CSIE 5111: Introduction to Mathematical Logic

Tony Tan
Department of Computer Science and Information Engineering
College of Electrical Engineering and Computer Science
National Taiwan University

Table of contents

Lesson 0: Preliminaries

Theme: Review of some essential mathematical backgrounds.

1 Useful notations and facts from discrete mathematics

1.1 Equivalence relations

A binary relation R over X is called an *equivalence relation*, if it satisfies the following conditions.

- Reflexive: $(x,x) \in R$, for every $x \in X$.
- Symmetric: $(x,y) \in R$ if and only if (y,x), for every $x,y \in X$.
- Transitive: for every $x,y,z \in X$, if $(x,y) \in R$ and $(y,z) \in R$, then $(x,z) \in R$.

For $x \in X$, the *equivalence class* of x in R is defined as:

$$[x]_R \quad := \quad \{y \mid (x,y) \in R\}$$

Lemma 0.1 *Let R be an equivalence relation over X. Then, the following holds:*

- $[x]_R = [y]_R$ *if and only if* $(x,y) \in R$.
- *If* $[x]_R \neq [y]_R$, *then* $[x]_R \cap [y]_R = \emptyset$.

Theorem 0.2 *Let R be an equivalence relation over X. Then, the equivalence classes of R* partition *X, i.e., every member of X belongs to exactly one equivalence class of R.*

1.2 Countable and uncountable sets

Let $\mathbb{N}$ be the set of natural numbers $\{0,1,2,\ldots\}$. A set X is *countable*, if there is an injective function from X to $\mathbb{N}$. Otherwise, it is called an *uncountable* set.

Theorem 0.3 *The following sets are all countable.*

(1) The set $\mathbb{Z} = \{\ldots,-2,-1,0,1,2,\ldots\}$ *of integers.*

(2) The set $\mathbb{N}^k$*, for every integer* $k \geqslant 1$.

(3) The set $\mathbb{N}^* := \bigcup_{k\geqslant 1} \mathbb{N}^k$.

Theorem 0.4 *The set* $2^{\mathbb{N}}$ *is uncountable.*

1.3 Poset (partially ordered set)

Let X be a set and R be a binary relation on X. The set X is a poset (w.r.t. R), if R is reflexive, anti-symmetric* and transitive.

Definition 0.5 An element $m \in X$ is a *maximal* element in a poset X (w.r.t. R), if for every $x \in X$ and $x \neq m$, $(m,x) \notin R$.

*A binary relation R on X is *anti-symmetric*, if the following holds: for every $a,b \in X$, if both (a,b) and (b,a) are in R, then $a = b$.

Definition 0.6 A subset C of X is a *chain* in X (w.r.t. R), if for every $x, y \in C$, either $(x, y) \in R$, or $(y, x) \in R$. A chain C is *bounded*, if there is $z \in X$ such that for every $x \in C$, $(x, z) \in R$.

The three statements below are equivalent and they are usually taken as "axioms" in mathematics.

Axiom of choice: Let I be a set such that each $i \in I$ is associated with a set A_i. There is a function $f : I \to \bigcup A_i$ such that for every $i \in I$, $f(i) \in A_i$.

Zorn's lemma: Let (A, R) be a poset such that every chain in A is bounded. There is an element $m \in A$ such that for every $x \in A$ and $x \neq m$, $(m, x) \notin R$.

Well-ordering theorem: Every set can be *well-ordered*. That is, for every set A, there is a total order relation R on A, that is, it satisfies the following conditions:

- Antisymmetry: for every $a, b \in A$, if $(a, b), (b, a) \in R$, then $a = b$;
- Transitive: if $(a, b), (b, c) \in R$, then $(a, c) \in R$;
- Totality: for every $a, b \in A$, either $(a, b) \in R$ or $(b, a) \in R$,

such that for every nonempty subset $B \subseteq A$ has a minimal element (w.r.t. R).

There is a kind of contradiction here: the axiom of choice is viewed as obviously "correct," while the well-ordering theorem is obviously "false," and there are mixed opinions about Zorn's lemma.

2 Basic propositional calculus (Boolean logic)

Throughout this class, `T` and `F` are special symbols denoting *true* and *false*, respectively. The symbols $\neg$, $\wedge$, $\vee$, $\to$ and $\leftrightarrow$ denote the *negation*, *and*, *or*, *implication* and *iff* operators on $\{\mathtt{T}, \mathtt{F}\}$, respectively, which are defined as follows.

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

p	$\neg p$
T	F
F	T

p	q	$p \to q$
T	T	T
T	F	F
F	T	T
F	F	T

p	q	$p \leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

Let $PV = \{p_1, p_2, \ldots\}$ to be a *countable* set of *propositional variables*.† Sometimes we also write p, q, or $q_1, q_2, \ldots$ to denotes propositional variables. Elements in PV are also called *atomic formulas*.

Definition 0.7 A well formed formula (wff) is a formula built up inductively as follows.

- Every propositional variable $p \in PV$ is a wff.

†For simplicity, we only consider PV a countable set. Although in general such assumption is not necessary, it will simplify our discussions a lot.

- If α and β are wffs, so are $(\neg\alpha)$, $(\alpha \wedge \beta)$, $(\alpha \vee \beta)$, $(\alpha \to \beta)$ and $(\alpha \leftrightarrow \beta)$.

Usually we will use the term formula to mean wff.

The *negation* of a propositional variable p is $\neg p$. A *literal* is either a propositional variable or its negation. A formula is in *conjunctive normal form* (CNF), if it is of the form:

$$(\ell_{0,0} \vee \cdots \vee \ell_{0,n_0}) \quad \wedge \quad (\ell_{1,0} \vee \cdots \vee \ell_{1,n_1}) \quad \wedge \quad \cdots \quad \wedge \quad (\ell_{k,0} \vee \cdots \vee \ell_{k,n_k}),$$

where each $\ell_{i,j}$ is a literal.

A formula is in *disjunctive normal form* (DNF), if it is of the form:

$$(\ell_{0,0} \wedge \cdots \wedge \ell_{0,n_0}) \quad \vee \quad (\ell_{1,0} \wedge \cdots \wedge \ell_{1,n_1}) \quad \vee \quad \cdots \quad \vee \quad (\ell_{k,0} \wedge \cdots \wedge \ell_{k,n_k}).$$

An *assignment* is a function that maps each propositional variable in PV to either $\mathtt{T}$ or $\mathtt{F}$. The value of a formula α under an assignment w is defined inductively as follows.

- $w(\alpha) = w(p)$, if α is propositional variable p.
- $w(\neg\alpha) = \neg w(\alpha)$.
- $w(\alpha \wedge \beta) = w(\alpha) \wedge w(\beta)$.
- $w(\alpha \vee \beta) = w(\alpha) \vee w(\beta)$.
- $w(\alpha \to \beta) = w(\alpha) \to w(\beta)$.
- $w(\alpha \leftrightarrow \beta) = w(\alpha) \leftrightarrow w(\beta)$.

Definition 0.8

- An assignment w is a *satisfying assignment* for a formula α, denoted by $w \models \alpha$, if $w(\alpha) = \mathtt{T}$. We also say that w is a *model* of α.
- Likewise, w is a *satisfying assignment* (or, a *model*) for a set X of formulas, denoted by $w \models X$, if $w \models \alpha$, for every $\alpha \in X$.
- A formula α is *satisfiable*, if it has a satisfying assignment, and accordingly, a set X of formulas is *satisfiable*, if it has a satisfying assignment.
- Two formulas α and β are equivalent, if for every assignment w, $w(\alpha) = w(\beta)$.

Sometimes we omit the brackets, when they are irrelevant. For example, $\alpha \wedge (\beta \wedge \gamma)$ and $(\alpha \wedge \beta) \wedge \gamma$ are equivalent, so the brackets can be omitted, and written simply as $\alpha \wedge \beta \wedge \gamma$.

Theorem 0.9 (Distributivity law for $\wedge$ and $\vee$) *For every formulas α, β, γ, the following holds.*

- $\alpha \wedge (\beta \vee \gamma)$ *and* $(\alpha \wedge \beta) \vee (\alpha \wedge \gamma)$ *are equivalent.*
- $\alpha \vee (\beta \wedge \gamma)$ *and* $(\alpha \vee \beta) \wedge (\alpha \vee \gamma)$ *are equivalent.*

A formula α using only atomic formulas $p_1, \ldots, p_n$ defines a function $f_\alpha : \{\mathtt{T}, \mathtt{F}\}^n \to \{\mathtt{T}, \mathtt{F}\}$, where for every $(v_1, \ldots, v_n) \in \{\mathtt{T}, \mathtt{F}\}^n$

$$f_\alpha(v_1, \ldots, v_n) = v \quad \textit{if and only if} \quad \left\{ \begin{array}{l} \text{under the assignment } w \\ \text{where } w(p_i) = v_i, \text{ for each } i = 1, \ldots, n, \\ w(\alpha) = v. \end{array} \right.$$

Definition 0.10 A set Γ of operators is *complete*, if for every integer $n \geqslant 1$, for every function $g : \{\mathtt{T}, \mathtt{F}\}^n \to \{\mathtt{T}, \mathtt{F}\}$, there is a formula α using only operators from Γ such that $f_\alpha = g$.

Theorem 0.11

(a) For every function $g : \{\mathtt{T}, \mathtt{F}\}^n \to \{\mathtt{T}, \mathtt{F}\}$, there is a formula α in DNF such that $f_\alpha = g$.

(b) Similarly, for every function $g : \{\mathtt{T}, \mathtt{F}\}^n \to \{\mathtt{T}, \mathtt{F}\}$, there is a formula α in CNF such that $f_\alpha = g$.

Corollary 0.12 *The set $\{\neg, \wedge, \vee\}$ is complete.*

Exercises

(1) Let $\mathbb{R}$ be the set of real numbers. Define a relation R, where $(x, y) \in R$ if and only if $x < y$. Prove that $\mathbb{R}$ is a poset w.r.t. R.[‡]

(2) Give an example of a bounded chain in the poset $(\mathbb{R}, \leqslant)$ as defined in question 4.

(3) Give an example of an unbounded chain in the poset $(\mathbb{R}, \leqslant)$.

(4) Let A be a set and $\mathcal{F}$ be a collection of subsets of A. Define a relation R on elements of $\mathcal{F}$:

$$(x, y) \in R \quad \text{if and only if} \quad x \subseteq y$$

Prove that $\mathcal{F}$ is a poset w.r.t. R.[§]

(5) Give an example of a poset $(\mathcal{F}, \subseteq)$ in which every chain is bounded.

(6) Give an example of a poset $(\mathcal{F}, \subseteq)$ in which there is an unbounded chain.

(7) Consider a poset $(\mathcal{F}, \subseteq)$ where $\mathcal{F}$ is a collection of subsets of a set A. Suppose that for every chain C in $\mathcal{F}$, the set $\bigcup C$ is in $\mathcal{F}$.

Assuming Zorn's lemma, prove that there is an element $M \in \mathcal{F}$ such that there is no $X \in \mathcal{F}$ where $M \subsetneq X$.

(8) Write down the equivalent formulas for $x \leftrightarrow y$ in DNF and CNF.

(9) Write down the formulas in DNF and CNF for the following function $f(p, q, r)$:

p	q	r	$f(p,q,r)$
F	F	F	F
T	F	F	F
F	T	F	F
T	T	F	T
F	F	T	F
T	F	T	T
F	T	T	T
T	T	T	F

(10) Prove that $\{\neg, \wedge\}$ and $\{\neg, \vee\}$ are complete.

[‡]The poset $\mathbb{R}$ w.r.t. the relation $\leqslant$ is usually written as $(\mathbb{R}, \leqslant)$.

[§]The poset $\mathcal{F}$ w.r.t. the relation $\subseteq$ is usually written as $(\mathcal{F}, \subseteq)$.

(11) Define the operators NAND and NOR, denoted by $p \mathbin{\overline{\wedge}} q$ and $p \mathbin{\underline{\vee}} q$, respectively, as follows.

p	q	$p \mathbin{\overline{\wedge}} q$
T	T	F
T	F	T
F	T	T
F	F	T

p	q	$p \mathbin{\underline{\vee}} q$
T	T	F
T	F	F
F	T	F
F	F	T

That is, $p \mathbin{\overline{\wedge}} q$ is equivalent to $\neg(p \wedge q)$ and $p \mathbin{\underline{\vee}} q$ is equivalent to $\neg(p \vee q)$. Prove that $\{\overline{\wedge}\}$ and $\{\underline{\vee}\}$ are complete.

(12) Prove part (b) of Theorem 0.11.

Appendix: Basic set theoretic notations

Sets:

- A *set* is a collection of things, which are called its members or elements.

  $a \in X$ (read: a is in X, or a belongs to X) means a is a member or an element of X. $a \notin X$ means that a is not a member of X.
- An empty set is denoted by $\emptyset$.
- X is a *subset* of Y, denoted by $X \subseteq Y$, if every element of X is also an element of Y.

  X is a proper subset of Y, denoted by $X \subsetneq Y$, if $X \neq Y$ and $X \subseteq Y$.
- For two sets X and Y, we write $X \cap Y$ and $X \cup Y$ to denote their intersection and union, respectively.
- Let X be a set whose elements are also sets. Then, $\bigcup X$ and $\bigcap X$ denote the following.

$$\begin{aligned} \bigcup X &:= \{a \mid a \text{ belongs to an element in } X\} \\ \bigcap X &:= \{a \mid a \text{ belongs to every element in } X\} \end{aligned}$$

- The cartesian product between two sets X and Y is the following.

$$X \times Y \quad := \quad \{(a,b) \mid a \in X \text{ and } b \in Y\}.$$

  We write X^n to denote $X \times \cdots \times X$ (X appears n times).

Relations:

- A *relation* R over two sets X, Y is a subset of $X \times Y$.
- A *binary relation* R over X is a subset of $X \times X$.
- An *n-ary relation* R over X is a subset of X^n.

Functions:

- A relation R over X, Y is a *function* or a *mapping*, if for every $x \in X$, there is exactly one $y \in Y$ such that $(x,y) \in R$.

  In this case, we will say R is a function from X to Y, or R maps X to Y. We denote it by $R : X \to Y$.
- We will usually use the letters $f, g, h, \ldots$ to represent functions. As usual, we write $f(x)$ to denote the element y in which $(x,y) \in f$.
- A function $f : X \to Y$ is an *injective* function, if for every $y \in Y$, there is at most one $x \in X$ such that $f(x) = y$. An injective functions is also called an *injection*.
- A function $f : X \to Y$ is a *surjective* function, if for every $y \in Y$, there is at least one $x \in X$ such that $f(x) = y$.
- A function $f : X \to Y$ is a *bijection*, if it is both injective and surjective.

Lesson 1: Compactness theorem for propositional calculus

Theme: Logical consequences, compactness theorem and its applications.

1 Logical consequences

Definition 1.1 A formula α is a logical consequence of a formula β, denoted by $\beta \models \alpha$, if every satisfying assignment of β is also a satisfying assignment of α. If $\alpha \models \beta$ and $\beta \models \alpha$, we write $\alpha \models\dashv \beta$.

Definition 1.2 We say that α is a *logical consequence* of a set X of formulas, denoted by $X \models \alpha$, if every satisfying assignment of X is also a satisfying assignment of α.

We write $X \not\models \alpha$, if it is not the case that $X \models \alpha$.

Theorem 1.3 *$X \models \alpha$ if and only if $X \cup \{\neg\alpha\}$ is not satisfiable.*

2 Compactness theorem

We say that a set X is *finitely satisfiable*, if every finite subset of X is satisfiable.

Lemma 1.4 *Suppose X is finitely satisfiable. Then, for every formula α, at least one of $X \cup \{\alpha\}$ or $X \cup \{\neg\alpha\}$ is finitely satisfiable.*

Theorem 1.5 (Compactness theorem for countable set) *A set X is satisfiable if and only if it is finitely satisfiable.*

Proof. The "only if" direction is trivial. We show the "if" direction. Suppose X is finitely satisfiable. Let $\alpha_1, \alpha_2, \dots$ be an enumeration of all possible formulas. For every integer $i \geqslant 0$, we define a set Δ_i as follows.

$$\begin{array}{rcl} \Delta_0 & := & X \\ \Delta_i & := & \begin{cases} \Delta_{i-1} \cup \{\alpha_i\}, & \text{if } \Delta_{i-1} \cup \{\alpha_i\} \text{ is finitely satisfiable} \\ \Delta_{i-1} \cup \{\neg\alpha_i\}, & \text{otherwise} \end{cases} \end{array}$$

Let $\Delta := \bigcup_{i \geqslant 0} \Delta_i$.

Claim 1 *The set Δ is finitely satisfiable.*

Consider the following assignment w, where for each atomic formula p,

$$w(p) \quad := \quad \begin{cases} \mathtt{T}, & \text{if } p \in \Delta \\ \mathtt{F}, & \text{if } \neg p \in \Delta \end{cases}$$

Claim 2 *The assignment w is a satisfying assignment for Δ. That is, $w \models \Delta$.*

Since $X \subseteq \Delta$, w is also a satisfying assignment of X. Hence, X is satisfiable. This completes our proof. ■

3 Two applications of compactness theorem

3.1 Four-colorability of (infinite) planar graphs

An (undirected) graph G is a pair (V, E) with E being a symmetrical binary relation on V. That is, $E \subseteq V \times V$, where $(u, v) \in E$ if and only if $(v, u) \in E$. The elements of V are called *vertices*, and the elements of E are called *edges*. A subgraph G' of G is a graph (V', E'), where $V \subseteq V'$ and $E \subseteq E'$.

A graph $G = (V, E)$ is 4-colorable, if there is a function $\xi : V \to \{1, 2, 3, 4\}$ such that whenever $(u, v) \in E$, $\xi(u) \neq \xi(v)$.

Lemma 1.6 *Let G be a graph. Then, G is 4-colorable, if and only if every finite subgraph of G is 4-colorable.*

Proof. (Sketch) The "only if" direction is trivial. The proof of the "if" direction is as follows. Let $G = (V, E)$.

For each $a \in V$, we have four atomic formulas $p_{a,1}, p_{a,2}, p_{a,3}, p_{a,4}$. Define the set X_G that contains the following formulas for each $a \in V$:

$$p_{a,1} \vee p_{a,2} \vee p_{a,3} \vee p_{a,4}$$

$$\bigwedge_{1 \leqslant i < j \leqslant 4} \neg(p_{a,i} \wedge p_{a,j})$$

$$\bigwedge_{1 \leqslant i \leqslant 4} \neg(p_{a,i} \wedge p_{b,i}) \qquad \text{where } (a, b) \in E$$

Then, G is 4-colorable if and only if X_G is satisfiable. Since every finite subgraph G' of G is 4-colorable, $X_{G'}$ is satisfiable, for every finite subgraph G' of G. This means X_G is finitely satisfiable (why?). By Theorem 1.5, G is 4-colorable. ■

Theorem 1.7 below is a well known result whose proof we will not discuss in the class.

Theorem 1.7 (Four color theorem) *Every* finite *planar graph is 4-colorable.*[*]

Corollary 1.8 *Every (finite or infinite) planar graph is 4-colorable.*

3.2 The marriage problem

Let R be a relation over X, Y. For an element $a \in X$, we define $R(a) = \{b \in Y \mid (a, b) \in R\}$. Likewise, for a subset $X_0 \subseteq X$, $R(X_0) = \{b \in Y \mid \text{ there is } a \in X_0 \text{ such that } (a, b) \in R\}$.

Theorem 1.9 below is a standard result in discrete mathematics, and we will not discuss its proof in the class.

