

4. Logical Equivalence

4.1 Definition

Two formulas ϕ, ψ are **logically equivalent**

if $\phi \models \psi$ and $\psi \models \phi$,

i.e. if for every valuation v , $\tilde{v}(\phi) = \tilde{v}(\psi)$.

Notation: $\phi \models \psi$

Exercise $\phi \models \psi$ if and only if $\models (\phi \leftrightarrow \psi)$

4.2 Lemma

(i) For any formulas ϕ, ψ

$$(\phi \vee \psi) \models \neg(\neg\phi \wedge \neg\psi)$$

(ii) Hence every formula is logically equivalent to one without '∨'.

Proof:

(i) Either use truth tables
or observe that, for any valuation v :

$$\begin{aligned}\tilde{v}(\neg(\neg\phi \wedge \neg\psi)) &= F \\ \text{iff } \tilde{v}(\neg\phi \wedge \neg\psi) &= T && \text{by tt } \neg \\ \text{iff } \tilde{v}(\neg\phi) = \tilde{v}(\neg\psi) &= T && \text{by tt } \wedge \\ \text{iff } \tilde{v}(\phi) = \tilde{v}(\psi) &= F && \text{by tt } \neg \\ \text{iff } \tilde{v}(\phi \vee \psi) &= F && \text{by tt } \vee\end{aligned}$$

(ii) Induction on the length of the formula ϕ :

Clear for length 1

For the induction step observe that

$$\text{If } \psi \models \psi' \text{ then } \neg\psi \models \neg\psi'$$

and

$$\text{If } \phi \models \phi' \text{ and } \psi \models \psi' \text{ then } (\phi \star \psi) \models (\phi' \star \psi'),$$

where $\star$ is any binary connective.

(Use (i) if $\star = \vee$)

□

4.3 Some sloppy notation

We are only interested in formulas
up to logical equivalence:

If A, B, C are formulas then

$$((A \vee B) \vee C) \text{ and } (A \vee (B \vee C))$$

are different formulas, but logically equivalent.
So here - up to logical equivalence -
bracketting doesn't matter.
Hence

- Write $(A \vee B \vee C)$ or even $A \vee B \vee C$ instead.
- More generally, if $A_1, \dots, A_n$ are formulas,
write $A_1 \vee \dots \vee A_n$ or $\bigvee_{i=1}^n A_i$
for some (any) correctly bracketed version.
- Similarly $\bigwedge_{i=1}^n A_i$.

4.4 Some logical equivalences

Let A, B, A_i be formulas. Then

1. $\neg(A \vee B) \models \models (\neg A \wedge \neg B)$

So, inductively,

$$\neg \bigvee_{i=1}^n A_i \models \models \bigwedge_{i=1}^n \neg A_i$$

This is called *De Morgan's Laws*.

2. like 1. with $\vee$ and $\wedge$ swapped everywhere

3. $(A \rightarrow B) \models \models (\neg A \vee B)$

4. $(A \vee B) \models \models ((A \rightarrow B) \rightarrow B)$

5. $(A \leftrightarrow B) \models \models ((A \rightarrow B) \wedge (B \rightarrow A))$

5. Adequacy of the Connectives

The connectives $\neg$ (unary) and $\rightarrow, \wedge, \vee, \leftrightarrow$ (binary) are the *logical part* of our language for propositional calculus.

Question:

- Do we have enough connectives?
- Can we express everything which is logically conceivable using only these connectives?
- Does our language $\mathcal{L}$ recover all potential truth tables?

Answer: yes

5.1 Definition

(i) We denote by V_n the set of all functions

$$v : \{p_0, \dots, p_{n-1}\} \rightarrow \{T, F\}$$

i.e. of all partial valuations, only assigning values to the first n propositional variables. Hence $\#V_n = 2^n$.

(ii) An **n -ary truth function** is a function

$$J : V_n \rightarrow \{T, F\}$$

There are precisely 2^{2^n} such functions.

(iii) If a formula $\phi \in \text{Form}(\mathcal{L})$ contains only prop. variables from the set $\{p_0, \dots, p_{n-1}\}$ – write ' $\phi \in \text{Form}_n(\mathcal{L})$ ' – then ϕ determines the truth function

$$\begin{array}{ccc} J_\phi : V_n & \rightarrow & \{T, F\} \\ v & \mapsto & \tilde{v}(\phi) \end{array}$$

i.e. J_ϕ is given by the truth table for ϕ .

