Foundation of Logic

What is logic?

"Logic is the beginning of wisdom, not the end"

The branch of philosophy concerned with analysing the patterns of reasoning by which a conclusion is drawn from a set of premises, without reference to meaning or context is known as logic.

(*Collins English Dictionary*)

Why is logic important?

- Logic is a **formal method for reasoning**.
- Logic is a formal language for **deducing** knowledge from a small number of explicitly stated **premises** (or hypotheses, axioms, facts).
- Logic provides a formal framework for **representing knowledge.**
- Logic differentiates between the **structure** and **content** of an argument.

What is an argument?

- An argument is just a **sequence of statements.**
- Some of these statements, the **premises**, are assumed to be true and serve as a basis for accepting another statement of the argument, called the **conclusion.**

Deduction and Inference

- If the conclusion is justified, based solely on the premises, the process of reasoning is called **deduction.**
- If the validity of the conclusion is based on *generalisation* from the premises, based on strong but inconclusive evidence, the process is called **inference** (sometimes called **induction**).

Two examples

- **Deductive** argument:

  "Alexandria is a port or a holiday resort. Alexandria is not a port. Therefore, Alexandria is a holiday resort."

- **Inductive** argument

  "Most students who did not do the tutorial questions will fail the exam. John did not do the tutorial questions. Therefore John will fail the exam."

Foundation of Logic

Mathematical Logic is a tool for working with compound statements. It includes:

- A formal language for expressing them.
- A methodology for objectively reasoning about their truth or falsity.
- It is the foundation for expressing formal proofs in all branches of mathematics.

We will talk about two logical systems:

- Propositional logic
- Predicate logic

Propositional Logic

Propositional Logic **is the logic of compound statements built from simpler statements using so-called** ***Boolean connectives.***

Some applications in computer science:

- **Design of digital electronic circuits.**
- **Expressing conditions in programs.**
- **Queries to databases & search engines.**

Definition: A ***proposition*** **is simply:**

- **a** ***statement*** **(*i.e.*, a declarative sentence)**
  - ***with some definite meaning***
- **having a** ***truth value*** **that's either** ***true*** **(T) or** ***false*** **(F)**
  - **it is never both, neither, or somewhere "in between!"**
  - **however, you might not** ***know*** **the actual truth value.**

Propositions in Propositional Logic

- **Simple types of statements, called propositions, are treated as atomic building blocks for more complex statements.**

- **Atoms: *p*, *q*, *r*, …**
  (Corresponds with simple English sentences, e.g.
  `I had salad for lunch')

- **Complex propositions : built up from atoms using operators: $p \wedge q$**
  (Corresponds with compound English sentences, e.g., "I had salad for lunch and I had steak for dinner.")

Some Popular Boolean Operators

Formal Name	Nickname	Arity	Symbol
Negation operator	NOT	Unary	$\neg$
Conjunction operator	AND	Binary	$\wedge$
Disjunction operator	OR	Binary	$\vee$
Exclusive-OR operator	XOR	Binary	$\oplus$
Implication operator	IMPLIES	Binary	$\rightarrow$
Biconditional operator	IFF	Binary	$\leftrightarrow$

The Negation Operator

The unary *negation operator* "$\neg$" (*NOT*) transforms a prop. into its *negation*.

E.g. If p = "I have brown hair."

then $\neg p$ = "I do **not** have brown hair."

The *truth table* for NOT:

p	$\neg p$
T	F
F	T

T :≡ True; F :≡ False

The Conjunction Operator

The binary *conjunction operator* "∧" (*AND*) combines two propositions to form their logical *conjunction*.

***E.g.* If p="I will have salad for lunch." and q="I will have steak for dinner.", then $p \wedge q$="I will have salad for lunch and I will have steak for dinner."**

- **Note that a conjunction $p_1 \wedge p_2 \wedge \ldots \wedge p_n$ of n propositions will have 2^n rows in its truth table.**

p	q	$p \wedge q$
F	F	F
F	T	F
T	F	F
T	T	T

The Disjunction Operator

The binary *disjunction operator* "$\lor$" (*OR*) combines two propositions to form their logical *disjunction*.

***p*="My car has a bad engine."**

***q*="My car has a bad carburator."**

$p \lor q$="Either my car has a bad engine, or my car has a bad carburetor."

- Note that $p \lor q$ means that *p* is true, or *q* is true, **or both** are true!
- So, this operation is also called *inclusive or,* because it **includes** the possibility that both *p* and *q* are true.

p	q	$p \lor q$
F	F	F
F	T	**T**
T	F	**T**
T	T	T

Nested Propositional Expressions

- Use parentheses to *group sub-expressions*:
  "I just saw my old friend, and either he's grown or I've shrunk." = $f \wedge (g \vee s)$
  - $(f \wedge g) \vee s$ would mean something different
  - $f \wedge g \vee s$ would be ambiguous
- By convention, "$\neg$" takes *precedence* over both "$\wedge$" and "$\vee$".

Tautologies/Contradictions/Contigencies

A *tautology* is a compound proposition that is **true** *no matter what* the truth values of its atomic propositions are!

Ex. $p \vee \neg p = 1$

A *contradiction* is a compound proposition that is **false** no matter what the truth values of its atomic propositions are!

Ex. $p \wedge \neg p = 0$

All other propositions are *contingencies*:

Some rows give T, others give F.

Logical Equivalence

Compound proposition *p* is *logically equivalent* to compound proposition *q*, written $p \Leftrightarrow q$, **IFF** *p* and *q* contain the same truth values in all rows of their truth tables

We will also say: they express the same truth function (= the same function from values for atoms to values for the whole formula).

