Logic and Computability *

Artie Khovanov

Compiled on June 5, 2023

These are some notes for the Cambridge Mathematical Tripos Part III course *Logic and Computability* in Michaelmas 2022. This is NOT a verbatim copy of the lectured material: I've edited the content to help me understand it. As a result, any errors are mine alone.

I'm actively maintaining these notes. If you want to report typos or mistakes, please email aik31@cam.ac.uk or message me on Discord at `FM22#2007`.

Contents

1 Proof Theory

1.1 Intuitionistic Logic

Say $\neg\varphi$ means $\varphi \to \bot$, where $\bot$ is a proposition with no proof. To prove the conjunction $\varphi \wedge \psi$ is to give proofs of φ and ψ. To prove the implication $\varphi \to \psi$

*Based on the lectures under the same name taught by Dr J. Siqueira in Michaelmas 2022.

is to give a procedure that converts proofs of φ into proofs of ψ. Proving the disjunction $\varphi \vee \psi$ requires giving *either* a proof of φ or a proof of ψ. Classically, we need not know which of φ and ψ is true, but intuitionistically we *must* choose one to give a proof of.

In fact, the **law of the excluded middle** (LEM), that is, $\varphi \vee \neg\varphi$, is not intuitionistically valid in general. The (informal) reason is that, if it were true, there would be some procedure for deciding which of φ or $\neg\varphi$ to give a proof of. But this would require knowing whether φ holds in advance!

Theorem 1.1 (Diaconescu)**.** *The LEM can be intuitionistically deduced from the axiom of choice (in a standard set theory).*

This is because intuitionistic existence $\exists x.\varphi(x)$ yields a concrete term x, along with a proof of $\varphi(x)$. This is crucial in the following proof.

Proof. Let φ be a proposition. Consider the sets $A = \{x \in \{0,1\} \mid \varphi \vee (x = 0)\}$ and $B = \{x \in \{0,1\} \mid \varphi \vee (x = 1)\}$. Since $0 \in A$ and $1 \in B$, these sets are nonempty. Let f be a choice function on $\{A, B\}$, so that $f(A) \in A$ and $f(B) \in B$. Therefore $\varphi \vee (f(A) = 0)$ and $\varphi \vee (f(B) = 1)$. Since we obviously have a proof of $(f(A) = 0) \vee (f(A) = 1)$ and so on, we have the following cases:

1. We have a proof that $f(A) = 1$, so $\varphi \vee (1 = 0)$ is valid, so we have a proof of φ (because of what disjunction means).
2. We have a proof that $f(B) = 0$, so similarly we have a proof of φ.
3. We have a proof that $f(A) = 0$ and $f(B) = 1$, in which case we obtain a proof of $\neg\varphi$. Indeed, given a proof of φ, we have $A = B = \{0,1\}$ (by extensionality), so $0 = f(A) = f(B) = 1$. Thus we have a proof of $\bot$.

□

Note that, in terms of the set theory, this proof only used the axioms of separation and extensionality, so this issue comes up in any reasonable set theory, not just (intuitionistic) ZF.

To prove $\forall x.\varphi(x)$ is to give a procedure that converts terms x into proofs of $\varphi(x)$. Note that $\forall$ and $\exists$ are *not* related by $\exists x.\varphi(x) \iff \neg(\forall x.\neg\varphi(x))$ as they are in classical logic.

Here are some reasons to care about this viewpoint:

- Intuitionistic proofs are *more informative*: they carry computable content with them.
- Intuitionistic logic is more general – we assume less.
- There is a more diverse range of concepts, since things which are classically equivalent are often not intuitionistically equivalent (e.g. finiteness).
- Intuitionistic logic is the logic of topoi, so intuitionistic proofs work everywhere (not just in classical topoi).

Viewing logical connectives as operators on proofs is the **Brouwer-Heyting-Kolmogorov (BHK) interpretation**.

Example 1.2. Informally, the proposition

$$(P \to Q) \to (\neg Q \to \neg P)$$

can be proved by starting with a proof of P, applying the proof of $P \to Q$ to get a proof of Q, and then applying the proof of $\neg Q$ to obtain a contradiction.

1.1.1 Natural Deduction Rules for Intuitionistic Propositional Calculus (IPC)

Here Γ is the collection of open assumptions, and Γ, A means $\Gamma \cup \{A\}$. We have introduction and elimination rules, labelled I and E respectively.

$$(\wedge - I) \quad \frac{\Gamma \vdash A \qquad \Gamma \vdash B}{\Gamma \vdash A \wedge B}$$

$$(\vee - I) \quad \frac{\Gamma \vdash A}{\Gamma \vdash A \vee B} \quad \frac{\Gamma \vdash B}{\Gamma \vdash A \vee B}$$

$$(\wedge - E) \quad \frac{\Gamma \vdash A \wedge B}{\Gamma \vdash A} \quad \frac{\Gamma \vdash A \wedge B}{\Gamma \vdash B}$$

$$(\vee - E) \quad \frac{\Gamma \vdash A \vee B \qquad \Gamma, A \vdash C \qquad \Gamma, B \vdash C}{\Gamma \vdash C}$$

$$(\to - I) \quad \frac{\Gamma, A \vdash B}{\Gamma \vdash A \to B}$$

$$(\to - E) \quad \frac{\Gamma \vdash A \to B \qquad \Gamma \vdash A}{\Gamma \vdash B}$$

$$(\neg - I) \quad \frac{\Gamma, A \vdash \bot}{\Gamma \vdash \neg A}$$

$$(\neg - E) \quad \frac{\Gamma \vdash \neg A \qquad \Gamma \vdash A}{\Gamma \vdash \bot}$$

$$(\bot - E) \quad \frac{\Gamma \vdash \bot}{\Gamma \vdash P}$$

$$(\text{Ax}) \quad \frac{}{\Gamma, A \vdash A}$$

We obtain classical logic by adding either

$$(\text{LEM}) \ \frac{}{\Gamma \vdash A \vee \neg A} \text{ (for each A), or (reductio ad absurdum) } \frac{\Gamma, \neg A \vdash \bot}{A}.$$

We write

$$(\text{A}) \ \frac{\begin{array}{c}[A] \\ \vdots \\ X\end{array}}{\text{B}}$$

to mean that, if we can prove $A \vdash X$, then we can prove B by discharging A; that is, by applying a deduction rule which removes it from the set of assumptions, such as $(\to - I)$ or $(\neg - I)$. This makes proofs with these introduction rules less tedious to write out. Formally, we use (Ax) to introduce the assumption A, keep track of A through the proof, and finally discharge A at the end.

1.1.2 Natural Deduction Rules for Intuitionistic Predicate Calculus (IQC)

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

$$(\exists - E) \quad \frac{\Gamma \vdash \exists x.\varphi(x) \qquad \Gamma, \varphi(x) \vdash A}{\Gamma \vdash A}$$

$$(\forall - I) \quad \frac{\Gamma \vdash \varphi}{\Gamma \vdash \forall x.\varphi(x)}, \text{ where } x \text{ is not free in } \Gamma.$$

$$(\forall - E) \quad \frac{\Gamma \vdash \forall x.\varphi(x)}{\Gamma \vdash \varphi[x = t]}$$

Example 1.3. We want to prove $A \wedge B \to B \wedge A$.

$$\dfrac{(\wedge - E)\dfrac{[A \wedge B]}{A} \qquad (\wedge - E)\dfrac{[A \wedge B]}{B}}{(\wedge - I)\dfrac{B \wedge A}{(\to - I)\dfrac{}{A \wedge B \to B \wedge A}}}$$

Example 1.4. We can also prove the Hilbert-style axioms

$$\varphi \to (\psi \to \varphi) \text{ and } (\varphi \to (\psi \to \chi)) \to ((\varphi \to \psi) \to (\varphi \to \chi))$$

by repeatedly applying arrow-introduction and elimination:

$$\dfrac{\dfrac{[\varphi] \qquad [\psi]}{\psi \to \varphi}}{\varphi \to (\psi \to \varphi)}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{[\varphi \to (\psi \to \chi)] \qquad [\varphi]}{\psi \to \chi} \qquad \dfrac{[\varphi \to \psi] \qquad [\varphi]}{\psi}}{\chi}}{\varphi \to \chi}}{(\varphi \to \psi) \to (\varphi \to \chi)}}{(\varphi \to (\psi \to \chi)) \to ((\varphi \to \psi) \to (\varphi \to \chi))}$$

If we have a natural deduction proof of φ from a set of open assumptions Γ, we write $\Gamma \vdash \varphi$. We may subscript the $\vdash$ with 'IPC', 'IQC', 'CQC', etc to specify the deduction rules used. If Γ is empty, we say φ is **intuitionistically valid**.

Lemma 1.5. *Suppose $\Gamma \vdash_{IPC} \varphi$. Then $\Gamma, \psi \vdash_{IPC} \varphi$ for any proposition ψ. Moreover, if p is a primitive proposition and ψ is any proposition, then*

$$\Gamma[p = \psi] \vdash_{IPC} \varphi[p = \psi].$$

Proof. Induction on the size of the proof. □

1.2 Simply typed λ-calculus

Suppose we have a set Π of simple types, generated by the grammar

$$\Pi = \mathcal{U} \mid \Pi \to \Pi$$

where $\mathcal{U}$ is a countable set of type variables. Types of the second form are called function types. Suppose we also have an infinite set V of (term) variables.

Definition 1.6. The set Λ_Π of **simply-typed λ-terms** is defined by the grammar

$$\Lambda_\Pi = V \mid \lambda V : \Pi.\Lambda_\Pi \mid \Lambda_\Pi \Lambda_\Pi$$

The formation of terms by the second clause in the grammar is called **λ-abstraction**, and formation by the third clause, **λ-application**.

The idea is that, for example, $\lambda x : \mathbb{Z}.x^2$ should represent the function $x \to x^2$ on $\mathbb{Z}$.

Simply-typed λ-terms come with a typability relation $\Vdash$, which gives rules to make a type judgement on terms given a context.

Definition 1.7. A **context** is a set of pairs $\{x_1 : \tau_1, \ldots, x_n : \tau_n\}$, where the x_i are distinct (term) variables and the τ_i are types. The **domain** $\operatorname{dom}\Gamma$ of a context Γ is the set of variables appearing in it, and its **range** $|\Gamma|$ is the set of types appearing in it.

A context assigns types to the variables in a λ-term. Write C for the set of all contexts. Given a context $\Gamma \in C$, write $\Gamma, x : \tau$ for $\Gamma \cup \{x : \tau\}$.

Definition 1.8. The **typability relation** $\Vdash \subseteq C \times \Lambda_\Pi \times \Pi$ is given by the following rules:

1. For every $\Gamma \in C$, $x \notin \operatorname{dom}\Gamma$ and $\tau \in \Pi$, have $\Gamma, x : \tau \Vdash x : \tau$.
2. Let $\Gamma \in C$, $x \notin \operatorname{dom}\Gamma$, $\sigma, \tau \in \Pi$ and $M \in \Lambda_\Pi$. If $\Gamma, x : \sigma \Vdash M : \tau$, then $\Gamma \Vdash (\lambda x : \sigma.M) : (\sigma \to \tau)$.
3. Let $\Gamma \in C$, $\sigma, \tau \in \Pi$, and $M, N \in \Lambda_\Pi$. If $\Gamma \Vdash M : (\sigma \to \tau)$ and $\Gamma \Vdash N : \sigma$, then $\Gamma \Vdash (MN) : \tau$.

If x appears on the left of a λ-abstraction, it is called a bound variable; otherwise, it is a free variable. We can replace all occurrences of a bound variable with another variable not appearing in the term; this is called α-equivalence. We will not define these notions precisely in this course because it's not worth it.

Definition 1.9. If M and N are λ-terms and x is a variable, define the **substitution** of x by N in M, written $M[x \coloneqq N]$, by

- $x[x \coloneqq N] = N$.
- $y[x \coloneqq N] = y$ for $y \neq x$.
- $(PQ)[x \coloneqq N] = P[x \coloneqq N]Q[x \coloneqq N]$.
- $(\lambda y : \sigma.P)[x \coloneqq N] = \lambda y : \sigma.(P[x \coloneqq N])$, where $x \neq y$ and y is not free in N.

The β**-reduction relation** $\to_\beta$ is the smallest relation on Λ_Π closed under the following rules:

1. $(\lambda x : \sigma.P)Q \to_\beta P[x \coloneqq Q]$
2. If $P \to_\beta P'$, then $\lambda x : \sigma.P \to_\beta \lambda x : \sigma.P'$ for all $x \in V$ and $\sigma \in \Pi$.
3. If $P \to_\beta P'$, then $PZ \to_\beta P'Z$ and $ZP \to_\beta ZP'$ for all $Z \in \Lambda_\Pi$.

The β**-equivalence relation** $\equiv_\beta$ is the smallest equivalence relation containing $\to_\beta$.

Example 1.10. Suppose we have a type $\mathbb{Z}$ of integers. Then

$$(\lambda x : \mathbb{Z}.(\lambda y : \tau.x))2 \to_\beta \lambda y : \tau 2.$$

When applying a β-reduction, the original term $(\lambda x : \sigma.P)Q$ is called a **redex** and its result $P[x \coloneqq Q]$ is called its β**-contraction**. Write $M \twoheadrightarrow_\beta N$ if M reduces to N after (potentially) multiple applications of β-contraction.

Also have η-reduction

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

which we will not discuss in detail.

We have the following bracketing conventions:

- KLM means $(KL)M$
- $\lambda x : \sigma \lambda y : \tau.M$ means $\lambda x : \sigma(\lambda y : \tau.M)$
- $\lambda x : \sigma.MN$ means $\lambda x : \sigma.(MN)$

Lemma 1.11 (Free Variable Lemma)**.** *Suppose* $\Gamma \Vdash M : \sigma$*. Then*

1. *If* $\Gamma' \supseteq \Gamma$*, then* $\Gamma' \Vdash M : \sigma$*.*
2. *The free variables of* M *all occur in* Γ*.*
3. *There is a context* $\Gamma^* \subseteq \Gamma$ *comprising exactly the free variables occurring in* M*, such that* $\Gamma^* \Vdash M : \sigma$*.*

Lemma 1.12 (Generation Lemma)**.**

1. *If* $\Gamma \Vdash x : \sigma$*, then* $x : \sigma \in \Gamma$*.*
2. *If* $\Gamma \Vdash MN : \sigma$*, then there is a type* τ *such that* $\Gamma \Vdash M : \tau \to \sigma$ *and* $\Gamma \Vdash N : \tau$*.*
3. *If* $\Gamma \Vdash (\lambda x : \tau.M) : \sigma$*, then there is a type* ρ *such that* $\Gamma, x : \tau \Vdash M : \rho$*, and* $\sigma = \rho \to \tau$*.*

Lemma 1.13 (Substitution Lemma)**.**

1. *If* $\Gamma \Vdash M : \sigma$ *and* $\alpha \in \Pi$*, then* $\Gamma[\alpha \coloneqq \tau] \Vdash M : \sigma[\alpha \coloneqq \tau]$*.*
2. *If* $\Gamma, x : \tau \Vdash M : \sigma$ *and* $\Gamma \Vdash N : \tau$*, then* $\Gamma \Vdash M[x \coloneqq N] : \sigma$*.*

The proofs of these are all easy inductions on the grammars.