5.2 Theorem

*Our language $\mathcal{L}$ is adequate,
i.e. for every n and every truth function
 $J : V_n \rightarrow \{T, F\}$ there is some $\phi \in \text{Form}_n(\mathcal{L})$
with $J_\phi = J$.
(In fact, we shall only use the connectives $\neg, \wedge, \vee$.)*

Proof: Let $J : V_n \rightarrow \{T, F\}$ be any n -ary truth function.

If $J(v) = F$ for all $v \in V_n$ take $\phi := (p_0 \wedge \neg p_0)$.
Then, for all $v \in V_n$: $J_\phi(v) = \tilde{v}(\phi) = F = J(v)$.

Otherwise let $U := \{v \in V_n \mid J(v) = T\} \neq \emptyset$.
For each $v \in U$ and each $i < n$ define the formula

$$\psi_i^v := \begin{cases} p_i & \text{if } v(p_i) = T \\ \neg p_i & \text{if } v(p_i) = F \end{cases}$$

and let $\psi^v := \bigwedge_{i=0}^{n-1} \psi_i^v$.

Then for any valuation $w \in V_n$ one has the following equivalence ($\star$):

$$\begin{aligned} \tilde{w}(\psi^v) = T & \text{ iff } \begin{array}{l} \text{for all } i < n : \\ \tilde{w}(\psi_i^v) = T \end{array} & (\text{by tt } \wedge) \\ & \text{ iff } w = v & (\text{by def. of } \psi_i^v) \end{aligned}$$

Now define $\phi := \bigvee_{v \in U} \psi^v$.

Then for any valuation $w \in V_n$:

$$\begin{aligned} \tilde{w}(\phi) = T & \text{ iff for some } v \in U : \tilde{w}(\psi^v) = T & (\text{by tt } \vee) \\ & \text{ iff for some } v \in U : w = v & (\text{by } (\star)) \\ & \text{ iff } w \in U \\ & \text{ iff } J(w) = T \end{aligned}$$

Hence for all $w \in V_n$: $J_\phi(w) = J(w)$, i.e. $J_\phi = J$.

□

5.3 Definition

- (i) A formula which is a conjunction of p_i 's and $\neg p_i$'s is called a **conjunctive clause**
 - e.g. ψ^v in the proof of 5.2
- (ii) A formula which is a disjunction of conjunctive clauses is said to be in **disjunctive normal form** ('dnf')
 - e.g. ϕ in the proof of 5.2

In Theorem 5.2 we constructed, for a given J ,

Formula ϕ with $J_\phi = J$,

$\phi := \bigvee_{v \in U} \psi^v$, where $\psi^v := \bigwedge_{i=0}^{n-1} \psi_i^v$, where $\psi_i^v \in \{p_i, \neg p_i\}$.

So the proof, in fact, establishes the following Corollary:

5.4 Corollary - 'The dnf-Theorem'

For any truth function

$$J : V_n \rightarrow \{T, F\}$$

*there is a formula $\phi \in \text{Form}_n(\mathcal{L})$ in **dnf** with $J_\phi = J$.*

In particular, every formula is logically equivalent to one in dnf.

5.5 Definition

Suppose S is a set of (truth-functional) connectives – so each $s \in S$ is given by some truth table.

- (i) Write $\mathcal{L}[S]$ for the language with connectives S instead of $\{\neg, \rightarrow, \wedge, \vee, \leftrightarrow\}$ and define $\text{Form}(\mathcal{L}[S])$ and $\text{Form}_n(\mathcal{L}[S])$ accordingly.
- (ii) We say that S is **adequate** (or **truth functionally complete**) if for all $n \geq 1$ and for all n -ary truth functions J there is some $\phi \in \text{Form}_n(\mathcal{L}[S])$ with $J_\phi = J$.

5.6 Examples

1. $S = \{\neg, \wedge, \vee\}$ is adequate (Theorem 5.2)
2. Hence, by Lemma 4.2(i), $S = \{\neg, \wedge\}$ is adequate:

$$\phi \vee \psi \models \neg(\neg\phi \wedge \neg\psi)$$

Similarly, $S = \{\neg, \vee\}$ is adequate:

$$\phi \wedge \psi \models \neg(\neg\phi \vee \neg\psi)$$

3. Can express $\vee$ in terms of $\rightarrow$, so $\{\neg, \rightarrow\}$ is adequate (Problem set #2).
4. $S = \{\vee, \wedge, \rightarrow\}$ is **not** adequate, because any $\phi \in \text{Form}(\mathcal{L}[S])$ has T in the top row of $\text{tt } \phi$, so no such ϕ gives $J_\phi = J_{\neg p_0}$ (#2).
5. There are precisely two binary connectives, say $\uparrow$ and $\downarrow$ such that $S = \{\uparrow\}$ and $S = \{\downarrow\}$ are adequate (Problem set #2).