Theorem 1.9 (Hall's marriage theorem) *For a relation R over X, Y, where X is finite, the following are equivalent.*

- *R contains an injective function f, i.e., there is a function $f \subseteq R$ such that f is injective.*
- *For every subset $X_0 \subseteq X$, $|X_0| \leqslant |R(X_0)|$.*

Theorem 1.10 *For a relation R over X, Y, where X is* infinite *and for every $a \in X$, $|R(a)|$ is finite, the following are equivalent.*

[*] A graph $G = (V, E)$ is planar, if there is a function $f : V \to \mathbb{R}^2$, and there is curve between every two points $f(u)$ and $f(v)$, whenever $(u, v) \in E$, such that no two curves "cross" each other.

- *R contains an injective function f, i.e., there is a function $f \subseteq R$ such that f is injective.*
- *For every* finite *subset $X_0 \subseteq X$, $|X_0| \leqslant |R(X_0)|$.*

Proof. (Sketch) The implication from the first to the second item is immediate. The proof for the other direction is almost the same as in Lemma 1.6. Let R be a relation over X, Y, where for every $a \in X$, $|R(a)|$ is finite.

For each $a \in X$, we have atomic formulas $p_{a,b_1}, p_{a,b_2}, \ldots, p_{a,b_n}$, where $R(a) = \{b_1, \ldots, b_n\}$. Define the set X_R that contains the following formulas for each $a \in X$:

$$\bigvee_{b \in R(a)} p_{a,b}$$

$$\neg(p_{a,b} \ \wedge \ p_{a,c}), \qquad \text{where } b \neq c$$

R has the desired injection if and only if X_R is satisfiable. Using Theorems 1.9 and 1.5, the proof can proceed in a similar manner as in Lemma 1.6. ■

Exercises

(0) Is Lemma 1.4 still correct if we allow the set PV of propositional variables to be uncountable?

Our proof for the compactness theorem in the lecture depends on the fact that there are only countably many formulas. If there are uncountably many propositional variables, there are uncountably many formulas, and our proof is no longer valid. Here we are going to present a proof that still holds for uncountably many formulas, i.e., *X is satisfiable if and only if X is finitely satisfiable*, where PV can be an uncountable set.

As usual, the "only if" part is trivial. The proof for the "if" part is as follows. Let X be finitely satisfiable. Define the collection $\mathcal{F}$ of sets of formulas as follows.

$$Y \in \mathcal{F} \quad \text{if and only if} \quad X \subseteq Y \text{ and } Y \text{ is finitely satisfiable.}$$

(1) Prove that $(\mathcal{F}, \subseteq)$ is a poset.

(2) Let K be a chain in $(\mathcal{F}, \subseteq)$. Prove that $\bigcup K$ is finitely satisfiable.

(3) Prove that there is a maximal set $M \in \mathcal{F}$, i.e., M is a set in $\mathcal{F}$ such that there is no $Y \in \mathcal{F}$ where $M \subsetneq Y$.

Hint: Use Zorn's lemma stated in Lesson 1.

(4) Prove that for every $p \in PV$, either p or $\neg p$ is in M.

(5) Prove that M is satisfiable, and hence, so is X.

Lesson 2: Proof system in propositional calculus

Theme: The notion of provability in propositional calculus.

1 Proofs in propositional calculus

Let X be a set of formulas and α be a formula. We say that α is provable/derivable from X, denoted by $X \vdash \alpha$, if it can be obtained inductively according to the following rules.

Initial Segment (IS): $\alpha \vdash \alpha$

Monotonicity Rule (MR): $\dfrac{X \vdash \alpha}{Y \vdash \alpha}$ for every $Y \supseteq X$

And Combine Rule (ACR): $\dfrac{X \vdash \alpha \quad \text{and} \quad X \vdash \beta}{X \vdash \alpha \wedge \beta}$

And Split Rule (ASR): $\dfrac{X \vdash \alpha \wedge \beta}{X \vdash \alpha \quad \text{and} \quad X \vdash \beta}$

Contradiction Rule (CR): $\dfrac{X \vdash \alpha \quad \text{and} \quad X \vdash \neg\alpha}{X \vdash \beta}$ for every formula β

Negation Rule (NR): $\dfrac{X, \alpha \vdash \beta \quad \text{and} \quad X, \neg\alpha \vdash \beta}{X \vdash \beta}$

Sometimes we will also say "α can be proved from X" when $X \vdash \alpha$. We write $X \nvdash \alpha$, if it is not the case that $X \vdash \alpha$.

Remark 2.1 To avoid clutter, we write $\alpha \vdash \alpha$ to denote $\{\alpha\} \vdash \alpha$, whereas $X, \alpha \vdash \beta$ means $X \cup \{\alpha\} \vdash \beta$. We also write $\{\alpha_1, \ldots, \alpha_n\} \vdash \alpha$ to denote $\alpha_1, \ldots, \alpha_n \vdash \alpha$ and $\vdash \alpha$ to denote $\emptyset \vdash \alpha$.

Remark 2.2 Note that in the proof system above, we only use the operators $\neg$ and $\wedge$. In the following a formula $\alpha \to \beta$ is to be interpreted as an abbreviation for $\neg(\alpha \wedge \neg\beta)$, and likewise, $\alpha \vee \beta$ for $\neg(\neg\alpha \wedge \neg\beta)$.

Example 2.3 (Elimination of Negation) $\dfrac{X, \neg\alpha \vdash \alpha}{X \vdash \alpha}$

1. $X, \neg\alpha \vdash \alpha$. (Supposition)
2. $X, \alpha \vdash \alpha$. (Initial Segment and Monotonicity Rule)
3. $X \vdash \alpha$. (Negation Rule on 1 and 2)

Example 2.4 (Reductio ad Absurdum) $\dfrac{X, \neg\alpha \vdash \beta \quad \text{and} \quad X, \neg\alpha \vdash \neg\beta}{X \vdash \alpha}$

1. $X, \neg\alpha \vdash \beta$. (Supposition)
2. $X, \neg\alpha \vdash \neg\beta$. (Supposition)
3. $X, \neg\alpha \vdash \alpha$. (Contradiction Rule on 1 and 2)
4. $X, \alpha \vdash \alpha$. (Initial Segment and Monotonicity Rule)
5. $X \vdash \alpha$. (Negation Rule on 3 and 4)

Example 2.5 (Cut Rule) $\dfrac{X \vdash \alpha \quad \text{and} \quad X, \alpha \vdash \beta}{X \vdash \beta}$

1. $X \vdash \alpha$. (Supposition)
2. $X, \alpha \vdash \beta$. (Supposition)
3. $X, \neg\alpha \vdash \neg\alpha$. (Initial Segment and Monotonicity Rule)
4. $X, \neg\alpha \vdash \alpha$. (Monotonicity Rule on 1)
5. $X, \neg\alpha \vdash \beta$. (Contradiction Rule on 3 and 4)
6. $X \vdash \beta$. (Negation Rule on 2 and 5)

Example 2.6 (Elimination of $\to$) $\dfrac{X \vdash \alpha \to \beta}{X, \alpha \vdash \beta}$

1. $X \vdash \alpha \to \beta$. (Supposition)
2. $X, \alpha, \neg\beta \vdash \alpha$. (Initial Segment and Monotonicity Rule)
3. $X, \alpha, \neg\beta \vdash \neg\beta$. (Initial Segment and Monotonicity Rule)
4. $X, \alpha, \neg\beta \vdash \alpha \wedge \neg\beta$. (And Combine Rule on 2 and 3)
5. $X, \alpha, \neg\beta \vdash \neg(\alpha \wedge \neg\beta)$. (Monotonicity Rule on 1)
6. $X, \alpha, \neg\beta \vdash \beta$. (Contradiction Rule on 4 and 5)
7. $X, \alpha, \beta \vdash \beta$. (Initial Segment and Monotonicity Rule)
8. $X, \alpha \vdash \beta$. (Negation Rule on 6 and 7)

Example 2.7 (Introduction of $\to$) $\dfrac{X, \alpha \vdash \beta}{X \vdash \alpha \to \beta}$

1. $X, \alpha \vdash \beta$. (Supposition)
2. $X, \alpha, \alpha \wedge \neg\beta \vdash \beta$. (Monotonicity Rule on 1)
3. $X, \alpha \wedge \neg\beta \vdash \alpha \wedge \neg\beta$. (Initial Segment and Monotonicity Rule)
4. $X, \alpha \wedge \neg\beta \vdash \alpha$. (And Split Rule on 3)
5. $X, \alpha \wedge \neg\beta \vdash \beta$. (Cut rule on 4 and 2)
6. $X, \alpha \wedge \neg\beta \vdash \neg\beta$. (And Split Rule on 3)
7. $X \vdash \alpha \to \beta$. (Reductio ad Absurdum on 5 and 6)

Theorem 2.8 (Finiteness theorem for $\vdash$) *If $X \vdash \alpha$, then there is a* finite *set $X_0 \subseteq X$ such that $X_0 \vdash \alpha$.*

Exercises

(1) Prove that $\dfrac{X, \alpha \vdash \neg\alpha}{X \vdash \neg\alpha}$.

(2) Prove that $\dfrac{X \vdash \alpha \quad \text{and} \quad X \vdash \alpha \to \beta}{X \vdash \beta}$.

(3) Prove that $\dfrac{X \vdash \alpha \to \beta \quad \text{and} \quad X \vdash \beta \to \gamma}{X \vdash \alpha \to \gamma}$.

(4) Prove that $\dfrac{X \vdash \neg\neg\alpha}{X \vdash \alpha}$.

(5) Prove that $\dfrac{X \vdash \alpha}{X \vdash \neg\neg\alpha}$.

(6) Prove that $\dfrac{X, \alpha \vdash \beta}{X, \neg\neg\alpha \vdash \beta}$.

(7) Prove that $\dfrac{X \vdash \alpha \to \beta}{X \vdash \neg\beta \to \neg\alpha}$.

Note that $\alpha \to \beta$ is an abbreviation for $\neg(\alpha \wedge \neg\beta)$, whereas $\neg\beta \to \neg\alpha$ for $\neg(\neg\beta \wedge \neg\neg\alpha)$.

Lesson 3: Completeness of propositional calculus

Theme: The equivalence between provability and logical consequences (completeness of propositional calculus).

Definition 3.1 A set X is *inconsistent*, if there is α such that $X \vdash \alpha$ and $X \vdash \neg\alpha$. Otherwise, we say that X is *consistent*.

Lemma 3.2 *For every set X of formulas and for every formula α, the following holds.*

(a) $X \vdash \alpha$ if and only if $X \cup \{\neg\alpha\}$ is inconsistent.

(b) $X \vdash \neg\alpha$ if and only if $X \cup \{\alpha\}$ is inconsistent.

Definition 3.3 A set X is *maximally consistent*, if it is consistent and for every $Y \supsetneq X$, Y is inconsistent.

Lemma 3.4 *Every consistent set X can be extended to a maximally consistent set. That is, for every consistent set X, there is a maximally consistent set Y such that $Y \supseteq X$.*

Lemma 3.5 *A maximally consistent set X has the following property: For every α,*

$$X \vdash \neg\alpha \quad \textit{if and only if} \quad X \nvdash \alpha.$$

Lemma 3.6 *A maximally consistent set X is satisfiable.*

Proof. (Sketch) Define the following assignment w, where for every atomic proposition p:

$$w(p) \quad := \quad \begin{cases} \mathtt{T}, & \text{if } X \vdash p \\ \mathtt{F}, & \text{if } X \vdash \neg p \end{cases}$$

We have to show that for every $\alpha \in X$, $w(\alpha) = \mathtt{T}$. It is sufficient to show the following.

$$X \vdash \alpha \quad \text{if and only if} \quad w(\alpha) = \mathtt{T}.$$

The proof is by induction on α. ■

Theorem 3.7 (Completeness of propositional calculus) *$X \vdash \alpha$ if and only if $X \models \alpha$.*

Proof. The "only if" direction is straightforward. We prove the "if" direction by showing that $X \nvdash \alpha$ implies $X \not\models \alpha$.

Suppose $X \nvdash \alpha$. This means that $X \cup \{\neg\alpha\}$ is consistent. By Lemma 3.4, we can extend it to a maximally consistent set Y. Lemma 3.6 implies Y is satisfiable, and hence, $X \cup \{\neg\alpha\}$ is also satisfiable, which further implies that $X \not\models \alpha$ (why?). This completes our proof. ■

There are six rules in our proof system. Where do we use each of them in our proof of completeness theorem?

Lesson 4: First-order logic, part 1

Theme: Mathematical structures and the syntax of first-order logic.

For the rest of this course, we fix three pairwise disjoint sets L_r, L_f, L_c of symbols.

- Elements in L_r are called *relational* symbols. Each symbol $R \in L_r$ is associated with a positive integer, which is called its *arity* and denoted by $\mathrm{ar}(R)$.
- Elements in L_f are called *operation/function* symbols. Each symbol $f \in L_f$ is associated with a positive integer, which is called its *arity* and denoted by $\mathrm{ar}(f)$.
- Elements in L_c are called *constant* symbols.

We usually write $R_1, R_2, \ldots$ for the elements of L_r; $f_1, f_2, \ldots$ for the elements of L_f; and $c_1, c_2, \ldots$ for the elements of L_c.

1 Mathematical structures

Definition 4.1 Let $L = \{R_1, \ldots, R_m, f_1, \ldots, f_n, c_1, \ldots, c_k\}$ be a finite subset of $L_r \cup L_f \cup L_c$. An L-structure is $\mathcal{A} = (A, R_1^{\mathcal{A}}, \ldots, R_m^{\mathcal{A}}, f_1^{\mathcal{A}}, \ldots, f_n^{\mathcal{A}}, c_1^{\mathcal{A}}, \ldots, c_k^{\mathcal{A}})$, where

- A is a set of elements, called the *domain*, or the *universe* of $\mathcal{A}$;
- each $R_i^{\mathcal{A}}$ is a relation over A of arity $\mathrm{ar}(R_i)$, i.e., $R_i^{\mathcal{A}} \subseteq A^{\mathrm{ar}(R_i)}$;
- each $f_i^{\mathcal{A}}$ is a function over A of arity $\mathrm{ar}(f_i)$, i.e., $f : A^{\mathrm{ar}(f_i)} \to A$;
- each $c_i^{\mathcal{A}}$ is an element of A.

The set L is called the *signature/vocabulary* of $\mathcal{A}$. If A is finite, then $\mathcal{A}$ is called a finite structure. Otherwise, it is an infinite structure.

The superscripts $\mathcal{A}$ in $R_i^{\mathcal{A}}, f_i^{\mathcal{A}}, c_i^{\mathcal{A}}$ are to indicate that we are talking about R_i, f_i, c_i in the structure $\mathcal{A}$. When $\mathcal{A}$ is clear from the context, we will usually omit the superscript, and write only $\mathcal{A} = (A, R_1, \ldots, R_m, f_1, \ldots, f_n, c_1, \ldots, c_k)$. We will usually write $\bar{a}$ to denote $\bar{a} = (a_1, \ldots, a_n)$ for some appropriate n. For example, we will simply write $\bar{a} \in R$, where we assume that $\bar{a} = (a_1, \ldots, a_l)$ and l is the arity of R. Likewise, we write $f(\bar{a})$ assuming that $\bar{a}$ is of length $\mathrm{ar}(f)$. We will also write $R(\bar{a})$ to mean $\bar{a} \in R$.

Remark 4.2 Usually structures are denoted by calligraphic fonts $\mathcal{A}, \mathcal{B}, \ldots$, and their domains by the standard Roman letters $A, B, \ldots$.

Remark 4.3 In definition 4.1 above, a structure is defined over a *finite* vocabulary, and that is usually the case. Sometimes though a structure can be defined over an *infinite* vocabulary, or even *uncountably infinite* vocabulary.

Definition 4.4 Let $\mathcal{A}$ and $\mathcal{B}$ be two structures over the same vocabulary L. The structure $\mathcal{A}$ is called a *substructure* of $\mathcal{B}$, if the following holds.

- $A \subseteq B$.
- $c^{\mathcal{A}} = c^{\mathcal{B}}$, for every constant symbol $c \in L$.

- $R^{\mathcal{A}} = A^{\mathrm{ar}(R)} \cap R^{\mathcal{B}}$, for every relation symbol $R \in L$.
- $f^{\mathcal{A}} = A^{\mathrm{ar}(f)+1} \cap f^{\mathcal{B}}$, for every function symbol $f \in L$.

The structure $\mathcal{B}$ is called an *extension* of $\mathcal{A}$.

Definition 4.5

- A *relational structure* is an L-structure, where $L \subseteq L_r$, i.e., without any function or constant.
- An *algebraic structure*, or in short, an *algebra*, is an L-structure, where $L \subseteq L_f \cup L_c$, i.e., without any relation.

Example 4.6 Some instances of infinite structures.

- $\mathcal{N}_0 = (\mathbb{N}, \leqslant)$.
- $\mathcal{N}_1 = (\mathbb{N}, 0, +)$.
- $\mathcal{N}_2 = (\mathbb{N}, 0, 1, +, \times)$.
- $\mathcal{N}_3 = (\mathbb{N}, 0, 1, +, \times, \leqslant)$.
- $\mathcal{R}_0 = (\mathbb{R}, \leqslant)$.
- $\mathcal{R}_1 = (\mathbb{R}, 0, +)$.
- $\mathcal{R}_2 = (\mathbb{R}, 0, 1, +, \times)$.
- $\mathcal{R}_3 = (\mathbb{R}, 0, 1, +, \times, \leqslant)$.

Example 4.7 Some instances of finite structures.

- $\mathcal{Z}_m = (\mathbb{Z}_m, 0, +_{\text{mod } m})$, where $+_{\text{mod } m}$ is addition modulo m.
- $\mathcal{Z}_p^* = (\mathbb{Z}_p, 0, 1, +_{\text{mod } p}, \times_{\text{mod } p})$, for a prime number p, where $\times_{\text{mod } p}$ is multiplication modulo p.
- $\mathcal{B} = (\{\mathtt{T}, \mathtt{F}\}, \wedge, \vee, \to, \leftrightarrow, \neg)$.

Example 4.8 A *graph* is a structure $\mathcal{A} = (A, E)$, where $E \subseteq A \times A$. It is usually written as $G = (V, E)$.

Definition 4.9 Let $\mathcal{A}, \mathcal{B}$ be L-structures. A *homomorphism* h from $\mathcal{A}$ to $\mathcal{B}$, denoted by $h : \mathcal{A} \to \mathcal{B}$, is a function $h : A \to B$ such that for every $R, f, c \in L$,

- $h(f^{\mathcal{A}}(\bar{a})) = f^{\mathcal{B}}(h(\bar{a}))$, for every $\bar{a} \in A^{\mathrm{ar}(f)}$,
- $h(c^{\mathcal{A}}) = c^{\mathcal{B}}$,
- for every $\bar{a} \in A^{\mathrm{ar}(R)}$, if $R^{\mathcal{A}}(\bar{a})$, then $R^{\mathcal{B}}(h(\bar{a}))$.

Here, $h(\bar{a}) = (h(a_1), \ldots, h(a_l))$, where $\bar{a} = (a_1, \ldots, a_l)$.

Definition 4.10 Let $h : \mathcal{A} \to \mathcal{B}$ be a homomorphism.

- h is a *strong* homomorphism, if h is a homomorphism and in addition, for every relation $R \in L$, for every $\bar{a} \in A^{\mathrm{ar}(R)}$,
$$\text{if } R^{\mathcal{B}}(h(\bar{a})), \text{ then there is } \bar{a}' \in A^{\mathrm{ar}(R)} \text{ such that } h(\bar{a}) = h(\bar{a}') \text{ and } R^{\mathcal{A}}(\bar{a}').$$
- h is an *embedding*, if it is an injective and strong homomorphism.
- h is an *isomorphism*, if it is a strong and bijective homomorphism.
- h is called *automorphism*, if h is an isomorphism and $\mathcal{B} = \mathcal{A}$, i.e., h is a bijection from A to A itself.

