{A} finite set: alphabet
- $\tau \subset Q \times \mathcal{A} \times Q$ set of transitions
- $q_0 \in Q$ initial state

computation in M : sequence $q_0 a_1 q_1 a_2 q_2 \dots a_n q_n$ where $q_{i-1} a_i q_i \in \tau$ for $1 \leq i \leq n$

- label of the computation: $a_1 \dots a_n$
- successful computation: $q_n \in F$
- M **accepts** a string $a_1 \dots a_n$ if there is a successful computation in M labeled by $a_1 \dots a_n$

Language recognized by M

= *sequences of instructions (programs) that run on M*

$$\mathcal{L}_M = \{w \in \mathcal{A}^* \mid w \text{ accepted by } M\}$$

can think of \mathcal{L}_M as *what M computes* (formal languages theory)
with current state as output of sequence of instruction with input
initial state

the only *memory* in an FSA is the *states* it has

Graphical description of FSA

Transition diagram: oriented finite labelled graph Γ with vertices $V(\Gamma) = Q$ set of states and $E(\Gamma) = \tau$, with $e_{q,a,q'}$ an edge from v_q to $v_{q'}$ with label $a \in \mathfrak{A}$; label vertex q_0 with $-$ and all final states vertices with $+$

- computations in $M \Leftrightarrow$ paths in Γ starting at v_{q_0}
- an oriented labelled finite graph with at most one edge with a given label between given vertices, and only one vertex labelled $-$ is the transition diagram of some FDA

deterministic FSA

for all $q \in Q$ and $a \in \mathfrak{A}$, there is a unique $q' \in Q$ with $(q, a, q') \in \tau$

\Rightarrow function $\delta : Q \times \mathfrak{A} \rightarrow Q$ with $\delta(q, a) = q'$, *transition function*

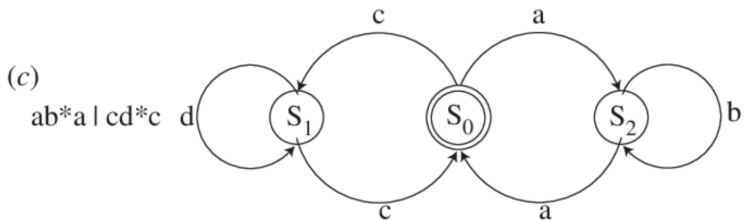
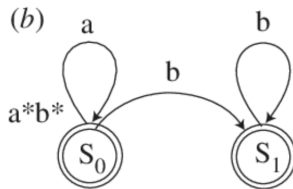
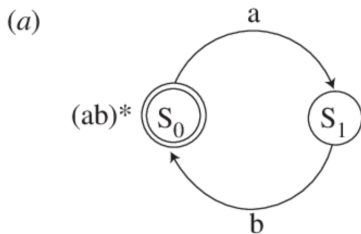
determines $\delta : Q \times \mathfrak{A}^* \rightarrow Q$ by $\delta(q, \epsilon) = q$ and

$\delta(q, wa) = \delta(\delta(q, w), a)$ for all $w \in \mathfrak{A}^*$ and $a \in \mathfrak{A}$

if $q_0 a_1 q_1 \dots a_n q_n$ computation in M then $q_n = \delta(q_0, a_1 \dots a_n)$

non-deterministic: multivalued transition functions also allowed

FSA examples



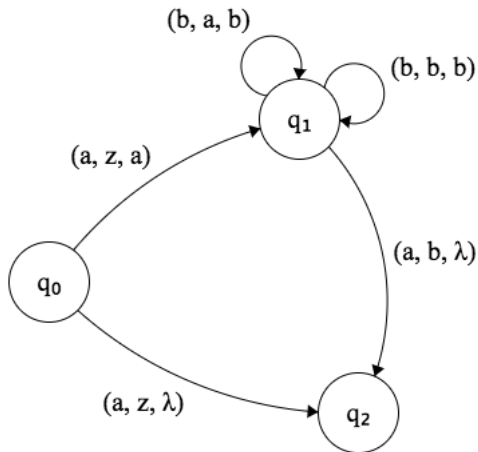
pushdown stack automaton (PDA)

$$M = (Q, F, \mathcal{A}, \Gamma, \tau, q_0, z_0)$$

- Q finite set of possible states
- F subset of Q : the final states
- \mathcal{A} finite set: alphabet
- Γ finite set: *stack alphabet*
- $\tau \subset Q \times (\mathcal{A} \cup \{\epsilon\}) \times \Gamma \times Q \times \Gamma^*$ finite subset: set of transitions
- $q_0 \in Q$ initial state
- $z_0 \in \Gamma$ start symbol

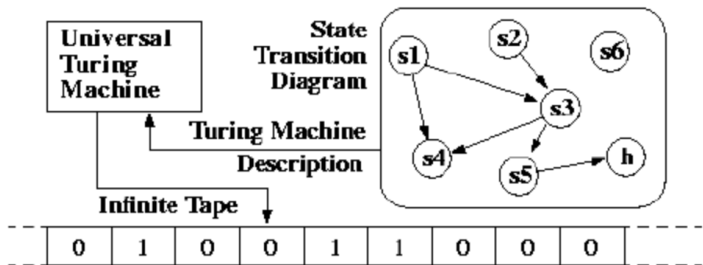
- it is a FSA $(Q, F, \mathfrak{A}, \tau, q_0)$ together with a stack Γ^*
- the transitions are determined by the first symbol in the stack, the current state, and a letter in $\mathfrak{A} \cup \{\epsilon\}$
- the transition adds a new (finite) sequence of symbols at the beginning of the stack Γ^*
- a **configuration** of M is an element of $Q \times \mathfrak{A}^* \times \Gamma^*$
- given $(q, a, z, q', \alpha) \in \tau \subset Q \times (\mathfrak{A} \cup \{\epsilon\}) \times \Gamma \times Q \times \Gamma^*$ the corresponding transition is from a configuration $(q, aw, z\beta)$ to a configuration $(q', w, \alpha\beta)$
- computation in M : a chain of transitions $c \rightarrow c'$ between configurations $c = c_1, \dots, c_n = c'$ where each $c_i \rightarrow c_{i+1}$ a transition as above
- computation stops when reach final state or empty stack