Proposition 1.14 (Subject Reduction)**.** *If* $\Gamma \Vdash M : \sigma$ *and* $M \twoheadrightarrow_\beta N$*, then* $\Gamma \Vdash N : \sigma$*.*

Proof. Induction on the derivation $M \twoheadrightarrow_\beta N$, by the generation and substitution lemmas. □

Proposition 1.15 (Church-Rosser)**.** *Suppose that* $\Gamma \Vdash M : \sigma$*. If* $M \twoheadrightarrow_\beta N_1$ *and* $M \twoheadrightarrow_\beta N_2$*, then there exists* $L \in \Lambda_\Pi$ *such that* $N_1 \twoheadrightarrow_\beta L$*,* $N_2 \twoheadrightarrow_\beta L$ *and* $\Gamma \Vdash L : \sigma$*.*

The proof of Church-Rosser is the same as in the untyped case, and is therefore postponed.

$N \in \Lambda_\Pi$ is **in β-normal form** if it has no nontrivial β-reductions. If $M \twoheadrightarrow_\beta N$, then N is called the **β-normal form** of M. Indeed,

Corollary 1.16. *β-normal forms, when they exist, are unique.*

Proposition 1.17.

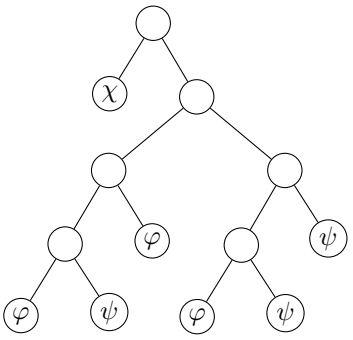
1. *If* $\Gamma \Vdash M : \sigma$ *and* $\Gamma \Vdash M : \tau$*, then* $\sigma = \tau$*.*
2. *If* $\Gamma \Vdash M : \sigma$*,* $\Gamma \Vdash N : \tau$ *and* $M \equiv_\beta N$*, then* $\sigma = \tau$*.*

Proof.

1. Easy induction.
2. The β-equivalence relation comprises a finite sequence of alternating upward and downward $\twoheadrightarrow_\beta$ relations. By induction on the number of these using (untyped) Church-Rosser, we get a term L which both M and N reduce to. By subject reduction, $\Gamma \Vdash L : \sigma$ and $\Gamma \Vdash L : \tau$, so, by (1), $\sigma = \tau$.

□

We want to show that, in fact, *every* $M \in \Lambda_\Pi$ admits a reduction to β-normal form. The idea is to view simple types as binary trees. The leaves are the type variables, and nodes (A, B) represent types $A \to B$. For example, consider the type $\chi \to (((\varphi \to \psi) \to \varphi) \to ((\varphi \to \psi) \to \psi))$. It has the following associated binary tree:



Example 1.18. There is no way to properly assign a type to $\lambda x : \tau.xx$.

Although typability is preserved by β-reduction, the converse is not true: there exist terms which are not themselves typable, but which β-reduce to typable terms.

For convenience, we denote types in λ-terms by superscripts: for example, a redex can be written

$$(\lambda x : \sigma.P^{\tau})^{\sigma\to\tau}Q^{\mu}.$$

Definition 1.19. The **height function** is the map $h : \Pi \to \mathbb{N}$ defined recursively by

1. $h(\tau) = 0$ for $\tau \in \mathcal{U}$
2. $h(\sigma \to \tau) = 1 + \max\{h(\sigma), h(\tau)\}$.

Equivalently, the height of a type is the height of its associated tree.

Define the **height** of a redex to be the height of its λ-abstraction, that is,

$$h((\lambda x : \sigma.P^{\tau})Q) \coloneqq h(\sigma \to \tau).$$

Define the **height** $h(M)$ of a λ-term M to be the maximal height of redexes appearing in M.

Theorem 1.20 (Weak Normalisation)**.** *Let $\Gamma \Vdash M : \sigma$. Then there is a finite chain of reductions*

$$M = M_0 \to_\beta M_1 \to_\beta \cdots \to_\beta M_n$$

such that M_n is in β-normal form.

Therefore, if $M \in \Lambda_\Pi$ is typable, its β-normal form exists, is unique, and can be computed: we have an effective procedure for finding β-normal form.

The reason this seems hard is that reductions can create new redexes, whose heights can (in principle) be arbitrarily large. In fact, we will show that, by always reducing the rightmost redex, the heights of any new redexes are bounded. So the proof proceeds by "taming the hydra" by induction on the complexity of M.

Proof. Define a function $m : \Delta_\Pi \to \mathbb{N}^2$:

- Set $m(M) = (0, 0)$ if M is in β-normal form.
- Otherwise, set $m(M) = (h(M), \mathrm{redex}(M))$, where $\mathrm{redex}(M)$ is the number of redexes of height $h(M)$ appearing in M.

We will induct on the lexicographic ordering of $m(M)$.

Indeed, suppose $\Gamma \Vdash M : \sigma$ is not in β-normal form, and let Δ be the rightmost redex of height $h(M)$. Reducing Δ may give rise to new redexes. All new redexes arise in one of the following ways:

- Writing $\Delta = (\lambda x : \tau.P)Q$, the redexes occurring in Q also appear in the reduction of Δ. Since Δ is the rightmost redex of height $h(\Delta)$, redexes occurring in Q have strictly smaller height.
- If
$$\Delta = (\lambda x : (\rho \to \mu). \dots xP^{\rho} \dots)^{(\rho\to\mu)\to\nu}(\lambda y : \rho.Q^{\mu}),$$
then Δ reduces to $\dots(\lambda y : \rho.Q^{\mu})P^{\rho}\dots$, which has height
$$h(\rho \to \mu) < h((\rho \to \mu) \to \nu) = h(\Delta).$$

- Suppose we have

  $$\Delta = (\lambda x : \tau.\lambda y : \rho : R^{\mu})^{\tau\to(\rho\to\mu)} P^{\tau} \text{ with } M = \ldots \Delta^{\rho\to\mu} Q^{\rho} \ldots$$

  If Δ reduces to some $\lambda y : \rho.R_1^{\mu}$, we then get a new redex

  $$(\lambda y : \rho.R_1^{\mu})Q^{\rho}$$

  of height $h(\rho \to \mu) < h(\tau \to (\rho \to \mu)) = h(\Delta)$.

- Suppose we have

  $$\Delta = (\lambda x : (\rho \to \mu).x)^{(\rho\to\mu)\to(\rho\to\mu)}(\lambda y : \rho.P^{\mu}) \text{ with } M = \ldots \Delta^{\rho\to\mu} Q^{\rho} \ldots.$$

  The reduction then yields a new redex $(\lambda y : \rho.P)Q$ of height

  $$h(\rho \to \mu) < h((\rho \to \mu) \to (\rho \to \mu)) = h(\Delta).$$

Now, Δ itself is gone, lowering the count by one, and any new redexes created have strictly smaller height. Hence the β-reduction of M at Δ has strictly smaller complexity m. □

A consequence is that β-equivalence of typable terms is decidable.

Theorem 1.21 (Strong Normalisation)**.** *Let $\Gamma \Vdash M : \sigma$. Then there is no infinite reduction path $M \to_\beta M_1 \to_\beta M_2 \to_\beta \ldots$*

We will not prove strong normalisation in this course.

1.3 Curry-Howard Correspondence

We can view propositions as types, and proofs as λ-terms in the simply typed λ calculus. We will start by exhibiting a correspondence between the simply typed λ-calculus and the **implication fragment** $IPC(\to)$ of IPC: that is, the fragment of IPC with one connective $\to$ and three deduction rules $(\to -I)$, $(\to -E)$ and (Ax).

Set $\mathcal{U}$ to be the set of primitive propositions of $IPC(\to)$; then the grammars generating Π and the propositions of $IPC(\to)$ are identical.

Proposition 1.22 (Curry-Howard correspondence for $IPC(\to)$)**.** *Let Γ be a context for $\lambda(\to)$, as above. Then*

1. *If $\Gamma \Vdash M : \varphi$, then $|\Gamma| \vdash_{IPC(\to)} \varphi$.*
2. *If $\Gamma \vdash_{IPC(\to)} \varphi$, then there is some $M \in \Lambda_{\Pi}^{\to}$ such that*

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

Proof.

1. Induction on the derivation of $\Gamma \Vdash M : \varphi$.

   If x is a variable not occurring in Γ' and the derivation is of the form $\Gamma', x : \varphi \Vdash x : \varphi$, then we need to prove that $|\Gamma', x : \varphi| \vdash_{\mathrm{IPC}(\to)} \varphi$. But $\varphi \vdash_{\mathrm{IPC}(\to)} \varphi$ by (Ax) and $|\Gamma', x : \varphi| = |\Gamma'| \cup \{\varphi\}$.

If the derivation is of the form $\Gamma \Vdash \lambda x : \sigma : N^\tau$, then $\varphi = \sigma \to \tau$, and we must have $\Gamma, x : \sigma \Vdash N : \tau$. By induction, we have $|\Gamma, x : \sigma| \vdash_{\mathrm{IPC}(\to)} \tau$; that is, $|\Gamma|, \sigma \vdash_{\mathrm{IPC}(\to)} \tau$. By $(\to -I)$, have $|\Gamma| \vdash_{\mathrm{IPC}(\to)} \sigma \to \tau$.

If the derivation is of the form $\Gamma \Vdash (PQ) : \varphi$, then we must have $\Gamma \Vdash P : (\sigma \to \varphi)$ and $\Gamma \Vdash Q : \sigma$. By induction, we have $|\Gamma| \vdash_{\mathrm{IPC}(\to)} \sigma \to \varphi$ and $|\Gamma| \vdash_{\mathrm{IPC}(\to)} \sigma$. By $(\to -E)$, we have $|\Gamma| \vdash_{\mathrm{IPC}(\to)} \varphi$.

2. Induction on the derivation of $|\Gamma| \vdash_{\mathrm{IPC}(\to)} \varphi$. Write $\Delta = \{(x_\psi : \psi) \mid \psi \in \Gamma\}$.

   If the derivation is of the form $\dfrac{}{\Gamma, \varphi \vdash \varphi}$, with $\varphi \notin \Gamma$ (wlog), then $\Delta, x_\varphi : \varphi \Vdash x_\varphi : \varphi$.

   If the derivation is of the form $\dfrac{\Gamma \vdash \psi \to \varphi \qquad \Gamma \vdash \psi}{\Gamma \vdash \varphi}$, then, by induction, we have $\Delta \Vdash M : \psi \to \varphi$ and $\Delta \vdash N : \psi$ for some $M, N \in \Lambda^{\to}_{\Pi}$. Then $\Delta \Vdash MN : \varphi$.

   If the derivation is of the form $\dfrac{\Gamma, \varphi \vdash \psi}{\Gamma \vdash \varphi \to \psi}$, then we have two cases:

   (i) If $\varphi \in \Gamma$, then, by induction, we have $\Delta \Vdash M : \psi$ for some $M \in \Lambda^{\to}_{\Pi}$. By weakening, we have $\Delta, x : \varphi \Vdash M : \psi$, where $x \notin \operatorname{dom} \Delta$. But then $\Delta \Vdash (\lambda x : \varphi . M) : \varphi \to \psi$.

   (ii) If $\varphi \notin \Gamma$, then, by induction, we have $\Delta, x_\varphi : \varphi \Vdash M : \psi$ for some $M \in \Lambda^{\to}_{\Pi}$. Then $\Delta \Vdash (\lambda x_\varphi : \varphi . M) : \varphi \to \psi$.

□

Examples 1.23. Let φ and ψ be primitive propositions, and consider the λ-term

$$\lambda f : (\varphi \to \psi) \to \varphi . \lambda g : \varphi \to \psi . g(fg)$$

By reading this λ-term, we can check it is of type

$$((\varphi \to \psi) \to \varphi) \to ((\varphi \to \psi) \to \psi),$$

which means it encodes a proof of that proposition in IPC($\to$). Explicitly, we get

$$\dfrac{\dfrac{\dfrac{f : [(\varphi \to \psi) \to \varphi] \qquad g : [\varphi \to \psi]}{fg : \varphi} \qquad g : [\varphi \to \psi]}{g(fg) : \psi}}{(\lambda g : \varphi \to \psi . g(fg)) : (\varphi \to \psi) \to \psi}}{(\lambda f : (\varphi \to \psi) \to \varphi . \lambda g : \varphi \to \psi . g(fg)) : ((\varphi \to \psi) \to \varphi) \to ((\varphi \to \psi) \to \psi)}$$

Definition 1.24. The **full simply typed λ-calculus** is an extension of the λ-calculus we have seen so far.

The set of types is given by the grammar

$$\Pi \coloneqq \mathcal{U} \mid \Pi \to \Pi \mid \underbrace{\Pi \times \Pi}_{\text{product types}} \mid \underbrace{\Pi + \Pi}_{\text{coproduct types}} \mid \underbrace{0}_{\text{initial type}} \mid \underbrace{1}_{\text{terminal type}}$$

where $\mathcal{U}$ is a set of type variables.

The set of terms is then given by the grammar

$$\begin{aligned}\Lambda_\Pi \coloneqq & V \mid \lambda V : \Pi.\Lambda_\Pi \mid \langle \Lambda_\Pi, \Lambda_\Pi \rangle \mid \pi_1(\Lambda_\Pi) \mid \pi_2(\Lambda_\Pi) \mid \iota_1(\Lambda_\Pi) \mid \iota_2(\Lambda_\Pi) \mid \\ & \mathrm{case}(\Lambda_\Pi; V.\Lambda_\Pi; V.\Lambda_\Pi) \mid \star \mid !_\Pi \Lambda_\Pi\end{aligned}$$

where V is a set of (term) variables and $\star$ is a constant.

These come with (additional) typing rules

$$\frac{\Gamma \Vdash M : \psi \times \varphi}{\Gamma \Vdash \pi_1(M) : \psi}$$

$$\frac{\Gamma \Vdash M : \psi \times \varphi}{\Gamma \Vdash \pi_2(M) : \varphi}$$

$$\frac{\Gamma \Vdash M : \psi \qquad \Gamma \Vdash M : \varphi}{\Gamma \Vdash \langle M, N \rangle : \psi \times \varphi}$$

$$\frac{\Gamma \Vdash M : \psi}{\Gamma \Vdash \iota_1(M) : \psi + \varphi}$$

$$\frac{\Gamma \Vdash M : \varphi}{\Gamma \Vdash \iota_2(M) : \psi + \varphi}$$

$$\frac{\Gamma \Vdash L : \psi + \varphi \qquad \Gamma, x : \psi \Vdash M : \rho \qquad \Gamma, y : \varphi \Vdash N : \rho}{\Gamma \Vdash \mathrm{case}(L; x : M; y : N) : \rho}$$

$$\frac{}{\Gamma \Vdash \star : 1}$$

$$\frac{\Gamma \Vdash M : 0}{\Gamma \Vdash !_\varphi M : \varphi} \text{ for each } \varphi \in \Pi$$

Product types record pairs of terms, and coproduct types record terms of *either* type. The type 1 is inhabited (up to β-equivalence) by exactly one term, whilst the existence of a term of type 0 makes every type inhabited. Terms of type 0 have no free variables.

These properties are enforced by the following new reduction rules:

- Projection: $\pi_1\langle M, N \rangle \to_\beta M$ and $\pi_2\langle M, N \rangle \to_\beta N$.
- Pairing: $\langle \pi_1 M, \pi_2 M \rangle \to_\eta M$.
- Definition by cases:

$$\begin{aligned}\mathrm{case}(\iota_1(M); x : K; y : L) &\to_\beta K[x \coloneqq M]; \\ \mathrm{case}(\iota_2(M); x : K; y : L) &\to_\beta L[x \coloneqq M].\end{aligned}$$

- Terminal type: If $\Gamma \Vdash M : 1$, then $M \to_\eta \star$.

We extend the Curry-Howard correspondence to associate

- 0 to $\bot$,
- $\varphi \times \psi$ to $\varphi \wedge \psi$,
- $\varphi + \psi$ to $\varphi \vee \psi$.