2 The syntax of first-order logic

2.1 Variables and terms

We reserve a set VAR of *first-order variables*. We usually write $x_1, x_2, \ldots, y_1, y_2, \ldots, z_1, z_2, \ldots$ to denote elements in VAR. When it is clear from the context, we will simply say *variables*, instead of *first-order variables*.

In the following let L be a finite subset of $L_r \cup L_f \cup L_c$. *Terms over* L, or, L-terms, are defined inductively as follows.

- A variable $x \in$ VAR is an L-term.
- A constant symbol $c \in L$ is an L-term.
- If $f \in L$ is a function symbol of arity n, and $t_1, \ldots, t_n$ are L-terms, then $f(t_1, \ldots, t_n)$ is an L-term.

The variable x and the constant c are called *atomic* L*-terms*. The set of all L-terms is denoted by Term(L). When there is no confusion, we will omit L, and simply write *terms*, instead of L*-terms*.

The set of variables used in a term t is the set $\mathrm{var}(t)$ defined as follows.

- For a constant symbol $c \in L$, $\mathrm{var}(c) = \emptyset$.
- For a variable $x \in$ VAR, $\mathrm{var}(x) = \{x\}$.
- For a term of the form $f(t_1, \ldots, t_n)$, $\mathrm{var}(f(t_1, \ldots, t_n)) = \mathrm{var}(t_1) \cup \cdots \cup \mathrm{var}(t_n)$.

2.2 First-order formulas

First-order (FO) formulas over the signature/vocabulary L are defined inductively as follows.

- If s and t are terms over L, then $(s \approx t)$ is an FO formula over L.
- If $t_1, \ldots, t_n$ are terms over L, and $R \in L$ is a relation symbol of arity n, then $R(t_1, \ldots, t_n)$ is an FO formula over L.
- If α and β are FO formulas over L, then so are $\neg\alpha$, $\alpha \wedge \beta$ and $\alpha \vee \beta$.
- If α is an FO formula over L, and $x \in$ VAR, then $\forall x(\alpha)$ is also an FO formula over L.
- If α is an FO formula over L, and $x \in$ VAR, then $\exists x(\alpha)$ is also an FO formula over L.

FO formulas of the form $s \approx t$ and $R(t_1, \ldots, t_n)$ are called *atomic* FO *formulas*. The set of all FO formulas over L is denoted by FO[L]. We will write $s \not\approx t$ as an abbreviation for $\neg(s \approx t)$.

To avoid clutter, we will usually write only *formulas* to mean FO *formulas*. When the signature L is clear, we will also omit mentioning it. So the word *formula* means an FO formula over some signature L which can be derived from the context.

The *quantifier rank* of a formula α, denoted by $\mathrm{qr}(\alpha)$, is defined inductively as follows.

- The quantifier rank of an atomic formula is zero.
- $\mathrm{qr}(\neg\beta) = \mathrm{qr}(\beta)$.
- $\mathrm{qr}(\beta \wedge \gamma) = \mathrm{qr}(\beta \vee \gamma) = \max(\mathrm{qr}(\beta), \mathrm{qr}(\gamma))$.
- $\mathrm{qr}(\forall x\ \beta) = \mathrm{qr}(\exists x\ \beta) = \mathrm{qr}(\beta) + 1$.

The set of *free variables* of a formula α, denoted by $\mathrm{free}(\alpha)$, is defined inductively as follows.

- If α is an atomic formula $s \approx t$, then $\mathrm{free}(\alpha) = \mathrm{var}(s) \cup \mathrm{var}(t)$.

- If α is an atomic formula $R(t_1, \ldots, t_n)$, then $\text{free}(\alpha) = \text{var}(t_1) \cup \cdots \cup \text{var}(t_n)$.
- $\text{free}(\neg\beta) = \text{free}(\beta)$.
- $\text{free}(\beta \wedge \gamma) = \text{free}(\beta \vee \gamma) = \text{free}(\beta) \cup \text{free}(\gamma)$.
- $\text{free}(\forall x\ \beta) = \text{free}(\exists x\ \beta) = \text{free}(\beta) - \{x\}$.

Formulas without free variables are called *sentences*, or *closed formulas*. Otherwise, they are called *open formulas*. A formula without any quantifier is called a *quantifier free* formula.

Sometimes, we will write $\varphi(x_1, \ldots, x_n)$ to indicate that the free variables in φ are $x_1, \ldots, x_n$. When n is unspecified, we write $\varphi(\bar{x})$. For formula of the form $\forall x\ \beta$, we say that x is a *bound variable* in $\forall x\ \beta$. Likewise, we say that x is a *bound variable* in $\exists x\ \beta$. In both cases, we say that β is the scope of x, and that x is bounded by a quantifier.

2.3 Substitutions

Simple substitutions. Let t be a term and x a variable. Let s be a term. The term $t[s/x]$ is the term obtained by substituting s to the variable x in t. Formally, it is defined inductively as follows.

- $x[s/x] = s$ and $y[s/x] = y$, if $y \neq x$.
- $c[s/x] = c$, where c is a constant symbol.
- $f(t_1, \ldots, t_n)[s/x] = f(t_1[s/x], \ldots, t_n[s/x])$, where f is a function symbol of arity n.

The formula $\alpha[s/x]$ is the formula obtained by substituting the free variable x in α with the term s. Formally, it is defined inductively as follows.

- $\big(t_1 \approx t_2\big)[s/x] \ = \ \big(t_1[s/x] \approx t_2[s/x]\big)$.
- $R(t_1, \ldots, t_n)[s/x] \ = \ R(t_1[s/x], \ldots, t_n[s/x])$.
- $(\neg\alpha)[s/x] \ = \ \neg(\alpha[s/x])$.
- $\big(\alpha \wedge \beta\big)[s/x] \ = \ \alpha[s/x] \ \wedge \ \beta[s/x]$.
- $\big(\alpha \vee \beta\big)[s/x] \ = \ \alpha[s/x] \ \vee \ \beta[s/x]$.
- $\big(\forall y\ \alpha\big)[s/x] \ = \ \begin{cases} \forall y\ \alpha[s/x] & \text{if } y \neq x \\ \forall y\ \alpha & \text{if } y = x \end{cases}$.
- $\big(\exists y\ \alpha\big)[s/x] \ = \ \begin{cases} \exists y\ \alpha[s/x] & \text{if } y \neq x \\ \exists y\ \alpha & \text{if } y = x \end{cases}$.

Collision-free substitution. A substitution s/x is *collision-free* in a formula α, if the following holds.

- s/x is collision-free in the atomic formulas $t_1 \approx t_2$ and $R(t_1, \ldots, t_n)$.
- s/x is collision-free in $\neg\alpha$ if and only if it is collision-free in α.
- s/x is collision-free in $\alpha \wedge \beta$ if and only if it is collision-free in both α and β. Likewise, it is collision free in $\alpha \vee \beta$ if and only if it is collision-free in both α and β.
- s/x is collision-free in $\forall y\ \alpha$ if and only if $y \notin \text{var}(s)$ and it is collision-free in α.

  Likewise, s/x is collision-free in $\exists y\ \alpha$ if and only if $y \notin \text{var}(s)$ and it is collision-free in α.

Simultaneous substitutions. For a formula $\alpha(x_1, \ldots, x_n)$ and terms $\bar{t} = (t_1, \ldots, t_n)$, $\alpha[\bar{t}/\bar{x}]$ denotes a substitution in which each x_i is substituted with t_i. Such a substitution $\alpha[\bar{t}/\bar{x}]$ is called a *simultaneous substitution*. It is collision-free, if each t_i/x_i is collision-free.

Lesson 5: First-order logic, part 2

Theme: The semantics of first-order logic.

1 Valuations

Recall that VAR is a set of variables. Let $\mathcal{A}$ be a structure.

- A *valuation* in a structure $\mathcal{A}$ is a function $\mathsf{val} : \mathrm{VAR} \to A$.
- For $\bar{a} = (a_1, \ldots, a_n)$, where each $a_i \in A$, and $\bar{x} = (x_1, \ldots, x_n)$, where $x_1, \ldots, x_n$ are all different variables, we write $\mathsf{val}[\bar{x} \mapsto \bar{a}]$ to denote the valuation val', where for every $y \in \mathrm{VAR}$,

$$\mathsf{val}'(y) \quad = \quad \begin{cases} \mathsf{val}(y), & \text{if } y \notin \{x_1, \ldots, x_n\} \\ a_i, & \text{if } y = x_i \end{cases}$$

  Sometimes we write $[\bar{x} \mapsto \bar{a}]$ to denote a valuation val such that $\mathsf{val}(x_i) = a_i$.

2 Interpretations/models

An *interpretation* is a pair $(\mathcal{A}, \mathsf{val})$, where $\mathcal{A}$ is a structure and val is a valuation. Quite often, interpretations are also called *models*.

In an interpretation $(\mathcal{A}, \mathsf{val})$, each term t is associated with an element $t^{\mathcal{A}}[\mathsf{val}]$ defined inductively as follows.

- $x^{\mathcal{A}}[\mathsf{val}] = \mathsf{val}(x)$, where $x \in \mathrm{VAR}$.
- $c^{\mathcal{A}}[\mathsf{val}] = c^{\mathcal{A}}$, where c is a constant symbol.
- $f(t_1, \ldots, t_n)^{\mathcal{A}}[\mathsf{val}] = f^{\mathcal{A}}(t_1^{\mathcal{A}}[\mathsf{val}], \ldots, t_n^{\mathcal{A}}[\mathsf{val}])$.

$t^{\mathcal{A}}[\mathsf{val}]$ reads *the term t in structure $\mathcal{A}$ according to valuation val.*

As usual, when the structure $\mathcal{A}$ is clear from the context, we will simply write $t[\mathsf{val}]$, instead of $t^{\mathcal{A}}[\mathsf{val}]$.

Given an FO formula φ, and an interpretation $(\mathcal{A}, \mathsf{val})$, we define $(\mathcal{A}, \mathsf{val}) \models \varphi$ (read: $(\mathcal{A}, \mathsf{val})$ is an interpretation/a model of φ, or that φ holds in $(\mathcal{A}, \mathsf{val})$) inductively as follows.

- $(\mathcal{A}, \mathsf{val}) \models s \approx t$, if and only if $s^{\mathcal{A}}[\mathsf{val}] = t^{\mathcal{A}}[\mathsf{val}]$.
- $(\mathcal{A}, \mathsf{val}) \models R(t_1, \ldots, t_n)$, if and only if $(t_1^{\mathcal{A}}[\mathsf{val}], \ldots, t_n^{\mathcal{A}}[\mathsf{val}]) \in R^{\mathcal{A}}$.
- $(\mathcal{A}, \mathsf{val}) \models \neg\alpha$, if and only if it is *not true* that $(\mathcal{A}, \mathsf{val}) \models \alpha$.
- $(\mathcal{A}, \mathsf{val}) \models \alpha \wedge \beta$, if and only if $(\mathcal{A}, \mathsf{val}) \models \alpha$ and $(\mathcal{A}, \mathsf{val}) \models \beta$.
- $(\mathcal{A}, \mathsf{val}) \models \alpha \vee \beta$, if and only if $(\mathcal{A}, \mathsf{val}) \models \alpha$ or $(\mathcal{A}, \mathsf{val}) \models \beta$.
- $(\mathcal{A}, \mathsf{val}) \models \exists x\ \alpha$, if and only if there is $a \in A$ such that $(\mathcal{A}, \mathsf{val}[x \mapsto a]) \models \alpha$.
- $(\mathcal{A}, \mathsf{val}) \models \forall x\ \alpha$, if and only if for every $a \in A$, $(\mathcal{A}, \mathsf{val}[x \mapsto a]) \models \alpha$.

We write $(\mathcal{A}, \mathsf{val}) \not\models \varphi$, when it is not true that $(\mathcal{A}, \mathsf{val}) \models \varphi$.

Note that whether $(\mathcal{A}, \mathsf{val}) \models \varphi(x_1, \ldots, x_n)$ depends only on $\mathcal{A}$ (obviously!) and the images of $x_1, \ldots, x_n$ under val. In other words, the value $\mathsf{val}(y)$ does not matter for every $y \notin \{x_1, \ldots, x_n\}$. To avoid clutter, we write $(\mathcal{A}, a_1, \ldots, a_n) \models \varphi(x_1, \ldots, x_n)$, to mean that $(\mathcal{A}, \mathsf{val}) \models \varphi$, where val is a valuation function that maps each x_i to a_i. In particular, if α is a sentence, the valuation val is dispensable in the determination of $(\mathcal{A}, \mathsf{val}) \models \alpha$. So, for a sentence α, we simply write $\mathcal{A} \models \alpha$.

A formula φ *is satisfiable*, if φ has an interpretation/model.

3 Some examples

Example 5.1 Let $\mathcal{A} = (A, \text{plus}^{\mathcal{A}}, 0^{\mathcal{A}})$ be the structure with signature $\{\text{plus}, 0\}$ defined as follows.

- $A = \{0, 1, 2, \dots, 8\}$,
- plus is a binary function/operator, where $\text{plus}^{\mathcal{A}}(x, y) = x + y \bmod 9$,
- $0^{\mathcal{A}} = 0$.

Here are some formulas that hold/not hold in $\mathcal{A}$.

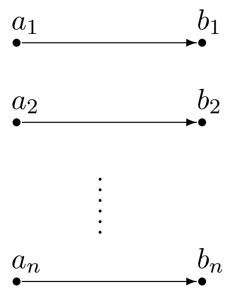
- $\mathcal{A}, (x, y, z) \mapsto (3, 5, 8) \models \text{plus}(x, y) \approx z$. Can I say that $\mathcal{A} \models \text{plus}(3, 5) \approx 8$?
- $\mathcal{A}, (x, y) \mapsto (1, 2) \not\models \text{plus}(x, y) \approx 0$.

  This is equivalent to say that $\mathcal{A}, (x, y) \mapsto (1, 2) \models \neg(\text{plus}(x, y) \approx 0)$, or, $\mathcal{A}, (x, y) \mapsto (1, 2) \models \text{plus}(x, y) \not\approx 0$.
- $\mathcal{A}, z \mapsto 0 \models \forall x\ \text{plus}(x, z) \approx x$.
- $\mathcal{A}, z \mapsto 1 \models \forall x\ \text{plus}(x, z) \not\approx x$. Can I say that $\mathcal{A} \models \forall x\ \text{plus}(x, 1) \not\approx x$?
- $\mathcal{A} \models \forall x\ \text{plus}(x, 0) \approx x$.
- $\mathcal{A} \models \forall x \exists y\ \text{plus}(x, y) \approx 0$.
- $\mathcal{A} \models \forall x \Big(x \not\approx 0 \ \rightarrow\ \big(\exists y\ x \not\approx y\ \wedge\ \text{plus}(x, y) \approx 0 \big) \Big)$.

Example 5.2 Let $\mathcal{B} = (B, E^{\mathcal{B}})$ be the following structure, where $\text{ar}(E) = 2$:

- $B = \{a_1, b_1, \dots, a_n, b_n\}$,
- $E^{\mathcal{B}} = \{(a_1, b_1), (a_2, b_2), \dots, (a_n, b_n)\}$.

The relation $E^{\mathcal{B}}$ can be illustrated as follows.



Here are some examples of formulas that hold/not hold in $\mathcal{B}$.

- $\mathcal{B}, (x, y) \mapsto (a_1, b_1) \models E(x, y)$. Can I say $\mathcal{B} \models E(a_1, b_1)$?
- $\mathcal{B}, (x, y) \mapsto (a_1, b_3) \not\models E(x, y)$.
- $\mathcal{B} \models \exists x \exists y\ E(x, y)$.
- $\mathcal{B} \not\models \exists x E(x, x)$, which can be rewritten as $\mathcal{B} \models \neg \exists x E(x, x)$
- $\mathcal{B} \models \forall x \exists y \Big(E(x, y)\ \wedge\ \forall z\ \big(E(x, z) \rightarrow y \approx z \big) \Big)$.

Example 5.3 Let $\mathcal{Z} = (\mathbb{Z}, \mathrm{succ}^{\mathcal{Z}}, \mathrm{plus}^{\mathcal{Z}}, 0^{\mathcal{Z}})$ be the structure with signature $\{\mathrm{plus}, \mathrm{succ}, 0\}$ defined as follows.

- $\mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$,
- $\mathrm{succ}^{\mathcal{Z}}$ is a binary relation, where $(x, y) \in \mathrm{succ}^{\mathcal{Z}}$ if and only if $y = x + 1$,
- $\mathrm{plus}^{\mathcal{Z}}$ is a binary operator, where $\mathrm{plus}^{\mathcal{Z}}(x, y) = x + y$,
- $0^{\mathcal{Z}} = 0$.

Here are some formulas that hold/not hold in $\mathcal{Z}$.

- $\mathcal{Z}, (x, y, z) \mapsto (3, 5, 8) \models \mathrm{plus}(x, y) \approx z$.
- $\mathcal{Z}, (x, y) \mapsto (1, 2) \not\models \mathrm{plus}(x, y) \approx 0$.
- $\mathcal{Z}, z \mapsto 0 \models \forall x\ \mathrm{plus}(x, z) \approx x$.
- $\mathcal{Z}, z \mapsto 1 \models \forall x\ \mathrm{plus}(x, z) \not\approx x$.
- $\mathcal{Z} \models \forall x\ \mathrm{plus}(x, 0) \approx x$.
- $\mathcal{Z} \models \forall x \exists y\ \mathrm{plus}(x, y) \approx 0$.
- $\mathcal{Z} \models \forall x \exists y\ \mathrm{succ}(x, y)\ \wedge\ x \not\approx y$.
- $\mathcal{Z} \models \forall x \exists y\ \mathrm{succ}(x, y)\ \wedge\ \Big(\forall z \big(\mathrm{succ}(x, z)\ \rightarrow\ y \approx z\big)\Big)$.
- $\mathcal{Z} \models \forall x \forall y \forall z \forall w \Big(\big(\mathrm{succ}(x, z)\ \wedge\ \mathrm{succ}(w, y)\big)\ \rightarrow\ \mathrm{plus}(x, y) \approx \mathrm{plus}(z, w)\Big)$.

4 Two little theorems

Theorem 5.4 *Let $h : \mathcal{A} \to \mathcal{B}$ be an isomorphism. Then, for every formula $\varphi(\bar{x})$,*

$$(\mathcal{A}, \bar{a}) \models \varphi(\bar{x}) \quad \textit{if and only if} \quad (\mathcal{B}, h(\bar{a})) \models \varphi(\bar{x})$$

(Recall that $\bar{x}$ and $\bar{a}$ stands for a vector of variables and elements, respectively, which we tacitly assume to be of the same length.)

A $\forall$-sentence (read: a universal sentence) is a sentence of the form:

$$\forall x_1\ \cdots\ \forall x_n\ \varphi, \tag{1}$$

where φ is quantifier free. Likewise, an $\exists$-sentence (read: an existential sentence) is a sentence of the form:

$$\exists x_1\ \cdots\ \exists x_n\ \varphi, \tag{2}$$

where φ is quantifier free. As usual, we will simply write $\forall \bar{x}\ \varphi$ or $\exists \bar{x}\ \varphi$, instead of Eq. (1) and (2), respectively.

Theorem 5.5 *Let $\mathcal{A} \subseteq \mathcal{B}$.*

- *For every $\forall$-sentence ψ, if $\mathcal{B} \models \psi$, then $\mathcal{A} \models \psi$.*
- *For every $\exists$-sentence ψ, if $\mathcal{A} \models \psi$, then $\mathcal{B} \models \psi$.*

Exercise set 1

In the following E, R, T, S are relational symbols, f, g are function symbols and $c, c_1, c_2, \ldots$ are constant symbols.