Example



a transition labelled (a, b, c) between vertex q_i and q_j means read letter a on string, read letter b on top of memory stack, remove b and place c at the top of the stack: move from configuration $(q_i, aw, b\alpha)$ to configuration $(q_j, w, c\alpha)$

Turing Machine



again a FSA with an additional memory mechanism (as in the case of pushdown stack), this time more flexible: here the FSA changes depending on the computation, but all can be represented inside a **universal** Turing machine

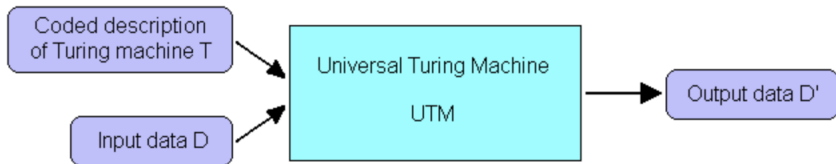
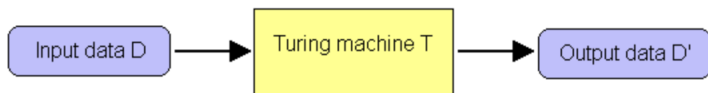
Turing machine $T = (Q, F, \mathcal{A}, I, \tau, q_0)$

- Q finite set of possible states
- F subset of Q : the final states
- \mathcal{A} finite set: alphabet (with a distinguished element B *blank symbol*)
- $I \subset \mathcal{A} \setminus \{B\}$ input alphabet
- $\tau \subset Q \times \mathcal{A} \times Q \times \mathcal{A} \times \{L, R\}$ transitions with $\{L, R\}$ a 2-element set
- $q_0 \in Q$ initial state

$qaq'a'L \in \tau$ means T is in state q , reads a on next square in the tape, changes to state q' , overwrites the square with new letter a' and moves one square to the left

- *tape description* for T : triple (a, α, β) with $a \in \mathfrak{A}$, $\alpha : \mathbb{N} \rightarrow \mathfrak{A}$, $\beta : \mathbb{N} \rightarrow \mathfrak{A}$ such that $\alpha(n) = B$ and $\beta(n) = B$ for all but finitely many $n \in \mathbb{N}$ (sequences of letters on tape right and left of a)
- *configuration* of T : (q, a, α, β) with $q \in Q$ and (a, α, β) a tape description
- configuration c' from c in a single move if either
 - $c = (q, a, \alpha, \beta)$, $qaq'a'L \in \tau$ and $c' = (q', \beta(0), \alpha', \beta')$ with $\alpha'(0) = a'$ and $\alpha'(n) = \alpha(n-1)$, and $\beta'(n) = \beta(n+1)$
 - $c = (q, a, \alpha, \beta)$, $qaq'a'R \in \tau$ and $c' = (q', \alpha(0), \alpha', \beta')$ with $\alpha'(n) = \alpha(n+1)$, and $\beta'(0) = a'$, $\beta'(n) = \beta(n-1)$
- *computation* $c \rightarrow c'$ in T starting at c and ending at c' : finite sequence $c = c_1, \dots, c_n = c'$ with c_{i+1} from c_i by a single move
- computation *halts* if c' *terminal configuration*, $c' = (q, a, \alpha, \beta)$ with no element in τ starting with qa

Universal Turing Machine

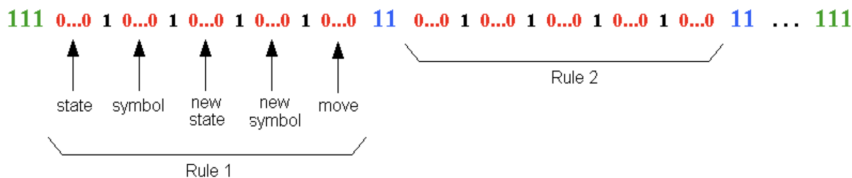


UTM simulates the behavior of a given Turing machine on input data by inputting also a coded description of the machine transition rules

universal: U for any Turing machine M there is a computable encoding $E(M)$ such that for all x input $U(E(x), x) = M(x)$

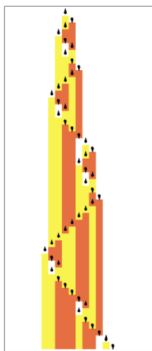
to code a Turing machine in a Universal Turing Machine, need to code the instructions (rules) $\tau \subset Q \times \mathcal{A} \times Q \times \mathcal{A} \times \{L, R\}$

Example:



there are very *small* universal Turing machines (e.g. Wolfram's 2-state 3-symbol Turing machine)

Example: a 2-state 3-symbol Universal Turing Machine



- transitions (states A , B and symbols $0, 1, 2$)