We will not prove the full correspondence (the proof would obviously be tedious and uninteresting).

Example 1.25. Consider the following proof of $(\varphi \wedge \chi) \to (\psi \to \varphi)$:

$$\dfrac{\dfrac{\dfrac{[\varphi \wedge \chi]}{\varphi} \qquad \psi}{\psi \to \varphi}}{(\varphi \wedge \chi) \implies (\psi \to \varphi)}$$

We can decorate this proof with the corresponding λ-terms, working from the top down:

$$\dfrac{\dfrac{\dfrac{p : [\varphi \wedge \chi]}{\pi_1(p) : \varphi} \qquad b : \psi}{\lambda b : \psi.\pi_1(p) : \psi \to \varphi}}{\lambda p : \varphi \times \chi.\lambda b : \psi.\pi_1(p) : (\varphi \wedge \chi) \to (\psi \to \varphi)}$$

To summarise, we have the following correspondence:

λ-calculus	IPC
(primitive) types	(primitive) propositions
variables	hypotheses
λ-terms	proofs
type constructors	connectives
type inhabitation	provability
term reduction	proof normalisation

1.4 Semantics for IPC

Definition 1.26. A **lattice** is a set L equipped with commutative, associative binary operations $\vee$ (meet) and $\wedge$ (join) satisfying the absorption laws

$$a \vee (a \wedge b) = a;\ a \wedge (a \vee b) = a;\ \text{for all } a, b \in L.$$

A lattice is:

- **Distributive** if

$$a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c) \text{ for all } a, b, c \in L.$$

- **Bounded** if there are elements $\bot, \top \in L$ such that

$$a \vee \bot = a;\ a \wedge \top = a \text{ for all } a \in L.$$

- **Complemented** if it is bounded and, for every $a \in L$, there is some **complement** $\neg a \in L$ such that $a \wedge (\neg a) = \bot$ and $a \vee (\neg a) = \top$.

A **Boolean algebra** is a complemented distributive lattice.

Note that the absorption laws force $\wedge$ and $\vee$ to be idempotent. Moreover, a lattice admits a partial ordering induced by $\wedge$: $a \leq b \iff a \wedge b = a$.

Example 1.27. The power set $\mathcal{P}A$ of a set A is a boolean algebra with meet $\cap$, join $\cup$, top element A, bottom element $\emptyset$ and complement $A \setminus (-)$. The induced ordering is inclusion $\subseteq$.

Proposition 1.28. *Let L be a bounded lattice with induced relation $\leq$. Then $\leq$ is indeed a partial order; it has least element $\bot$ and greatest element $\top$. Further, for $a, b \in L$ we have $a \wedge b = \inf\{a, b\}$ and $a \vee b = \sup\{a, b\}$.*

Proof. Fix $a, b, c \in L$.

Since $a \wedge a = a$, we have $a \leq a$.

If $a \leq b$ and $b \leq c$, then $a \wedge b = a$ and $b \wedge c = b$; then $a \wedge c = (a \wedge b) \wedge c = a \wedge (b \wedge c) = a \wedge b = a$, so $a \leq c$.

If $a \leq b$ and $b \leq a$, then $a = a \wedge b = b \wedge a = b$.

Hence $\leq$ is indeed a partial order.

If $c \leq a$ and $c \leq b$, then $c = a \wedge c = b \wedge c$, so $c \wedge (a \wedge b) = (c \wedge a) \wedge b = c \wedge b = c$, so $c \leq a \wedge b$. Also, $a \wedge (a \wedge b) = (a \wedge a) \wedge b = a \wedge b$, so $a \leq a \wedge b$ (and by symmetry $b \leq a \wedge b$). Hence $a \wedge b = \inf\{a, b\}$; dually, $a \vee b = \sup\{a, b\}$.

By the boundedness conditions, $\bot$ and $\top$ are, respectively, the least and greatest elements of L wrt $\leq$. □

Observe that a finite poset has a bounded lattice structure since binary suprema and infima always exist.

Classically, we say that $\Gamma \vDash T$ if, for every valuation $v : P \to \{0, 1\}$ with $v(\Gamma) = \{1\}$, we have also $v(T) = 1$. Now, we can view valuations as a extensions of a set map on the primitive propositions to a map of Boolean algebras. What happens if we put a more complex Boolean algebra in the codomain of v instead of $\{0, 1\}$? Could we prove a completeness theorem for IPC?

In general, this cannot work: Boolean algebras are complemented lattices, so, using the standard logical interpretation of $\neg$ as 'not', the LEM holds by definition.

We need a more general structure than a Boolean algebra if we hope to model IPC, yet this structure must also be able to model implication. A Boolean lattice does this by defining $a \Rightarrow b$ in terms of $\vee$ and the complement $(-)^*$. We take a more direct approach.

Definition 1.29. A **Heyting algebra** H is a bounded lattice equipped with a binary operation $\Rightarrow$ such that, for $a, b, c \in H$, we have $(a \wedge b) \leq c$ iff $a \leq (b \Rightarrow c)$. A **morphism of Heyting algebras** is a map commuting with $\vee$, $\wedge$ and $\Rightarrow$.

Examples 1.30.

1. A Boolean algebra is a Heyting algebra, with $(a \Rightarrow b) = \neg a \vee b$. In particular, power sets are Heyting algebras. Note that $\neg a = (a \Rightarrow \bot)$.

2. Every topology on a set X is a Heyting algebra, with $U \Rightarrow V = (X \setminus U \cup V)^\circ$.
3. A finite distributive lattice is a Heyting algebra (ES2).
4. A bounded total order (with top element 1) is a Heyting algebra, with
$$x \Rightarrow y = \begin{Bmatrix} 1 & x \leq y \\ y & \text{o/w} \end{Bmatrix}.$$
5. Let $T \subseteq \mathcal{L}$ be a propositional theory (with connectives $\wedge$, $\vee$, $\Rightarrow$ and constant $\bot$). Consider the equivalence relation $\sim$ on $\mathcal{L}$ by $A \sim B$ iff $T \vdash_{\text{IPC}} A \Leftrightarrow B$, and descend the Heyting algebra operations on $\mathcal{L}$ to $\mathcal{L}/\sim$. We show on ES2 that the result is a Heyting algebra; it is called the **Lindenbaum-Tarski algebra** of T. For $T = \emptyset$, we get the **free Lindenbaum-Tarski algebra** on $\mathcal{L}$.

Definition 1.31. Let H be a Heyting algebra, and let $\mathcal{L}$ be a propositional language generated by a set P of primitive propositions. Then an H-valuation is a function $v : P \to H$, which is extended to $\mathcal{L}$ recursively by setting $v(\bot) = \bot$, $v(P \wedge Q) = v(P) \wedge v(Q)$, $v(P \vee Q) = v(P) \vee v(Q)$, and $v(P \to Q) = P \Rightarrow Q$.

A proposition $P \in \mathcal{L}$ is **H-valid** if $v(P) = \top$ for all H-valuations v, and is an **H-consequence** of a (finite) set of propositions Γ if $v(\bigwedge \Gamma) \leq v(P)$ (written $\Gamma \vDash_H P$).

A valuation descends to a Heyting-algebra morphism on the free Lindenbaum-Tarski algebra.

Lemma 1.32 (Soundness of Heyting semantics)**.** *Let H be a Heyting algebra and $v : \mathcal{L} \to H$ a valuation. If $\Gamma \vdash_{IPC} A$, then $\Gamma \vDash_{H,v} A$.*

From now on, $\vdash$ means $\vdash_{\text{IPC}}$ unless otherwise specified.

Proof. Induction on the derivation of the proof.

(Ax): $v(\bigwedge \Gamma \wedge A) = v(\bigwedge \Gamma) \wedge v(A) \leq v(A)$.

$(\wedge - I)$: Let $\varphi = B \wedge C$, and suppose $\Gamma \vdash B$ and $\Gamma \vdash C$. By induction, $v(\bigwedge \Gamma) \leq v(B)$ and $v(\bigwedge \Gamma) \leq v(C)$, so
$$v(\bigwedge \Gamma) = v(\bigwedge \Gamma) \wedge v(\bigwedge \Gamma) \leq v(B) \wedge v(C) \leq v(B \wedge C) = v(\varphi).$$

$(\to - I)$: Let $\varphi = (B \to C)$, and suppose $\Gamma, B \vdash C$. By induction, $v(\bigwedge \Gamma) \wedge v(B) \leq v(C)$. But this is equivalent to saying that
$$v(\bigwedge \Gamma) \leq (v(B) \Rightarrow v(C)) = v(B \to C) = v(\varphi).$$

$(\vee - I)$: Let $\varphi = B \vee C$, and suppose $\Gamma \vdash B$. By induction, $v(\bigwedge \Gamma) \leq v(B)$, but $v(B) \leq v(B) \vee v(C) = v(\varphi)$.

$(\wedge - E)$: Suppose $\Gamma \vdash A \wedge B$; by induction, $v(\bigwedge \Gamma) \leq v(A) \wedge v(B)$. But $v(A) \wedge v(B) \leq v(A), v(B)$.

$(\to -E)$: Suppose $\Gamma \vdash A \to B$ and $\Gamma \vdash A$; by induction, $v(\bigwedge \Gamma) \leq (v(A) \Rightarrow v(B))$ and $v(\bigwedge \Gamma) \leq v(A)$. Thus $v(\bigwedge \Gamma) = v(\bigwedge \Gamma) \wedge v(A) \leq v(B)$.

$(\vee - E)$: Suppose $\Gamma \vdash A \vee B$, $\Gamma, A \vdash C$ and $\Gamma, B \vdash C$. By induction, $v(\bigwedge \Gamma) \leq v(A) \vee v(B)$ and $v(\bigwedge \Gamma) \wedge v(A) \leq v(B), v(C)$. Then

$$v(\bigwedge \Gamma) = v(\bigwedge \Gamma) \wedge (v(A) \vee v(B)) = (v(\bigwedge \Gamma) \wedge v(A)) \vee (v(\bigwedge \Gamma) \wedge v(B)) \leq v(C) \vee v(C) = v(C),$$

since Heyting algebras are distributive lattices (see example sheet).

$(\bot - E)$: Suppose $\Gamma \vdash \bot$. By induction, $v(\bigwedge \Gamma) \leq v(\bot) = \bot$. By minimality, $v(\bigwedge \Gamma) = \bot$, but $\bot \leq v(P)$ for any proposition P. □

Example 1.33. LEM is not intuitionistically valid. Indeed, let p be a primitive proposition, and consider the Heyting algebra given by the topology on the Sierpiński space $S = \{0, 1\}$ (with $\{1\}$ open).

We can define a valuation v with $v(p) = \{1\}$, in which case

$$v(\neg p) = (Sv(p))^\circ = \{0\}^\circ = \emptyset,$$

and so $v(p \vee \neg p) = \{1\} \neq S = \top$. By soundness, there is no proof of $p \vee \neg p$ in IPC.

Example 1.34 (Pierce's Law)**.** The proposition $((p \to q) \to p) \to p$ is not intuitionistically valid, either. Indeed, take the valuation on $\mathbb{R}^2$ (with the Euclidean topology) mapping $p \to \mathbb{R}^2 \setminus \mathbf{0}$ and $q = \emptyset$.

Classical completeness states that that $\Gamma \vdash_{\text{CPC}} \varphi$ iff $\Gamma \vDash_Z \varphi$. Now, there is no *finite* Heyting algebra H such that $\Gamma \vdash_{\text{IPC}}$ iff $\Gamma \vDash_H \varphi$. However, if φ is a H-consequence of Γ in *all* Heyting algebras H, then we do indeed have a proof of φ from Γ in IPC.

Theorem 1.35 (Completeness of Heyting semantics)**.** *A formula φ is H-valid in all Heyting algebras H such that $\vDash_H \Gamma$ if and only if $\Gamma \vdash_{\text{IPC}} \varphi$. In particular, φ is H-valid in all Heyting algebras if and only if $\vDash \text{IPC}\varphi$.*

Proof. One direction follows from soundness. Conversely, the Lindenbaum-Tarski algebra $\mathcal{L}/\sim$ of Γ is a Heyting algebra, as shown on ES2. The map $v : P \to \mathcal{L}/\sim$ by $p \to [p]$ extends to a valuation $\mathcal{L} \to \mathcal{L}/\sim$; by induction, this is just the natural quotient $\psi \to [\psi]$. Since $\vDash_{\mathcal{L}/\sim, v} \Gamma$, we have $\vDash_{\mathcal{L}/\sim, v} \varphi$. Then $[\varphi] = \top$, so we have a proof $\vdash \varphi$ by definition of the relation $\sim$. □

Finding Heyting algebra counterexamples can be difficult, so we will now build a simpler class of Heyting algebras out of posets, and show that this class satisfies a form of completeness.

Definition 1.36. Let S be a poset, and fix $a \in S$. The **partial upset** of $a \in S$ is $\uparrow a = \{b \in S \mid a \leq b\}$. A subset $U \subseteq S$ is a **terminal segment** if $\uparrow c \subseteq U$ for all $c \in U$. Write $T(S)$ for the set of terminal segments of S.

Proposition 1.37. *Let S be a poset. Then $T(S)$ can be given the structure of a Heyting algebra.*

Note that Heyting algebra structures on posets are unique (exercise).

Proof. Order $T(S)$ by inclusion; since terminal segments are closed under $\cup$ and $\cap$, set meet and join, respectively, to these operations. Then $\emptyset$ and S are terminal segments, and so bottom and top elements. It remains to define Heyting implication.

Let $U, V \in T(S)$. Define

$$(U \Rightarrow V) = \{x \in S \mid {\uparrow}x \cap U \subseteq V\}$$

Then, if $U, V, W \in T(S)$, we have that $W \subseteq (U \Rightarrow V)$ iff

$$({\uparrow}w \cap U \subseteq V \ \forall w \in W), \text{ iff } (\forall w \in W, u \in U : (u \in v \text{ if } w \leq u)).$$

But $U \in T(S)$, so $w \leq u$ implies that $u \in W$. This makes the last condition equivalent to saying $U \cap W \subseteq V$. □

Definition 1.38. Let v be a $T(S)$-valuation. For a proposition φ and $s \in S$, say s **forces** φ (written $s \vDash \varphi$) if $s \in v(\varphi)$. The relation $\vDash$ is called the **forcing relation**.

Note that φ is valid (under v) iff every $s \in S$ forces φ. Since every $v(\varphi)$ is a terminal segment, $\vDash$ satisfies the **persistence property**: if $s \vDash \varphi$ and $s \leq s'$, then also $s' \vDash \varphi$.

We can interpret S as the set of all possible worlds, ordered by knowledge. Indeed, $\vDash$ is well-behaved with respect to this idea.

Proposition 1.39. *Let v be a $T(S)$-valuation. Then $\vDash$ satisfies the following:*

(i) For every primitive proposition p and $s \in S$, $s \vDash p$ iff $s \in v(p)$.

(ii) There is no $s \in S$ with $s \vDash \bot$.

(iii) $s \vDash \varphi \wedge \psi$ iff $s \vDash \varphi$ and $s \vDash \psi$.

(iv) $s \vDash \varphi \vee \psi$ iff $s \vDash \varphi$ or $s \vDash \psi$.

(v) $s \vDash (\varphi \to \psi)$ iff, for every $s' \geq s$, $s' \vDash \varphi$ implies that $s \vDash \psi$.

Proof.

(i) By definition.

(ii) $v(\bot) = \bot = \emptyset$

(iii) v preserves finite meets.

(iv) v preserves finite joins.

(v) $s \vDash (\varphi \to \psi)$ iff $s \in (v(\varphi) \Rightarrow v(\psi))$ iff $(s{\uparrow}) \cap v(\varphi) \leq v(\psi)$. This is equivalent to saying that, for all $s' \geq s$, $s' \in v(\varphi)$ implies that $s \in v(\psi)$.

