b. Induction (LfP 50–3)

b.i. The principle of mathematical induction

The principle of mathematical induction is a property of natural numbers.

b.i.1. Natural numbers

The natural numbers are the non-negative integers: $0, 1, 2, 3, \ldots$. We write $\mathbb{N}$ for the set of natural numbers—i.e.:

$$\mathbb{N} = \{0, 1, 2, 3, \ldots\}$$

We'll reserve the variables m and n for natural numbers.

b.i.2. Weak and Strong induction and the Least Number Principle.

There are three common ways to formulate the principle of induction, and variants of it.

The 'weak' principle of induction (Ind).

If (i) $F(0)$ and (ii) for each $n \in \mathbb{N}$: $F(n)$ implies $F(n+1)$, then $F(n)$ for each $n \in \mathbb{N}$

$(F(0) \wedge \forall n(F(n) \rightarrow F(n+1))) \rightarrow \forall n F(n)$

The 'strong' principle of induction (S-Ind).

If, for each $n \in \mathbb{N}$, $F(m)$ for all $m < n$ implies $F(n)$, then $F(n)$ for all $n \in \mathbb{N}$

$\forall n((\forall m < n)F(m) \rightarrow F(n)) \rightarrow \forall n F(n)$

The least number principle (LNP).

If M is a non-empty set of natural numbers, M has a least element.

Remarks.

- Ind is included as one of the axioms in standard axiomatizations of the canonical theory of arithmetic—Peano arithmetic (PA).
- This serves to distinguish $\mathbb{N}$, from e.g. the set of *all* integers, non-negative and negative,

  $$\mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$$

  e.g. $\{-1, -2, -3, \ldots\}$ is a subset of $\mathbb{Z}$ with no least element.
- Ind, S-Ind, and LNP are provably equivalent (given the other axioms of PA)
- We shall help ourselves to all three in informal proofs—usually S-Ind.
- Consequently, the background theory where we 'do metatheory' is stronger than classical logic—also containing some classical mathematics.

b.ii. Induction in arithmetic

We'll usually think of induction as a proof technique.

b.ii.1. Proof by induction on n

Proof by induction on n. One standard way to show that every $n \in \mathbb{N}$ is such that $F(n)$ is by induction on n:

- *Base case.* Prove $F(0)$.
- *Induction hypothesis.* Assume $F(n)$
- *Induction step.* Prove $F(n+1)$ (given the induction hypothesis)

Remark. Once you've completed the induction step, you've completed the proof. But sometimes you may wish to flag this with something like "Consequently $F(n)$ holds for every $n \in \mathbb{N}$, by induction".

b.ii.2. Example: summing series

Worked Example. Show that for any $n \in \mathbb{N}$:

$$0^2 + 1^2 + 2^2 + \cdots + n^2 = \frac{n(n+1)(2n+1)}{6}$$

b.iii. Induction in logic

Induction in logic is often strong induction 'on complexity of formulas'.

b.iii.1. Complexity

One standard way to measure the complexity of a formula is simply to count how many connectives it contains:

Definition of complexity. Let the complexity of a wff ϕ be the number of connectives in ϕ—we'll sometimes write this as $C[\phi]$.

Remark. This is a complexity measure for PL, and other propositional languages. When we come to predicate logic, we'll count quantifiers as well as connectives.

Worked Example. Compute: (i) $C[P]$; (ii) $C[\sim(P \wedge Q)]$.

b.iii.2. Proof by induction on complexity of ϕ

Proof by induction on the complexity of ϕ. One standard way to show that every formula ϕ is such that $G(\phi)$ is to reason as follows.

- *Base case.* Show $G(\alpha)$ holds for any atomic formula α.
- Let ϕ be an arbitrary formula.
- *Induction hypothesis.* Assume $G(\psi)$ holds for all ψ with complexity less than ϕ.
- *Induction step.* Prove $G(\phi)$ (given the induction hypothesis).

Remark. This is an application of S-Ind on the following property of natural numbers:

$$F(n) := G(\phi) \text{ holds for all formulas } \phi \text{ with } C[\phi] \leqslant n.$$

b.iii.3. Example: syntactic properties

Worked Example. Show that every PL-wff has the same number of left brackets as right brackets.

Remarks.

- For PL, the induction step often splits into cases: (i) $\phi = \sim\psi$, (ii) $\phi = (\psi_1 \to \psi_2)$.
- Applying the induction hypothesis relies on the fact that $C[\psi], C[\psi_1], C[\psi_2] < C[\phi]$.

b.iii.4. Example: semantic properties

Worked Example. Let $\mathscr{I}_\#$ be the trivalent interpretation that assigns every sentence letter #. Show that $\mathrm{KV}_{\mathscr{I}_\#}(\phi) = \#$ for each wff ϕ.

Remark.

- The induction step in the three-valued setting typically calls for four cases: (i) $\phi = \sim\psi$, (ii) $\phi = (\psi_1 \to \psi_2)$, (iii) $\phi = (\psi_1 \wedge \psi_2)$, (iv) $\phi = (\psi_1 \vee \psi_2)$.

c. Expressive Adequacy in PL (LfP 3.1)

c.i. Symbolizing truth functions

We often think of the truth tables as somehow 'giving the meaning' of a connective. Truth functions and the notion of symbolizing provide a way to make this precise.

c.i.1. Truth functions

Definition of truth function. An (*n-place*) *truth function* is a function that maps each n-tuple of truth values to a truth value.

Notation. We use lowercase t (and t_1, t' etc.) to stand for truth values—since we're working with PL, we only have two truth values, 1 or 0.