	A	B
0	P1,R,B	P2,L,A
1	P2,L,A	P2,R,B
2	P1,L,A	P0,R,A



- proof of universality by Alex Smith, 2007

Church Thesis (or Church-Turing thesis)

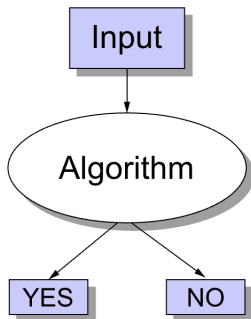
equivalence of different notions of computability

- general recursive functions
- functions computable by a Turing machine (that on input $x \in \mathcal{D}(f)$ outputs $f(x)$ and on input $x \notin \mathcal{D}(f)$ does not halt)
 - at input $x \in \mathcal{D}(f)$ (coded as a sequence of tape instructions) the Turing Machine halts with value $f(x)$ stored in memory (on tape)
 - at input $x \notin \mathcal{D}(f)$ the Turing Machine never halts
- λ -definable functions (computable in Church's λ -calculus)

how about **non-computable functions**?

Halting Problem

- **decision problem** (Entscheidungsproblem) of determining, knowing a computer program and an input, whether the program will halt on that input or continue to run forever



- computer program modeled as computation in a Turing machine

key idea of proof:

- suppose there is a total recursive function $\text{halts}(P)$ that returns true if program P halts (on a given input) and returns false otherwise
- then can construct another function g

```
def g() -> None:
    if halts(g):
        loop_forever()
```

- if $\text{halts}(g) = 1$ then this loops forever (giving $\text{halts}(g) = 0$) while if $\text{halts}(g) = 0$ (not halting) in fact halts, so contradiction
- **note** how this has the same flavor as the Epimenides paradox (e.g. “this sentence is a lie”)

more rigorous argument

- fix an enumeration P_n of all programs on a fixed Turing machine
- define $h(n, x) = 1$ if program P_n halts on input x and $h(n, x) = 0$ otherwise
- consider an *arbitrary* total computable $f(n, m)$ and partial function

$$g(n) = \begin{cases} 0 & f(n, n) = 0 \\ \text{undefined} & \text{otherwise} \end{cases}$$

- there is a program $P(g)$ that computes g (some P_m in list)
- if $f(n, n) = 0$ then the program computing g halts
- if $f(n, n) = 1$ then the program computing g does not halt
- so $f(m, m) \neq h(m, m)$ for *all* total computable f
- **note** in this form it becomes a **diagonal argument**

Kolmogorov complexity famous non-computable function

- Let T_U be a **universal Turing machine** (a Turing machine that can simulate any other arbitrary Turing machine: reads on tape both the input and the description of the Turing machine it should simulate)
- Given a string w in an alphabet \mathfrak{A} , the **Kolmogorov complexity**

$$\mathcal{K}_{T_U}(w) = \min_{P: T_U(P)=w} \ell(P),$$

minimal length of a program that outputs w

- **universality**: given any other Turing machine T

$$\mathcal{K}_T(w) = \mathcal{K}_{T_U}(w) + c_T$$

shift by a bounded constant, independent of w ; c_T is the Kolmogorov complexity of the program needed to describe T for T_U to simulate it

- conditional Kolmogorov complexity

$$\mathcal{K}_{T_U}(w | \ell(w)) = \min_{P: T_U(P, \ell(w))=w} \ell(P),$$

where the length $\ell(w)$ is given and made available to machine T_U

$$\mathcal{K}(w | \ell(w)) \leq \ell(w) + c,$$

because if know $\ell(w)$ then a possible program is just to write out w : then $\ell(w) + c$ is just number of bits needed for transmission of w plus print instructions

- upper bound

$$\mathcal{K}_{T_U}(w) \leq \mathcal{K}_{T_U}(w | \ell(w)) + 2 \log \ell(w) + c$$

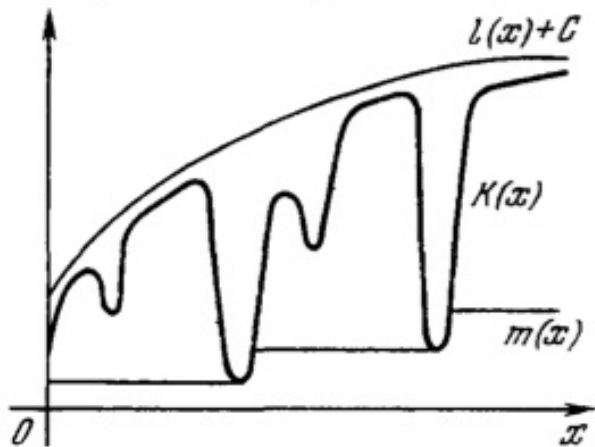
if don't know a priori $\ell(w)$ need to signal end of description of w (can show for this suffices a "punctuation method" that adds a program of length $\leq 2 \log \ell(w)$)

- any **program** that produces a description of w is an **upper bound** on Kolmogorov complexity $\mathcal{K}_{T_U}(w)$
- think of Kolmogorov complexity in terms of **data compression**
- shortest description of w is also its **most compressed form**
- can obtain **upper bounds** on Kolmogorov complexity using **data compression algorithms**
- finding upper bounds is easy... but **NOT lower bounds**

Kolmogorov complexity is a non-computable function

- suppose list programs P_k (increasing lengths) and run through T_U : if machine halts on P_k with output w then $\ell(P_k)$ is an upper bound on $\mathcal{K}_{T_U}(w)$
- but... there can be an earlier P_j in the list such that T_U has not yet halted on P_j
- if eventually halts and outputs w then $\ell(P_j)$ is a better approximation to $\mathcal{K}_{T_U}(w)$
- would be able to compute $\mathcal{K}_{T_U}(w)$ if can tell exactly on which programs P_k the machine T_U halts
- but... **halting problem is unsolvable** (halting function is non-computable)