□

Corollary 1.40. *Let φ be a proposition, and let $s \in S$. Then*

1. *$s \vDash \neg\varphi$ iff $s' \nVdash \varphi$ for every $s' \geq s$.*
2. *$s \vDash \neg\neg\varphi$ iff, for each $s' \geq s$, there is some $s'' \geq s'$ with $s'' \vDash \varphi$.*

Intuitively, a world not believing in φ does not mean it believes in $\neg\varphi$ – it might be indifferent to φ.

Proof. Same idea as the last proposition. □

Abstracting away the valuation, we get the following definition.

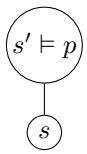
Definition 1.41. Let P be a set of primitive propositions. A **Kripke model** is a triple $(S, \leq, \vDash)$, where $(S, \leq)$ is a poset and $\vDash \subseteq S \times P$ is a relation satisfying the **persistence property**: if $s \vDash p$, then $s' \vDash p$ for all $s' \geq s$.

The elements of S are called **worlds** or **states**, and its order $\leq$ is called the **accessibility relation**; the relation $\vDash$ is called the **forcing relation**.

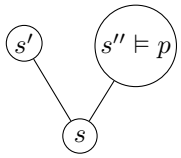
We extend $\vDash$ recursively to an **extended forcing relation** $\vDash \subseteq S \times \mathcal{L}$ using the clauses (ii)-(v) in the last proposition. By an easy induction, the persistence property for $\vDash$ in fact holds on all $\mathcal{L}$.

Write $S \vDash \varphi$ if $s \vDash \varphi$ for all $s \in S$.

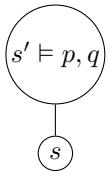
Example 1.42. Consider the following Kripke models (they clearly satisfy persistence). Once we have proved soundness of IPC for these models, they will provide easy-to-construct counterexamples to non-intuitionistically-valid statements.



Since $s \leq s' \vDash p$, $s \nvDash \neg p$, but also $s \nvDash p$, so $s \nvDash p \vee \neg p$. It is also the case that $s \vDash \neg\neg p$, so $s \nvDash (\neg p \to p)$.



Since $s' \geq s$ does not have a greater world forcing p, $s \nvDash \neg\neg p$. Since $s'' \geq s$ forces p, $s \nvDash \neg p$. Hence $s \nvDash (\neg\neg p \vee \neg p)$.



We will show $s \nvDash (p \to q) \to (\neg p \vee q)$. Indeed, since all worlds above s forcing p also force q, $s \vDash p \to q$. On the other hand, $s \nvDash q$ and $s \nvDash \neg p$ (as $s \leq s' \vDash p$), so $s \nvDash \neg p \vee q$.

Definition 1.43. Let H be a Heyting algebra. A **filter** on H is a nonempty terminal segment closed under binary meets. Explicitly:

$$F \neq \emptyset;\ x \leq y,\ x \in F \implies y \in F;\ x, y \in F \implies x \wedge y \in F.$$

A filter F is **proper** if $F \subsetneq H$, and a proper filter is **prime** if, for every disjunction $x \vee y \in F$, either $x \in F$ or $y \in F$.

Suppose F is a (proper) filter with $x \notin F$. We will show on the example sheet that there is a prime filter $G \supseteq F$ that still doesn't contain x.

Lemma 1.44. *Let H be a Heyting algebra and v a H-valuation. Then there is a Kripke model $(S, \leq, \vDash)$ such that, for every proposition φ, we have $\vDash_{H,v} \varphi$ iff $S \vDash \varphi$.*

Proof. Let S be the set of all prime filters of H (ordered by inclusion). For a primitive proposition p, say that $F \vDash p$ iff $v(p) \in F$.

Claim: $F \vDash \varphi$ iff $v(\varphi) \in F$ for *any* proposition.

This is an easy induction (by definition of a prime filter), apart from the implication case, which we will cover here.

Indeed, suppose that we have a (prime) filter F with $F \vDash (\psi \to \varphi)$, but $v(\psi \to \varphi) \notin F$. Let G' be the least filter containing both F and $v(\psi)$; we will show on the example sheet that

$$G' = \{b \mid \exists f \in F : f \wedge v(\psi) \leq b\}.$$

Now, $v(\varphi) \notin G'$: otherwise, we can find some $f \in F$ such that $f \wedge v(\psi) \leq v(\varphi)$. Then $f \leq v(\psi \to \varphi)$, so $v(\psi \to \varphi) \in F_{\#}$ as F is a filter. In particular, G' is a proper filter, so there is a prime filter G extending G', and so containing $v(\psi)$, yet not containing $v(\varphi)$. Now, by induction, $G \vDash \psi$; since G also contains F, have also $G \vDash (\psi \to \varphi)$ by (extended) persistence. Hence $G \vDash \varphi$ (by induction); but then $v(\varphi) \in G$.$_{\#}$

Conversely suppose that $v(\psi \to \varphi) \in F$ and G is a prime filter extending F and forcing ψ. By the induction hypothesis, $v(\psi) \in G$; since $G \supseteq F$, also $v(\psi \to \varphi) \in G$. But then

$$v(\varphi) \geq v(\psi) \wedge v(\psi \to \varphi)$$

since G is a filter; hence $v(\varphi) \in G$. By induction, $G \vDash \varphi$.

Now we prove the proposition. If $\vDash_{H,\varphi} \varphi$, then $v(\varphi) = \top$. Hence φ lies in every filter of H, and so $S \vDash \varphi$. Conversely, suppose $S \vDash \varphi$, and suppose $\nvDash_{H,\varphi} \varphi$. Then $v(\varphi) \neq \top$, so $\{\top\}$ is a proper filter of H not containing φ. But we can extend $\{\top\}$ to a prime filter G that still doesn't contain $v(\varphi)$, so then $G \nvDash \varphi$.$_{\#}$ □

Theorem 1.45 (Completeness of Kripke semantics)**.** *Let φ be a proposition. Then $\Gamma \vdash_{IPC} \varphi$ iff, for all Kripke models $(S, \leq, \vDash)$ with $S \vDash \Gamma$, we also have $S \vDash \varphi$.*

Proof. If $\Gamma \vdash_{\mathrm{IPC}} \varphi$ and $S \vDash \Gamma$, then $S \vDash \varphi$ by induction on the grammar.

Suppose $\Gamma \nvdash_{\mathrm{IPC}} \varphi$. Then there is a Heyting algebra H and valuation v such that $\vDash_{H,v} \Gamma$ but $\nvDash_{H,v} \varphi$. By the last lemma, $S \vDash_{H,v} \Gamma$ but $S \nvDash_{H,v} \varphi$. □

1.5 Negative Translations

We try to convert classical proofs into intuitionistic ones. Of course, this is not always possible, so we will end up proving an intuitionistically weaker statement.

Definition 1.46. The negative translation of a proposition φ, written φ^N, is defined by:

- If p is a primitive proposition, then $p^n = \neg\neg p$.
- $(\varphi \wedge \psi)^N = \varphi^N \wedge \psi^N$.
- $(\varphi \to \psi)^N = \varphi^N \to \psi^N$.
- $(\varphi \vee \psi)^N = \neg(\neg\varphi^N \wedge \neg\psi^N)$.
- $(\neg\varphi)^N = \neg\varphi^N$.

Recall that a proposition is classically valid iff it is valid in every Boolean algebra. We therefore want to investigate the relationship between Heyting algebras and Boolean algebras.

Lemma 1.47. *Let H be a Heyting algebra. The map $\varphi \to \neg\neg\varphi$ preserves $\wedge$ and $\to$.*

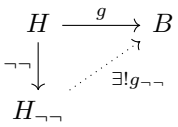
Proof. Example sheet. □

Definition 1.48. Let H be a Heyting algebra. The **regularisation** of H is the subset $H_{\neg\neg} \coloneqq \{x \in H \mid \neg\neg x = x\}$ of **regular elements**.

Note that, in particular, the last lemma guarantees that disjunctions and implications all lie in $H_{\neg\neg}$.

Lemma 1.49 (Regularisation lemma)**.** *$H_{\neg\neg}$ is a Boolean algebra satisfying the following universal property.*

Let $\neg\neg : H \to \neg\neg H$ by $\varphi \to \neg\neg\varphi$ be the natural map into $H_{\neg\neg}$. For every Heyting morphism $g : H \to B$, where B is a Boolean algebra, there is a unique Boolean morphism $g_{\neg\neg} : H_{\neg\neg} \to B$ such that $g_{\neg\neg} \circ (\neg\neg) = g$.



Proof. Give $H_{\neg\neg}$ the order on H, so that $\wedge$, $\bot$ and $\to$, all of which commute with $\neg\neg$, remain the same as in H. It remains to define disjunctions. Indeed, it is easy to check that

$$\sup\{a, b\} = \neg\neg(a \vee b) \coloneqq a \vee_{\neg\neg} b$$

in $H_{\neg\neg}$. We then show on the example sheet that, since all elements of $H_{\neg\neg}$ are regular, $H_{\neg\neg}$ is a Boolean algebra.

Given a Heyting morphism $g : H \to B$, where B is a Boolean algebra, define $(g_{\neg\neg} : H_{\neg\neg} \to B) \coloneqq g|_{H_{\neg\neg}}$. This clearly preserves $\wedge$, $\to$, $\bot$ and $\top$; it remains to check $\vee$:

$$g_{\neg\neg}(a \vee_{\neg\neg} b) = g(\neg\neg(a \vee b)) = \underbrace{\neg\neg(g(a) \vee g(b)) = g(a) \vee g(b)}_{B \text{ boolean}}.$$

Hence $g_{\neg\neg}$ is a Boolean morphism. We also have $g_{\neg\neg} \circ (\neg\neg) = g$:

$$g_{\neg\neg}(\neg\neg x) = g(\neg\neg x) = \neg\neg g(x) = g(x).$$

Finally, if $h : H_{\neg\neg} \to B$ is some other Boolean morphism satisfying the same universal property, then

$$h(a) = h(\neg\neg a) = g(a) = g_{\neg\neg}(a) \text{ for } a \in H_{\neg\neg},$$

so in fact $h = g_{\neg\neg}$. □

In particular, if S is a set then regularisation of the free Heyting algebra on S is the free Boolean algebra on S.

Theorem 1.50 (Glivenko)**.** *Let φ and ψ be propositions. Then*

$$\vdash_{CPC} (\varphi \to \psi) \text{ if and only if } \vdash_{IPC} (\neg\neg\varphi \to \neg\neg\psi).$$

Proof. Suppose $\vdash_{\text{CPC}} (\varphi \to \psi)$, and let H be a Heyting algebra. Then $\varphi \to \psi = \top$ in the Boolean algebra $H_{\neg\neg}$. Since the inclusion $H_{\neg\neg} \overset{i}{\hookrightarrow} H$ strictly preserves $\leq$ and $\to$, we have

$$\top = i(\top) = i(\varphi \to \psi) = \varphi \to \psi = (\neg\neg(\varphi \to \psi)) = \neg\neg\varphi \to \neg\neg\psi$$

in H. This means $\vdash_{\text{IPC}} (\neg\neg\varphi \to \neg\neg\psi)$.

The converse is clear. □

Corollary 1.51. *Let φ be a proposition. Then $\vdash_{CPC} \varphi$ iff $\vdash_{IPC} \varphi^N$.*

Proof. Easy induction on the grammar, by the last theorem. □

Note that, in particular, IPC and CPC agree on the validity of negative theorems $\neg\varphi$.

Corollary 1.52. *CPC is inconsistent iff IPC is inconsistent.*

Proof. If CPC is inconsistent, then there is some φ such that $\vdash_{\text{CPC}} \varphi$ and $\vdash_{\text{CPC}} \neg\varphi$. But then $\vdash_{\text{IPC}} \neg\neg\varphi$ and $\vdash_{\text{IPC}} \neg\varphi$, so IPC is inconsistent.

The converse is clear. □

2 Computability

2.1 Recursive functions and λ-computability.

Definition 2.1. The class of **recursive functions** of form $\mathbb{N}^k \rightharpoonup \mathbb{N}$ is the smallest class of functions containing the following basic functions:

- the i^{th} projection $\Pi_i^m : (n_1, \ldots, n_m) \to n_i$;
- the successor $S^+ : n \to n + 1$;
- the zero map $n \to 0$;

and closed under the following operations:

- composition: from $g : \mathbb{N}^k \rightharpoonup \mathbb{N}$ and $h_1, \ldots, h_k : \mathbb{N}^m \rightharpoonup \mathbb{N}$, produce $f : \mathbb{N}^k \rightharpoonup \mathbb{N}$ given by

$$f(n_1, \ldots, n_k) = f(\mathbf{n}) = g(h_1(\mathbf{n}), \ldots, h_k(\mathbf{n}));$$

- primitive recursion: from $g : \mathbb{N}^m \rightharpoonup \mathbb{N}$ and $h : \mathbb{N}^{k+m} \rightharpoonup \mathbb{N}$, produce $f : \mathbb{N}^{k+1} \rightharpoonup \mathbb{N}$ given by

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

- minimisation: from $g : \mathbb{N}^{k+1} \to \mathbb{N}$, produce $f : \mathbb{N}^k \rightharpoonup \mathbb{N}$ given by

$$f(\mathbf{n}) = \begin{Bmatrix} \min\{m \in \mathbb{N} \mid f(m, \mathbf{n}) = 0\} & f(m, \mathbf{n}) = 0 \text{ for some } m \in \mathbb{N} \\ \uparrow & \text{otherwise} \end{Bmatrix}.$$

In the last case, we write $f(\mathbf{n}) = \mu m.(g(m, \mathbf{n}) = 0)$.

Let V be an infinite set of variables. The terms of the **untyped λ-calculus** are given by the grammar

$$\Lambda \coloneqq V \mid \lambda V.\Lambda \mid \Lambda\Lambda$$

The notions of α-equivalence, β, η-reduction, and so on, apply as with the simply-typed λ-calculus.

Example 2.2. Consider the λ-terms $\omega = \lambda x.xx$ and $\Omega = \omega\omega$. Then

$$\Omega = (\lambda x.xx)(\lambda x.xx) \to_\beta (\lambda x.xx)(\lambda x.xx)$$

so Ω β-reduces to itself. Since Ω is not in β-normal form, we have an infinite chain of β-reductions. In fact, Ω does not have a β-normal form.

This means that, unlike the simply typed λ-calculus, the untyped version is *not* strongly normalisable, nor do β-normal forms always exist (this will correspond to the fact that some algorithms don't terminate). However, we do still have the Church-Rosser theorem.

Theorem 2.3 (Church-Rosser)**.** *Let $L, M_1, M_2 \in \Lambda$, and let $L \twoheadrightarrow_\beta M_1, M_2$. Then there is some $N \in \Lambda$ such that $M_1, M_2 \twoheadrightarrow_\beta N$.*

Proof. Example sheet. □

This has some important consequences: β-equivalent terms must both β-reduce to some common term, and so β-normal forms, when they exist, are unique. Further, if two terms have different BNFs, then they cannot be β-equivalent. For example, $\lambda x.x \not\equiv_\beta \lambda x.\lambda y.x$

Intuitively, we view λ-terms as functions. We want to show that the λ-calculus can in fact encode *all* computation: that is, all recursive functions. Immediately, we have a problem: how do we represent the naturals $\mathbb{N}$?

Definition 2.4. Let $n \in \mathbb{N}$. Its corresponding **Church numeral** c_n is the λ-term

$$c_n \coloneqq \lambda s.\lambda z.\underbrace{s(s\ldots(s}_{n \text{ times}} z)\ldots) = \lambda s.\lambda z.s^n z$$

Intuitively, c_n maps a function s to its n^{th} composite.