Examples.

$$N : 1 \mapsto 0 \qquad\qquad C : \langle 1,1\rangle \mapsto 1$$
$$0 \mapsto 1 \qquad\qquad \langle 1,0\rangle \mapsto 0$$
$$\langle 0,1\rangle \mapsto 1$$
$$\langle 0,0\rangle \mapsto 1$$

Remark. Each n-ary truth function is uniquely determined by the corresponding $n + 1$-column truth-table, and vice versa.

c.i.2. Symbolizing

Definition of symbolizing (LfP 68)**.** A wff $\phi(P_1, \ldots, P_n)$ symbolizes an n-place truth function f iff $\phi(P_1, \ldots, P_n)$ contains the sentence letters $P_1, \ldots, P_n$, and no others, and for each PL-interpretation $\mathscr{I}$:

$$\mathrm{V}_{\mathscr{I}}(\phi) = f(\mathscr{I}(P_1), \ldots, \mathscr{I}(P_n))$$

Remark. It's helpful to write the wff here as $\phi(P_1, \ldots, P_n)$ as a reminder that it contains precisely these sentence letters.

Examples.

- $\sim P_1$ symbolizes N
- $P_1 \to P_2$ symbolizes C

Remark. When $\phi(P_1, \ldots, P_n)$ symbolizes f, $\phi(P_1, \ldots, P_n)$ and f have the same truth table.

c.ii. DNF Theorem

Question. Which truth functions can we symbolize in PL?

To start with take $\sim$, $\wedge$, and $\vee$ as primitive.

> **DNF Theorem.** *Every truth function is symbolized by some wff or other containing only $\sim$, $\wedge$ and $\vee$—in fact, by a sentence in Disjunctive Normal Form (DNF).*

> **Definition of DNF.** A literal is a sentence letter or its negation. A sentence ϕ is in DNF if it is a disjunction or one or more conjunctions of one or more literals.

> *Worked Example.* Find a DNF-sentence that symbolizes C.

Proof of DNF theorem. Let an n-ary truth-function f be given.
Clearly, there are only finitely many n-tuples of truth-values s.t. $f(t_1, \ldots, t_n) = 1$.[1]

Case 1. There are 0 such n-tuples. Set $\delta = (P_1 \wedge \sim P_1) \vee \ldots \vee (P_n \wedge \sim P_n)$.
Clearly this symbolizes the truth-function that maps each n-tuple to 0 and is in DNF.

Case 2. There are 1 or more such n-tuples.
Enumerate the n-tuples that are mapped to 1: $\langle t_1^1, \ldots, t_n^1 \rangle, \ldots, \langle t_1^k, \ldots, t_n^k \rangle$.

Define literals θ_j^i as follows ($i = 1, \ldots k, j = 1, \ldots n$):

$$\theta_j^i = \begin{cases} P_j \text{ if } t_j^i = 1 \\ \sim P_j \text{ if } t_j^i = 0 \end{cases}$$

Note that, by construction, $V_{\mathscr{I}}(\theta_j^i) = 1$ iff $V_{\mathscr{I}}(P_j) = t_j^i$ (under any $\mathscr{I}$). $(\star)$

Define a DNF-sentence as follows:

$$\delta = (\theta_1^1 \wedge \ldots \wedge \theta_n^1) \vee \ldots \vee (\theta_1^k \wedge \ldots \wedge \theta_n^k).$$

Remains to prove: δ symbolizes f: $V_{\mathscr{I}}(\delta) = 1$ iff $f(\mathscr{I}(P_1), \ldots, \mathscr{I}(P_n)) = 1$ for each $\mathscr{I}$.

$$\begin{aligned} V_{\mathscr{I}}(\delta) = 1 &\text{ iff } V_{\mathscr{I}}(\theta_1^i \wedge \ldots \wedge \theta_n^i) = 1 \text{ for some } i = 1, \ldots, k \\ &\text{ iff } V_{\mathscr{I}}(\theta_1^i) = 1; \ldots; V_{\mathscr{I}}(\theta_n^i) = 1 \text{ for some } i = 1, \ldots, k \\ &\text{ iff } \mathscr{I}(P_1) = t_1^i; \ldots; \mathscr{I}(P_n) = t_n^i \text{ for some } i = 1, \ldots, k \qquad (\text{by } (\star)) \\ &\text{ iff } \langle \mathscr{I}(P_1), \ldots, \mathscr{I}(P_n) \rangle = \langle t_1^i, \ldots, t_n^i \rangle \text{ for some } i = 1, \ldots, k \\ &\text{ iff } f(\mathscr{I}(P_1), \ldots, \mathscr{I}(P_n)) = f(t_1^i, \ldots, t_n^i) = 1 \end{aligned}$$

□

[1] After all, for each n, there are only 2^n n-tuples of truth values.

c.iii. Expressive Adequacy

Definition of adequacy (LfP 69)**.** A set of connectives S is (expressively) *adequate* iff every truth function is symbolized by a wff containing only connectives in S.

c.iii.1. Demonstrating adequacy

The usual strategy to show S is adequate is to show its members can simulate $\sim$, $\wedge$ and $\vee$, and then appeal to DNF.

Worked Example. Show $\{\sim, \rightarrow\}$ is adequate.

c.iii.2. Demonstrating inadequacy

The usual strategy to show S is inadequate is to show:

(i) all the truth functions that can be symbolized with sentences built from S-connectives have a certain property F.

(ii) not all truth functions have the property F.

Worked Example. $\{\wedge, \vee, \rightarrow, \leftrightarrow\}$ is *not* adequate.