Observations: various non-computability results rely on the halting problem; Kolmogorov complexity is a “mild” non-computability case because it has good computable upper bound approximations



with $m(x) = \min_{y \geq x} K(y)$

Computable functions and Representability

- **numerical language:** \mathcal{L} first-order language containing symbol 0, unary function S
- **key example:** $\mathcal{L}_{ar} = \{0, S, +, \cdot, <\}$
- $\underline{\mathbb{N}}$: set of 9 Peano axioms:

$$\underline{\mathbf{N1}} \quad \forall x (Sx \neq 0)$$

$$\underline{\mathbf{N2}} \quad \forall x \forall y (Sx = Sy \rightarrow x = y)$$

$$\underline{\mathbf{N3}} \quad \forall x (x + 0 = x)$$

$$\underline{\mathbf{N4}} \quad \forall x \forall y (x + Sy = S(x + y))$$

$$\underline{\mathbf{N5}} \quad \forall x (x \cdot 0 = 0)$$

$$\underline{\mathbf{N6}} \quad \forall x \forall y (x \cdot Sy = x \cdot y + x)$$

$$\underline{\mathbf{N7}} \quad \forall x (x \neq 0)$$

$$\underline{\mathbf{N8}} \quad \forall x \forall y (x < Sy \leftrightarrow x < y \vee x = y)$$

$$\underline{\mathbf{N9}} \quad \forall x \forall y (x < y \vee x = y \vee y < x)$$

- these are all true in $\mathcal{N} = \langle \mathbb{N}, 0, S, +, \cdot, < \rangle$ standard structure
- of \mathcal{X} is a structure with $\mathcal{X} \models \underline{\mathbb{N}}$ then there is a unique homomorphism $\iota : \mathcal{N} \rightarrow \mathcal{X}$, which is an *embedding* with (forall $a \in X$ and $n \in \mathbb{N}$)
 - 1 if $a <_{\mathcal{X}} \iota(n)$ then $a = \iota(m)$ for some $m < n$
 - 2 if $a \notin \iota(\mathbb{N})$ then $\iota(n) <_{\mathcal{X}} a$

Representable relations

Definition. Let L be a numerical language, and Σ a set of L -sentences. A relation $R \subseteq \mathbf{N}^m$ is said to be Σ -representable, if there is an L -formula $\varphi(x_1, \dots, x_m)$ such that for all $(a_1, \dots, a_m) \in \mathbf{N}^m$ we have

$$(i) \quad R(a_1, \dots, a_m) \implies \Sigma \vdash \varphi(S^{a_1}0, \dots, S^{a_m}0)$$

$$(ii) \quad \neg R(a_1, \dots, a_m) \implies \Sigma \vdash \neg \varphi(S^{a_1}0, \dots, S^{a_m}0)$$

Such a $\varphi(x_1, \dots, x_m)$ is said to *represent* R in Σ or to Σ -represent R . Note that if $\varphi(x_1, \dots, x_m)$ Σ -represents R and Σ is consistent, then for all $(a_1, \dots, a_m) \in \mathbf{N}^m$

$$\begin{aligned} R(a_1, \dots, a_m) &\iff \Sigma \vdash \varphi(S^{a_1}0, \dots, S^{a_m}0), \\ \neg R(a_1, \dots, a_m) &\iff \Sigma \vdash \neg \varphi(S^{a_1}0, \dots, S^{a_m}0). \end{aligned}$$

A function $F : \mathbf{N}^m \rightarrow \mathbf{N}$ is Σ -representable if there is a formula $\varphi(x_1, \dots, x_m, y)$ of L such that for all $(a_1, \dots, a_m) \in \mathbf{N}^m$ we have

$$\Sigma \vdash \varphi(S^{a_1}0, \dots, S^{a_m}0, y) \leftrightarrow y = S^{F(a_1, \dots, a_m)}0.$$

relation R representable iff characteristic function χ_R representable

Representability theorem for computable functions

- every computable function $F : \mathbb{N}^n \rightarrow \mathbb{N}$ is $\underline{\mathbb{N}}$ -representable
- every computable relation $R \subseteq \mathbb{N}^m$ is $\underline{\mathbb{N}}$ -representable

steps:

- 1 functions $+$: $\mathbb{N}^2 \rightarrow \mathbb{N}$, \cdot : $\mathbb{N}^2 \rightarrow \mathbb{N}$, χ_{\leq} : $\mathbb{N}^2 \rightarrow \mathbb{N}$ are $\underline{\mathbb{N}}$ -representable
- 2 coordinate functions I_i^n are $\underline{\mathbb{N}}$ -representable
- 3 compositions: if $G : \mathbb{N}^m \rightarrow \mathbb{N}$ and $H_1, \dots, H_m : \mathbb{N}^n \rightarrow \mathbb{N}$ are $\underline{\mathbb{N}}$ -representable then $\underline{\mathbb{N}}$ -representable composition

$$F(a) = G(H_1(a), \dots, H_m(a))$$

- 4 if $G : \mathbb{N}^{n+1} \rightarrow \mathbb{N}$ is $\underline{\mathbb{N}}$ -representable and for all $a \in \mathbb{N}^n \exists x \in \mathbb{N}$ with $G(a, x) = 0$ then $\underline{\mathbb{N}}$ -representable minimization