Example 2.5. Intuitively, $c_0 = \lambda s.\lambda z.z$ maps a function s to the identity map $z \to z$. Then $c_1 = \lambda s.\lambda z.sz$ is the identity map on functions.

Using this encoding of $\mathbb{N}$, we can encode (some) *functions* $\mathbb{N}^k \rightharpoonup \mathbb{N}$.

Definition 2.6. A function $f : \mathbb{N}^k \rightharpoonup \mathbb{N}$ is **definable** if there is a λ-term $F \in \Lambda$ such that $Fc_{n_1}c_{n_2}\ldots c_{n_k} \equiv_\beta c_{f(\mathbf{n})}$ for all $\mathbf{n} \in \mathbb{N}^k$ such that $f(\mathbf{n})\downarrow$.

First, observe that the core arithmetic functions are definable.

Proposition 2.7 (Rosser)**.** *Consider the following λ-terms:*

- $A_+ \coloneqq \lambda x.\lambda y.\lambda s.\lambda z.xs(ysz)$.
- $A_\times \coloneqq \lambda x.\lambda y.\lambda s.x(ys)$.
- $A_e \coloneqq \lambda x.\lambda y.yx$.

Then, for all $n, m \in \mathbb{N}$, we have $A_+c_nc_m = c_{n+m}$, $A_\times c_nc_m = c_{n\times m}$, and, for $m > 0$, $A_ec_nc_m = c_{n^m}$.

Proof. We will prove the result for A_+; the other two are analogous computations. First, note that

$$c_nsz = (\lambda f\lambda x f^n(x))sz \equiv_\beta s^n z.$$

Then we compute

$$\begin{aligned} A_+c_nc_m &= (\lambda x\lambda y\lambda s\lambda z.xs(ysz)).c_nc_m \\ &\equiv_\beta (\lambda y\lambda s\lambda.zc_ns(ysz))c_m \\ &\equiv_\beta \lambda s\lambda z.c_ns(c_msz) \\ &\equiv_\beta \lambda s.\lambda z.s^n(s^m z) \\ &= \lambda s.\lambda z.s^{n+m}z = c_{n+m} \end{aligned}$$

□

Similarly, we can encode some logic.

Proposition 2.8. *Consider the following λ-terms:*

- $\top \coloneqq \lambda x.\lambda y.x.$
- $\bot \coloneqq \lambda x.\lambda y.y.$
- $(\textit{if } B \textit{ then } P \textit{ else } Q) \coloneqq BPQ.$

Then, for $P, Q \in \Lambda$, we have $(\textit{if } \top \textit{ then } P \textit{ else } Q) \equiv_\beta P$ and $(\textit{if } \bot \textit{ then } P \textit{ else } Q) \equiv_\beta Q$.

Proof. Easy computation. □

Using this, we can encode (classical) logical connectives:

- $\neg P \coloneqq$ if P then $\bot$ else $\top$.
- $\wedge PQ \coloneqq$ if P then (if Q then $\top$ else $\bot$) else $\bot$.
- $\vee PQ \coloneqq$ if P then $\top$ else (if Q then $\top$ else $\bot$).

We can also encode pairs: define

$$[P, Q] \coloneqq \lambda x.xPQ;$$

then $[P, Q]\top \equiv_\beta P$ and $[P, Q]\bot \equiv_\beta Q$, so $\top$ and $\bot$ act as projections. By induction, we can define arbitrary n-tuples. Note, however, that, in general, $[M\top, M\bot] \not\equiv_\beta M$.

We would like to define λ-terms recursively, within the λ-calculus. We can view such a term as the solution to the fixed point equation $F = (\lambda x.M)F$.

Definition 2.9. A **combinator** is a λ-term with no free variables.

Theorem 2.10. *There is a combinator Y such that, for all $F \in \Lambda$,*

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

Proof. Define

$$Y = \lambda f.(\lambda x.f(xx))\lambda x.f(xx)$$

For $F \in \Lambda$, we can compute

$$YF \equiv_\beta (\lambda x.F(xx))\lambda x.F(xx) \equiv_\beta F(\lambda x.F(xx)\lambda x.F(xx)) = F(YF).$$

□

Definition 2.11. A term $Z \in \Lambda$ satisfying $F(ZF) \equiv_\beta ZF$ or all $F \in \Lambda$ is called a **fixed-point combinator**.

Corollary 2.12. *Let $M \in \Lambda$ and $f \in V$. Then there is a term $F \in \Lambda$ such that*

$$F \equiv_\beta M[f \coloneqq F]$$

Proof. Take $F = Y\lambda f.M$. We can then compute

$$F \equiv_\beta (\lambda f.M)Y\lambda f.M = (\lambda f.M)F \equiv_\beta M[f \coloneqq F]$$

□

Example 2.13. Suppose $P \in \Lambda$ encodes a predicate: that is, $Pc_n \equiv_\beta \top$ or $\bot$ for all $n \in \mathbb{N}$. We want a λ-term encoding the map taking a number to the next number that satisfies the predicate.

Fist consider the term

$$M \coloneqq \lambda f.\lambda x.(\text{if } Px \text{ then } x \text{ else } f(Sx))$$

where S encodes the successor function (we are about to show this exists). Informally, M takes a program f and input x; if Px holds, it returns n, and if it does not, it returns $f(x+1)$. Clearly, we want to run M on itself. We can achieve this using the λ-term $F = YM$. Indeed,

$$Fc_n \equiv_\beta YMc_n \equiv_\beta M(YM)c_n = MFc_n \equiv_\beta (\text{if } Pc_n \text{ then } c_n \text{ else } Fc_{n+1})$$

We have now encoded enough computational concepts to show that, in fact, the λ-calculus captures *all* computation.

Lemma 2.14. *The basic functions are λ-definable.*

Proof. The i^{th} projection is defined by

$$\pi_k^i \coloneqq \lambda x_1 \dots \lambda x_n.x_i$$

The successor function is defined by

$$S \coloneqq \lambda x \lambda s \lambda z.s(xsz)$$

The zero map is defined by

$$Z \coloneqq \lambda x.c_0$$

□

Lemma 2.15. *The class of total λ-definable functions is closed under composition.*

Proof. Let G define $g : \mathbb{N}^k \to \mathbb{N}$, and let $H_1, \dots, H_k$ define $h_1, \dots, h_k : \mathbb{N}^m \to \mathbb{N}$. Then their composite $\mathbf{n} \to g(h_1(\mathbf{n}), \dots, h_k(\mathbf{n}))$ is defined by the λ-term

$$F = \lambda x_1, \dots, \lambda x_m.(G(H_1 x_1 \dots x_m) \dots (H_k x_1 \dots x_m))$$

□

Lemma 2.16. *The class of total λ-definable functions is closed under primitive recursion.*

Proof. Suppose $f : \mathbb{N}^{m+1} \to \mathbb{N}$ is obtained from $h : \mathbb{N}^{m+2} \to \mathbb{N}$ and $g : \mathbb{N}^m \to \mathbb{N}$ by primitive recursion, with

$$f(0, \mathbf{n}) = g(\mathbf{n}) \text{ and } f(k+1, \mathbf{n}) = h(f(k, \mathbf{n}), k, \mathbf{n}).$$

Let G and H define g and h, respectively. We want a λ-term that keeps track of the pair $(k, f(k, \mathbf{n}))$. Indeed, define

$$\mathcal{T} \coloneqq \lambda p.[S(p\top), H(p\bot)(p\top)x_1 \dots x_n]$$

with the x_i free. Recall that applying a pair to $\top$ and $\bot$ acts as the first and second projection on the pair, respectively; then $\mathcal{T}$ acts on pairs of Church numerals by updating the iteration data, with inputs $x_1, \ldots, x_n$ for the last m entries of H.

Then f is definable by

$$F \coloneqq \lambda x \lambda x_1 \ldots \lambda x_m . x \mathcal{T} [c_0, G x_1 \ldots x_m] \bot$$

Indeed,

$$F c_k c_{n_1} \ldots c_{n_m} = c_k \mathcal{T} [c_0, G c_{n_1} \ldots c_{n_m}] \bot \equiv_\beta \mathcal{T}^k [c_0, G c_{n_1} \ldots c_{n_m}] \bot$$

Then

$$\mathcal{T}^k [c_0, H(G c_{n_1} \ldots c_{n_m})] = [c_k, c_{f(k,\mathbf{n})}],$$

so we are done. □

Lemma 2.17. *The class of total λ-definable functions is closed under (total) minimisation.*

Proof. Suppose G λ-defines a function $g : \mathbb{N}^{k+1} \to \mathbb{N}$ such that, for any $\mathbf{n} \in \mathbb{N}^k$, there is some $k \in \mathbb{N}$ such that $g(g, k\mathbf{n}) = 0$. Let $f : \mathbb{N}^k \to \mathbb{N}$ be defined from g by minimisation:

$$f(\mathbf{n}) = \mu m . (g(m, \mathbf{n}) = 0).$$

We use the idea of a fixed-point combinator to write an algorithm that searches for the least k such that $g(k, \mathbf{n}) = 0$. First, define

$$\texttt{zero?} \coloneqq \lambda x . x (\lambda y . \bot) \top$$

This satisfies $\texttt{zero?} c_0 \equiv_\beta \top$ and $\texttt{zero?} c_k \equiv_\beta \bot$. Next, define

$$\texttt{search} \coloneqq \lambda f . \lambda g \lambda k \lambda x_1 \ldots \lambda x_n . \text{if } (\texttt{zero?}(g k x_1 \ldots x_m)) \text{ then } k \text{ else } f(g(Sk) x_1 \ldots x_m)$$

If g represents a function $\tilde{g} : \mathbb{N}^k \to \mathbb{N}$ and $x_1, \ldots, x_n$ are Church numerals, then this term does the following. If $\tilde{g}(k, \mathbf{x})$ is zero, it returns k; otherwise, it applies f to $\tilde{g}(k+1, \mathbf{x})$. If we can run `search` on itself, therefore, then we are done. Indeed, set

$$F \coloneqq \lambda x_1 \ldots \lambda x_n . (Y\ \texttt{search}) G c_0 x_1 \ldots x_m$$

This clearly represents F, since

$$(Y\ \texttt{search}) G c_0 x_1 \ldots x_m \equiv_\beta \texttt{search} (Y\ \texttt{search}) G c_0 x_1 \ldots x_m$$

and G represents $g : \mathbb{N}^k \to \mathbb{N}$. □

Therefore we have the following theorem.

Theorem 2.18. *Every total recursive function is λ-definable.*

In fact, the partial recursive functions are all λ-definable, but the proofs are more complicated.

Theorem 2.19. *If a partial function $\mathbb{N}^k \rightharpoonup \mathbb{N}$ is λ-definable, then it is recursive.*

Proof sketch. We assign Gödel numbers $\lceil T \rceil$ to λ-terms T in the same way we encode strings over a finite alphabet.

Then consider a function $N(t)$ which, on input t, checks if t is the Gödel number of a λ-term (possible since the grammar is context-free and so computable). If $t = \lceil T \rceil$, then N returns the code of the BNF of T if it exists, and is undefined otherwise. We have an effective algorithm for weak normalisability, so BNFs are semi-computable, and so N is partial recursive. We also have partial recursive functions that map $n \to \lceil c_N \rceil$ and vice versa.

Now, suppose f is a partial function defined by $F \in \Delta$. We can compute $f(\mathbf{n})$ by the following procedure.

Compute $\lceil c_{n_1} \rceil, \dots, \lceil c_{n_m} \rceil, \lceil F \rceil$; then compute the code $\lceil F c_{n_1} \dots c_{n_m} \rceil \coloneqq \lceil T \rceil$ of their concatenation. Now, compute the code of the BNF of T, if it exists (otherwise the computation doesn't halt). If the code exists, if it is a Church numeral c_k; then return k.

This BNF exists iff $f(\mathbf{n})$ is defined; if it *is* defined, then the BNF of T is the Church numeral $c_{f(\mathbf{n})}$ (since F defines f). Hence this algorithm computes f. $\square$

2.2 Gödel's incompleteness theorem

Recall that $X \subseteq \mathbb{N}$ is **decidable** or **recursive** if χ_X is recursive, and **semi-decidable** or **recursively enumerable** (re) if φ_X is recursive, where

$$\varphi(x) = \begin{cases} 1 & \text{if } x \in X \\ \uparrow & \text{if } x \notin X \end{cases}.$$

X is re if any of the following equivalent conditions hold:

(i) X is the image of some partial recursive function.

(ii) X is the image of some total recursive function.

(iii) X is the set of values on which some partial recursive function is defined.

Note that X is recursive iff both X and $\mathbb{N} \setminus X$ are re. Using some effective Gödel numbering, we can talk about sets of things other than naturals being recursive or re. Since the set of recursive functions is countable, fix an enumeration $\{f_0, f_1, \dots\}$ of them for what comes next.

Example 2.20. $W = \{(i, x) \mid f_i(x)\!\downarrow\} \subseteq \mathbb{N}^2$ is re but not recursive.

Definition 2.21. A first order language $\mathcal{L}$ is **recursive** if the set of valid $\mathcal{L}$-formulae is computable. An $\mathcal{L}$-theory T is **recursive** if its set of axioms is computable (as a subset of $\mathcal{L}$).

We say T is **decidable** if the set $\{\varphi \in \mathcal{L} \mid T \vDash \varphi\}$ is computable; that is, if there is an algorithm to decide whether an $\mathcal{L}$-proposition φ is T-valid.

From now on, we will assume our language $\mathcal{L}$ is recursive.

Theorem 2.22 (Craig)**.** *Let T be a first order theory with an re set of axioms. Then T has a recursive axiomatisation.*

Proof. Let $f : \mathbb{N} \to \mathcal{L}$ be a total recursive function enumerating the axiom set of T, so that $T = \operatorname{im} f$. Let $\psi_n = \bigwedge_{k=1}^{n} f(n)$, and form the axiom set $T^* = \{\psi_n \mid n \in \mathbb{N}\}$. Now, T^* and T have the same deductive closure, so T^* axiomatises T. Since any formula $\varphi \in \mathcal{L}$ has finite length, we can check whether $\varphi \in T^*$ using the following algorithm.

First, check if it is of the form $\bigwedge_{k=1}^{n} \chi$ for some $\chi \in \mathcal{L}$. If not, halt and say that $\varphi \notin T^*$. Else check whether $\chi = f(n)$ (symbol-by-symbol).

Hence T^* is a recursive axiomatisation of T. □

Lemma 2.23. *The set of total recursive functions is not re.*

Proof. Suppose we have a total recursive function $f : \mathbb{N} \to \mathbb{N}$ enumerating the set of (codes for) total recursive functions. Define $g : \mathbb{N} \to \mathbb{N}$ by

$$g(n) = \lfloor f(n) \rfloor(n) + 1.$$

Then g is total recursive, but cannot be encoded by $f(m)$ for any $m \in \mathbb{N}$: indeed, we would have

$$g(m) = \lfloor f(m) \rfloor(m) + 1 = g(m) + 1.\#$$

□

We can now prove a version of Gödel's first incompleteness theorem.

Theorem 2.24 (Gödel's incompleteness theorem)**.** *Let $\mathcal{T}$ be a theory in the language of arithmetic with an re set of axioms. Then there is a statement that is true in $\mathbb{N}$, yet not provable from $\mathcal{T}$. In particular, if $\mathbb{N}$ is a model of $\mathcal{T}$, then $\mathcal{T}$ is incomplete.*

The difficult part of the proof is to show that the language of arithmetic can encode totality of recursive functions; this follows from the fact that it can encode Turing machines.

Proof. Since the axioms are re, $\mathcal{T}$ has a recursive axiomatisation. The set of theorems of $\mathcal{T}$ is therefore re.