(1) Determine the quantifier rank of each of the following formulas.

$$\begin{array}{lcl}
\beta_1 & := & \forall x \exists y \big(z \not\approx y \ \wedge \ R(x,y)\big) \\
\beta_2 & := & \forall x \big(x \not\approx y \ \wedge \ \exists y \ R(x,y)\big) \\
\beta_3 & := & \Big(\forall z \big(\exists z \ z \not\approx y\big)\Big) \ \wedge \ f(z) \approx z \\
\beta_4 & := & \forall z \Big(z \approx y \ \wedge \ \exists z \big(f(z) \approx g(z)\big)\Big) \\
\beta_5 & := & \exists y \forall x \big(R(z, g(z,y)) \ \wedge \ T(y) \ \rightarrow \ \exists z \forall y \ x \approx f(x, g(y,z))\big) \\
\beta_6 & := & x \not\approx f(c,z) \ \wedge \ \forall z \forall x \Big(R(x,c,c,y) \ \wedge \ f(x,z) \approx c \ \wedge \ \exists y \big(f(x,y) \ \wedge \ g(z,y)\big)\Big)
\end{array}$$

(2) Determine the free variables of each of the formulas above.

(3) Determine the result of each of the following substitutions.

- $z/f(z,z,x)$ in β_1.
- $y/g(c,c)$ in β_2.
- $z/f(x,y,z)$ in β_3.
- y/z in β_4.
- $z/f(c,z,x)$ in β_5.
- $(x,y,z)/(x,x,x)$ in β_6.

Which substitutions are collision-free?

Exercise set 2: The notion of congruence

In this exercise, we will study the notion of congruence on structures. Let Z be set, and $\sim$ be an equivalence relation on Z. For a positive integer n, define a binary relation $\sim^n$ on Z^n as follows.

$$(a_1, \ldots, a_n) \sim^n (b_1, \ldots, b_n) \quad \text{if and only if} \quad a_i \sim b_i, \ \text{for each } i \in \{1, \ldots, n\}.$$

(4) Prove that $\sim^n$ is an equivalence relation.

The relation $\sim^n$ is called the extension of $\sim$ to Z^n.

(5) Prove that $[\bar{a}]_{\sim^n} = [a_1]_\sim \times [a_2]_\sim \times \cdots \times [a_n]_\sim$, where $\bar{a} = (a_1, \ldots, a_n)$.

When it is clear from the context, we will simply use the same symbol $\sim$, instead of $\sim^n$. That is, we will write $\bar{a} \sim \bar{b}$ to mean the extension of $\sim$ to Z^n, instead of $\bar{a} \sim^n \bar{b}$.

Let $\mathcal{A}$ be an L-structure. A *congruence* in $\mathcal{A}$ is an equivalence relation $\sim$ on A such that for every function symbol $f \in L$, the following holds.

$$\text{If } \bar{a} \sim \bar{b}, \ \text{then } f(\bar{a}) \sim f(\bar{b}).$$

(6) Let $\sim$ be a congruence in an L-structure $\mathcal{A}$. The *factor of $\mathcal{A}$ modulo $\sim$* is a structure $\mathcal{B}$ such that

- $B = A/{\sim} = \{[a]_\sim \mid a \in A\}$,

- $([a_1]_\sim, \ldots, [a_l]_\sim) \in R^{\mathcal{B}}$ if and only if $R^{\mathcal{A}} \cap [\bar{a}]_{\sim^l} \neq \emptyset$, for every relation symbol $R \in L$ of arity l,
- $c_i^{\mathcal{B}} = [c_i^{\mathcal{A}}]_\sim$,
- $f^{\mathcal{B}}([a_1]_\sim, \ldots, [a_l]_\sim) = [f^{\mathcal{A}}(a_1, \ldots, a_l)]_\sim$, for every function symbol $f \in L$ of arity l.

Prove that this definition is sound. That is, show that

(i) if $[(a_1, \ldots, a_l)]_{\sim^l} = [(b_1, \ldots, b_l)]_{\sim^l}$, then $([a_1]_\sim, \ldots, [a_l]_\sim) = ([b_1]_\sim, \ldots, [b_l]_\sim)$,
(ii) if $([a_1]_\sim, \ldots, [a_l]_\sim) = ([b_1]_\sim, \ldots, [b_l]_\sim)$, then $f^{\mathcal{B}}([a_1]_\sim, \ldots, [a_l]_\sim) = f^{\mathcal{B}}([b_1]_\sim, \ldots, [b_l]_\sim)$.

The factor of $\mathcal{A}$ modulo $\sim$ is denoted by $\mathcal{A}/\sim$.

(7) For a congruence $\sim$ in $\mathcal{A}$, the *canonical homomorphism* $\kappa : \mathcal{A} \to \mathcal{A}/\sim$ is defined by $\kappa(a) = [a]_\sim$. Prove that κ is a strong and surjective homomorphism.

Exercise set 3: Skolem normal form

Two formulas φ_1 and φ_2 are *equi-satisfiable*, if

$$\varphi_1 \text{ is satisfiable} \quad \text{if and only if} \quad \varphi_2 \text{ is satisfiable.}$$

(8) Consider a sentence ψ over a vocabulary L of the form:

$$\psi \quad := \quad \exists x_1 \cdots \exists x_n \ \varphi$$

Pick n "new" constant symbols $c_1, \ldots, c_n \notin L$. Show that $\varphi[(x_1, \ldots, x_n)/(c_1, \ldots, c_n)]$ and ψ are equi-satisfiable.

(9) Consider a sentence ψ over a vocabulary L of the form:

$$\psi \quad := \quad \forall x_1 \cdots \forall x_n \exists y \ \varphi$$

Pick a "new" arity n function symbol $f \notin L$. Show that $\forall x_1 \cdots \forall x_n \varphi[y/f(x_1, \ldots, x_n)]$ and ψ are equi-satisfiable.

(10) Consider a sentence ψ over a vocabulary L of the form:

$$\psi \quad := \quad \forall x_1 \cdots \forall x_n \exists y_1 \cdots \exists y_m \ \varphi,$$

where φ does not start with existential quantifiers. Prove that there is a sentence of the form:

$$\psi' \quad := \quad \forall x_1 \cdots \forall x_n \ \varphi'$$

such that φ' does not start with existential quantifiers, and ψ and ψ' are equi-satisfiable.

(11) **(Skolem normal form)** Consider a sentence ψ over a vocabulary L of the form:

$$\psi \quad := \quad Q_1 x_1 \cdots Q_n x_n \ \varphi, \tag{3}$$

where each Q_i is a quantifier (either $\forall$ or $\exists$), and φ is quantifier free. Prove that there is $\forall$-sentence ψ' (over different vocabulary L') such that ψ and ψ' are equi-satisfiable.

Note 1: The $\forall$-sentence ψ' is called the *Skolem normal form* of ψ.

Note 2: Formulas of the form (3) are often called formulas in *Prenex Normal Form* (PNF). We will show later on that every formula can be converted into PNF.

Lesson 6: Logical consequences and theories

Theme: Logical consequences and first-order theories.

1 Logical consequences

Definition 6.1 Let X be a set of formulas. We write $(\mathcal{A}, \mathsf{val}) \models X$, if $(\mathcal{A}, \mathsf{val}) \models \varphi$, for every $\varphi \in X$.

Definition 6.2 A formula β is a logical consequence of a formula α, denoted by $\alpha \models \beta$, if every model of α is also a model of β. If $\alpha \models \beta$ and $\beta \models \alpha$, we write $\alpha \models\!\dashv \beta$, or $\alpha \equiv \beta$.

One example is $\forall x\ \varphi \models \exists x\ \varphi$. (Recall that the domain of a structure is never empty.)

Definition 6.3 We say that α is a *logical consequence* of a set X of formulas, denoted by $X \models \alpha$, if every model of X is also a model of α. More formally, $X \models \alpha$ means that for every model $(\mathcal{A}, \mathsf{val})$, if $(\mathcal{A}, \mathsf{val}) \models X$, then $(\mathcal{A}, \mathsf{val}) \models \alpha$.

We write $X \not\models \alpha$, if it is not the case that $X \models \alpha$.

Definition 6.4 A *sentence* φ is *valid*, if $\models \varphi$. In other words, φ is valid, if $\mathcal{A} \models \varphi$, for every structure $\mathcal{A}$.*

Some conventions to read the notations:

- $(\mathcal{A}, \mathsf{val}) \models X$ is read as "$(\mathcal{A}, \mathsf{val})$ is a model of X."
- $\alpha \models \beta$ is also read as "α implies β."
- $\alpha \equiv \beta$ is also read as "α and β are equivalent."

Theorem 6.5 *$X \models \varphi$ if and only if $X \cup \{\neg\varphi\}$ is not satisfiable.*

Proposition 6.6 *For every formulas α and β, the following holds.*

$$\begin{array}{rcll}
\neg\forall x\ \alpha & \equiv & \exists x\ \neg\alpha & \\
\neg\exists x\ \alpha & \equiv & \forall x\ \neg\alpha & \\
\alpha\ \wedge\ \forall x\ \beta & \equiv & \forall x\big(\alpha\ \wedge\ \beta\big) & \textit{when } x \textit{ is not free in } \alpha \\
\alpha\ \wedge\ \exists x\ \beta & \equiv & \exists x\big(\alpha\ \wedge\ \beta\big) & \textit{when } x \textit{ is not free in } \alpha
\end{array}$$

Definition 6.7 Every formula is in *Prenex Normal Form* (PNF), if is is of the form:

$$Q_1 x_1 \cdots Q_n x_n\ \varphi,$$

where φ is quantifier-free, and each $Q_i \in \{\forall, \exists\}$.

Theorem 6.8 *Every formula is equivalent to another formula in PNF.*

*Recall that $\models \varphi$ is the abbreviation for $\emptyset \models \varphi$.

2 First-order theories

Definition 6.9

- A set T of sentences is called a *theory*, if it is closed under logical consequences, i.e., for every sentence φ, if $T \models \varphi$, then $\varphi \in T$.
- A theory T is *complete*, if for every sentence φ, either $\varphi \in T$ or $\neg\varphi \in T$.

Definition 6.10

- For a set X of sentences, $\mathrm{Model}(X) := \{\mathcal{A} \mid \mathcal{A} \models X\}$.
- For a set X of sentences, $\mathrm{Cn}(X) := \{\varphi \mid X \models \varphi\}$.
- For a set $\mathcal{K}$ of structures, $\mathrm{Th}(\mathcal{K}) := \{\varphi \mid \varphi$ holds in every structure in $\mathcal{K}\}$.

Theorem 6.11 *For a set $\mathcal{K}$ of structures, and a set X of sentences, the following holds.*

- $\mathcal{K} \subseteq \mathit{Model}(\mathit{Th}(\mathcal{K}))$.
- *$\mathit{Th}(\mathcal{K})$ is a theory.*
- $\mathit{Cn}(X) = \mathit{Th}(\mathit{Model}(X))$.

Definition 6.12 A theory T is *finitely axiomatizable*, if there is a finite set Σ such that $T = \mathrm{Cn}(\Sigma)$.

Remark 6.13 If $\mathrm{Cn}(T)$ is finitely axiomatizable, then there is a finite subset $T_0 \subseteq T$ such that $\mathrm{Cn}(T_0) = \mathrm{Cn}(T)$.

Exercises

(1) Show that $\exists x \forall y\ \varphi \not\models \forall x \exists y\ \varphi$.

That is, give a model $\mathcal{A}$ and a formula φ such that $\mathcal{A} \models \exists x \forall y\ \varphi$. but $\mathcal{A} \not\models \forall x \exists y\ \varphi$.

(2) Give a set $\mathcal{K}$ of sentences such that $\mathcal{K} \neq \mathrm{Model}(\mathrm{Th}(\mathcal{K}))$.

(3) Let $\mathcal{K} = \{\mathcal{A}\}$, i.e., it consists of only one structure $\mathcal{A}$. Prove that $\mathrm{Th}(\mathcal{K})$ is complete.

(4) Give a set $\mathcal{K}$ of structures such that $\mathrm{Th}(\mathcal{K})$ is not complete.

(5) Let T be a complete theory and let $\mathcal{A} \models T$. Prove that for every sentence α, $\mathcal{A} \models \alpha$ if and only if $T \models \alpha$.

We denote by $\mathcal{A} \cong \mathcal{B}$, if $\mathcal{A}$ is isomorphic to $\mathcal{B}$, i.e., there is an isomorphism from $\mathcal{A}$ to $\mathcal{B}$. Two structures $\mathcal{A}$ and $\mathcal{B}$ are *elementarily equivalent*, written as $\mathcal{A} \equiv \mathcal{B}$, if for every sentence φ,

$$\mathcal{A} \models \varphi \quad \text{if and only if} \quad \mathcal{B} \models \varphi.$$

(6) Prove that if $\mathcal{A} \cong \mathcal{B}$, then $\mathcal{A} \equiv \mathcal{B}$.

(7) Let $\mathcal{K}$ be a set of structures such that for every $\mathcal{A}, \mathcal{B} \in \mathcal{K}$, we have $\mathcal{A} \cong \mathcal{B}$. Prove that $\mathrm{Th}(\mathcal{K})$ is complete.

Appendix

The converse of question (6) does not hold in general. That is, $\mathcal{A} \equiv \mathcal{B}$ does not necessarily imply $\mathcal{A} \cong \mathcal{B}$. Consider, for example, the following two structures.

- $\mathcal{R} = (\mathbb{R}, <^{\mathcal{R}})$, where $<^{\mathcal{R}}$ is the standard ordering in $\mathbb{R}$.
- $\mathcal{Q} = (\mathbb{Q}, <^{\mathcal{Q}})$, where $<^{\mathcal{Q}}$ is the standard ordering in $\mathbb{Q}$.

It is known that $\mathcal{R} \equiv \mathcal{Q}$, but $\mathcal{R}$ is *not* isomorphic to $\mathcal{Q}$, since $\mathbb{R}$ is uncountable, but $\mathbb{Q}$ is countable.

In general it is not a trivial matter to determine whether two structures are elementarily equivalent. It usually involves a technique called *Ehrenfeucht-Fraïssé* game, which we will not cover in this course.

Lesson 7: Proof system in first-order logic

Theme: The notion of provability in first-order logic.

1 Proofs in first-order logic

Throughout this note, L is a fixed vocabulary. For a formula α, we denote by $\mathrm{var}(\alpha)$ to be the set of *all* variables in α (both free and quantified).

Let X be a set of formulas and α be a formula (over L). We say that α can be provable from X, or α is derivable from X, denoted by $X \vdash_L \alpha$, if it can be obtained inductively according to the following rules.

Initial Rule (IR):
$$\begin{array}{lll} X \vdash_L \alpha & & \text{if } \alpha \in X \\ X \vdash_L t \approx t & & \text{for every } L\text{-term } t \end{array}$$

Monotonicity Rule (MR):
$$\frac{X \vdash_L \alpha}{Y \vdash_L \alpha} \qquad \text{for every } Y \supseteq X$$

And Combine Rule (ACR):
$$\frac{X \vdash_L \alpha \quad \text{and} \quad X \vdash_L \beta}{X \vdash_L \alpha \wedge \beta}$$

And Split Rule (ASR):
$$\frac{X \vdash_L \alpha \wedge \beta}{X \vdash_L \alpha \quad \text{and} \quad X \vdash_L \beta}$$

Contradiction Rule (CR):
$$\frac{X \vdash_L \alpha \quad \text{and} \quad X \vdash_L \neg\alpha}{X \vdash_L \beta} \qquad \text{for every } \beta$$

Negation Rule (NR):
$$\frac{X, \alpha \vdash_L \beta \quad \text{and} \quad X, \neg\alpha \vdash_L \beta}{X \vdash_L \beta}$$

Specialisation Rule (SR):
$$\frac{X \vdash_L \forall x\ \alpha}{X \vdash_L \alpha[t/x]} \qquad \text{where } [t/x] \text{ is collision-free in } \alpha$$

Generalisation Rule (GR):
$$\frac{X \vdash_L \alpha[y/x]}{X \vdash_L \forall x\ \alpha} \qquad \text{where } y \notin \mathrm{free}(X) \cup \mathrm{var}(\alpha)$$

Equality Rule (ER):
$$\frac{X \vdash_L s \approx t \quad \text{and} \quad X \vdash_L \alpha[s/x]}{X \vdash_L \alpha[t/x]} \qquad \text{where } \alpha \text{ is atomic}$$

We write $X \not\vdash_L \alpha$, if it is not the case that $X \vdash_L \alpha$.

Remark 7.1 When there is no confusion, we will omit writing L, and thus, write only $\vdash$, instead of $\vdash_L$.

We will also follow the writing convention from the proof system in the propositional calculus. We write $\alpha \vdash \alpha$ to denote $\{\alpha\} \vdash \alpha$, whereas $X, \alpha \vdash \beta$ means $X \cup \{\alpha\} \vdash \beta$. As before, $\vdash \alpha$ to denote $\emptyset \vdash \alpha$.

Theorem 7.2 (Finiteness theorem for $\vdash$) *If $X \vdash \alpha$, then there is a* finite *set $X_0 \subseteq X$ such that $X_0 \vdash \alpha$.*

Example 7.3 $\dfrac{X \vdash s \approx t \quad \text{and} \quad X \vdash s \approx t'}{X \vdash t \approx t'}$

1. $X \vdash s \approx t$. (supposition)
2. $X \vdash s \approx t'$. (supposition)

Let $x \notin \text{var}(t')$ and $\alpha := x \approx t'$. So (2) is actually $X \vdash \alpha[s/x]$.

3. $X \vdash \alpha[t/x]$. (Equality Rule on 1 and 2)

$\alpha[t/x]$ is precisely $t \approx t'$.

Example 7.4 $\dfrac{X \vdash s \approx t}{X \vdash t \approx s}$

1. $X \vdash s \approx t$. (supposition)
2. $X \vdash s \approx s$. (Initial rule)
3. $X \vdash t \approx s$. (Example 7.3 on 1 and 2)

Example 7.5 $\dfrac{X \vdash t \approx s \quad \text{and} \quad X \vdash s \approx t'}{X \vdash t \approx t'}$

1. $X \vdash t \approx s$. (supposition)
2. $X \vdash s \approx t'$. (supposition)
3. $X \vdash s \approx t$. (Example 7.4 on 1)
4. $X \vdash t \approx t'$. (Example 7.3 on 3 and 3)

Example 7.6 $\dfrac{X \vdash t_i \approx s}{X \vdash f(t_1, \ldots, t_k) \approx f(t_1, \ldots, t_{i-1}, s, t_{i+1,\ldots,t_k})}$

1. $X \vdash t_i \approx s$. (supposition)
2. $X \vdash f(t_1, \ldots, t_k) \approx f(t_1, \ldots, t_k)$. (Initial rule)

Let $x \notin \text{var}(t_1) \cup \cdots \cup \text{var}(t_k) \cup \text{var}(s)$ and $\alpha := f(t_1, \ldots, t_k) \approx f(t_1, \ldots, t_{i-1}, x, t_{i+1}, \ldots, t_k)$. So (2) is actually $X \vdash \alpha[t_i/x]$.

3. $X \vdash \alpha[s/x]$. (Equality Rule on 1 and 2)

$\alpha[s/x]$ is precisely $f(t_1, \ldots, t_k) \approx f(t_1, \ldots, t_{i-1}, s, t_{i+1,\ldots,t_k})$.

Example 7.7 $\dfrac{X \vdash t_i \approx s \quad \text{and} \quad X \vdash R(t_1, \ldots, t_k)}{X \vdash R(t_1, \ldots, t_{i-1}, s, t_{i+1,\ldots,t_k})}$.

In the following $X \vdash (t_1, \ldots, t_k) \approx (s_1, \ldots, s_k)$ denotes $X \vdash t_i \approx s_i$, for each $i \in \{1, \ldots, k\}$.