$$F(a) = \mu x (G(a, x) = 0)$$

- formula $x_1 = x_2$ represents $\{(a, b) \in \mathbb{N}^2 : a = b\}$ in \underline{N}
 - if $a = b$ then $\underline{N} \vdash (S^a(0) = S^b(0))$ and if $a \neq b$ then for every \mathcal{X} have $\mathcal{X} \models (S^a(0) \neq S^b(0))$ (by $\iota : \mathcal{N} \rightarrow \mathcal{X}$) and so $\underline{N} \vdash (S^a(0) \neq S^b(0))$
- $x_1 + x_2$ represents $+ : \mathbb{N}^2 \rightarrow \mathbb{N}$
 - for $a, b, c \in \mathbb{N}$ with $a + b = c$ have $\mathcal{X} \models (S^a(0) + S^b(0) = S^c(0))$ in every model \mathcal{X} (by $\iota : \mathcal{N} \rightarrow \mathcal{X}$) so $\underline{N} \vdash (S^a(0) + S^b(0) = S^c(0))$
- $x_1 \cdot x_2$ analogous
- $x_1 < x_2$ represents $\{(a, b) \in \mathbb{N}^2 : a < b\}$ as in $x_1 = x_2$ case
- from these $(x_1 < x_2) \vee (x_1 = x_2)$ represents $\{(a, b) \in \mathbb{N}^2 : a \leq b\}$ so χ_{\leq} representable
- term $t_i^n(x_1, \dots, x_n) := x_i$ represents coordinate function I_i^n

- **composition:** $F = G(H_1, \dots, H_m)$ is represented by

$$\theta(x_1, \dots, x_n, z) := \exists y_1 \dots \exists y_m \left(\left(\bigwedge_{i=1}^m \varphi_i(x_1, \dots, x_n, y_i) \right) \wedge \psi(y_1, \dots, y_m, z) \right)$$

where $\phi_i(x_1, \dots, x_n, y_i)$ represents H_i and $\psi(y_1, \dots, y_m, z)$ represents G

Put $a = (a_1, \dots, a_n)$ and let $c = F(a)$. We have to show that

$$\underline{N} \vdash \theta(S^a 0, z) \leftrightarrow z = S^c 0$$

where $S^a 0 := (S^{a_1} 0, \dots, S^{a_n} 0)$. Let $b_i = H_i(a)$ and put $b = (b_1, \dots, b_m)$. Then $F(a) = G(b) = c$. Therefore, $\underline{N} \vdash \psi(S^b 0, z) \leftrightarrow z = S^c 0$ and

$$\underline{N} \vdash \varphi_i(S^a 0, y_i) \leftrightarrow y_i = S^{b_i} 0, \quad (i = 1, \dots, m)$$

Argue in models to conclude : $\underline{N} \vdash \theta(S^a 0, z) \leftrightarrow z = S^c 0$.

- same type of argument for **minimization**:

$F(a) = \mu x(G(a, x) = 0)$ represented by

$$\psi(x_1, \dots, x_n, y) :=$$

$$\varphi(x_1, \dots, x_n, y, 0) \wedge \forall w((w < y) \rightarrow \neg \varphi(x_1, \dots, x_n, w, 0))$$

where G is represented by $\varphi(x_1, \dots, x_n, y, z)$

Converse also true: if \mathcal{L} numerical and Σ computable consistent set of \mathcal{L} -sentences then every Σ -representable $R \subseteq \mathbb{N}^n$ is computable

so in particular $f : \mathbb{N}^n \rightarrow \mathbb{N}$ computable iff \mathbb{N} -representable

this converse statement requires **Gödel numbering**

Gödel numbering

- first-order language \mathcal{L} with sets Φ, \mathcal{R} *finite*
- **symbols**: variables $v_0, v_1, \dots, v_n \dots$; logical symbols; functions and relations $f \in \Phi$ and $R \in \mathcal{R}$
- assign a **symbol number** in \mathbb{N} to each symbol (injectively)
- **variables**: $SN(v_i) = 2i$ even numbers
- remaining *finite* set of logical symbols, functions, and relations mapped injectively to a subset of odd numbers
- then assign a numbering to **terms**, inductively by $t = F(t_1, \dots, t_k)$

numbering of terms and formulae

$$\ulcorner t \urcorner = \begin{cases} \langle \text{SN}(\mathbf{v}_i) \rangle & \text{if } t = \mathbf{v}_i, \\ \langle \text{SN}(F), \ulcorner t_1 \urcorner, \dots, \ulcorner t_n \urcorner \rangle & \text{if } t = Ft_1 \dots t_n. \end{cases}$$

The Gödel number $\ulcorner \varphi \urcorner$ of an L -formula φ is given recursively by

