

MA410 Artificial Intelligence - Review of Logic

1 Symbols (L^AT_EX code given in brackets)

1.1 Logical connectives

\wedge (`\wedge`): meet, and, conjunction

\vee (`\vee`): join, or, disjunction

\neg (`\neg`): not, negation

\rightarrow (`\rightarrow`): if ... then ... , implication

\leftrightarrow (`\leftrightarrow`): ... if and only if ... , double-implication

1.2 Less common logical connectives

\uparrow (`\uparrow`): nand, alternative denial, *not both*

\downarrow (`\downarrow`): nor, joint denial, *neither ... nor ...*

\oplus (`\oplus`): xor, eor, exclusive or, *one or the other but not both*

1.3 Other symbols

$\top, 1$ (`\top`): true, tautology

$\perp, 0$ (`\bot`): false, contradiction

\models (`\models`): entails, models, logically follows from

\vdash (`\vdash`): can be derived by using a proof procedure

\equiv (`\equiv`): is equivalent to

\therefore (`\therefore`): therefore

\forall (`\forall`): for all, universal

\exists (`\exists`): there exists, existential

\leq (`\leq`): less than or equal to

2 Boolean Algebra

a set: A binary operators: \vee, \wedge unary operator: \neg elements: 0, 1

for all elements $a, b, c \in A$, the following axioms hold:

associativity	$a \vee (b \vee c) = (a \vee b) \vee c$	$a \wedge (b \wedge c) = (a \wedge b) \wedge c$
commutativity	$a \vee b = b \vee a$	$a \wedge b = b \wedge a$
absorption	$a \vee (a \wedge b) = a$	$a \wedge (a \vee b) = a$
distributivity	$a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$	$a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$
complements	$a \vee \neg a = 1$	$a \wedge \neg a = 0$

Partial-order relation \leq can be defined, where

$$a \leq b \text{ iff } a = a \wedge b \text{ or equivalently } a \leq b \text{ iff } b = a \vee b$$

3 Summary of Propositional Logic

Based on simple truth-definite statements.

3.1 Axiomatic System

An axiom is a proposition that is not proved or demonstrated but whose truth is taken for granted. It serves as a starting point for deducing and inferring other truths.

Σ : Alphabet of symbols

wff: Set of well-formed formulae

$A \in \mathbf{wff}$: Set of axioms

R: A set of rules of deduction.

3.2 Acceptable statements (Well-formed Formulae - wff for short)

- every atomic proposition
- $\neg P, P \vee Q, P \wedge Q, P \rightarrow Q$ given that P, Q are wff
- those found by finitely many uses of the above rules.

3.3 Deduction and Proof

Given a knowledge base, we can prove things to be true by using:

- A truth table
- Axiomatic logic (Modus Ponens) : $((A \rightarrow B) \wedge A) \rightarrow B$
- Resolution (Proof by contradiction, Reductio Ad Absurdum): $(\neg A \wedge \dots \wedge \dots \rightarrow 0) \rightarrow A$

3.3.1 Some truth table examples

p	q	$p \rightarrow q$	p	q	$p \vee q$	$(p \vee q) \vee (p \rightarrow q)$	p	q	$\neg q$	$\neg q \wedge (p \rightarrow q)$	$p \wedge \neg q \wedge (p \rightarrow q)$
0	0	1	0	0	0	1	0	0	1	1	0
0	1	1	0	1	1	1	0	1	0	0	0
1	0	0	1	0	1	1	1	0	1	0	0
1	1	1	1	1	1	1	1	1	0	0	0

3.3.2 Results from Truth Tables

Tautology All values in truth table give 1.

Contradiction All values in truth table give 0.

Contingency Neither a tautology nor a contradiction.

3.4 Some rules and definitions

Let ϕ, ψ, χ be some wffs.