Example 7.8 $\dfrac{X \vdash (t_1, \ldots, t_k) \approx (s_1, \ldots, s_k)}{X \vdash f(t_1, \ldots, t_k) \approx f(s_1, \ldots, s_k)}$

Example 7.9 $\dfrac{X \vdash (t_1, \ldots, t_k) \approx (s_1, \ldots, s_k) \quad \text{and} \quad X \vdash R(t_1, \ldots, t_k)}{X \vdash R(s_1, \ldots, s_k)}$.

Lemma 7.10 *Let t be a term, and $x \notin var(t)$. Then, the following holds.*

(a) $\vdash \exists x\ t \approx x$. (Here $\exists x\ t \approx x$ stands for $\neg \forall x\ t \not\approx x$.)

(b) $\vdash \exists x\ x \approx x$. (Here $\exists x\ x \approx x$ stands for $\neg \forall x\ x \not\approx x$.)

Proof. We prove item (a).

1. $\forall x\ t \not\approx x \vdash \forall x\ t \not\approx x$. (Initial rule)
2. $\forall x\ t \not\approx x \vdash (t \not\approx x)[t/x]$. (Specialisation Rule on 1)
3. $\forall x\ t \not\approx x \vdash t \not\approx t$. $\big((t \not\approx x)[t/x] = t \not\approx t\big)$
4. $\forall x\ t \not\approx x \vdash t \approx t$. (Initial Rule)
5. $\forall x\ t \not\approx x \vdash \neg \forall x\ t \not\approx x$. (Contradiction Rule on 3 and 4)
6. $\neg \forall x\ t \not\approx x \vdash \neg \forall x\ t \not\approx x$. (Initial Rule)
7. $\vdash \neg \forall x\ t \not\approx x$. (Negation Rule on 5 and 6)

Part (b) can be proved in a similar manner starting with $\forall x\ x \not\approx x \vdash x \not\approx x$ and $\forall x\ x \not\approx x \vdash x \approx x$.
■

2 Precursors to the soundness of $\vdash$

Proposition 7.11 *Let α be a formula, and $y \notin var(\alpha)$. Then, the following holds.*

- $\alpha[y/x][x/y] = \alpha$.
- $\forall z\ \alpha \equiv \forall y\ \alpha[y/z]$.

Proposition 7.12 *Let $(\mathcal{A}, \mathsf{val})$ be an interpretation. Let t be a term. Suppose that $t^{\mathcal{A}}[\mathsf{val}] = b$.*

(a) For every term s,

$$s[t/x]^{\mathcal{A}}[\mathsf{val}] \quad = \quad s^{\mathcal{A}}[\mathsf{val}[x \mapsto b]].$$

(b) For every term s_1, s_2,

$$(\mathcal{A}, \mathsf{val}[x \mapsto b]) \models s_1 \approx s_2 \quad \text{if and only if} \quad (\mathcal{A}, \mathsf{val}) \models (s_1 \approx s_2)[t/x].$$

(c) For a relation R and terms $s_1, \ldots, s_m$,

$$(\mathcal{A}, \mathsf{val}[x \mapsto b]) \models R(s_1, \ldots, s_m) \quad \text{if and only if} \quad (\mathcal{A}, \mathsf{val}) \models R(s_1, \ldots, s_m)[t/x].$$

Proposition 7.13 *Let $(\mathcal{A}, \mathsf{val})$ be an interpretation. Let α be a formula, and $[t/x]$ be collision-free in α. Suppose $t^{\mathcal{A}}[\mathsf{val}] = b$. Then, $(\mathcal{A}, \mathsf{val}[x \mapsto b]) \models \alpha$ if and only if $(\mathcal{A}, \mathsf{val}) \models \alpha[t/x]$.*

Proof. The proof is by induction on α. The base case is when α is atomic formula, i.e., of the form $s_1 \approx s_2$ or $R(s_1, \ldots, s_n)$. This has been settled in Proposition 7.12 parts (b) and (c).

For the induction step, we have three cases: α is of the form $\neg\beta$, or $\beta \wedge \gamma$, or $\forall z\ \beta$. The first two cases are easy. We consider the case when α is $\forall z\ \beta$.

We first prove the "only if" direction. By definition,

$$(\mathcal{A}, \mathsf{val}[x \mapsto b]) \quad \models \quad \forall z\ \beta \tag{1}$$

if and only if for every $a \in A$,

$$(\mathcal{A}, \mathsf{val}[x \mapsto b][z \mapsto a]) \quad \models \quad \beta \tag{2}$$

Now, $[t/x]$ is collision-free in α, which by definition, t does not contain z and $[t/x]$ is collision-free in β. Since t does not contain z, we have:

$$t^{\mathcal{A}}[\mathsf{val}[z \mapsto a]] \quad = \quad t^{\mathcal{A}}[\mathsf{val}] \quad = \quad b \tag{3}$$

CAUTION: if t contains z, Equation 3 may not hold. That is why we need $[t/x]$ to be collision-free in α.

So by the induction hypothesis on Equation 2, we have that for every $a \in A$:

$$(\mathcal{A}, \mathsf{val}[z \mapsto a]) \quad \models \quad \beta[t/x] \tag{4}$$

This means that $(\mathcal{A}, \mathsf{val}) \models \forall z\ \beta[t/x]$, and therefore,

$$(\mathcal{A}, \mathsf{val}) \quad \models \quad \alpha[t/x] \tag{5}$$

The "if" direction can be proved in a similar manner via (5)$\Rightarrow$(4)$\Rightarrow$(3)$\Rightarrow$(2). ■

Exercises

We are going to show that our proof system is sound, as stated formally below.

> **(Soundness theorem for $\vdash$)** *If $X \vdash \alpha$, then $X \models \alpha$.*

We are going to show that each rule in our proof system is sound.

(1) Prove that the Initial Rule (IR) is sound, i.e., for every set X,
- $X \models \alpha$, for every $\alpha \in X$.
- $X \models t \approx t$, for every term t.

(2) Prove that the Monotonicity Rule (MR) is sound, i.e., for every set X, if $X \models \alpha$, then $Y \models \alpha$, for every $Y \supseteq X$.

(3) Prove that the And Combine Rule (ACR) is sound, i.e., for every set X, for every formulas α and β, if $X \models \alpha$ and $X \models \beta$, then $X \models \alpha \wedge \beta$.

(4) Prove that the And Split Rule (ASR) is sound, i.e., for every set X, for every formulas α and β, if $X \models \alpha \wedge \beta$, then $X \models \alpha$ and $X \models \beta$.

(5) Prove that the Contradiction Rule (CR) is sound, i.e., for every set X, for every formula α, if $X \models \alpha$ and $X \models \neg\alpha$, then $X \models \beta$, for every formula β.

(6) Prove that the Negation Rule (NR) is sound, i.e., for every set X, for every formulas α and β, if $X, \alpha \models \beta$ and $X, \neg\alpha \models \beta$, then $X \models \beta$.

(7) Prove that the Specialisation Rule (SR) is sound, i.e., for every set X, for every formula α, if $X \models \forall x\ \alpha$, and $[t/x]$ is collision-free in α, then $X \models \alpha[t/x]$.

(8) Prove that the Generalisation Rule (GR) is sound, i.e., for every set X, for every formula α, for every variable $y \notin \mathrm{free}(X) \cup \mathrm{var}(\alpha)$, if $X \models \alpha[y/x]$, then $X \models \forall x\ \alpha$.

(9) Prove that the Equality Rule (ER) is sound, i.e., for every set X, for every atomic formula α, for every terms s and t, if $X \models s \approx t$ and $X \models \alpha[s/x]$, then $X \models \alpha[t/x]$.

(10) Finally, conclude that $\vdash$ is sound. That is, for every set X, for every formula α, if $X \vdash \alpha$, then $X \models \alpha$.

Hint: For questions (7)–(9), use Propositions 7.11, 7.12 and 7.13.

Lesson 8: Gödel's completeness theorem*

Theme: Consistent set, Henkin set and the equivalence between the notions of $\vdash$ and $\models$.

1 Consistent sets

Let L be a vocabulary, and let $X \subseteq \text{FO}[L]$. The set X is *inconsistent*, if there is a formula α such that $X \vdash \alpha$ and $X \vdash \neg\alpha$. By the contradiction rule, this also means that X is inconsistent if $X \vdash \beta$, for every formula β.

We say that X is consistent, if X is not inconsistent. It is maximally consistent, if it is consistent and for every set $Y \subseteq \text{FO}[L]$ and $Y \supsetneq X$, Y is inconsistent.

2 Constants elimination

Let c be a constant symbol and z be a variable. For a formula α, we write $\alpha\frac{z}{c}$ to denote the formula obtained by replacing every constant symbol c in α by z. For a set X, we write $X\frac{z}{c}$ to denote the set $\{\alpha\frac{z}{c} \mid \alpha \in X\}$.

Lemma 8.1 *Suppose $X \vdash_L \alpha$. Let c be a constant in L, and L' denote $L - \{c\}$. Then, there is a finite subset $X_0 \subseteq X$ and a variable $z \notin var(X_0) \cup var(\alpha)$,*

$$X_0\frac{z}{c} \quad \vdash_{L'} \quad \alpha\frac{z}{c}.$$

Proof. (Sketch) Suppose $X \vdash_L \alpha$. By the finiteness theorem of $\vdash$, there is a finite set $X_0 \subseteq X$ such that $X_0 \vdash_L \alpha$. Let $z \notin \text{var}(X_0) \cup \text{var}(\alpha)$.

Claim 1 $X_0\frac{z}{c} \vdash_{L'} \alpha\frac{z}{c}$.

The claim can be proved by induction on the length of the proof of $X_0 \vdash_L \alpha$. ∎

Lemma 8.2 *Suppose $X \vdash \alpha[c/x]$ and c does not appear in X and α. Then, $X \vdash \forall x\ \alpha$.*

Proof. Suppose $X \vdash \alpha[c/x]$, where c does not appear in X and α.

By Lemma 8.1, there is a finite subset $X_0 \subseteq X$ such that $X_0\frac{z}{c} \vdash \alpha[c/x]\frac{z}{c}$, where $z \notin \text{var}(X_0) \cup \text{var}(\alpha[c/x])$.

Now, since c does not appear in X, $X_0\frac{z}{c} = X_0$. So,

$$X_0 \quad \vdash \quad \alpha[c/x]\frac{z}{c}.$$

Moreover, c does not appear in α. So $\alpha[c/x]\frac{z}{c} = \alpha[z/x]$. Thus,

$$X_0 \quad \vdash \quad \alpha[z/x].$$

Since z does not appear in X_0 and α, by generalisation rule, we have $X_0 \vdash \forall x\ \alpha$. Lemma 8.2 follows immediately by monotonicity rule. ∎

For a variable $x \in \text{VAR}$ and $\alpha \in \text{FO}[L]$, we define a "new" constant $c_{x,\alpha} \notin L$. We define the following formula $\alpha^x \in \text{FO}[L \cup \{c_{x,\alpha}\}]$.

$$\alpha^x \quad := \quad \neg\forall x\ \alpha \quad \wedge \quad \alpha[c_{x,\alpha}/x]$$

*Similar material can be obtained from Section 3.2 in the textbook *A Concise Introduction to Mathematical Logic* (3rd ed.) by Wolfgang Rautenberg.

Lemma 8.3 *Let L be a vocabulary. Define the set Γ_L of formulas as follows.*[†]

$$\Gamma_L \quad := \quad \{\neg\alpha^x \mid x \in \mathit{VAR} \text{ and } \alpha \in \mathrm{FO}[L]\}$$

If a set X is consistent, then so is $X \cup \Gamma_L$.

Proof. Let X be a consistent set. Suppose to the contrary that $X \cup \Gamma_L$ is inconsistent. That is, there is φ such that

$$X \cup \Gamma_L \vdash \varphi \quad \text{and} \quad X \cup \Gamma_L \vdash \neg\varphi$$

Thus, $X \cup \Gamma_L \vdash \mathtt{F}$, where $\mathtt{F}$ denotes $\varphi \wedge \neg\varphi$. By finiteness theorem, there is a finite subset $X_0 \subseteq X$ such that

$$X_0, \neg\alpha_1^{x_1}, \ldots, \neg\alpha_{n-1}^{x_{n-1}}, \neg\alpha_n^{x_n} \ \vdash \ \mathtt{F}. \tag{1}$$

We can assume that n is minimal in the sense that $X_0, \neg\alpha_1^{x_1}, \ldots, \neg\alpha_i^{x_i} \ \nvdash \ \mathtt{F}$, for every $i < n$. By Contradition Rule on (1),

$$X_0, \neg\alpha_1^{x_1}, \ldots, \neg\alpha_{n-1}^{x_{n-1}}, \neg\alpha_n^{x_n} \ \vdash \ \alpha_n^{x_n}. \tag{2}$$

By Initial Rule and Monotonicity Rule,

$$X_0, \neg\alpha_1^{x_1}, \ldots, \neg\alpha_{n-1}^{x_{n-1}}, \alpha_n^{x_n} \ \vdash \ \alpha_n^{x_n}. \tag{3}$$

By Negation Rule on (2) and (3),

$$X_0, \neg\alpha_1^{x_1}, \ldots, \neg\alpha_{n-1}^{x_{n-1}} \ \vdash \ \alpha_n^{x_n}. \tag{4}$$

Let us denote by $x := x_n$, $\alpha := \alpha_n$ and $c := c_{x,\alpha}$. Thus,

$$X_0, \neg\alpha_1^{x_1}, \ldots, \neg\alpha_{n-1}^{x_{n-1}} \quad \vdash \quad \neg\forall x\ \alpha \quad \wedge \quad \alpha[c_{x,\alpha}/x]. \tag{5}$$

By And Split Rule on (5)

$$X_0, \neg\alpha_1^{x_1}, \ldots, \neg\alpha_{n-1}^{x_{n-1}} \quad \vdash \quad \neg\forall x\ \alpha \tag{6}$$

$$X_0, \neg\alpha_1^{x_1}, \ldots, \neg\alpha_{n-1}^{x_{n-1}} \quad \vdash \quad \alpha[c_{x,\alpha}/x]. \tag{7}$$

Since $c_{x,\alpha}$ does not appear in X_0 and in each of $\alpha_i^{x_i}$, by Lemma 8.2 on (7), we have

$$X_0, \neg\alpha_1^{x_1}, \ldots, \neg\alpha_{n-1}^{x_{n-1}} \quad \vdash \quad \forall x\ \alpha. \tag{8}$$

But (6) and (8) imply that $X_0, \neg\alpha_1^{x_1}, \ldots, \neg\alpha_{n-1}^{x_{n-1}}$ is inconsistent, which contradicts the assumption that n is minimal. ■

[†]Note that Γ_L is a set of formulas over the vocabulary $L \cup \{c_{x,\alpha} \mid \alpha \in \mathrm{FO}[L],\ x \in \mathrm{VAR}\}$.

3 Henkin sets

Definition 8.4 A set $X \subseteq \mathrm{FO}[L]$ is called a *Henkin* set, if it satisfies the following properties.

(H1) $X \vdash \neg\alpha$ if and only if $X \nvdash \alpha$. Or, equivalently, $X \vdash \alpha$ if and only if $X \nvdash \neg\alpha$.

(H2) $X \vdash \forall x\ \alpha$ if and only if $X \vdash \alpha[c/x]$ for every constant $c \in L$.

Proposition 8.5 *If X is a Henkin set over vocabulary L, then for each L-term t, there is a constant $c \in L$ such that $X \vdash t \approx c$.*

Proof. Let X be a Henkin set over vocabulary L. By Example 7.10, we have $\vdash \neg\forall x\ t \not\approx x$, when $x \notin \mathrm{var}(t)$. By Monotonicity Rule, $X \vdash \neg\forall x\ t \not\approx x$. Since X is Henkin, by (H1), we have

$$X \ \nvdash\ \forall x\ t \not\approx x.$$

By (H2), for some constant c,

$$X \ \nvdash\ t \not\approx c.$$

By (H1),

$$X \ \vdash\ t \approx c.$$

This completes our proof of Proposition 8.5. ■

Lemma 8.6 *For every consistent set $X \subseteq \mathrm{FO}[L]$, there is a Henkin set $Y \supseteq X$, where $Y \subseteq \mathrm{FO}[L \cup C]$, for some set C of "new" constants not in L.*

Proof. Let $X \subseteq \mathrm{FO}[L]$ be a consistent set. For each integer $i \geqslant 0$, we define the sets Γ_i, Δ_i, L_i and C_i as follows.

$$\Delta_0 \ :=\ X \qquad L_0 \ :=\ L \qquad C_0 \ :=\ \emptyset \qquad \Gamma_0 \ :=\ \emptyset$$

For each $i > 0$,

$$\begin{array}{rcl} C_i & := & \{c_{x,\alpha} \mid x \in \mathrm{VAR} \text{ and } \alpha \in \mathrm{FO}[L_{i-1}]\} \\ L_i & := & L_{i-1}\ \cup\ C_i \\ \Gamma_i & := & \left\{ \neg\alpha^x \ \middle|\ \alpha^x := \neg\forall x\ \alpha\ \wedge\ \alpha[c_{x,\alpha}/x] \text{ where } \alpha \in \mathrm{FO}[L_{i-1}] \text{ and } c_{x,\alpha} \in C_i \right\} \\ \Delta_i & := & \Delta_{i-1}\ \cup\ \Gamma_i \end{array}$$

Now, let $\Delta := \bigcup_{i>0} \Delta_i$ and $L' := \bigcup_{i>0} L_i$.

Consider the poset $(\mathcal{F}, \subseteq)$, where

$$\mathcal{F} \ :=\ \{Z \mid \Delta \subseteq Z \subseteq \mathrm{FO}[L'] \text{ and } Z \text{ is consistent}\}.$$

Claim 2 *Let K be a chain in $(\mathcal{F}, \subseteq)$. Then, $\bigcup K$ is consistent.*

Proof. (of Claim 2) Proceeds like the one in propositional calculus. ■

By Zorn's lemma, there is a maximal consistent set $Y \in \mathcal{F}$. We will now show that that Y is Henkin.

Claim 3 *Y satisfies (H1), i.e., $Y \vdash \neg\alpha$ if and only if $Y \nvdash \alpha$.*

Proof. (of Claim 3) For the "only if" direction, suppose $Y \vdash \neg\alpha$. Since Y is consistent, $Y \nvdash \alpha$.

For the "if" direction, suppose $Y \nvdash \alpha$, which means that $\alpha \notin Y$. Since Y is maximal, $Y \cup \{\alpha\}$ is not consistent. So,

$$Y, \alpha \quad \vdash \quad \neg\alpha.$$

By Initial Rule,

$$Y, \neg\alpha \quad \vdash \quad \neg\alpha.$$

By Negation Rule,

$$Y \quad \vdash \quad \neg\alpha.$$

This completes our proof of Claim 3. ■

Claim 4 *Y satisfies (H2), i.e., $Y \vdash \forall x\ \alpha$ if and only if $Y \vdash \alpha[c/x]$ for every constant $c \in L'$.*

Proof. (of Claim 4) For the "only if" direction, suppose $Y \vdash \forall x\ \alpha$. Let $c \in L'$. Now $[c/x]$ is collision-free in α. By Specialisation Rule, $Y \vdash \alpha[c/x]$.

For the "if" direction, suppose $Y \vdash \alpha[c/x]$ for every constant $c \in L'$. Let $\alpha \in \mathrm{FO}[L_n]$. So, in particular for $c \in C_n$,

$$Y \quad \vdash \quad \alpha[c/x]. \tag{9}$$

Now, suppose to the contrary that $Y \nvdash \forall x\ \alpha$. By (H1),

$$Y \quad \vdash \quad \neg\forall x\ \alpha. \tag{10}$$

By And Combine Rule on (9) and (10),

$$Y \quad \vdash \quad \neg\forall x\ \alpha \quad \wedge \quad \alpha[c/x] \tag{11}$$

Note that the right side of (11) is simply α^x. So, $Y \vdash \alpha^x$.