$$\ulcorner \varphi \urcorner = \begin{cases} \langle \text{SN}(\top) \rangle & \text{if } \varphi = \top, \\ \langle \text{SN}(\perp) \rangle & \text{if } \varphi = \perp, \\ \langle \text{SN}(=), \ulcorner t_1 \urcorner, \ulcorner t_2 \urcorner \rangle & \text{if } \varphi = (t_1 = t_2), \\ \langle \text{SN}(R), \ulcorner t_1 \urcorner, \dots, \ulcorner t_m \urcorner \rangle & \text{if } \varphi = Rt_1 \dots t_m, \\ \langle \text{SN}(\neg), \ulcorner \psi \urcorner \rangle & \text{if } \varphi = \neg\psi, \\ \langle \text{SN}(\vee), \ulcorner \varphi_1 \urcorner, \ulcorner \varphi_2 \urcorner \rangle & \text{if } \varphi = \varphi_1 \vee \varphi_2, \\ \langle \text{SN}(\wedge), \ulcorner \varphi_1 \urcorner, \ulcorner \varphi_2 \urcorner \rangle & \text{if } \varphi = \varphi_1 \wedge \varphi_2, \\ \langle \text{SN}(\exists), \ulcorner x \urcorner, \ulcorner \psi \urcorner \rangle & \text{if } \varphi = \exists x \psi, \\ \langle \text{SN}(\forall), \ulcorner x \urcorner, \ulcorner \psi \urcorner \rangle & \text{if } \varphi = \forall x \psi. \end{cases}$$

with $\langle a_1, \dots, a_n \rangle$ the sequence number for $a_i \in \mathbb{N}$ (injective so terms and formulae can be identified uniquely by their Gödel numbering)

variable substitution and Gödel numbering

- x a variable, φ, ψ formulae and t, τ terms in numerical language \mathcal{L}
- function $\text{Sub} : \mathbb{N}^3 \rightarrow \mathbb{N}$ with $\text{Sub}(a, b, c)$ given by (with $\text{Vble} =$ set of Gödel numbers of variables)

$$\begin{cases} c & \text{if } \text{Vble}(a) \text{ and } a = b, \\ \langle (a)_0, \text{Sub}((a)_1, b, c), \dots, \text{Sub}((a)_n, b, c) \rangle & \text{if } a = \langle (a)_0, \dots, (a)_n \rangle \text{ with } n > 0 \text{ and} \\ & (a)_0 \neq \text{SN}(\exists), (a)_0 \neq \text{SN}(\forall), \\ \langle \text{SN}(\exists), (a)_1, \text{Sub}((a)_2, b, c) \rangle & \text{if } a = \langle \text{SN}(\exists), (a)_1, (a)_2 \rangle \text{ and } (a)_1 \neq b, \\ \langle \text{SN}(\forall), (a)_1, \text{Sub}((a)_2, b, c) \rangle & \text{if } a = \langle \text{SN}(\forall), (a)_1, (a)_2 \rangle \text{ and } (a)_1 \neq b, \\ a & \text{otherwise} \end{cases}$$

- the function $\text{Sub} : \mathbb{N}^3 \rightarrow \mathbb{N}$ is computable and satisfies

$$\text{Sub}(\ulcorner t \urcorner, \ulcorner x \urcorner, \ulcorner \tau \urcorner) = \ulcorner t(x/\tau) \urcorner$$

$$\text{Sub}(\ulcorner \varphi \urcorner, \ulcorner x \urcorner, \ulcorner \tau \urcorner) = \ulcorner \varphi(x/\tau) \urcorner$$

some computable sets of Gödel numbers

- computable sets

- (1) $Vble := \{\ulcorner x \urcorner : x \text{ is a variable}\}$
- (2) $Term := \{\ulcorner t \urcorner : t \text{ is an } L\text{-term}\}$
- (3) $AFor := \{\ulcorner \varphi \urcorner : \varphi \text{ is an atomic } L\text{-formula}\}$
- (4) $For := \{\ulcorner \varphi \urcorner : \varphi \text{ is an } L\text{-formula}\}$

- also computable sets (with computable Sub)

- (1) $Fr := \{(\ulcorner \varphi \urcorner, \ulcorner x \urcorner) : x \text{ occurs free in } \varphi\} \subseteq \mathbb{N}^2$
- (2) $FrSub := \{(\ulcorner \varphi \urcorner, \ulcorner x \urcorner, \ulcorner \tau \urcorner) : \tau \text{ is free for } x \text{ in } \varphi\} \subseteq \mathbb{N}^3$
- (3) $PrAx := \{\ulcorner \varphi \urcorner : \varphi \text{ is a propositional axiom}\} \subseteq \mathbb{N}$
- (4) $Eq := \{\ulcorner \varphi \urcorner : \varphi \text{ is an equality axiom}\} \subseteq \mathbb{N}$
- (5) $Quant := \{\ulcorner \psi \urcorner : \psi \text{ is a quantifier axiom}\} \subseteq \mathbb{N}$
- (6) $MP := \{(\ulcorner \varphi_1 \urcorner, \ulcorner \varphi_1 \rightarrow \varphi_2 \urcorner, \ulcorner \varphi_2 \urcorner) : \varphi_1, \varphi_2 \text{ are } L\text{-formulas}\} \subseteq \mathbb{N}^3$
- (7) $Gen := \{(\ulcorner \varphi \urcorner, \ulcorner \psi \urcorner) : \psi \text{ follows from } \varphi \text{ by the generalization rule}\} \subseteq \mathbb{N}^2$
- (8) $Sent := \{\ulcorner \varphi \urcorner : \varphi \text{ is a sentence}\} \subseteq \mathbb{N}$

- also computable Prf_{Σ} set of Gödel numbers of proofs from Σ , if Σ computable (i.e. $\ulcorner \Sigma \urcorner \subset \mathbb{N}$ computable)

Representability implies Computability

- computable function $\text{Num} : \mathbb{N} \rightarrow \mathbb{N}$ with

$$\text{Num}(a) = \ulcorner S^a(0) \urcorner$$

recursion $\text{Num}(0) = \ulcorner 0 \urcorner$ and

$\text{Num}(a+1) = \langle \text{SN}(S), \text{Num}(a) \rangle$ shows computability