3.4.1 Some Basic Rules

$$\text{B1: } \phi \vee \neg\phi \equiv 1$$

$$\text{B2: } \phi \wedge \neg\phi \equiv 0$$

$$\text{B3: } \phi \vee 1 \equiv 1$$

$$\text{B4: } \phi \wedge 1 \equiv \phi$$

$$\text{B5: } \phi \vee 0 \equiv \phi$$

$$\text{B6: } \phi \wedge 0 \equiv 0$$

3.4.2 Involution

$$\text{IV: } \neg(\neg\phi) \equiv \phi$$

3.4.3 Idempotency

$$\text{Im1: } \phi \wedge \phi \equiv \phi$$

$$\text{Im2: } \phi \vee \phi \equiv \phi$$

3.4.4 Implication Conversion

$$\text{I1: } \phi \rightarrow \psi \equiv \neg\phi \vee \psi$$

$$\text{I2: } \phi \leftrightarrow \psi \equiv (\neg\phi \vee \psi) \wedge (\phi \vee \neg\psi)$$

3.4.5 DeMorgan's laws

$$\text{DM1: } \neg(\phi \vee \psi) \equiv \neg\phi \wedge \neg\psi$$

$$\text{DM2: } \neg(\phi \wedge \psi) \equiv \neg\phi \vee \neg\psi$$

3.4.6 Breaking up into clauses

CNF: If C_1, \dots, C_n are clauses of the form $d_1 \vee \dots \vee d_{k_i}$ ($1 \leq i \leq n$) where each d_j is a literal¹, then $C_1 \wedge C_2 \wedge \dots \wedge C_n$ is said to be in the Conjunctive Normal Form.

DNF: If D_1, \dots, D_m are clauses of the form $c_1 \wedge \dots \wedge c_{k_i}$ ($1 \leq i \leq m$) where each c_j is a literal, then $D_1 \vee D_2 \vee \dots \vee D_m$ is said to be in the Disjunctive Normal Form.

3.4.7 Resolution rules

$$\text{R1: } \phi \wedge (\neg\phi \vee \psi) \models \psi \quad \text{also} \quad \phi \wedge (\phi \rightarrow \psi) \models \psi$$

$$\text{R2: } (\phi \vee \psi) \wedge (\neg\phi \vee \chi) \models \psi \vee \chi$$

¹an atomic proposition or its negation

4 Summary of Predicate Logic

4.1 Logic Formulae

constant (e.g. $n, m, \text{pat}, \text{mary}, \dots$)

predicate (e.g. $\text{smart}, \text{will_pass}, \dots$)

variable (e.g. $x, y, \text{Student}, \text{Example}, \dots$)

function (e.g. $f(x), g(x, y, z), \text{family}(\text{pat}, \text{mary}, \text{john}), \dots$)

quantifiers (universal \forall , existential \exists)

logical connectives ($\wedge, \vee, \neg, \rightarrow, \leftrightarrow$)

term: a variable, constant or function

atom: p or $p(t_1, \dots, t_n)$ where p is a predicate and each t_i is a term

4.2 Well-formed Formulae (wff)

- every atom
- $\neg P, P \vee Q, P \wedge Q, P \rightarrow Q$ where P, Q are wff
- $\forall x P, \exists x Q$ are wff where x is a variable and P, Q are wff
- those found by finitely many uses of the above rules.

4.3 Negation and Quantifiers

- $\neg(\forall x A(x)) \equiv (\exists x)(\neg A(x))$
- $\neg(\exists x A(x)) \equiv (\forall x)(\neg A(x))$

4.4 Types of variables

Bound variable x : $\forall x f(\dots x \dots)$.

Free variable x : Where it is not bound to a formula.

4.5 Types of formulae

Closed formula no free variables (also called a *sentence*)

Ground formula a proposition (i.e. no variables)

4.6 Knowledge Base (KB)

- the set of all clauses, axioms
- mapping of domain and predicates to real world (true or false)
- $KB \models g$ meaning g logically follows from KB. It is true in every model of KB.

4.7 Proof Systems

- A proof procedure is a way of proving theorems
- $KB \vdash g$ means that g can be derived from KB by using a proof procedure
- A proof procedure is **sound** if $KB \vdash g \implies KB \models g$
- A proof procedure is **complete** if $KB \models g \implies KB \vdash g$
- Bottom-up, top-down proof procedures