Observe that the terms $\bar{n} = SS \ldots S0$ representing standard naturals form a decidable subset of the language. Now, there is some formula $TR(x)$ such that, in $\mathbb{N}$, the sentence $TR(\bar{n})$ has the interpretation "n is a code for a total recursive function". The set

$$S = \{n \in \mathbb{N} \mid \mathcal{T} \vdash TR(\bar{n})\}$$

is then re: enumerate the theorems of $\mathcal{T}$, and check if each has form $TR(\bar{n})$.

By the last lemma, S cannot be the set of all total recursive functions, so there must be some $m \notin S$ coding a total recursive function. Then $\mathbb{N} \vDash TR(m)$, yet $\mathcal{T} \nvdash TR(\bar{m})$. □

Since $\mathbb{N}$ is a model of PA, we have shown that PA is incomplete.

2.3 Computable models of PA

Are there any concrete nonstandard models of PA? When we talk about 'concrete' models of arithmetic, we generally mean *computable* ones. We will show the only computable model of PA is the standard model $\mathbb{N}$.

Even though PA is incomplete, we *can* prove completeness for certain classes of formulae.

Definition 2.25. A formula of arithmetic is Δ_0 if its quantifiers are bounded: that is, if it is of the form $(\exists x < t)\varphi(x)$ or $(\forall x < t)\varphi(x)$, where φ is Δ_0 or quantifier-free.

All Δ_0 sentences are PA-equivalent to some quantifier-free sentence. Indeed, replace each bounded quantifier with a finite conjunction or disjunction.

Example 2.26. $\forall x < 3.\varphi(x)$ is PA-equivalent to $\varphi(0) \wedge \varphi(1) \wedge \varphi(2)$.

Theorem 2.27. *Let φ be a Δ_0 sentence. If $\mathbb{N} \vDash \varphi$, then $PA \vdash \varphi$.*

Proof. By the above discussion, it suffices to prove the result for quantifier-free formulae. Proceed by induction on the formula; conjunction and disjunction follow easily. It remains to consider atomic formulae and their negations.

Now, the closed terms in the language of arithmetic are all equivalent to some standard natural 0, $S0$, $SS0$, and so on. The atomic sentences therefore have form either $n = m$ or $n < m$, for closed terms n and m. We can reduce any atomic sentence which not provable in PA to a PA-equivalent (unprovable) sentence ψ of form $0 = n'$, $0 < m'$ or $0 > k'$ by removing applications of S (by the successor axiom). If $n' = 0$ or $m' \neq 0$, then $PA \vdash \psi;_{\#}$ hence $n' \neq 0$ or $m' = 0$, and so $\mathbb{N} \nvDash \psi$.

A similar argument gives the result for negations of atomic formulae. □

In particular, nonstandard models of PA satisfy the same Δ_0 sentences that $\mathbb{N}$ does.

Definition 2.28. A formula $\varphi(x)$ of arithmetic is Σ_1 if there is a Δ_0-formula $\psi(x, y)$ such that

$$PA \vdash \varphi(x) \leftrightarrow (\exists y)\psi(x, y).$$

Note that the previous theorem can be extended to Σ_1 sentences: if $\mathbb{N} \vDash (\exists y)\psi(k, y)$, where ψ is Δ_0, then there is some $n \in \mathbb{N}$ such that $\mathbb{N} \vDash \psi(k, n)$; since ψ is Δ_0, we have $PA \vdash \psi(\bar{k}, \bar{n})$, and so $PA \vdash (\exists y)\psi(\bar{k}, y)$.

Proposition 2.29. *A subset $S \subseteq \mathbb{N}$ is re iff it is definable by a Σ_1 formula.*

Proof. If S is re, then there is a Turing machine T that halts iff its input is in S. Now, there is a Δ_0 formula $\varphi(n, k)$ which holds iff T halts on input n in at most k steps. Then S is definable by the Σ_1-formula $(\exists k)\varphi(n, k)$.

Conversely, if S is defined by a Σ_1 formula which is PA-equivalent to $(\exists y)\psi(x, y)$, where $\psi(x, y)$ is a Δ_0 formula, then we can iterate over $n \in \mathbb{N}$, and check if $\psi(x, \bar{n})$ holds. Since Δ_0 formulae are decidable, this yields a recursive function with halting set S. □

A similar argument shows that every recursive function is definable by a Σ_1 formula.

Let f be a total recursive function. Then there is a formula $\varphi(\mathbf{x}, y)$ satisfying

$$PA \vdash \forall \mathbf{x} \exists! y \; \varphi(\mathbf{x}, y);$$
$$PA \vdash \forall y : \varphi(\bar{\mathbf{n}}, y) \leftrightarrow y = \overline{f(\mathbf{n})};$$

that is, f is **provably total**. We can therefore extend (the language of) PA by arbitrary recursive functions. For example, we can extend the language of PA by a unary function symbol π_x (provably) agreeing with the n^{th}-prime function $n \to \pi_n$ on the standard naturals.

To show that $\mathbb{N}$ is the only computable model of PA, we will encode subsets of $\mathbb{N}$ within a countable model of PA. We will show that, if PA is nonstandard, then it must encode some non-recursive subset. On the other hand, we will show that, if the model has recursive $+$, then the sets it encodes are recursive.

Definition 2.30. A countable structure M in the language of arithmetic is **recursive** if there are recursive functions $\oplus, \otimes : \mathbb{N}^2 \to \mathbb{N}$, a recursive relation $\preceq \subseteq \mathbb{N}^2$, and $n_0, n_1 \in \mathbb{N}$, such that

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

Let M be a model of PA. Then $\mathbb{N}$ embeds into M by

$$n \to (\underbrace{S \dots S}_{n} 0) \coloneqq \bar{n};$$

these terms form the **standard naturals** in M. Note that the standard naturals form an initial segment: indeed, the formula

$$x < \bar{n} \to (x = \bar{0} \vee x = \bar{1} \vee \dots \vee x = \overline{n-1})$$

is provable in PA. As a result, if M is nonstandard, then it must have an element e which is *not* in the image of the embedding. Call such an element a **nonstandard number**.

Lemma 2.31 (Overspill)**.** *Let M be a model of PA, and let $\varphi(x)$ be an formula of arithmetic. If $M \vDash \varphi(n)$ for all standard naturals n, then there is a nonstandard natural e in M such that $M \vDash \varphi(e)$.*

Proof. Since M is nonstandard it contains a nonstandard natural e. Suppose that φ holds in M for exactly the standard naturals; then in particular $M \nvDash \varphi(e)$. Now, $M \vDash \varphi(0)$; since $n+1$ is a standard natural iff n is, $M \vdash (\varphi(n) \to \varphi(n+1))$. By PA-induction, $M \vDash \forall k.\varphi(k)$, and so $M \vDash k$ for all numbers k in M. But then $M \vDash \varphi(e)._{\#}$ □

Now, fix some $m \in \mathbb{N}$ and a property $\varphi(x)$ of the naturals. Then the (standard) number

$$c \coloneqq \prod \{\pi_k \mid k < m, \varphi(k)\}$$

satisfies the formula

$$(\forall k < m)(\varphi(k) \iff \pi_k \text{ divides } c).$$

We can think of c as a code for the numbers less than m with property φ.

In $\mathbb{N}$, c can only code for finite subsets. In a nonstandard model, however, numbers can have infinitely many prime divisors.

Definition 2.32. A subset $S \subseteq \mathbb{N}$ is **canonically coded** in a model M of PA if there is some $c \in M$ such that

$$S = \{n \in \mathbb{N} \mid M \vDash \pi_{\bar{n}} \text{ divides } c\}.$$

In principle, we could replace "$\pi_{\bar{n}}$ divides c" with some other formula $\varphi(x, c)$. It turns out that we do not get any new subsets by doing this; i.e., $\{n \in \mathbb{N} \mid M \vDash \varphi(\bar{n}, c)\}$ is canonically coded. We will sketch a proof of this.

Proposition 2.33. *Let $\varphi(x, y)$ be a formula, and let M be a nonstandard model of PA. For $\tilde{b} \in M$, there is some $c \in M$ such that, for $n \in \mathbb{N}$,*

$$M \vDash \varphi(\bar{n}, \tilde{b}) \leftrightarrow (\pi_{\bar{n}} \textit{ divides } c).$$

Proof sketch. For $n \in \mathbb{N}$,

$$PA \vdash \forall b \exists a (\forall x < \bar{n}) \varphi(x, b) \leftrightarrow (\pi_x \text{ divides } a);$$

the proof follows from the product-of-primes encoding discussed above. By overspill, there is a nonstandard number w in M such that

$$M \vDash \forall b \exists a (\forall x < w) \varphi(x, b) \leftrightarrow (\pi_x \text{ divides } a).$$

Therefore, given our $\tilde{b}$, there must be some number c such that

$$M \vDash (\forall x < w) \varphi(x, b) \leftrightarrow (\pi_x \text{ divides } c).$$

Then $M \vDash \bar{n} < w$ for $n \in \mathbb{N}$, so

$$M \vDash \varphi(\bar{n}, b) \leftrightarrow (\pi_{\bar{n}} \text{ divides } c).$$

□

Definition 2.34. Say $A, B \subseteq \mathbb{N}$ are **recursively inseparable** if they are disjoint and, for any recursive $C \supseteq A$, we have $B \cap C \neq \emptyset$.

Proposition 2.35. *There exist re $A, B \subseteq \mathbb{N}$ which are recursively inseparable.*

Proof. Let $\{\varphi_n \mid n \in \mathbb{N}\}$ be a recursive enumeration of the set of recursive functions. Define

$$A = \{n \in \mathbb{N} \mid \varphi_n(n) = 0\} \text{ and } B = \{n \in \mathbb{N} \mid \varphi_n(n) = 1\};$$

these are disjoint, and re by a zigzag process.

Now, let $C \supseteq A$ be recursive, and let χ_C be its characteristic function. Since χ_C is recursive, it is re, so $\chi_C = \varphi_u$ for some $u \in \mathbb{N}$; since $\chi_C(u) \in \{0, 1\}$, either $u \in A$ or $u \in B$. If $u \in A$, then $\chi_C(u) = 0$, so $u \notin C$; but $C \supseteq A$.$_{\#}$ Therefore $u \in B$, so $\chi_C(u) = 1$ and so $u \in C$. Hence $u \in B \cap C$. □

Lemma 2.36. *Let M be a nonstandard model of PA. Then there is a non-recursive set X which is canonically coded in M.*

Proof. Let $A, B \subseteq \mathbb{N}$ be recursively inseparable re sets. Then there are Σ_1-formulae $(\exists m)a(m,x)$ and $(\exists m)b(m,x)$ defining A and B, respectively. Note that a and b are Δ_0-formulae.

Fix $n \in \mathbb{N}$. Since $A \cap B = \emptyset$, we have

$$\mathbb{N} \vDash (\forall v < n)(\forall w < n)(\forall x < n)\neg(a(v,x) \wedge b(w,x)).$$

This is a Δ_0-formula, so, by Δ_0-completeness of PA, we have that

$$M \vDash (\forall v < \bar{n})(\forall w < \bar{n})(\forall x < \bar{n})\neg(a(v,x) \wedge b(w,x)).$$

By overspill, there is some nonstandard $c \in M$ such that

$$M \vDash (\forall v < c)(\forall w < c)(\forall x < c)\neg(a(v,x) \wedge b(w,x)). \qquad (\star)$$

Now, the following set is canonically coded:

$$X \coloneqq \{n \in \mathbb{N} \mid M \vDash (\exists v < c)a(v,\bar{n})\}.$$

We have $A \supseteq X$. Indeed, if $n \in A$, then $\mathbb{N} \vDash (\exists m)a(m,n)$, so there is some $m \in \mathbb{N}$ such that $\mathbb{N} \vDash a(m,n)$. But a is a Δ_0-formula, so by completeness $M \vDash a(\bar{m},\bar{n})$. Since c is nonstandard, $M \vDash c > \bar{m}$, and so $M \vDash (\exists v < c)a(v,\bar{n})$. Hence $x \in X$.

We also have $B \cap X = \emptyset$. If $n \in B$, then there is some $m \in \mathbb{N}$ such that $\mathbb{N} \vDash b(m,n)$; proceeding as before, $M \vDash (\exists w < c)b(w,\bar{n})$. By $(\star)$, we deduce

$$M \vDash \neg(\exists v < c)a(v,\bar{n}).$$

Since $v > \bar{m}$ for all $m \in \mathbb{N}$, $n \notin X$.

Since A and B are recursively inseparable, X cannot be recursive. But X is canonically coded. □

Theorem 2.37 (Tennenbaum)**.** *Let $M = (\tilde{M}, \oplus, \otimes, \preceq, n_0, n_1)$ be a countable nonstandard model of PA. Then $\oplus$ is not recursive.*

Proof. Since M is countable (and infinite, by PA), we can assume by a bijection that $\tilde{M} = \mathbb{N}$, $n_0 = 0$ and $n_1 = 1$. By the last lemma, let M canonically code the non recursive set

$$X = \{n \in \mathbb{N} \mid M \vDash (\pi_{\bar{n}} \text{ divides } c)\}.$$

Since PA proves that

$$\pi_m \cdot x = \underbrace{x + x + \cdots + x}_{\pi_n \text{ times}},$$

we have that

$$M \vDash \pi_{\bar{n}} \cdot y = \underbrace{y + y + \cdots + y}_{\pi_{\bar{n}} \text{ times}}.$$

We can therefore conclude that $n \in X$ iff there is some $d \in \mathbb{N}$ such that

$$c = \underbrace{d \oplus d \oplus \cdots \oplus d}_{\pi_n \text{ times}}.$$

Suppose $\oplus$ is recursive. Then, searching through $\mathbb{N}$ in its natural order, we can check recursively if each $d \in \mathbb{N}$ satisfies the following disjunction:

$$\bigvee_{k=0}^{\pi_n - 1} c = \underbrace{x \oplus x \oplus \cdots \oplus x}_{\pi_n \text{ times}} \oplus \underbrace{1 \oplus 1 \oplus \cdots \oplus 1}_{k \text{ times}}.$$

Moreover, this search must always terminate. Indeed, PA proves the division lemma, so there is a (unique) $d \in \mathbb{N}$ and $r \preceq \bar{\pi}_n$ such that $c = (x \otimes \bar{\pi}_n) \oplus r$. But PA also proves

$$(\forall x).(x < \bar{\pi}_n \leftrightarrow x = 0 \vee x = 1 \vee x = 1 + 1 \vee \cdots \vee x = \underbrace{1 + 1 + \cdots + 1}_{\pi_n - 1 \text{ times}}),$$

so this value of d satisfies the disjunction above.

If d satisfies the first disjunct, then $n \in X$; if d does not (and hence satisfies some other disjunct), then $n \notin X$. Hence X is recursive.$_{\#}$

Therefore $\oplus$ cannot be not computable. □

We can show $\otimes$ cannot be recursive in a similar way.

3 Duality Theory

Definition 3.1. Let P and Q be preordered sets. An **adjunction** between them is a pair of order-preserving maps $f : P \to Q$ and $g : Q \to P$ such that, for all $x \in P$ and $y \in Q$, we have $f(x) \leq_Q y$ iff $x \leq_P g(y)$. We say f is **left adjoint** to g, and g is **right adjoint** to f.

Write P^{op} for P with the reverse ordering. An adjunction between P^{op} and Q is called a **Galois connection**.