However, $\neg\alpha^x \in Y$. So, $Y \vdash \neg\alpha^x$, which means Y is inconsistent. This contradicts the fact that $Y \in \mathcal{F}$, which means that Y is consistent. Therefore, $Y \vdash \forall x\ \alpha$, and this completes the proof of Claim 4. ■

Claims 3 and 4 state that Y is Henkin, and this completes our proof of Lemma 8.6. ■

Lemma 8.7 *Every Henkin set is satisfiable.*

Proof. This will be proved in the exercise. ■

4 The completeness theorem for FO

Theorem 8.8 (Gödel's completeness theorem) *$X \models \alpha$ if and only if $X \vdash \alpha$.*

Proof. The "if" direction is the soundness theorem. For the "only if" direction, we show if $X \nvdash \alpha$, then $X \not\models \alpha$. Suppose $X \nvdash \alpha$. Then, $X \cup \{\neg\alpha\}$ is consistent[‡]. By Lemmas 8.6 and 8.7, there is a Henkin set $Y \supseteq X \cup \{\neg\alpha\}$ and Y is satisfiable. This means $X \cup \{\neg\alpha\}$ is satisfiable, and therefore, $X \not\models \alpha$. ■

[‡]Lemma 3.2 can be easily proved for a set X of first-order formulas.

Exercises

In questions (1)-(8) below we are going to show that every Henkin set is satisfiable. Let Y be a Henkin set and C be the set of all the constants that appear in Y. We associate each constant $c \in C$ with an element a_c. Different constants $c \neq c'$ are associated with different elements $a_c \neq a_{c'}$. Consider the set U.

$$U \quad := \quad \{a_c \mid c \in C\}$$

Define a relation $\sim$ on U as follows.

$$a_c \sim a_{c'} \quad \text{if and only if} \quad Y \vdash c \approx c'$$

(1) Prove that $\sim$ is an equivalence relation on U. (Note this is not a trivial question.)

Let $[a_c]$ denote the equivalence class of a_c w.r.t. $\sim$. The structure $\mathcal{A} = (A, R_1, \ldots, f_1, \ldots, c_1, \ldots)$ is defined as follows.

- $A = \{[a_c] \mid a_c \in U\}$.
- $c_i = [a_{c_i}]$.
- $R_i([a_{c_1}], \ldots, [a_{c_n}])$ if and only if $Y \vdash R(c_1, \ldots, c_n)$.
- $f_i([a_{c_1}], \ldots, [a_{c_n}]) = [a_c]$, if $Y \vdash f_i(c_1, \ldots, c_n) \approx c$.

(2) Prove that the definition of R_i is well defined.

That is, if $([a_{c_1}], \ldots, [a_{c_n}]) = ([a_{d_1}], \ldots, [a_{d_n}])$, then,

$$Y \vdash R(c_1, \ldots, c_n) \quad \text{if and only if} \quad Y \vdash R(d_1, \ldots, d_n)$$

(3) Prove that the definition of f_i is well defined.

That is,

- for every $c_1, \ldots, c_n \in C$, there is c such that $Y \vdash f_i(c_1, \ldots, c_n) \approx c$, and
- if $([a_{c_1}], \ldots, [a_{c_n}]) = ([a_{d_1}], \ldots, [a_{d_n}])$, then $f_i([a_{c_1}], \ldots, [a_{c_n}]) = f_i([a_{d_1}], \ldots, [a_{d_n}])$.

Consider the following valuation $\mathsf{val} : \text{VAR} \to A$, where $\mathsf{val}(x) = [a_c]$, where $Y \vdash x \approx c$.

(4) Prove that val is well defined.

(5) Prove that for every term t, if $Y \vdash t \approx c$, then $t^{\mathcal{A}}[\mathsf{val}] = [a_c]$.

Next, we will show that Y is satisfiable, i.e., $(\mathcal{A}, \mathsf{val}) \models \alpha$, for every $\alpha \in Y$.

(6) Prove that $(\mathcal{A}, \mathsf{val}) \models s \approx t$, for every atomic formula $s \approx t \in Y$.

(7) Prove that $(\mathcal{A}, \mathsf{val}) \models R(s_1, \ldots, s_n)$, for every atomic formula $R(s_1, \ldots, s_n) \in Y$.

(8) Prove that $(\mathcal{A}, \mathsf{val}) \models \alpha$, for every $\alpha \in Y$, and hence, Y is satisfiable.

Compactness theorem states that X is satisfiable if and only if X is finitely satisfiable.

(9) Use the completeness theorem to prove the compactness theorem for FO.

Lesson 9: Löwenheim-Skolem theorem and categorical sets

Theme: Cardinalities of first-order structures.

1 Cardinal numbers

- Two sets A *and* B *have the same cardinality*, if there is a bijection from A to B, denoted by $|A| = |B|$.
- In the same spirit, $|A| \leqslant |B|$, if there is an injective function from A to B.
- $|A| < |B|$, if $|A| \leqslant |B|$ and $|A| \neq |B|$.

For $i \in \{0, 1, 2, \ldots\}$, we define $\aleph_i$ and $\beth_i$ as follows.

- Both $\aleph_0$ and $\beth_0$ denote $\mathbb{N}$.
- For each $i \geqslant 1$, $\aleph_i$ denotes the minimal set such that $|\aleph_i| > |\aleph_{i-1}|$.
- For each $i \geqslant 1$, $\beth_i$ denotes $2^{\beth_{i-1}}$.

Abusing the notation, we will often regard each $\aleph_i$ and $\beth_i$ as "cardinalities." So, when we write $A = \aleph_i$ and $A = \beth_i$, we mean $|A| = |\aleph_i|$ and $|A| = |\beth_i|$, respectively. Likewise, such abuse also applies for $<$ and $\leqslant$ comparisons.

Theorem 9.1 (Cantor's theorem) $|A| < |2^A|$, *for every set* A.

Cantor's theorem implies that the sequence $\beth_0, \beth_1, \beth_2, \ldots$ will never end, which in turn implies that the sequence $\aleph_0, \aleph_1, \aleph_2, \ldots$ will also never end. The so called *Continuum Hypothesis* (CH) states the following.

$$\aleph_1 \quad = \quad \beth_1$$

2 Löwenheim-Skolem theorem

Theorem 9.2 (Löwenheim-Skolem theorem) *If* $X \subseteq \mathrm{FO}[L]$ *is satisfiable, and* L *is countable, then* X *is satisfied by a countable structure.*

Theorem 9.3 (Downward Löwenheim-Skolem theorem) *If* $X \subseteq \mathrm{FO}[L]$ *is satisfiable, and* L *is of cardinality* λ*, then* X *is satisfied by a structure with cardinality* $\leqslant \lambda$.

Theorem 9.4 (Upward Löwenheim-Skolem-Tarski theorem) *If* $X \subseteq \mathrm{FO}[L]$ *is satisfiable, and* L *is of cardinality* λ*, then for every cardinal number* $\kappa \geqslant \lambda$*, there is a structure with cardinality* κ *that satisfies* X.

Corollary 9.5

(a) Let $X \subseteq \mathrm{FO}[L]$*, where* L *is countable. If* X *has an infinite model, then* X *has models of every infinite cardinality.*

(b) Let $\mathcal{A}$ *be an infinite structure for a countable vocabulary* L*. Then, for every infinite cardinal* λ*, there is a structure* $\mathcal{B}$ *of cardinality* λ*, such that* $\mathcal{A} \equiv \mathcal{B}$.

3 Categorical sets

A set X is *categorical*, if every two models of X is isomorphic.

Proposition 9.6 *If X has an infinite model, then X is not categorical.*

A theory T is $\aleph_0$-*categorical*, if all infinite countable models of T are isomorphic. A theory T is κ-*categorical*, if all models of T of cardinality κ are isomorphic.

Theorem 9.7 (Łoś-Vaught Test) *Let T be a theory over a countable vocabulary. Assume that T has no finite models.*

(a) If T is $\aleph_0$-categorical, then T is complete.

(b) If T is κ-categorical for some infinite cardinal κ, then T is complete.

4 The ZFC system

The ZFC system (**Z**ermelo-**F**raenkel-Axiom of **C**hoice) is a set of axioms that describe mathematics being founded entirely on set theory. The vocabulary has only *one* binary relation ε, which intuitively represents the standard relation $\in$.

The ZFC system consists of the following axioms.

Extensionality axiom: $\forall x\ \forall y\ \Big(\forall z\big(z\ \varepsilon\ x \leftrightarrow z\ \varepsilon\ y\big)\ \rightarrow\ x \approx y\Big)$.

Intuitively, this means that if x and y have the same members, then x and y are the same.

Separation axioms: $\forall x_1 \cdots \forall x_n\ \forall x\ \exists y\ \forall z\Big(z\ \varepsilon\ y \quad \leftrightarrow \quad \big(z\ \varepsilon\ x\ \wedge\ \varphi(z, x_1, \ldots, x_n)\big)\Big)$.

The formula φ is over the vocabulary $\{\varepsilon\}$. Intuitively, it means that for a set x, and a "property" φ, there is a set y that contains precisely the elements in x that satisfies φ.

Pairing axiom: $\forall x\ \forall y\ \exists z\ \forall w\ \Big(w\ \varepsilon\ z \quad \leftrightarrow \quad \big(w \approx x\ \vee\ w \approx y\big)\Big)$.

Intuitively, it means that for every two sets x and y, the set $\{x, y\}$ exists.

Union axiom: $\forall x\ \exists y\ \forall z\ \Big(z\ \varepsilon\ y \quad \leftrightarrow \quad \exists w\ \big(w\ \varepsilon\ x\ \wedge\ z\ \varepsilon\ w\big)\Big)$.

Intuitively, it means that for every set x, the set $\bigcup x$ exists.

Power set axiom: $\forall x\ \exists y\ \forall z\ \Big(z\ \varepsilon\ y \quad \leftrightarrow \quad \forall w\ \big(w\ \varepsilon\ z\ \rightarrow\ w\ \varepsilon\ x\big)\Big)$.

Intuitively, it means that for every set x, the set 2^x exists.

Infinity axiom: $\exists x\ \Big(\underline{\emptyset\ \varepsilon\ x}\ \wedge\ \forall y\ \big(y\ \varepsilon\ x \quad \rightarrow \quad \underline{y \cup \{y\}\ \varepsilon\ x}\big)\Big)$

Intuitively, it means that there is an infinite set containing $\hat{0}, \hat{1}, \hat{2}, \ldots$, where $\hat{0}$ stands for $\emptyset$, $\hat{1}$ stands for $\{\emptyset\}$, and $\hat{n} = \{\hat{1}, \ldots, \widehat{n-1}\}$.

Note that both $\underline{\emptyset\ \varepsilon\ x}$ and $\underline{y \cup \{y\}\ \varepsilon\ x}$ are abbreviations, where $\underline{\emptyset\ \varepsilon\ x}$ represents "$\emptyset \in x$," i.e., $\exists y\big(\forall z\ \neg(z\ \varepsilon\ y)\ \wedge\ y\ \varepsilon\ x\big)$, and $\underline{y \cup \{y\}\ \varepsilon\ x}$ represents "$y \cup \{y\} \in x$," which can be written in a similar manner.

Replacement axioms: $\forall x_1 \cdots \forall x_n$

$$\forall x\ \exists^{=1} y\ \varphi(x, y, x_1, \ldots, x_n) \quad \rightarrow \quad \forall u\ \exists v\ \forall y\ \Big(y\ \varepsilon\ v\ \leftrightarrow\ \exists x\ \big(\varphi(x, y, x_1, \ldots, x_n)\ \wedge\ x\ \varepsilon\ u\big)\Big)$$

Intuitively, this means that if for parameters $x_1, \ldots, x_n$, the formula $\varphi(x, y, x_1, \ldots, x_n)$ defines a map $x \mapsto y$, then the range of a set is again a set.

Axiom of choice: $\forall x$

$$\Big(\underline{\emptyset \ \not\varepsilon\ x} \ \wedge\ \forall u \forall v \begin{pmatrix} u\ \varepsilon\ x\ \wedge\ v\ \varepsilon\ x\ \wedge\ u \not\approx v \\ \rightarrow\ \underline{u \cap v \approx \emptyset} \end{pmatrix} \Big) \ \rightarrow\ \exists y\, \forall w \Big(w\ \varepsilon\ x\ \rightarrow\ \exists^{=1} z \big(\underline{z\ \varepsilon\ w \cap y} \big) \Big)$$

This states axiom of choice. As before, those underline represent abbreviations of first-order formula describing their respective intuitive meanings.

Remark 9.8 Assuming the consistency of ZFC, the following holds.

- ZFC + CH is consistent (Gödel 1940).
- ZFC + ¬CH is consistent (Cohen 1963).

That is, both CH and its negation cannot be proved from ZFC, provided that ZFC is consistent.

5 Skolem paradox

It is generally accepted that ZFC is consistent, although there is no way to prove it. In the following we are going to show an application of Löwenheim-Skolem theorem that yields a seemingly absurd result, called *Skolem* paradox.

Assuming its consistency, by Löwenheim-Skolem theorem, ZFC has a countable structure $\mathcal{A} = (A, \varepsilon^{\mathcal{A}})$. By the infinity axiom, there is an element $x \in A$ such that x is an infinite set. By power set axiom, $2^x \in A$. Now, by Cantor's theorem, we know that 2^x is uncountable. However, since A is countable, the set of elements related to 2^x (by relation ε) must be countable too (since they all must come from A). Does this mean that Cantor's theorem and Löwenheim-Skolem theorem contradict each other? Or, that ZFC is inconsistent?

Lesson 10: Gödel's incompleteness theorem, part. 1*

Theme: Robinson arithmetic and its arithmetization.

In this lesson and the next, we are only dealing with logic over vocabulary $\{\tilde{0}, \texttt{Succ}, +, \cdot\}$, where $\tilde{0}$ is a constant symbol intended to represent the number zero; Succ is a unary function intended to represent +1, i.e., $\texttt{Succ}(x) = x + 1$; and finally, + and $\cdot$ are intended to represent the standard addition and multiplication operands.

1 Robinson arithmetic

Robinson's arithmetic is a theory Q derived from the following axioms.

(Q1) $\forall x \ \big(\texttt{Succ}(x) \not\approx 0\big)$.

(Q2) $\forall x \forall y \ (\texttt{Succ}(x) \approx \texttt{Succ}(y) \ \rightarrow \ x \approx y)$.

(Q3) $\forall x \ \big(x \not\approx \tilde{0} \ \rightarrow \ \exists y \ x \approx \texttt{Succ}(y)\big)$.

(Q4) $\forall x \ \big(x + \tilde{0} \approx x\big)$.

(Q5) $\forall x \forall y \ \big(x + \texttt{Succ}(y) \approx \texttt{Succ}(x + y)\big)$.

(Q6) $\forall x \ \big(x \cdot \tilde{0} \approx \tilde{0}\big)$.

(Q7) $\forall x \forall y \ \big(x \cdot \texttt{Succ}(y) \approx (x \cdot y) + x\big)$.

Note that by its definition, Q is a finitely axiomatizable theory, and that Q is a *proper* subtheory of Th($\mathcal{N}$), where $\mathcal{N}$ is the standard structure $\mathcal{N} = (\mathbb{N}, 0, \texttt{Succ}, +, \cdot)$. What we call *number theory* usually refers to Th($\mathcal{N}$). Note that Th($\mathcal{N}$) is much stronger than Q. For example, $\forall x \ x \not\approx \texttt{Succ}(x)$ is not provable in Q.

In the following, we will often write $x \leqslant y$ as an abbreviation for $\exists z \ x + z \approx y$, and $x < y$ for $x \leqslant y \ \wedge \ x \not\approx y$.

Remark 10.1 For the rest of this lesson and the next, the proof system will always be in a theory $T \supseteq \mathsf{Q}$, with the sentences (Q1)–(Q7) above being included as axioms of T.

2 Arithmetization

We denote the set $\mathsf{Symb} = \{\neg, \wedge, \forall, (,), \approx, \tilde{0}, \texttt{Succ}, +, \cdot, x_0, x_1, x_2, \ldots\}$ In principle, we can assume that every formula is a string with symbols from Symb, and every proof is a sequence of formulas with a comma in between two formulas.

In this section we are going to see how to encode a formula φ as a number, and hence, a proof as a number too. For this purpose, we assign each symbol $\mathsf{s} \in \mathsf{Symb} \cup \{\mathbf{,}\}$ a number $\sharp\mathsf{s}$ as follows.

s	$\neg$	$\wedge$	$\forall$	(	)	$\approx$	$\tilde{0}$	Succ	+	$\cdot$	,	x_0	x_1	x_2	$\cdots$
$\sharp$s	1	2	3	4	5	6	7	8	9	10	11	12	13	14	$\cdots$

Let $\{p_0, p_1, \ldots\}$ be the set of all prime numbers with $p_0 < p_1 < \cdots$.

*Some of the material is taken from the textbook *A Concise Introduction to Mathematical Logic* (3rd ed.) by Wolfgang Rautenberg.

For a string $\mathsf{str} = \mathsf{s}_0 \cdots \mathsf{s}_n$ with each symbol s_i coming from $\mathsf{Symb} \cup \{,\}$, the *Gödel number* of str, denoted by $\sharp\mathsf{str}$ is the number:

$$\sharp\mathsf{str} \quad := \quad p_0^{\sharp\mathsf{s}_0} p_1^{\sharp\mathsf{s}_1} \cdots p_n^{\sharp\mathsf{s}_n}$$

The Gödel numbers of a formula φ and a proof ξ are defined as $\sharp\varphi$ and $\sharp\xi$, respectively, where φ and ξ are viewed as a string of symbols coming from $\mathsf{Symb} \cup \{,\}$.

Remark 10.2

- We can write a computer program IsFormula for the following task.
  - **Input:** A positive number N.
  - **Output:** Output `True`, if N "represents" a formula, i.e., N is the Gödel number of a formula. Otherwise, output `False`.

  Likewise, we can write a program IsSentence that checks whether an input number N represents a sentence.
- We can write a computer program $\text{IsProof}_{\mathsf{Q}}$ for the following task.
  - **Input:** A positive number N.
  - **Output:** Output `True`, if N represents a proof in Q. Otherwise, output `False`.
- We can write a computer program $\text{IsProofOf}_{\mathsf{Q}}$ for the following task.
  - **Input:** Two positive numbers N and M.
  - **Output:** Output `True`, if N represents a proof, M represents a formula, and N is a proof of M in Q. Otherwise, output `False`.

Definition 10.3 Let T be a theory such that $T = \mathrm{Cn}(\Sigma)$. We say that T is *recursively axiomatizable*, if there is a computer program IsAxiom_T for the following task.

- **Input:** A positive number N.
- **Output:** Output `True`, if N represents an axiom in T, i.e., N represents a sentence Σ. Otherwise, output `False`.

Remark 10.4

- We can write a computer program IsProofOf_T for the following task.
  - **Input:** Two positive numbers N and M.
  - **Output:** Output `True`, if N represents a proof in T, M represents a formula, and N is a proof of M in T. Otherwise, output `False`.

