

## 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:  $0, 1, 2, 3, \dots$ . We write  $\mathbb{N}$  for the set of natural numbers—i.e.:

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

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  $n \in \mathbb{N}$   
 $(F(0) \wedge \forall n(F(n) \rightarrow F(n + 1))) \rightarrow \forall nF(n)$

**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  $n \in \mathbb{N}$   
 $\forall n((\forall m < n)F(m) \rightarrow F(n)) \rightarrow \forall nF(n)$

**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,

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

e.g.  $\{-1, -2, -3, \dots\}$  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  $\phi$  be the number of connectives in  $\phi$ —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)  $C[\sim\sim((P \wedge Q) \vee R)]$ .

### b.iii.2. Proof by induction on complexity of $\phi$

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

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

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

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

### 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)  $\phi = (\psi_1 \rightarrow \psi_2)$ .
- 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  $\mathcal{I}_\#$  be the trivalent interpretation that assigns every sentence letter  $\#$ . Show that  $KV_{\mathcal{I}_\#}(\phi) = \#$  for each wff  $\phi$ .

*Remark.*

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

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

*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  $\phi(P_1, \dots, P_n)$  symbolizes an  $n$ -place truth function  $f$  iff  $\phi(P_1, \dots, P_n)$  contains the sentence letters  $P_1, \dots, P_n$ , and no others, and for each PL-interpretation  $\mathcal{I}$ :

$$\mathcal{V}_{\mathcal{I}}(\phi) = f(\mathcal{I}(P_1), \dots, \mathcal{I}(P_n))$$

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

*Examples.*

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

*Remark.* When  $\phi(P_1, \dots, P_n)$  symbolizes  $f$ ,  $\phi(P_1, \dots, P_n)$  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  $\phi$  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.  $f(t_1, \dots, t_n) = 1$ .<sup>1</sup>

*Case 1.* There are 0 such  $n$ -tuples. Set  $\delta = (P_1 \wedge \sim P_1) \vee \dots \vee (P_n \wedge \sim P_n)$ .

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:  $\langle t_1^1, \dots, t_n^1 \rangle, \dots, \langle t_1^k, \dots, t_n^k \rangle$ .

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

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

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

Define a DNF-sentence as follows:

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

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

$V_{\mathcal{I}}(\delta) = 1$  iff  $V_{\mathcal{I}}(\theta_1^i \wedge \dots \wedge \theta_n^i) = 1$  for some  $i = 1, \dots, k$   
 iff  $V_{\mathcal{I}}(\theta_1^i) = 1; \dots; V_{\mathcal{I}}(\theta_n^i) = 1$  for some  $i = 1, \dots, k$   
 iff  $\mathcal{I}(P_1) = t_1^i; \dots; \mathcal{I}(P_n) = t_n^i$  for some  $i = 1, \dots, k$  (by (★))  
 iff  $\langle \mathcal{I}(P_1), \dots, \mathcal{I}(P_n) \rangle = \langle t_1^i, \dots, t_n^i \rangle$  for some  $i = 1, \dots, k$   
 iff  $f(\mathcal{I}(P_1), \dots, \mathcal{I}(P_n)) = f(t_1^i, \dots, t_n^i) = 1$  □

<sup>1</sup>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.