Example 3.2. Let $\mathcal{S}$ be a set of structures and $\mathcal{F}$ a set of formulae within some logical framework. There is a satisfaction relation $\vDash \subseteq \mathcal{S} \times \mathcal{F}$. Given $S \subseteq \mathcal{S}$ and $F \subseteq \mathcal{F}$, we can consider the *theory* $\text{Th}(S)$ of S, given by

$$\text{Th}(S) = \{\varphi \in \mathcal{F} \mid (\forall M \in S)(M \vDash \varphi)\},$$

and the *models* of F, given by

$$\text{Mod}(F) = \{M \in \mathcal{S} \mid (\forall \varphi \in F)(M \vDash \varphi).\}$$

This induces a Galois connection

$$\mathcal{P}\mathcal{S}^{\text{op}} \underset{\text{Mod}}{\overset{\text{Th}}{\rightleftarrows}} \mathcal{P}\mathcal{F}$$

3.1 Finite duality

Definition 3.3. Let L be a lattice. An element $j \in L$ is **join-irreducible** if, whenever $j = \bigvee S$ for some (finite) $S \subseteq L$, in fact $j \in S$. **Meet-irreducible** elements are defined dually. Let $\mathcal{J}(L) \subseteq L$ be the sublattice of join-irreducible elements of L, and $\mathcal{M}(L)$ the sublattice of meet-irreducibles.

Definition 3.4. Let P be a poset. Let $\mathcal{U}(P) \subseteq \mathcal{P}(P)$ be the sublattice of terminal segments of P, and $\mathcal{D}(P)$ the sublattice of initial segments. In the case $P = \mathcal{P}(L)$, write $\mathcal{U}(L) = \mathcal{U}(\mathcal{P}(L))$ and $\mathcal{D}(L) = \mathcal{D}(\mathcal{P}(L))$.

Lemma 3.5. *Let L be a finite lattice, and let $a \not\leq b \in L$. Then there is some $j \in \mathcal{J}(L)$ such that $j \leq a$, but $j \not\leq b$.*

Proof. Let

$$\mathcal{T} = \{x \in L \mid x \leq a;\ x \not\leq b\}.$$

Now, $a \in \mathcal{T}$, so $\mathcal{T}$ is nonempty; by finiteness of L, we can find a minimal element $j \in \mathcal{T}$.

Then j is join-irreducible. Indeed, suppose $j = \bigvee S$ for some $S \subseteq L$. Since $\bigvee S \not\leq b$, there is some $s \in S$ such that $s \not\leq b$. But $s \leq j \leq a$ and $s \not\leq b$, so $s \in \mathcal{T}$; by minimality, in fact $s = j$. Hence $j \in S$. □

We want to show that a finite distributive lattice L has form $\mathcal{D}(Q)$ for some poset Q.

Indeed, consider the **Stone map**

$$\begin{aligned}(-)^* : L &\longrightarrow \mathcal{D}(\mathcal{J}(L)) \\ a &\longrightarrow \{j \in \mathcal{J}(L) \mid j \leq a\}\end{aligned}$$

This is clearly order preserving; by the last lemma, it is also *order-reflecting* (that is, $a^* \leq b^*$ implies $a \leq b$). We call this an **order embedding**. Therefore, whenever $(-)^*$ is surjective, it must be an order isomorphism.

Now, $(-)^*$ is not surjective in general, but it *is* surjective whenever L is finite and distributive.

Definition 3.6. An element $j \in L$ is **join-prime** if, for every finite $S \subseteq L$ with $j \leq \bigvee S$, there is some $s \in S$ with $j \leq s$.

If $j \in L$ is join-prime, and $j = \bigvee S$ for some finite $S \subseteq L$, then $j \leq s$ for some $s \in S$; but then $j = s$. Hence j is join-irreducible.

Proposition 3.7. *Let L be a finite lattice. TFAE:*

1. *L is distributive (equivalently, a Heyting algebra).*
2. *Every join-irreducible element of L is join-prime.*
3. *$(-)^*$ is surjective (and hence an order isomorphism).*

Proof.
$1 \Rightarrow 2$: Let j be join-irreducible, and let $j \leq \bigvee S$ for some $S \subseteq L$. By distributivity,

$$j = j \wedge \bigvee S = \bigvee_{s \in S} (j \wedge s);$$

since j is join-irreducible, $j = j \wedge s$ for some $s \in S$. Then $j \leq s$, so j is join-prime.

$2 \Rightarrow 3$: Let $D \in \mathcal{D}(\mathcal{J}(L))$ be an initial segment of $\mathcal{J}(L)$. For $j \in \mathcal{J}(L)$, we have that $j \in D$ iff $j \leq d$ for some $d \in D$; since j is join-prime, this occurs iff $j \leq \bigvee D$. Hence $D = (\bigvee D)^*$, and so $(-)^*$ is surjective.

$3 \Rightarrow 1$: L is isomorphic to a subset of $\mathcal{P}(L)$. Since powerset lattices are distributive, L must also be distributive. □

Every finite distributive lattice L has a **dual poset** $\mathcal{J}(L)$; every finite poset P has a **dual distributive lattice** $\mathcal{D}(P)$. We therefore have maps between (finite) distributive lattices and (finite) posets; we want to make these into functors.

Let $f : P \to Q$ be an order-preserving map of posets, and define

$$\begin{aligned} \mathcal{D}(f) : \mathcal{D}(Q) &\longrightarrow \mathcal{D}(P) \\ Q \supseteq D &\longrightarrow f^{-1}(D) \subseteq P \end{aligned}$$

This is well-defined since inverse images of order-preserving maps preserve initial segments for *any* preorder.

Proposition 3.8. *Let P and Q be finite posets, and let $h : \mathcal{D}(Q) \to \mathcal{D}(P)$ be a lattice homomorphism. Then there is a unique order-preserving map $f : P \to Q$ such that $h = \mathcal{D}(f)$.*

Proof. Since h preserves (finite) meets, it has a left adjoint $g : \mathcal{D}(P) \to \mathcal{D}(Q)$; we show this on the example sheet. Since h preserves (finite) joins, g preserves join-prime elements. Indeed, if $j \in \mathcal{D}(P)$ is join-prime and $g(j) \leq \bigvee S$ for some finite $S \subseteq \mathcal{D}(Q)$, then (by adjointness) $j \leq h(\bigvee S) = \bigvee h(s)$. Therefore $j \leq h(s)$ for some $s \in S$; then $g(j) \leq s$, so $g(j)$ is join-prime.

Now, down-sets $\downarrow p$ ($p \in P$) are join-irreducible in $\mathcal{D}(P)$: if $\downarrow p = S_1 \vee \cdots \vee S_n$, $p \in S_i$ for some i; then $\downarrow p = S_i$. Since $\mathcal{D}(P)$ is distributive, $\downarrow p$ is join-prime, and, therefore, so is $g(\downarrow p)$. This must be a down-set in $\mathcal{D}(Q)$. By uniqueness of maximal elements of down-sets, let $f(p)$ be the unique element of Q such that $g(\downarrow p) = \downarrow f(p)$. Then $f : P \to Q$ is order-preserving: if $p \leq p'$, then

$$f(p) \in \downarrow f(p) = g(\downarrow p) \subseteq g(\downarrow p') = \downarrow f(p'),$$

so $f(p) \leq f(p')$.

For any $E \in \mathcal{D}(Q)$ and $p \in P$, we have

$$p \in h(E) \iff \downarrow p \subseteq h(E) \iff g(\downarrow p) \subseteq E \iff \downarrow f(p) \in E \iff f(p) \in E,$$

so $h(E) = f^{-1}(E) = \mathcal{D}(f)(E)$, as required.

Uniqueness of f is clear. □

Theorem 3.9 (Birkhoff's Representation Theorem)**.** *The category of finite distributive lattices is equivalent to the category of finite posets.*

Proof. We have constructed a full, faithful and bijective functor $\mathcal{D}$ from the category of finite posets to the category of finite distributive lattices. □

3.2 Priestley duality

We want a duality theorem for *infinite* lattices. Here, the concept of join-irreducible elements is insufficient to describe the lattice.

Example 3.10. Let $L = \{\bot\} \oplus (\mathbb{N}^{\mathrm{op}})^2$, with the pointwise partial order on $(\mathbb{N}^{\mathrm{op}})^2$. This lattice lacks join-irreducible elements:

$$(n, m) = (n + 1, m) \vee (n, m + 1).$$

If j is a join-irreducible element in a distributive lattice, then it can be thought of as a limit of approximations by meets; we package these together in its up-set $\uparrow j$ to construct a dual poset. In a finite lattice, these meets always exist (since they must be finite); in an infinite lattice, they might not exist. In the infinite setting, we will use prime filters in place of meets to formalise this idea of approximations to an irreducible element; this is similar to how we use Cauchy sequences of rationals to formalise the idea of real numbers.

Let L be a lattice. A filter F on L is **principal** if it has form $\uparrow a$ for some $a \in L$. If $\bigwedge F \in F$ (note we need the meet to exist), then $F = \uparrow(\bigwedge F)$, and F is prime iff $\bigwedge F$ is join-irreducible. Suppose L is finite; then all meets exist, and all meets are finite meets, so all filters are principal and the prime filters are precisely the up-sets of join-irreducible elements.

Since bigger meets yield smaller elements, we order the set of prime filters on L by *reverse* inclusion.

Lemma 3.11 (Stone's prime filter lemma)**.** *Let F be a filter on a distributive lattice L, and let I be an ideal of L such that $F \cap I = \emptyset$. Then there is a prime filter G of L with $F \subseteq G$ and $G \cap I = \emptyset$.*

Proof. Exercise. □

Let L be a lattice. Write $\mathcal{F}'(L)$ for the poset of prime filters on L, under reverse inclusion.

Theorem 3.12 (Stone representation)**.** *Let L be a lattice. Then there is a lattice homomorphism*

$$(-)^* : L \to \mathcal{D}\mathcal{F}'(L)$$

which is injective iff L is distributive.

Proof. Define

$$a^* = \{F \in \mathcal{F}'(L) \mid a \in F\}.$$

Now, $a \wedge b \in F$ iff $a, b \in F$ for all filters F on L; similarly, $a \vee b \in F$ iff $a, b \in F$ for all *prime* filters on L. Therefore $(-)^*$ preserves binary meets and joins.

Further, if $\top$ and/or $\bot$ exist in L, then $\top^* = \mathcal{F}^*(L)$ and $\bot^* = \emptyset$ (since prime filters are proper). Therefore $(-)^*$ is a lattice homomorphism.

Suppose $(-)^*$ is injective. Then L is isomorphic to a sublattice of a distributive lattice, so it must be itself distributive. Conversely, suppose L is distributive, and take $a \neq b \in L$. Take $a \not\leq b$ (wlog); then the filter $\uparrow a$ is disjoint from the ideal $\downarrow b$. By Stone's lemma, find a filter $G \supseteq \uparrow a$ such that $G \cap \downarrow b = \emptyset$. But then $G \in a^* \setminus b^*$, so $a^* \neq b^*$. Hence $(-)^*$ is injective. □

What is $\mathrm{im}(-)^*$? It turns out to be the lattice of clopen initial segments of $\mathcal{F}'(L)$ for a particular topology on L.

Let X be a set. For each family $\mathcal{S}$ of subsets of X, there is a least topology containing it. This is the topology $\langle \mathcal{S} \rangle$ *generated by* $\mathcal{S}$; we say $\mathcal{S}$ forms a **subbasis** for $\langle \mathcal{S} \rangle$. Explicitly, $\langle \mathcal{S} \rangle$ is generated by arbitrary unions of finite intersections of elements of $\mathcal{S}$.

Lemma 3.13 (Alexander subbasis lemma)**.** *Let X be a topological space, and let $\mathcal{S}$ be a subbasis for the topology. If every open cover of X by sets in $\mathcal{S}$ has a finite subcover, then X is compact.*

Note that this requires choice.

Definition 3.14. An **ordered (topological) space** X is a topological space X equipped with a partial order $\leq$ such that $\leq \subseteq X \times X$ is closed (in the product topology).

Proposition 3.15. *Ordered spaces are Hausdorff.*

Proof. Since $\leq$ is closed, so is $\geq$, since $(x, y) \to (y, x)$ is continuous. But then $\Delta_X = \leq \cap \geq$ is closed in $X \times X$. □

Proposition 3.16. *Let X be a compact ordered space, and suppose $C \subseteq X$ is closed. Then $\uparrow C = \bigcup_{c \in C} \uparrow c$ and $\downarrow C = \bigcup_{c \in C} \downarrow c$ are closed. In particular, the principal ideals and filters $\downarrow x$ and $\uparrow x$ ($x \in X$) are closed.*

Proof. $(C \times X) \cap \leq$ is closed in $X \times X$. Since X is compact, the projections $X \times X \to X$ are closed, so $\uparrow C = \pi_2((C \times X) \cap \{\leq\})$ is closed. Dually, $\downarrow C$ is closed. The last part follows since singletons in a Hausdorff space are closed. □

Compact ordered spaces have quite a strong separation property.

Proposition 3.17. *Let X be a compact ordered space, and suppose $x, y \in X$ with $x \not\leq y$. Then there are disjoint open neighbourhoods U, V of x and y such that U is an initial segment and V is a terminal segment.*

Proof. By the last proposition, $\uparrow x$ and $\downarrow y$ are closed neighbourhoods; since $x \not\leq y$, they are disjoint. Since X is Hausdorff, and therefore normal, there are disjoint open neighbourhoods U and V of $\uparrow x$ and $\downarrow y$. Then $U' = (\downarrow(U^c))^c$ and $V = (\uparrow(V^c))^c$ have the desired properties. □

We now use $(-)^*$ to generate a topology on $\mathcal{F}'(L)$.

Definition 3.18. Let L be a distributive lattice. The **Priestley topology** τ^P on $\mathcal{F}'(L)$ is the topology generated by the subbasis

$$\mathcal{S} = \{a^* \mid a \in L\} \cup \{(a^*)^c \mid a \in L\}.$$

An open subset of τ^P is a union of finite intersections of elements of the form a^* or $(a^*)^c$.

Proposition 3.19. *Let L be a distributive lattice.*

(1) Then $\mathcal{F}'(L)$ equipped with τ^P is compact.

(2) If $F, G \in \mathcal{F}'(L)$ and $F \not\leq G$, then there is a clopen $U \in \mathcal{DF}'(L)$ such that $G \in U$ but $F \notin U$.

Proof.

(1) By Alexander's lemma, it suffices to show that any open cover of $\mathcal{F}'(L)$ by sub-basis elements has a finite subcover.

Let

$$\mathcal{F}'(L) = \bigcup_{a \in A} a^* \cup \bigcup_{b \in B} (b^*)^c$$

for some sets $A, B \subseteq L$. Therefore any prime filter G containing B must intersect A. Let F_B be the filter generated by B, and let I_A be the ideal generated by A; any prime filter that contains F_B must intersect I_A. By Stone's prime filter lemma, F_B must meet I_A.

Let $c \in F_B \cap I_A$; pick finite subsets $A' \subseteq A$ and $B' \subseteq B$ such that $c \leq \bigvee A'$ and $c \geq \bigwedge B'$. We can do this by the concrete characterisation of generated filters and ideals, and is an exercise. In particular, $\bigwedge B' \leq \bigvee A'$; applying the lattice homomorphism $(-)^*$, we have $\bigcap_{b \in B'} b^* \subseteq \bigcup_{a \in A'} a^*$. Then

$$\mathcal{F}'(L) = \bigcup_{a \in A'} a^* \cup \bigcup b \in B'(b^*)^c.$$

(2) Fix $F, G \in \mathcal{F}'(L)$, and suppose $F \not\leq G$; that is, $G \not\subseteq F$. Find $a \in L$ such that $a \in G$ but $a \notin F$. Then $U = a^*$ is a clopen initial segment containing G but not F.

□

In fact, when L is distributive, $(\mathcal{F}'(L), \tau^P, \leq)$ is a compact ordered space.