3 A sketch proof of the incompleteness theorem

Gödel's incompleteness theorem states that *for every consistent and recursively axiomatizable theory $T \supseteq \mathsf{Q}$, there is a sentence Ψ such that neither Ψ nor $\neg\Psi$ are provable in T.*

For an integer $N \geqslant 0$, let $\underline{N}$ denote the following term:

$$\underline{N} \quad := \quad \underbrace{\texttt{Succ}\cdots\texttt{Succ}}_{N \text{ times}}(\tilde{0})$$

Now, suppose that instead of being a computer program, the boolean function $\textsc{IsProofOf}_T(y, x)$ is a first-order formula that indicates y is a proof of x in T. So, for every sentence φ,

$$T \vdash \varphi \quad \textit{if and only if} \quad T \vdash \exists y \ \textsc{IsProofOf}_T(y, \underline{\sharp\varphi}). \tag{1}$$

Consider a sentence Ψ such that

$$T \quad \vdash \quad \Psi \leftrightarrow \Big(\forall y \neg \ \textsc{IsProofOf}_T(y, \underline{\sharp\Psi})\Big) \tag{2}$$

which is an abbreviation for:

$$T \quad \vdash \quad \Psi \rightarrow \Big(\forall y \neg \ \textsc{IsProofOf}_T(y, \underline{\sharp\Psi})\Big) \tag{3}$$

$$T \quad \vdash \quad \Big(\forall y \neg \ \textsc{IsProofOf}_T(y, \underline{\sharp\Psi})\Big) \rightarrow \Psi \tag{4}$$

From Equation (4), we can derive:[†]

$$T \quad \vdash \quad \neg\Psi \rightarrow \neg\Big(\forall y \neg \ \textsc{IsProofOf}_T(y, \underline{\sharp\Psi})\Big) \tag{5}$$

We now argue that neither $T \vdash \Psi$ nor $T \vdash \neg\Psi$.

- Suppose $T \vdash \Psi$.

  Applying modus ponens on $T \vdash \Psi$ and Equation (3), we have:[‡]

  $$T \quad \vdash \quad \forall y \neg \ \textsc{IsProofOf}_T(y, \underline{\sharp\Psi})$$

  which by Equation (1), means Ψ is not provable in T, contradicting supposition $T \vdash \Psi$.

- Suppose $T \vdash \neg\Psi$.

  Applying modus ponens on $T \vdash \neg\Psi$ and Equation (5),

  $$T \quad \vdash \quad \neg \ \forall y \neg \ \textsc{IsProofOf}_T(y, \underline{\sharp\Psi})$$

  which is equivalent to

  $$T \quad \vdash \quad \exists y \ \textsc{IsProofOf}_T(y, \underline{\sharp\Psi}).$$

  By Equation (1), it means $T \vdash \Psi$, contradicting the consistency of T.

Therefore, neither Ψ nor $\neg\Psi$ are provable in T, hence the incompleteness of T.

In this lesson and the next, we focus on the following two tasks in order to complete our proof above.

(a) Find the formula for $\textsc{IsProofOf}_T(y, x)$ using the vocabulary $\{\tilde{0}, \texttt{Succ}, +, \cdot\}$.

(b) Find the statement Ψ.

[†]Recall that in Lesson 4 we show if $X \vdash \alpha \rightarrow \beta$, then $X \vdash \neg\beta \rightarrow \neg\alpha$, which is called contrapositive.

[‡]Recall that in Lesson 4 we show if $X \vdash \alpha \rightarrow \beta$ and $X \vdash \alpha$, then $X \vdash \beta$, which is called modus ponens.

Appendix: The formal definition of recursive functions

We will formalize the notion of *recursive functions*, which are equivalent to the notion of computable functions. Recall that $\mathbb{N} = \{0, 1, 2, \ldots\}$. Let $\mathbf{F}_n$ be the set of all functions from $\mathbb{N}^n$ to $\mathbb{N}$, and let $\mathbf{F} := \bigcup_{n \geqslant 1} \mathbf{F}_n$.

μ-recursive functions, or shortly, recursive functions, are functions that are built inductively as follows.

- Base case: All three kinds of functions below are recursive.

  Constant function: $f(v_1, \ldots, v_n) = 0$.
  Successor function (on the i-component): $f(v_1, \ldots, v_n) = \mathtt{Succ}(v_i)$.
  Projection function (to the i-component): $f(v_1, \ldots, v_n) = v_i$.

- Induction step: All the functions built up from recursive functions using one of the rules below are recursive functions.

  Composition (Oc). If $h \in \mathbf{F}_m$ and $g_1, \ldots, g_m \in \mathbf{F}_n$ are recursive, then the following function f is also recursive. For every $\bar{a} = (a_1, \ldots, a_n) \in \mathbb{N}^n$,

  $$f(\bar{a}) \quad := \quad h(g_1(\bar{a}), \ldots, g_m(\bar{a})).$$

  We usually write $h[g_1, \ldots, g_m]$ to denote the function f constructed above.

  Primitive recursion (Op). If $g \in \mathbf{F}_n$ and $h \in \mathbf{F}_{n+2}$ are recursive functions, then so is $f \in \mathbf{F}_{n+1}$, defined as follows. For every $\bar{a} = (a_1, \ldots, a_n) \in \mathbb{N}^n$,

  $$\begin{aligned} f(\bar{a}, 0) \quad &:= \quad g(\bar{a}) \\ f(\bar{a}, \mathtt{Succ}(b)) \quad &:= \quad h(\bar{a}, b, f(\bar{a}, b)) \end{aligned}$$

  μ operation (Oμ). Let $g \in \mathbf{F}_{n+1}$ be such that for every $\bar{a} \in \mathbb{N}^n$, there is $b \in \mathbb{N}$, where $g(\bar{a}, b) = 0$. If g is computable, then so is the following function f. For every $\bar{a} = (a_1, \ldots, a_n) \in \mathbb{N}^n$,

  $$f(\bar{a}) \quad := \quad \text{the smallest } b \text{ such that } g(\bar{a}, b) = 0$$

  We write $f(\bar{a}) := \mu b[g(\bar{a}, b) = 0]$ to denote the function f as constructed above.

A recursive function obtained without using the $\mathbf{O}\mu$ rule is called a *primitive recursive* function.

Example 10.5

- The function $f_{add}(a, b) = a + b$ is recursive by an application of **Op** rule.

  $$f_{add}(a, 0) \ := \ a \qquad \text{and} \qquad f_{add}(a, \mathtt{Succ}(b) \ := \ \mathtt{Succ}(f_{add}(a, b))$$

- The functions $f_{mul}(a, b) = a \cdot b$ and $f_{exp}(a, b) = a^b$ are recursive.

  $$\begin{aligned} f_{mul}(a, 0) \ &:= \ 0 \qquad \text{and} \quad f_{mul}(a, \mathtt{Succ}(b)) \ := \ f_{add}(b, f_{mul}(a, b)) \\ f_{exp}(a, 0) \ &:= \ \mathtt{Succ}(0) \quad \text{and} \quad f_{exp}(a, \mathtt{Succ}(b)) \ := \ f_{mul}(a, f_{exp}(a, b)) \end{aligned}$$

- The function $f_{abs}(a, b) := |a - b|$ is recursive.
- The function $f_{div}(a, b) := 0$, if b divides a, and 1, otherwise, is recursive.

- The function $f_{prime}(n) := p_n$, where p_n is the n^{th} prime number, is recursive.

Theorem 10.6 (Church-Turing thesis) *If a function f is computable (by a "computer program"), then it is also (i) computable in λ-calculus; (ii) computable by a Turing machine; (iii) μ-recursive.*

In fact, the notions of λ-calculus, Turing machines, and μ-recursive are all equivalent. That is, a function is computable in λ-calculus if and only if it is computable by a Turing machine if and only if it is μ-recursive.

In his original paper,[§] Gödel showed the following.

- An explicit construction of the primitive recursive function for $\textsc{IsProofOf}(x, y)$ as specified in Remark 10.4.
- For every primitive recursive function $f : \mathbb{N}^n \to \mathbb{N}$, there is a formula $\alpha(x_1, \ldots, x_n, y)$ over vocabulary $\{\texttt{Succ}, +, \cdot, \tilde{0}\}$ such that

$$f(a_1, \ldots, a_n) = b \quad \text{if and only if} \quad T \vdash \alpha(\underline{a_1}, \ldots, \underline{a_n}, \underline{b}) \tag{6}$$

An explicit formula for $\textsc{IsProofOf}$ is conceptually not difficult, but long and tedious. In this class, having convinced ourselves that we can write a computer program for $\textsc{IsProofOf}(x, y)$, we can invoke Church-Turing thesis to arrive at the conclusion that $\textsc{IsProofOf}(x, y)$ is recursive. On the other hand, converting a recursive function f to a formula α as specified in Equation (6) involves a very nice piece of mathematics,[¶] and this will be our focus in our next lesson.

[§]Kurt Gödel, Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I (On formally undecidable propositions of Principia Mathematica and related systems I), *Monatshefte für Mathematik and Physik*, 38:173–198 (1931).

[¶]To be exact, expressing **Oc** and **O**μ rules in formulas over $\{\tilde{0}, \texttt{Succ}, +, \cdot\}$ is not difficult. The main difficulty is in expressing the **Op**.

Lesson 11: Gödel's incompleteness theorem, part. 2

Theme: Representability of recursive functions, fixed point lemma and Gödel's first incompleteness theorem.

1 Some preliminary results on Robinson's arithmetic Q

Recall that all our formulas are over the vocabulary $L_{ar} = \{\tilde{0}, \texttt{Succ}, +, \cdot\}$, and that for every integer $n \geqslant 0$, we write $\underline{n}$ to denote the term $\texttt{Succ}^n(\tilde{0})$, i.e., applying Succ on $\tilde{0}$ for n number of times. For a vector $\bar{a} = (a_1, \ldots, a_n)$ of integers, we will write $\underline{\bar{a}}$ to denote $(\underline{a_1}, \ldots, \underline{a_n})$.

By a straightforward induction on n and m, it is not that difficult to show that for every integers $n, m \geqslant 0$, the following holds.

(C1) $\mathsf{Q} \vdash (\texttt{Succ}(\underline{m}) + \underline{n}) \approx (\underline{m} + \texttt{Succ}(\underline{n}))$.

(C2) $\mathsf{Q} \vdash (\underline{m} + \underline{n}) \approx \underline{m + n}$.

(C3) $\mathsf{Q} \vdash (\underline{m} \cdot \underline{n}) \approx \underline{m \cdot n}$.

(C4) $\mathsf{Q} \vdash \underline{n} \not\approx \underline{m}$, for every $n \neq m$.

(C5) $\mathsf{Q} \vdash \underline{m} \leqslant \underline{n}$, for every $m \leqslant n$.

Recall that our vocabulary L_{ar} does not include $\leqslant$. The formula $\underline{m} \leqslant \underline{n}$ is actually an abbreviation for $\exists z\ \underline{m} + z \approx \underline{n}$.

(C6) $\mathsf{Q} \vdash \neg(\underline{m} \leqslant \underline{n})$, for every $m \nleqslant n$.

(C7) $\mathsf{Q}, x \leqslant \underline{n} \vdash (x \approx \tilde{0}) \lor (x \approx \underline{1}) \lor \cdots \lor (x \approx \underline{n})$.

(C8) $\mathsf{Q} \vdash (x \leqslant \underline{n}) \lor (\underline{n} \leqslant x)$.

All these statements show that the natural meaning of the standard operations like addition and multiplication are provable in Q, and hence, in any extension $T \supseteq \mathsf{Q}$.

Definition 11.1

- A formula φ is called a Δ_0-formula, if all its quantifiers are bounded quantifiers, i.e., of the form $(\forall x \leqslant t)\ \alpha$, where t is a term over L_{ar}.

  Intuitively $(\forall x \leqslant t)\ \alpha$ states "for every $x \leqslant t$, the formula α holds."
- A formula φ is called a Σ_1-formula, if it is of the form $\exists \bar{x}\ \psi$, where ψ is a Δ_0-formula.
- A formula φ is called a Π_1-formula, if it is of the form $\forall \bar{x}\ \psi$, where ψ is a Δ_0-formula.

Proposition 11.2 *Let t be a term over L_{ar} with free variables $x_1, \ldots, x_n$. For a valuation $\mathsf{val}\colon VAR \to \mathbb{N}$, consider the substitution $sub := [x_1/\underline{\mathsf{val}(x_1)}, \ldots, x_n/\underline{\mathsf{val}(x_n)}]$. Then,*

$$t^{\mathcal{N}}[\mathsf{val}] = m \quad \textit{if and only if} \quad \mathsf{Q} \vdash t[sub] \approx \underline{m}$$
$$t^{\mathcal{N}}[\mathsf{val}] \leqslant m \quad \textit{if and only if} \quad \mathsf{Q} \vdash t[sub] \leqslant \underline{m}$$

Proof. By straightforward induction on t together with (C1)–(C8) above. ■

Theorem 11.3 below will be very useful. It states that in order to check whether a Δ_0-sentence φ is provable in Q, it is sufficient to check whether it holds in $\mathcal{N}$. In other words, instead of looking for a proof of φ, we simply checks whether it holds in $\mathcal{N}$, which is a more convenient and intuitive system to work with.

Theorem 11.3 *For every Δ_0-formula $\varphi(\bar{x})$, where $\bar{x} = (x_1, \ldots, x_n)$, the following holds. For every $\bar{a} = (a_1, \ldots, a_n) \in \mathbb{N}^n$:*

$$\mathcal{N} \models \varphi(\underline{\bar{a}}) \quad \textit{if and only if} \quad \mathsf{Q} \vdash \varphi(\underline{\bar{a}}).$$

Proof. The proof is by induction on φ. The base case, when the atomic formula of the form $s \approx t$, can be deduced directly from Proposition 11.2.

The induction step consists of three cases.

Case 1: $\varphi(\bar{x})$ is $\neg\alpha(\bar{x})$.

$\mathcal{N} \models \varphi(\bar{a})$ if and only if $\mathcal{N} \not\models \alpha(\bar{a})$, if and only if $\mathsf{Q} \not\vdash \alpha(\underline{\bar{a}})$, if and only if $\mathsf{Q} \vdash \varphi(\underline{\bar{a}})$, with the second "if and only if" coming from the induction hypothesis.

Case 2: $\varphi(\bar{x})$ is $\alpha_1(\bar{x}) \wedge \alpha_2(\bar{x})$.

$\mathcal{N} \models \varphi(\bar{a})$ if and only if $\mathcal{N} \models \alpha_1(\bar{a}) \wedge \alpha_2(\bar{a})$, if and only if $\mathcal{N} \models \alpha_1(\bar{a})$ and $\mathcal{N} \models \alpha_2(\bar{a})$, if and only if $\mathsf{Q} \vdash \alpha_1(\underline{\bar{a}})$ and $\mathsf{Q} \vdash \alpha_2(\underline{\bar{a}})$, if and only if $\mathsf{Q} \vdash \varphi(\underline{\bar{a}})$, with the third "if and only if" coming from the induction hypothesis.

Case 3: $\varphi(\bar{x})$ is $\forall z \leqslant t\ \alpha(\bar{x}, z)$.

Let val denote the valuation that maps x_i to a_i, and *sub* denote the substitution that substitute x_i with $\underline{a_i}$. Let $M = t^{\mathcal{N}}[\mathsf{val}]$. By Proposition 11.2, we have $\mathsf{Q} \vdash t \approx \underline{M}$.

$\mathcal{N} \models \varphi(\bar{a})$ if and only if for every $m \leqslant M$,

$$\mathcal{N}, [\mathsf{val}, z \mapsto m] \quad \models \quad \alpha(\bar{a}, z),$$

which holds, if and only if for every $m \leqslant M$,

$$\mathsf{Q} \quad \vdash \quad \alpha[sub, z/\underline{m}],$$

which holds, if and only if

$$\mathsf{Q}, z \leqslant \underline{M} \quad \vdash \quad \alpha[sub, z],$$

which holds, if and only if

$$\mathsf{Q} \quad \vdash \quad (\forall z \leqslant \underline{M})\alpha(\bar{x}, z),$$

which holds, if and only if

$$\mathsf{Q} \quad \vdash \quad (\forall z \leqslant t)\alpha(\bar{x}, z).$$

The third "if and only if" comes from the induction hypothesis, while the fourth is from (C5) and (C7). The fifth comes from the fact that $(\forall z \leqslant \underline{M})\alpha(\bar{x}, z)$ is an abbreviation of $\forall z(z \leqslant \underline{M} \to \alpha(\bar{x}, z))$. The last one comes from $\mathsf{Q} \vdash t \approx \underline{M}$.

■

2 Representable functions

In the following let $\bar{x}$ be a vector of variables, and $\bar{a}$ be a vector of natural numbers with the same length as $\bar{x}$.

Representable functions in a theory $T \supseteq$ Q. A function $f : \mathbb{N}^k \to \mathbb{N}$ is called *representable* in a theory $T \supseteq \mathsf{Q}$, if there is a formula $\varphi(\bar{x}, y)$ such that $f(\bar{a}) = m$ if and only if $T \vdash \varphi(\underline{\bar{a}}, \underline{m})$. Note that this is equivalent to saying that $f(\bar{a}) = m$ if and only if $T \vdash y \approx \underline{m} \leftrightarrow \varphi(\underline{\bar{a}}, y)$.

It is Σ_1-representable, if the formula $\varphi(\bar{x})$ is Σ_1-formula, and the formula φ is called the representation formula for f.

Likewise, a relation $R \subseteq \mathbb{N}^k$ is called representable in a theory $T \supseteq \mathsf{Q}$, if there is a formula $\varphi(\bar{x})$ such that if $\bar{a} \in R$, then $T \vdash \varphi(\underline{\bar{a}})$; and if $\bar{a} \notin R$, then $T \vdash \neg\varphi(\underline{\bar{a}})$.

Arithmetical functions (functions representable in $\mathcal{N}$). A function $f : \mathbb{N}^k \to \mathbb{N}$ is called *arithmetical*, or *representable* in $\mathcal{N}$, if there is a formula $\varphi(\bar{x}, y)$ such that $f(\bar{a}) = m$ if and only if $\mathcal{N} \models \varphi(\underline{\bar{a}}, \underline{m})$. The notions of Σ_1-representable and Π_1-representable are defined similarly as above.

3 Representability of recursive functions

In this section we will show the following theorem.

Theorem 11.4 *Every recursive function f is representable by a Σ_1-formula in Q.*

The proof consists of two steps:

(1) We show that f is representable in $\mathcal{N}$ by a Σ_1-formula, as well as by a Π_1-formula.

(2) We show that it can be represented by a Σ_1-formula in Q.

Representing f in $\mathcal{N}$. The proof is by induction on f. The base case is as follows.

- f is the constant zero function, i.e., $f(\bar{x}) = 0$.
  Then, $\varphi(\bar{x}, y) := y \approx \tilde{0}$ is a Δ_0-formula representing f.
- f is the successor function of one of its component, i.e., $f(\bar{x}) = \mathtt{Succ}(x_i)$.
  Then, $\varphi(\bar{x}, y) := y \approx \mathtt{Succ}(x_i)$ is a Δ_0-formula representing f.
- f is the projection function to one of its components, i.e., $f(\bar{x}) = x_i$.
  Then, $\varphi(\bar{x}, y) := y \approx x_i$ is a Δ_0-formula representing f.

The induction step is as follows.

- Functions obtained from applying the composition rule **Oc**.
  Let $f = h[g_1, \ldots, g_m]$ be a function from $\mathbb{N}^n \to \mathbb{N}$, i.e., each $g_i : \mathbb{N}^n \to \mathbb{N}$ and $h : \mathbb{N}^m \to \mathbb{N}$.
  By the induction hypothesis, let α and γ_i be Σ_1-formulas representing h and γ_i, respectively.
  Both Σ_1-formula φ_1 and Π_1-formula φ_2 below represent f in $\mathcal{N}$.

$$\begin{aligned} \varphi_1(\bar{x}, z) \quad &:= \quad \exists y_1 \cdots \exists y_m \bigwedge_{1 \leqslant i \leqslant m} \gamma_i(\bar{x}, y_i) \quad \wedge \quad \alpha(y_1, \ldots, y_m, z) \\ \varphi_2(\bar{x}, z) \quad &:= \quad \forall u \ \big(\varphi_1(\bar{x}, u) \ \to \ u \approx z\big) \end{aligned}$$

- Functions obtained from applying the primitive recursive rule **Op**.

