Logic is a system/language to formalise arguments, expressions, and other methods of communicating.

Syntax – a logic will consist of symbols and rules for how the symbols must be used to construct well-formed formulas.

Semantics – the semantics of a logic provide the context (a domain of discourse) to interpret a formula, and give a meaning to the formula.

Rules of Inference – a logic provides rules of inference, i.e. formulas which can be semantically interpreted based on the interpretation of a set of premises. This is a system often used to reason about the syntax and semantics of a logic.

An **inference/proof system** is a collection of rules to construct formal proofs by reaching conclusions from a set of axioms. The rules of inference systems are defined in syntactically valid formulas independent of semantics, reliant only on syntax.

A system of logic is **sound** when its proof system cannot derive conclusions from a set of premises unless it is semantically entailed by (a consequence of) them.

A system of logic is **complete** when its proof system can derive every conclusions which is semantically entailed by its premises.

A system of logic is **consistent** when no contradiction cannot be derived by its proof system.

Boolean Variables – a variable which can be either True or False.

Alphabet – a nonempty set of symbols.

Truth Values – a Boolean value. The value true can be represented by the constants True, T, 1, or ⊤ (top formula, a given tautology, true in every valuation), these all always represent true regardless of semantics. The value false can be represented by the constants False, F, 0, or ⊥ (bottom formula, a given contradiction, false in every valuation), these all always represent false regardless of semantics.

Variable – a symbol from the alphabet which takes a value from the domain of discourse when semantically interpreted.

Domain (or Universe) of Discourse – the set of values which variables may take.

Propositional Logic

Proposition – a declarative statement or sentence (declares something that is), which must be either true or false.

Propositional Variables – letters/Boolean variables representing propositions under some set of semantics, when no semantics are provided they cannot be given a Boolean value as there is no associated proposition. Often represented by the letters $P, Q, R, S, \ldots$, or in lowercase (depending on convention).

Atomic Proposition – a single fact or single statement about a single thing. It is a proposition which cannot be broken down into a composition of other propositions. Represented by propositional letters.

Formula – all atomic propositions are formulas (specifically atomic formulas), the application of propositional/logical connectives to formulas are formulas, and nothing else is a formula.

Truth Table – a table showing all possible truth values of propositions, this includes each progressive stage of building up a formula with proper syntax in the grammar. The number of 'cases' of providing values to a formula with n atomic propositions is 2^n.

Syntax for Propositional Logic:

- Assuming there is an infinite alphabet, $A ::= P, Q, R, \ldots$, of propositional variables/letters (letters can be subscripted, for example P_1, P_2 to continue this into infinity).
- The standard grammar for all well-formed propositional formulas ϕ is:

$$\begin{aligned} \phi ::= {} & A \\ & \mid \neg\phi && \text{(negation)} \\ & \mid (\phi \wedge \phi) && \text{(conjunction)} \\ & \mid (\phi \vee \phi) && \text{(disjunction)} \\ & \mid (\phi \rightarrow \phi) && \text{(implication)} \\ & \mid (\phi \leftrightarrow \phi) && \text{(equivalence)} \end{aligned}$$

- The outermost brackets can clearly be dropped, and the brackets can be dropped where precedence is followed in a formula.
- Note: it can be shown that any well-formed formula can be expressed by only disjunctions and negations, only conjunctions and negations, or only implications and negations.
- Schematic letters, such as φ, ψ, χ are often used to represent formulas.
- **Precedence (highest to lowest):** negations, conjunctions, disjunctions, implications, equivalences. This would be considered the standard convention for precedence, although it may be that disjunctions and conjunctions have equal precedence and the same for implications and equivalences.

Logical Connectives:

- Note: where propositional letters have been used, they can be more generally considered as well-formed formula, the syntax and rules applies the same.
- Applied to one or more Boolean values or variables.

- **Negation:** unary connective NOT, symbol ¬.
  - $\neg P$ is true only when P is false (not true), and false only when P is true.
  - Truth table:

P	$\neg P$
F	T
T	F

- **Conjunction:** binary connective AND, symbol ∧.
  - $P \wedge Q$ is true only when both P and Q are true.
  - Truth table:

P	Q	$P \wedge Q$
F	F	F
F	T	F
T	F	F
T	T	T

- Also note a less clear English equivalent 'P but Q'.

- **Disjunction:** binary connective (inclusive or) OR, symbol $\vee$.
  - $P \vee Q$ is true only when at least one of P or Q are true.
  - Truth table:

P	Q	$P \vee Q$
F	F	F
F	T	T
T	F	T
T	T	T

  - **Exclusive Disjunction:** binary connective (exclusive or) XOR, symbols $\not\Leftrightarrow$, $\oplus$, $\not\leftrightarrow$.
    - $(P \vee Q) \wedge \neg(P \wedge Q) \equiv P \not\Leftrightarrow Q$
    - In English 'either P or Q', or 'P or Q, but not both'.
  - 'Neither P nor Q' is interpreted as $\neg(P \vee Q)$.