Ex. Prove that $p \vee q \Leftrightarrow \neg(\neg p \wedge \neg q)$.

p	q	$p \vee q$	$\neg p$	$\neg q$	$\neg p \wedge \neg q$	$\neg(\neg p \wedge \neg q)$
F	F	F	T	T	T	F
F	T	T	T	F	F	T
T	F	T	F	T	F	T
T	T	T	F	F	F	T

The *Exclusive Or* Operator

The binary *exclusive-or operator* "$\oplus$" (*XOR*) combines two propositions to form their logical "exclusive or".

p = "I will earn an A in this course,"

q = "I will drop this course,"

$p \oplus q$ = "I will either earn an A in this course, or I will drop it (but not both!)"

- Note that $p \oplus q$ means that p is true, or q is true, but **not both**!
- This operation is called *exclusive or,* because it **excludes** the possibility that both p and q are true.

p	q	$p \oplus q$
F	F	F
F	T	T
T	F	T
T	T	**F**

The *Implication* Operator

The *implication* $p \rightarrow q$ states that p implies q.

i.e., if p is true, then q is true; but if p is not true, then q could be either true or false.

e.g., let p = "You study hard."
q = "You will get a good grade."

$p \rightarrow q$ = "If you study hard, then you will get a good grade."

- $p \rightarrow q$ is **false** only when (p is true but q is **not** true)
- $p \rightarrow q$ does **not** say that p causes q!
- $p \rightarrow q$ does **not** require that p or q **are true**!

p	q	$p \rightarrow q$
F	F	T
F	T	T
T	F	**F**
T	T	T

Implications between real sentences

- "If this lecture ever ends, then the sun has risen this morning." *True* or *False*?
- "If Tuesday is a day of the week, then I am a penguin." *True* or *False*?
- "If 1+1=6, then Bush is president." *True* or *False*?
- "If the moon is made of green cheese, then I am richer than Bill Gates." *True* or *False*?

Biconditional Truth Table

- $p \leftrightarrow q$ means that p and q have the **same** truth value.
- Note this truth table is the exact **opposite** of $\oplus$'s!

  Thus, $p \leftrightarrow q$ means $\neg(p \oplus q)$
- $p \leftrightarrow q$ does **not** imply that p and q are true, or that either of them causes the other.

p	q	$p \leftrightarrow q$
F	F	T
F	T	F
T	F	F
T	T	T

Converse/Contrapositive

Some terminology, for an implication $p \rightarrow q$:

- Its *converse* is: $q \rightarrow p$.
- Its *contrapositive*: $\neg q \rightarrow \neg p$.

Logical Equivalences

- *Identity:* $p \wedge \mathbf{T} \Leftrightarrow p \qquad p \vee \mathbf{F} \Leftrightarrow p$
- *Domination:* $p \vee \mathbf{T} \Leftrightarrow \mathbf{T} \qquad p \wedge \mathbf{F} \Leftrightarrow \mathbf{F}$
- *Idempotence:* $p \vee p \Leftrightarrow p \qquad p \wedge p \Leftrightarrow p$
- *Double negation:* $\neg\neg p \Leftrightarrow p$
- *Commutativity:* $p \vee q \Leftrightarrow q \vee p \qquad p \wedge q \Leftrightarrow q \wedge p$
- *Associativity:* $(p \vee q) \vee r \Leftrightarrow p \vee (q \vee r)$

  $(p \wedge q) \wedge r \Leftrightarrow p \wedge (q \wedge r)$

More Equivalence Laws

- *Distributive*: $p\lor(q\land r) \Leftrightarrow (p\lor q)\land(p\lor r)$

  $p\land(q\lor r) \Leftrightarrow (p\land q)\lor(p\land r)$

- *De Morgan's Laws*:

  $\neg(p\land q) \Leftrightarrow \neg p \lor \neg q$

  $\neg(p\lor q) \Leftrightarrow \neg p \land \neg q$

- *Trivial tautology/contradiction*:

  $p \lor \neg p \Leftrightarrow \mathbf{T}$ $\quad$ $p \land \neg p \Leftrightarrow \mathbf{F}$

Augustus De Morgan (1806-1871)

Predicate Logic

- *Predicate logic* is an extension of propositional logic that permits quantification over classes of entities.
- Propositional logic (recall) treats *simple propositions* (sentences) as atomic entities.
- A 'predicate' is just a property
- Predicates define relationships between any number of entities using qualifiers:

  $\forall$ "for all", "for every"

  $\exists$ "there exists"

  Remember:

  $\forall$ *x* 'for every *x*', or 'for <u>A</u>ll *x*'

  $\exists$ *x* 'there is an *x*' or 'there <u>E</u>xists an *x*'

Applications of Predicate Logic

- It is one of the most-used formal notations for writing mathematical *definitions*, *axioms*, and *theorems*.
- For example, in *linear algebra*, a *partial order* is introduced saying that a relation R is *reflexive* and *transitive* – and these notions are defined using predicate logic.
- Basis for many Artificial Intelligence systems.
- Predicate-logic like statements are supported by some of the more sophisticated *database query engines.*

Propositional/Predicate logic

- In propositional logic, we could not simply say whether a formula is TRUE; what we could say is whether it is TRUE with respect to a given assignment of TRUE/FALSE to the Atoms in the formula
- e.g., $p \rightarrow q$ is TRUE with respect to the assignment p =TRUE, q = TRUE
- In predicate logic, we say that a formula is TRUE (FALSE) with respect to a **model.**