We can also endow the set of ideals or lattice homomorphisms with the Priestley topology.

Definition 3.20. Let $(X, \tau, \leq)$ be an ordered space. Then X is **totally order-disconnected** if, for every $x, y \in X$ with $x \not\leq y$, there is a clopen initial segment U with $y \in U$ but $x \notin U$. If X is also compact, it is called a **Priestley space**.

Definition 3.21. Let L be a lattice. The **Priestley dual space** of L is the Priestley space $\hat{L} = (\mathcal{F}'(L), \tau^P, \leq)$.

Theorem 3.22 (Priestley's Representation Theorem)**.** *Let L be a distributive lattice. Then*

$$L \cong \operatorname{Clp} \mathcal{D}(\hat{L}) \coloneqq \{\text{clopen sets in } \mathcal{D}(\hat{L})\}.$$

Proof. By the Stone representation theorem, $(-)^* : L \to \mathcal{D}(\mathcal{F}'(L))$ is injective. Since each a^* is clopen, $\operatorname{im}(-)^* \subseteq \operatorname{Clp} \mathcal{D}(\mathcal{F}'(L))$ is injective. It therefore suffices to show that $(-)^*$ is surjective.

Fix a clopen initial segment A of $\mathcal{F}'(L)$, and take $x, y \in \mathcal{F}'(L)$ with $y \in A$ but $x \notin A$. Since A is an initial segment, $x \not\leq y$; then $y \not\subseteq x$, so there is some $b_{x,y} \in L$ such that $b_{x,y} \in y$ but $b_{x,y} \notin x$.

Fix $y \in A$, and consider the open cover $\{(b^*_{x,y})^c \mid x \notin A\}$ of A^c. Since A is open and $\mathcal{F}'(L)$ is compact, A^c is compact, and so we have a finite subcover $\{(b^*_{x_i,y})^c \mid 1 \leq i \leq n_y\}$ for some $x_i \notin A$. Now, $\bigcap_{i=1}^{n_y} b^*_{x_i,y} \subseteq A$. Define $a_j \coloneqq \bigwedge_{i=1}^{n_y} b_{x_i,y}$; by construction, $y \in a^*_y \subseteq A$. Then $\{a^*_y \mid y \in A\}$ is an open cover of the compact (since closed) set A, so we have a finite subcover $\{a^*_{y_1}, \ldots, a^*_{y_m}\}$ of A. Define $a \coloneqq \bigvee_{i=1}^{m} a^*_{y_i}$; then

$$a^* = \bigcup a^*_{y_i} = A.$$

□

We can make the bijection $\hat{L} \to L$ into a functor. Let L and M be distributive lattices, and take an order-preserving map $f : \hat{L} \to \hat{M}$. This induces a map $f^{-1} : \mathcal{D}(\hat{L}) \to \mathcal{D}(\hat{M})$. Since f^{-1} preserves intersections and unions, it is a lattice homomorphism. Suppose further that f is continuous (wrt the Priestley topology); then f^{-1} restricts to a map on clopen sets. By the last proposition, conjugating by the Stone maps, we get a lattice homomorphism $h_f : M \to L$; it is the unique map making the square below commute.

$$\begin{array}{ccc} M & \xrightarrow{\quad h_f \quad} & L \\ {\scriptstyle (-)^*}\downarrow\simeq & & \simeq\downarrow{\scriptstyle (-)^*} \\ \operatorname{Clp}\mathcal{D}(\hat{M}) & \xrightarrow[f^{-1}]{} & \operatorname{Clp}\mathcal{D}(\hat{L}) \end{array}$$

Fix $m \in M$; explicitly, $h_f(m)$ is the (unique) element of L such that

$$(h_f(m))^* = f^{-1}(m^*).$$

Using the square above, we can check that $f \to h_f$ is functorial; then the Stone map $(-)^*$ is a natural isomorphism between the contravariant functors $\hat{L} \to L$ and $\operatorname{Clp} \mathcal{D}$.

In fact, *every* lattice homomorphism $h : M \to L$ is of this form (that is, the functor is full). Note that it is obviously faithful, as h_f determines f^{-1} and hence f.

Proposition 3.23. *Let L and M be distributive lattices, and let $h : M \to L$ be a lattice homomorphism. Then there is a unique continuous, order-preserving map $f : \hat{L} \to \hat{M}$ such that $h_f = h$.*

Proof. Let $x \in \hat{L}$ be a prime filter. Then there is a lattice homomorphism

$$M \xrightarrow{h} L \xrightarrow{\chi_x} 2.$$

Define

$$f(x) = \mathrm{coker}(\chi_x \circ f) = \{m \in M \mid h(m) \in x\}.$$

Now, $f(x) \in m^*$ iff $m \in f(x)$ iff $h(m) \in x$ iff $x \in (h(m))^*$; on the other hand, $f(x) \in m^*$ iff $x \in (h_f(m))^*$. Since $(-)^*$ is injective, we have $h = h_f$.

It remains to check that f is continuous and order-preserving. Indeed, for $y \in M$, we have $f^{-1}(y) = (h(y))^*$, so f is continuous as it sends a subbasis to open sets. If $x \leq x' \in \hat{L}$, then $x' \subseteq x$, so $\chi_{x'} \leq \chi_x$ (pointwise).Since h is a lattice homomorphism, and hence order-preserving, we have $\chi_{x'} \circ h \leq \chi_x \circ h$. By definition, then, we get $f(x) \leq f(x')$.

Uniqueness follows since h_f determines f. □

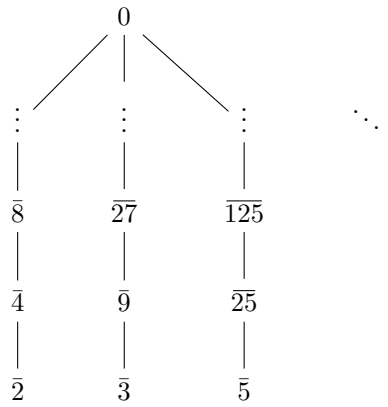
Theorem 3.24 (Priestley Duality)**.** *The categories of Priestley spaces, with continuous order-preserving maps between them, and bounded distributive lattices, with lattice homomorphisms between them, are equivalent.*

Proof. The functor given by $\hat{L} \to L$ and $f \to h_f$ is full, faithful and essentially surjective. □

Example 3.25. The set $\mathbb{N}$, ordered by divisibility, forms a bounded distributive lattice, with meet gcd, join lcm, top element 0 and bottom element 1. The prime filters of $\mathbb{N}$ are $\bar{0} = \{0\}$ and $\overline{p^k} = \{n \in \mathbb{N} \mid p^k \mid n\}$, where p is prime and $k \geq 1$.

Indeed, let $\bar{0} \neq x \in \hat{\mathbb{N}}$; let $m \in x$ be its least nonzero element. Since x is upward closed, $\uparrow m \subseteq x$. Conversely, let $0 \neq n \in x$. Then $0 < \gcd(n, m) \in x$; by minimality, $\gcd(n, m) = m$, and so $n \in \uparrow m$ (that is, $m \mid n$). Hence $x = \uparrow m$. It remains to show that m is a prime power. Since x is prime, it is proper, and so $m \geq 1$. Let p be a prime divisor of m, and write $m = p^k u$, where u is coprime to p (so $k \geq 1$). Then $\mathrm{lcm}(p^k, u) = m \in x$, so $p^k \in x$ or $u \in x$. If $p^k \in x = \uparrow m$, then $m \mid p^k$, so $m = p^k$. Otherwise, $u \in x$ and $m > p^k$; but then $u < m$. violating minimality.#

The ordering on $\hat{\mathbb{N}}$ is then given by $\overline{p^k} < \bar{0}$, and $\overline{p^k} \leq \overline{q^l}$ iff $p = q$ and $k \leq l$. A picture of $\hat{\mathbb{N}}$ is given below.



The subbasic open sets are given by $0^* = \{x \in \hat{\mathbb{N}} \mid 0 \in x\} = \bar{0}$ and, for a prime factorisation $n = p_1^{e_1} \dots p_r^{e_r}$, $n^* = \{x \in \hat{\mathbb{N}} \mid n \in x\} = \{p_i^k \mid 1 \leq i \leq r,\ k \leq e_i\}$. Observe that $\{\overline{p^k}\} = (p^k)^* \setminus (p^{k+1})^*$, so all singletons except $\bar{0}$ are clopen. Since every n^* apart from 0^* is finite, a clopen subset of $\hat{\mathbb{N}}$ contains $\bar{0}$ iff it is cofinite.

Example 3.26. Let V be a discrete topological space; let $X = 2^V$ with the product topology. Then X is an ordered space with respect to reverse inclusion. Explicitly, for $f, g \in 2^V$, we have $f \leq g$ iff $g(v) = 1 \implies f(v) = 1$. Call X the **ordered generalised Cantor space**.

Define

$$\langle v \to a \rangle \coloneqq \{f \in 2^V \mid f(v) = a\};$$

then $\{\langle v \to a \rangle \mid v \in V, a \in 2 = \{0, 1\}\}$ is a subbasis for the topology on X. If $f \not\leq g$, then there is some $v_0 \in V$ such that $g(v_0) = 1$ but $f(v_0) = 0$. Consider $U = \langle v_0 \to 1 \rangle$; then $g \in U$ but $f \notin U$. Now, U is clopen since its complement is in our subbasis; further, U is an initial segment: if $a \leq b \in U$, then $b(v_0) = 1$, so $a(v_0) = 1$ and so $a \in U$. Therefore U is totally order disconnected. By Tychonoff's theorem, X is compact; hence X is a Priestley space.

The lattice dual to X is in fact the free distributive lattice with generating set V.

This example implies that any finitely-generated distributive lattice L must be finite. Indeed, such a lattice must be a quotient of the free lattice L_V on a finite generating set V, whose dual is the finite space 2^V. Since $L_V \cong \mathrm{Cl}\mathcal{D}(2^V)$, L_V is finite, and so L is finite. This makes the proof of completeness for finite Heyting algebras much easier.

Closed subspaces of $\hat{L}$ correspond to quotients of L by congruences. Indeed, write $a \approx b$ for $(a, b) \in L \times L$, and define

$$\lceil a \approx b \rceil \coloneqq (a^* \cap b^*) \cup ((a^*)^c \cap (b^*)^c) = \{x \in \hat{L} \mid x \in a^* \iff x \in b^*\}.$$

A prime filter on L can be thought of as a lattice homomorphism $L \to 2$, or, more intuitively, a truth table for L. This definition yields the set of tables under which a and b have the same truth value. Conversely, given a 'truth table' (prime filter) $x \in \hat{L}$, we get a relation

$$\theta(x) \coloneqq \{a \approx b \mid x \in a^* \iff x \in b^*\} \subseteq L \times L.$$

We can extend these definitions to binary relations $\approx$ on L and subsets of $\hat{L}$, respectively, to get maps

$$\begin{array}{cc} \text{Relations on } L & \text{Subsets of } \hat{L} \\ R & \to \lceil R \rceil \\ \theta(S) & \leftarrow S \end{array}$$

where

$$\lceil R \rceil \coloneqq \bigcap_{aRb} \lceil aRb \rceil = \{x \in \hat{L} \mid \forall (a, b) \in R : (x \in a^* \iff x \in b^*)\} \text{ and}$$

$$\theta(S) \coloneqq \bigcap_{x \in S} \theta(x) = \{(a, b) \in L \times L \mid a^* \cap S = b^* \cap S\}.$$

These maps form a contravariant Galois connection, with $\lceil\cdot\rceil \dashv \theta$. This adjunction induces an equivalence between the fixed points of $\lceil\theta(\cdot)\rceil$ and those of $\theta(\lceil\cdot\rceil)$.

Proposition 3.27. *The fixed points of the adjunction are precisely the lattice congruences of L on the left, and the closed subspaces of $\hat{L}$ on the right.*

Proof. Note that, since each $\lceil a \approx b\rceil$ is clopen, each $\lceil R\rceil$ is closed.

Suppose $c = \lceil\theta(c)\rceil$; then c is closed. Conversely, suppose $c \subseteq \hat{L}$ is closed. Since $\lceil\cdot\rceil \dashv \theta$, we have $c \subseteq \lceil\theta(c)\rceil$; it remains to show the reverse inclusion. Indeed, suppose $x \notin c$. Now, $\mathcal{B} = \{a^* \setminus b^* \mid a, b \in L\}$ is a basis for the Priestley topology, so there are $a, b \in L$ such that $x \in a^* \setminus b^*$ and $c \cap (a^* \setminus b^*) = \emptyset$. From disjointness, we get that $(a \approx (a \wedge b)) \in \theta(c)$. But $x \in a^*$ and $x \notin (a \wedge b)^*$, so $x \notin \lceil\theta(c)\rceil$. Hence $c = \lceil\theta(c)\rceil$.

The proof on the left is similar. □

The adjunction therefore induces a correspondence between quotients of L and closed subsets of $\hat{L}$. More precisely, we get the theorem below, which we will not prove.

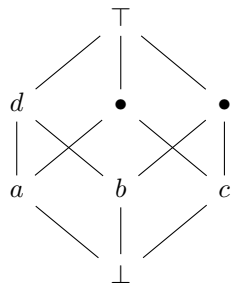
Theorem 3.28. *Let L be a distributive lattice.*

(1) Let R be a relation on L. Then the congruence generated by R is $\theta(\lceil R\rceil)$, and the Priestley dual of $L/\langle R\rangle = L/\theta(\lceil R\rceil)$ is order homeomorphic to the closed subspace $\lceil R\rceil \subseteq \hat{L}$.

(2) Let $S \subseteq \hat{L}$. Then $\bar{S} = \lceil\theta(S)\rceil$, and the lattice dual to $\bar{S}$ is isomorphic to $L/\theta(S)$.

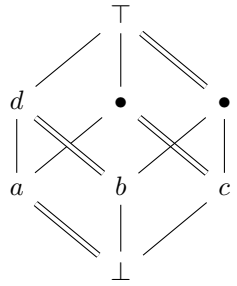
This, along with the previous example, allows us to (in principle) compute the Priestley dual of a lattice.

Example 3.29. Let $\mathcal{B}$ be the 8-element Boolean algebra below.

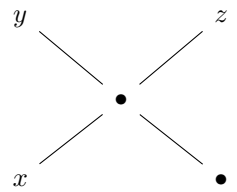


Its dual is $\hat{\mathcal{B}} = \{\uparrow a, \uparrow b, \uparrow c\}$. The equation $b \approx d$ corresponds to the (closed) subspace $\{\uparrow b, \uparrow c\}$, and the congruence generated by $b \approx d$ is shown by doubled

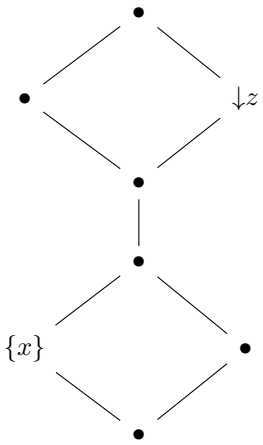
lines below.



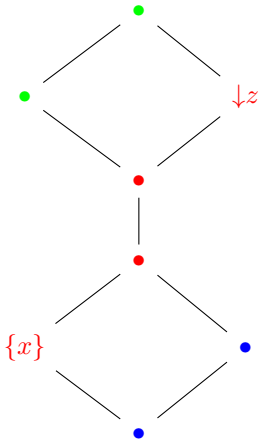
Example 3.30. Consider the poset whose Hasse diagram is drawn below.



Its dual lattice is



The subset $S = \{x, y\}$ gives the congruence whose equivalence classes yield the colouring below:



Also, $\bar{S}$ is given by $\downarrow z \approx \{x\}$.