- for any formula φ in \mathcal{L} : computable function

$$a \mapsto \ulcorner \varphi(S^a(0)) \urcorner = \text{Sub}(\ulcorner \varphi \urcorner, \ulcorner x \urcorner, \text{Num}(a))$$

- Σ consistent computable set of \mathcal{L} sentences
- $R \subseteq \mathbb{N}^n$ is Σ -represented by a formula $\varphi(x_1, \dots, x_n)$

$$R(a_1, \dots, a_n) \Leftrightarrow \Sigma \vdash \varphi(S^{a_1}(0), \dots, S^{a_n}(0))$$

- define $s_n : \mathbb{N}^n \rightarrow \mathbb{N}$ by

$$s(a_1, \dots, a_n) := \ulcorner \varphi(S^{a_1}(0), \dots, S^{a_n}(0)) \urcorner$$

- **computable** because $s_1(a_1) = \text{Sub}(\ulcorner \varphi \urcorner, \ulcorner x_1 \urcorner, \text{Num}(a_1))$ and

$$s_{i+1}(a_1, \dots, a_i, a_{i+1}) = \text{Sub}(s_i(a_1, \dots, a_i), \ulcorner x_{i+1} \urcorner, \text{Num}(a_{i+1}))$$

with

$$s_i(a_1, \dots, a_i) := \ulcorner \varphi(S^{a_1}(0), \dots, S^{a_i}(0), x_{i+1}, \dots, x_n) \urcorner$$

recursively enumerable (computably generated)

- relation (set) $R \subseteq \mathbb{N}^n$ **computably generated** (recursively enumerable) if there is a computable $Q \subseteq \mathbb{N}^{n+1}$ with

$$R(a) \Leftrightarrow \exists x Q(a, x)$$

- theory T for numerical language \mathcal{L} is **computably axiomatizable** if $T = T(\Sigma)$ with Σ computable
- Σ computable means $\ulcorner \Sigma \urcorner \subseteq \mathbb{N}$ computable

if Σ computable then $T(\Sigma)$ computably generated:

- theory $T(\Sigma)$ and its image $\ulcorner T(\Sigma) \urcorner \subseteq \mathbb{N}$ under Gödel numbering
- $a \in \ulcorner T(\Sigma) \urcorner$ iff $\text{Sent}(a)$ and $\exists b(\text{Prf}_\Sigma(b))$ with $a = \beta(b, \ell(b))$
- $\Sigma \vdash \varphi(S^{a_1}(0), \dots, S^{a_n}(0))$ gives $s(a_1, \dots, a_n) \in \ulcorner T(\Sigma) \urcorner$

decidable/undecidable

- theory T for numerical language \mathcal{L} is **decidable** if $\ulcorner T \urcorner \subseteq \mathbb{N}$ **computable**, undecidable otherwise

negation argument:

- for $R \subseteq \mathbb{N}^n$ both R and $\neg R$ are *recursively enumerable* then R is **computable**, because

$$R(a) \Leftrightarrow \exists x P(a, x) \quad \text{and} \quad \neg R(a) \Leftrightarrow \exists x Q(a, x)$$

for some $P, Q \subseteq \mathbb{N}^{n+1}$ (recursive enumerability) and for each $a \in \mathbb{N}^n$ there is $x \in \mathbb{N}$ with $(P \vee Q)(a, x)$ so

$$R(a) \Leftrightarrow P(a, \mu x (P \vee Q)(a, x)) \quad \text{computable}$$

complete computably axiomatizable theory T of a numerical language \mathcal{L} is *decidable*:

- $T = T(\Sigma)$ computably generated
- $a \notin \ulcorner T \urcorner$ means $a \notin \text{Sent}$ or $\langle \text{SN}(\neg), a \rangle \in \ulcorner T \urcorner$, i.e. $a \notin \text{Sent}$ or

$$\exists b (\text{Prf}_{\Sigma}(b) \text{ and } \beta(b, \ell(b)) = \langle \text{SN}(\neg), a \rangle)$$

so complement of $\ulcorner T \urcorner$ also computably generated

- so T **decidable** by negation argument

- we have obtained $s(a_1, \dots, a_n) \in \ulcorner T(\Sigma) \urcorner$ where Σ computable hence $T(\Sigma)$ computably generated, so there is some $R' \subseteq \mathbb{N}^{n+1}$

$$u \in \ulcorner T(\Sigma) \urcorner \Leftrightarrow \exists y R'(u, y)$$

- so get for $a = (a_1, \dots, a_n)$

$$R(a) \Leftrightarrow s(a) \in \ulcorner T(\Sigma) \urcorner \Leftrightarrow \exists y R'(s(a), y)$$

- this means $R \subseteq \mathbb{N}$ is *computably generated*
- but hypothesis that R is Σ -representable means that also $\neg R$ is Σ -representable so same argument also gives $\neg R$ is *computably generated*
- by negation argument then R is **computable**
- for Σ -representable functions $f : \mathbb{N}^n \rightarrow \mathbb{N}$ same argument applied to graph $\Gamma(f)$ so computable

Conclusion: $f : \mathbb{N}^n \rightarrow \mathbb{N}$ computable iff \mathbb{N} -representable

Church undecidability theorem:

- for any (finite) language \mathcal{L} extending $\mathcal{L}(\underline{\mathbb{N}})$ any consistent theory extending $T(\underline{\mathbb{N}})$ is **undecidable**

main steps of the argument:

- 1 produce an **enumeration** of computable sets $X \subseteq \mathbb{N}$
- 2 suppose T extending $T(\underline{\mathbb{N}})$ is decidable (i.e. $\ulcorner T \urcorner$ computable)
- 3 **Cantor diagonal argument**: produce an “antidiagonal” computable set that cannot be in the list enumerating all computable sets

Gödel Incompleteness: consequence of Church undecidability

- as shown, every complete and computably axiomatizable theory is decidable, so any computably axiomatizable theory extending $T(\underline{\mathbb{N}})$ **must be incomplete**

enumeration of computable sets

- Σ set of \mathcal{L} -sentences, x variable
- define binary relation $P^\Sigma \subseteq \mathbb{N}^2$

$$P^\Sigma(a, b) \Leftrightarrow \text{Sub}(a, \ulcorner x \urcorner, \text{Num}(b)) \in \ulcorner T(\Sigma) \urcorner$$

- equivalently for formula $\varphi(x)$ and $a = \ulcorner \varphi(x) \urcorner$

$$P^\Sigma(a, b) \Leftrightarrow \Sigma \vdash \varphi(S^b(0))$$

since we have

$$\text{Sub}(\ulcorner \varphi(x) \urcorner, \ulcorner x \urcorner, \ulcorner S^b(0) \urcorner) = \ulcorner \varphi(S^b(0)) \urcorner$$

and $\ulcorner \varphi(S^b(0)) \urcorner \in \ulcorner T(\Sigma) \urcorner$ so $\varphi(S^b(0)) \in T(\Sigma)$

- define $P^\Sigma(a) \subseteq \mathbb{N}$ by $b \in P^\Sigma(a)$ iff $P^\Sigma(a, b)$
- **Claim:** every computable $X \subseteq \mathbb{N}$ is of the form $P^\Sigma(a)$ for some $a \in \mathbb{N}$

computable $X = P^\Sigma(a)$ for some $a \in \mathbb{N}$:

- computable $X \subseteq \mathbb{N}$ is Σ -representable by some formula $\varphi(x)$
(so $X(b) \Rightarrow \Sigma \vdash \varphi(S^b(0))$)
- by completeness also $X(b) \Leftrightarrow \Sigma \vdash \varphi(S^b(0))$
- for $a = \ulcorner \varphi(x) \urcorner$ then have

$$X(b) \Leftrightarrow \Sigma \vdash \varphi(S^b(0)) \Leftrightarrow P^\Sigma(a, b)$$

this means $X = P^\Sigma(a)$

- so **enumeration** $\{P^\Sigma(a)\}_{a \in \mathbb{N}}$ of computable sets $X \subseteq \mathbb{N}$

Cantor (anti)diagonal argument

- set X and relation $P \subseteq X \times X$: antidiagonal set $Q \subset X$

$$Q(b) \Leftrightarrow \neg P(b, b)$$

- the unary relation Q **cannot be** of the form $P(a)$ for any $a \in X$
- because if $Q = P(a)$ then $Q(a) \Leftrightarrow P(a, a)$ (by definition of $P(a)$) but $Q(a) \Leftrightarrow \neg P(a, a)$ (by definition of $Q(a)$) so *contradiction*

Conclusion: Church undecidability:

- suppose that $\ulcorner T \urcorner$ computable
- then antidiagonal $Q^\Sigma(b) = \neg P^\Sigma(b, b)$ is **computable**:

$$Q^\Sigma(b) \Leftrightarrow \neg P^\Sigma(b, b) \Leftrightarrow \text{Sub}(b, \ulcorner x \urcorner, \text{Num}(b)) \notin \ulcorner T(\Sigma) \urcorner$$

(intersection of computable sets P^Σ , diagonal, complement of $\ulcorner T(\Sigma) \urcorner$)

- but by (anti)diagonal argument Q^Σ cannot be of the form $P^\Sigma(a)$ for any $a \in \mathbb{N}$ so not in the enumeration of all computable sets: **cannot be computable**
- contradiction so $\ulcorner T \urcorner$ non-computable **so T undecidable**

Gödel First/Second Incompleteness theorems

1 first incompleteness theorem

- for every formula $\phi(x)$ in $L(\underline{\mathbb{N}})$ there is a sentence ψ with

$$\Sigma \vdash \psi \Leftrightarrow \phi(S^{\ulcorner \psi \urcorner} 0)$$

- applied to formula $\neg \exists y \text{Prf}(y, x)$ with $\text{Prf}(y, x)$ meaning x is the Gödel numbering of a sentence ϕ and y is the Gödel numbering of a proof of ϕ
- this gives Gödel sentence G with property that

$$\Sigma \vdash G \Leftrightarrow \neg \exists y \text{Prf}(y, \ulcorner G \urcorner)$$

- so G is true iff there is no proof for G
- so there cannot be any proof for G (otherwise G would be false), so G is true but not provable in $T(\underline{\mathbb{N}})$

2 second incompleteness theorem

- the Gödel sentence is equivalent in $T(\underline{\mathbb{N}})$ to the sentence

$$\neg \exists y \text{Prf}(y, \ulcorner \perp \urcorner)$$

- this sentence expresses the property that the theory $T(\underline{\mathbb{N}})$ is formally consistent
- so result of theorem is that $T(\underline{\mathbb{N}})$ cannot prove its own consistency

Credits

Slides incorporate material taken from:

- Alexander S. Kechris, Michael A. Shulman, Andrés E. Caicedo, *Math/CS 6c Notes*, 2020
- Lou van den Dries, *Mathematical Logic Lecture Notes*, 2010