  This is the most challenging part. See the appendix for the details.

- Functions obtained from applying the rule $\mathbf{O}\mu$.

  Let $\bar{x} = (x_1, \ldots, x_n)$, and let $f(\bar{x}) := \mu y[g(\bar{x}, y) = 0]$. By the induction hypothesis, there is a Σ_1-formula $\alpha_1(\bar{x}, y, z)$, and Π_1-formula $\alpha_2(x, t, z)$ representing g in $\mathcal{N}$.

$$\begin{array}{lcll} \alpha_1(\bar{x}, y, z) & := & \exists \bar{v}\ \psi_1(\bar{x}, y, z, \bar{v}) & \qquad \text{where } \psi_1 \text{ is a } \Delta_0\text{-formula} \\ \alpha_2(\bar{x}, y, z) & := & \forall \bar{w}\ \psi_2(\bar{x}, y, z, \bar{w}) & \qquad \text{where } \psi_2 \text{ is a } \Delta_0\text{-formula} \end{array}$$

  Consider the formula φ_1 below.

$$\begin{array}{lcl} \varphi_1(\bar{x}, y) & := & \alpha_1(\bar{x}, y, \tilde{0}) \ \wedge \ (\forall z < y)\ \neg\alpha_2(\bar{x}, z, \tilde{0}) \\ & := & \exists \bar{v}\ \psi_1(\bar{x}, y, z, \bar{v}) \ \wedge \ (\forall z < y)\ \exists \bar{w}\ \neg\psi_2(\bar{x}, z, \tilde{0}, \bar{w}) \end{array}$$

  We have the following identity (can be easily proved) in $\mathcal{N}$:

$$\mathcal{N} \ \models \ (\forall z < y)\ \exists u\ \psi \ \equiv \ \exists u'(\forall z < y)\neg(\forall u < u')\neg\psi$$

  Therefore, the following Σ_1-formula φ_1' is equivalent to φ_1 in $\mathcal{N}$.

$$\begin{array}{lcll} \varphi_1'(\bar{x}, y) & := & \exists \bar{v}\ \psi_1(\bar{x}, y, z, \bar{v}) \ \wedge \ \exists \bar{w}'(\forall z < y)\ \psi_2'(\bar{x}, z, \tilde{0}) & \qquad \text{where } \psi_2' \text{ is } \Delta_0\text{-formula} \\ & := & \exists \bar{v} \exists \bar{w}' \Big(\psi_1(\bar{x}, y, z, \bar{v}) \ \wedge \ (\forall z < y)\ \psi_2'(\bar{x}, z, \tilde{0})\Big) & \end{array}$$

  Thus, φ_1' is the desired Σ_1-formula representing f in $\mathcal{N}$.

  A Π_1-formula φ_2 representing f can be obtained as follows.

$$\varphi_2(\bar{x}, y) \ := \ \forall u\ \big(\varphi_1'(\bar{x}, u) \ \to \ u \approx y\big)$$

Representing f in Q. Note that if f is representable in Q, then by monotonicity rule, it is representable in $T \supseteq \mathsf{Q}$.

Let $f : \mathbb{N}^n \to \mathbb{N}$ be a recursive function, and let $\varphi(\bar{x}, y) := \exists \bar{z}\ \psi(\bar{x}, y, \bar{z})$ be its representing formula in $\mathcal{N}$, where ψ is Δ_0-formula. That is, for every $\bar{a} \in \mathbb{N}^n$,

$$f(\bar{a}) \ = \ b \quad \text{if and only if} \quad \mathcal{N} \models \varphi(\underline{\bar{a}}, \underline{b}).$$

We have to show that for every $\bar{a} \in \mathbb{N}^n$,

$$f(\bar{a}) \ = \ b \quad \text{if and only if} \quad \mathsf{Q} \vdash \varphi(\underline{\bar{a}}, \underline{b}).$$

We start with the "if" part. Suppose $f(\bar{a}) = b$. Since φ represents f, for some $\bar{w}$,

$$\mathcal{N} \ \models \ \psi(\underline{\bar{a}}, \underline{b}, \underline{\bar{w}})$$

Since ψ is Δ_0-formula, by Theorem 11.3, we have $\mathsf{Q} \vdash \psi(\underline{\bar{a}}, \underline{b}, \underline{\bar{w}})$, and hence, $\mathsf{Q} \vdash \exists \bar{z}\ \psi(\underline{\bar{a}}, \underline{b}, \bar{z})$.

Now, we show the "only if" part. Suppose for some $\bar{w}$, $\mathsf{Q} \vdash \psi(\underline{\bar{a}}, \underline{b}, \underline{\bar{w}})$. Since $\mathcal{N} \models \mathsf{Q}$, we have that $\mathcal{N} \models \psi(\underline{\bar{a}}, \underline{b}, \underline{\bar{w}})$, and thus, $\mathcal{N} \models \exists \bar{z}\ \psi(\underline{\bar{a}}, \underline{b}, \bar{z})$. Therefore, $\mathcal{N} \models \varphi(\underline{\bar{a}}, \underline{b})$. Since φ represents f, we have $f(\bar{a}) = b$. This completes the proof of Theorem 11.4.

4 Fixed point lemma and Gödel's first incompleteness theorem

Recall that in order to prove Gödel's incompleteness theorem, we have to show that:

- Every recursive function is representable in Q.
- For a consistent and recursively axiomatizable theory $T \supseteq \mathsf{Q}$, there is a sentence Ψ such that $T \vdash \Psi \ \leftrightarrow\ \big(\forall y\ \neg\ \textsc{IsProofOf}_T(y, \underline{\sharp\Psi})\big)$,

We describe how to achieve the first part in the previous section. We will now describe how to achieve the second part.

For a variable x, define the function $\textsc{Subs}_x : \mathbb{N}^2 \to \mathbb{N}$ as follows.

$$\textsc{Subs}_x(N, m) \quad := \quad K$$

where K is "the formula" obtained by substituting variable x with the term $\underline{m}$ in "formula" N. Here, "the formulas" K and N refer to the formulas whose Gödel's numbers are K and N, respectively. It is not that difficult to think of a computer program for $\textsc{Subs}_x$. So, it is also a recursive function, and can be represented in a theory Q, and hence, in any extension $T \supseteq \mathsf{Q}$. Let $\Lambda_{\textsc{Subs}_x}(v_1, v_2, v_3)$ be a Σ_1-formula representing $\textsc{Subs}_x$.

Lemma 11.5 (Fixed point lemma) *Let $T \supseteq \mathsf{Q}$. For every formula $\alpha(z)$ over vocabulary $\{\tilde{0}, \mathtt{Succ}, +, \cdot\}$, there is a formula γ such that $T \vdash \gamma \leftrightarrow \alpha(\underline{\sharp\gamma})$.*

Proof. Due to the definition of $\Lambda_{\textsc{Subs}_x}(v_1, v_2, v_3)$, for every formula φ,

$$T \ \vdash\ \Lambda_{\textsc{Subs}_x}(\underline{\sharp\varphi}, \underline{n}, y) \ \leftrightarrow\ y \approx \underline{\sharp\varphi[x/\underline{n}]}$$

If we plug in n with $\sharp\varphi$ itself,

$$T \ \vdash\ \Lambda_{\textsc{Subs}_x}(\underline{\sharp\varphi}, \underline{\sharp\varphi}, y) \ \leftrightarrow\ y \approx \underline{\sharp\varphi[x/\underline{\sharp\varphi}]} \tag{1}$$

Let $\beta(x)$ be the following formula.

$$\beta(x) \quad := \quad \forall y\ \Big(\Lambda_{\textsc{Subs}_x}(x, x, y) \ \to\ \alpha[z/y]\Big)$$

Consider $\gamma := \beta[x/\underline{\sharp\beta}]$. That is,

$$\gamma \quad = \quad \forall y\ \Big(\Lambda_{\textsc{Subs}_x}(\underline{\sharp\beta}, \underline{\sharp\beta}, y) \ \to\ \alpha[z/y]\Big)$$

By (1),

$$T \ \vdash\ \gamma \quad \leftrightarrow \quad \forall y\ \Big(y \approx \underline{\sharp\beta[x/\underline{\sharp\beta}]} \ \to\ \alpha[z/y]\Big)$$

Since $\gamma = \beta[x/\underline{\sharp\beta}]$,

$$\begin{aligned} T \ &\vdash\ \gamma \quad \leftrightarrow \quad \forall y\ \Big(y \approx \underline{\sharp\gamma} \ \to\ \alpha[z/y]\Big) \\ T \ &\vdash\ \gamma \quad \leftrightarrow \quad \alpha(\underline{\sharp\gamma}) \end{aligned}$$

This completes the proof of fixed point lemma. ■

To wrap up, we state and prove formally Gödel's incompleteness theorem.

Theorem 11.6 (Gödel's incompleteness theorem) *For every consistent and recursively axiomatizable theory $T \supseteq \mathsf{Q}$, there is a sentence Ψ such that neither $T \vdash \Psi$ nor $T \vdash \neg\Psi$.*

Proof. Since T is recursively axiomatizable theory, we have a "computer program" on an input proof y, output x, which represents the conclusion of the proof y. By Church-Turing thesis, every "computer program" is equivalent to a recursive function, and by Theorem 11.4, a recursive function can be represented in Σ_1-formula in $T \supseteq \mathsf{Q}$. Thus, we have a Σ_1-formula $\textsc{IsProofOf}_T(y,x)$ which states that y is a proof of x. In particular, we also have the following formula.

$$\textsc{Provable}_T(x) \quad := \quad \exists y\ \textsc{IsProofOf}_T(y,x)$$

such that

$$T \ \vdash \ \varphi \leftrightarrow \textsc{Provable}_T(\underline{\sharp\varphi})$$

Consider the negation of $\textsc{Provable}_T(x)$, i.e., $\neg\textsc{Provable}_T(x)$. By fixed-point lemma, there is Ψ such that

$$T \ \vdash \ \Psi \leftrightarrow \neg\textsc{Provable}_T(\underline{\sharp\Psi})$$

which is simply

$$T \ \vdash \ \Psi \leftrightarrow \forall y\ \neg\textsc{IsProofOf}_T(y, \underline{\sharp\Psi})$$

Following the argument in Section 3 in Lesson 10, neither Ψ nor $\neg\Psi$ are provable in T. ■

Appendix: Representing the Op rule

The proof consists of two steps.

- First, we construct a function $G : \mathbb{N}^2 \to \mathbb{N}$ representable with Δ_0-formula such that for every n, for every sequence $c_0, \ldots, c_n$, there is c such that for all $i = 0, \ldots, n$, we have $G(c,i) = c_i$.
- Using the function G constructed, we can represent the **Op** rule with a Σ_1-formula.

Intuitively, the function G "encodes" every sequence element $(c_0, \ldots, c_n) \in \mathbb{N}^* = \bigcup_{i \geqslant 1} \mathbb{N}^i$ as a number c such that to retrieve an element c_i, we simply "access" $G(c,i)$.

Constructing the function G. Consider the following bijection $\wp : \mathbb{N}^2 \to \mathbb{N}$.

$$\wp(a,b) \quad := \quad a + \sum_{i=1}^{a+b} i \quad = \quad a \ + \ \frac{1}{2}(a+b)(a+b+1)$$

Note that $a, b \leqslant \wp(a,b)$, for every a,b. It is trivial that $\wp$ can be represented by a Δ_0-formula.

Let $F : \mathbb{N}^3 \to \mathbb{N}$ be the following function.

$$F(a,b,i) \quad := \quad \text{the remainder of } a \text{ divided by } 1 + (1+i)b$$

It is not that difficult to show that the function F is represented by a Δ_0-formula.

Let Proj_x and Proj_y be the following functions. For every $m \in \mathbb{N}$, if $\wp^{-1}(m) = (a,b)$,

$$\mathit{Proj}_x(m) \ := \ a \quad \text{and} \quad \mathit{Proj}_y(m) \ := \ b$$

Consider the following function $G : \mathbb{N}^2 \to \mathbb{N}$.

$$G(c,i) \ := \ F(\mathit{Proj}_x(c) \ , \ \mathit{Proj}_y(c) \ , \ i)$$

The function G can be represented with a Δ_0-formula as follows.

$$G(c,i) = m \quad \text{if and only if} \quad (\exists x \leqslant c)(\exists y \leqslant c)\Big(\underline{\wp(a,b) = c} \ \wedge \ \underline{F(a,b,i) = m}\Big)$$

The underlined parts denote abbreviations of the formulas that represent $\wp(a,b) = c$ and $F(a,b,i) = m$, respectively.

We will show that G is our desired function. In the following we write $a \mid b$ to denote that a divides b, i.e., when b is divided by a, there is no remainder. For two positive integers a, b, we say that a and b are coprime, if there is no prime p that divides both a and b.

Lemma 11.7 (Euclid) *If a and b are coprime, then there are $x, y \in \mathbb{N}$ such that $ax + 1 = by$.*

Theorem 11.8 (Chinese remainder theorem) *Let $c_0, \ldots, c_k, d_0, \ldots, d_k$ such that $c_i < d_i$. Let $d_1, \ldots, d_k$ be pairwise coprime. Then, there exists an integer $a \in \mathbb{N}$ such that $\mathrm{rem}(a, d_i) = c_i$, i.e. the remainder of a divided by d_i is c_i.*

Theorem 11.9 *For every n, for every sequence $c_0, \ldots, c_n$, there exist a, b such that for all $i = 0, \ldots, n$, we have $F(a,b,i) = c_i$.*

Since $G(\wp(a,b), i) = F(a,b,i)$, we have that for every sequence $c_0, \ldots, c_n$, there is c, which is $\wp(a,b)$ and greater than each c_i, such that for all $i = 0, \ldots, n$, we have $G(c,i) = c_i$.

Proof. Let $c_0, \ldots, c_n$ be a sequence of natural numbers. Consider the following two numbers M and K.

- $M := \max(n, c_0, \ldots, c_n)$.
- $b := \mathrm{lcm}(1, \ldots, M)$, where "lcm" is least common multiplier.

Let $d_i := 1 + (1+i)b$, for each $i = 0, \ldots, n$. Note that $d_i > c_i$.

We claim that $d_0, \ldots, d_n$ are pairwise coprime. Suppose to the contrary that there is a prime p that divides both d_i and d_j. Thus, $p \mid d_i - d_j = (i-j)b$. So, either $p|(i-j)$ or $p \mid b$.

Now, $i, j \leqslant M$, since b is the least common multiplier of all integers between 1 and M, we have $(i-j) \mid b$. This means that $p \mid b$. By definition of d_i, $b \mid (d_i - 1)$, which means $p \mid (d_i - 1)$. This is absurd, since $p \mid d_i$. So, there is such prime p that divides d_i and d_j. In other words, $d_0, \ldots, d_n$ are coprime.

By Theorem 11.8, there is a such that $\mathrm{rem}(a, d_i) = c_i$. By the definition of the function F, we have $F(a,b,i) = c_i$. By the construction, it is obvious that $\wp(a,b) > c_i$. ■

Representing functions obtained from applying Op rule. Let $g \in \mathbf{F}_n$ and $h \in \mathbf{F}_{n+2}$ be recursive functions.

- Let g be represented by a Σ_1-formula α_1, as well as a Π_1-formula α_2.
- Let h be represented by a Σ_1-formula β_1, as well as a Π_1-formula β_2.

Suppose $f \in \mathbf{F}_{n+1}$ is the function obtained via the **Op** rule as follows. For every $\bar{a} \in \mathbb{N}^n$,

$$f(\bar{a}, 0) \ := \ g(\bar{a}) \quad \text{and} \quad f(\bar{a}, \mathtt{Succ}(b)) \ := \ h(\bar{a}, b, f(\bar{a}, b))$$

The following formula represents f.

$$\varphi(\bar{x}, y, z) \quad := \quad \Big(y \approx \tilde{0} \;\rightarrow\; \alpha_1(\bar{x}, z)\Big) \quad \wedge \quad \exists z'(\forall y' < y)\Big(\underline{G(z', \mathtt{Succ}(y')) = h(\bar{x}, y', G(z', y'))}\Big)$$

Intuitively, the variable z' is such that for every $i \leqslant y$, $G(z', i) = f(\bar{x}, i)$.

Now, $\varphi(\bar{x}, y, z)$ can be rewritten into:

$$\begin{aligned} \varphi(\bar{x}, y, z) \quad := \quad & \Big(y \approx \tilde{0} \;\rightarrow\; \alpha_1(\bar{x}, z)\Big) \quad \wedge \\ & \exists z'(\forall y' < y)(\forall u < z')(\forall v < z') \\ & \qquad \Big(\underline{G(z', \mathtt{Succ}(y')) = u} \;\wedge\; \underline{G(z', y') = v} \;\rightarrow\; \beta_1(\bar{x}, y', u_2, u_1)\Big) \end{aligned}$$

By pulling all the existential quantifiers from β_1 and $\exists z'$ to the front of the formula, we obtain a Σ_1-formula. A Π_1-formula can be obtained via:

$$\varphi'(\bar{x}, y, z) \quad := \quad \forall w \; \varphi(\bar{x}, y, w) \rightarrow w \approx z$$

Lesson 12: Decision problems in FO

Theme: The complexity of some standard decision problems in FO.

From Gödel's incompleteness theorem, it is immediate that the following problem $\mathsf{SAT}(\mathcal{N})$ is undecidable, where $\mathcal{N}$ is the structure $\mathcal{N} = (\mathbb{N}, 0, \text{succ}, +, \cdot)$.

$\mathsf{SAT}(\mathcal{N})$	
Input:	An FO sentence φ over the vocabulary $L_{ar} = \{\tilde{0}, \texttt{Succ}, +, \cdot\}$.
Task:	Output `True`, if $\mathcal{N} \models \varphi$. Otherwise, output `False`.

Theorem 12.1 *The problem $\mathsf{SAT}(\mathcal{N})$ is undecidable.*

Consider the following evaluation problems.

$\mathsf{EVAL}(\mathrm{FO})$	
Input:	An FO sentence φ and a finite structure $\mathcal{A}$.
Task:	Output `True`, if $\mathcal{A} \models \varphi$. Otherwise, output `False`.

$\mathsf{EVAL}(\varphi)$, where φ is an FO sentence	
Input:	A finite structure $\mathcal{A}$.
Task:	Output `True`, if $\mathcal{A} \models \varphi$. Otherwise, output `False`.

Theorem 12.2

- *The problem $\mathsf{EVAL}(\mathrm{FO})$ is* PSPACE*-complete.*
- *For every* FO *sentence φ, the problem $\mathsf{EVAL}(\varphi)$ is in* PTIME.

Recall that a sentence φ is *satisfiable*, if there is a model $\mathcal{A}$ such that $\mathcal{A} \models \varphi$. A sentence is *finitely satisfiable*, if there is a finite model $\mathcal{A}$ such that $\mathcal{A} \models \varphi$. We will consider the following two problems.

$\mathsf{SAT}(\mathrm{FO})$	
Input:	An FO sentence φ.
Task:	Output `True`, if φ is satisfiable. Otherwise, output `False`.

$\mathsf{FIN\text{-}SAT}(\mathrm{FO})$	
Input:	An FO sentence φ.
Task:	Output `True`, if φ is finitely satisfiable. Otherwise, output `False`.

Theorem 12.3

- *Both $\mathsf{SAT}(\mathrm{FO})$ and $\mathsf{FIN\text{-}SAT}(\mathrm{FO})$ are undecidable.*
- *$\mathsf{SAT}(\mathrm{FO})$ is co-r.e (co-recursive enumerable).*
- *$\mathsf{FIN\text{-}SAT}(\mathrm{FO})$ is r.e (recursive enumerable).*