- **Implication:** binary connective (conditional) IF . . . THEN, symbols $\Rightarrow$, $\rightarrow$.
  - $P \Rightarrow Q$ can be interpreted in English in many ways:
    - P implies Q, i.e. if P is true, then Q must also be true (P only if Q).
    - P is a sufficient condition for Q. If P, it follows that Q.
    - Q is a necessary condition for P, i.e. P cannot be true if Q is not true.
  - In $P \Rightarrow Q$, P is known as the **hypothesis** (**antecedent** or **premise**) and Q the **conclusion** (**consequence** or **consequent**).
  - Truth table:

P	Q	$P \Rightarrow Q$	$\neg P \vee Q$
F	F	T	T
F	T	T	T
T	F	F	F
T	T	T	T

  - $P \Rightarrow Q$ is representative of $\neg P \vee Q$. This is the material implication (a rule of inference), which can be seen to hold from the truth table.
  - The negation of the implication is when the implication is false, hence, it occurs when $P \wedge \neg Q$.
  - The converse of $P \Rightarrow Q$ is $Q \Rightarrow P$.
  - The inverse of $P \Rightarrow Q$ is $\neg P \Rightarrow \neg Q$.
  - The converse and inverse are equivalent.
  - Contraposition of the implication:
    - Since $P \Rightarrow Q$ is the same as $Q \vee \neg P$, and hence $\neg Q \Rightarrow \neg P$.
    - $\neg Q \Rightarrow \neg P$ is the contrapositive form of $P \Rightarrow Q$.
  - Other common logical equivalences with implication:
    - $(P \Rightarrow Q) \vee (P \Rightarrow R) \equiv P \Rightarrow (Q \vee R)$
    - $(P \Rightarrow Q) \wedge (P \Rightarrow R) \equiv P \Rightarrow (Q \wedge R)$
    - $(P \Rightarrow R) \vee (Q \Rightarrow R) \equiv P \Rightarrow (Q \Rightarrow R) \equiv (P \wedge Q) \Rightarrow R$
    - $(P \Rightarrow R) \wedge (Q \Rightarrow R) \equiv (P \vee Q) \Rightarrow R$
  - It is possible to represent all other logical connectives using only implication, given a bottom formula $\perp$ which is always false.

- **Equivalence:** binary connective XNOR, symbols $\Leftrightarrow$, $\leftrightarrow$.
  - Also called a biconditional or a bi-implication..

- $(P \Leftrightarrow Q) \equiv (P \Rightarrow Q) \wedge (Q \Rightarrow P)$, hence P and Q are only equivalent when P has the same semantic value as Q.
- This can also be said as 'P if, and only if, Q', (often abbreviated 'iff') i.e. if P is true, then Q must also be true, and the converse, if Q is true, then P must also be true.
- Truth table:

P	Q	$P \Leftrightarrow Q$
F	F	T
F	T	F
T	F	F
T	T	T

- Other common logical equivalences with equivalence:
  - $(P \Leftrightarrow Q) \equiv (P \wedge Q) \vee (\neg P \wedge \neg Q)$
  - $(P \Leftrightarrow Q) \equiv (\neg P \Leftrightarrow \neg Q)$
  - $\neg(P \Leftrightarrow Q) \equiv (P \Leftrightarrow \neg Q) \equiv (\neg P \Leftrightarrow Q)$

- **Syntactic Laws and Logical Equivalences:**
  - **De Morgan's:**
    Shows how disjunctions and conjunctions should be negated, also allows a conjunction to be defined in terms of only negations and conjunctions, and vice-versa, i.e. $P \vee Q \equiv \neg(\neg P \wedge \neg Q)$.
    $\neg(P \wedge Q) \equiv \neg P \vee \neg Q$
    $\neg(P \vee Q) \equiv \neg P \wedge \neg Q$
  - **Identity:**
    $P \wedge T \equiv P$
    $P \vee F \equiv P$
  - **Domination:**
    $P \wedge F \equiv F$
    $P \vee T \equiv T$
  - **Idempotent**:
    $P \wedge P \equiv P$
    $P \vee P \equiv P$
  - **Double negation:**
    $\neg\neg P \equiv P$
  - **Commutativity:**
    $P \wedge Q \equiv Q \wedge P$
    $P \vee Q \equiv Q \vee P$
  - **Associative:**
    $P \wedge (Q \wedge R) \equiv (P \wedge Q) \wedge R$
    $P \vee (Q \vee R) \equiv (P \vee Q) \vee R$
  - **Distributive:**
    $P \wedge (Q \vee R) \equiv (P \wedge Q) \vee (P \wedge R)$
    $P \vee (Q \wedge R) \equiv (P \vee Q) \wedge (P \vee R)$
  - **Absorption:**
    $P \wedge (P \vee Q) \equiv P$
    $P \vee (P \wedge Q) \equiv P$
  - **Negation:**
    $P \wedge \neg P \equiv F$ (law of noncontradiction, two directly contradictory statements cannot be true at the same time)

$P \vee \neg P \equiv T$ (law of the excluded middle, every proposition is true or false)

- Notation for a sequence of disjunctions or conjunctions:

$$\bigwedge_{i=1}^{n} P_i \equiv P_1 \wedge P_2 \wedge \ldots \wedge P_n$$

$$\bigvee_{i=1}^{n} P_i \equiv P_1 \vee P_2 \vee \ldots \vee P_n$$

  - Which also allow for De Morgan's laws, which follow by the associative laws.

$$\neg \bigwedge_{i=1}^{n} P_i \equiv \bigvee_{i=1}^{n} \neg P_i$$

$$\neg \bigvee_{i=1}^{n} P_i \equiv \bigwedge_{i=1}^{n} \neg P_i$$

Propositional Semantics:

- **Valuation** – a propositional valuation is a function v which maps all propositional letters to a Boolean value, i.e. for the alphabet A as the set of all propositional letters, $v: A \to \{F, T\}$. A valuation is an assignment of truth values, i.e. semantics, to all atomic propositions (propositional letters).
- It is important to note that for n propositional letters in a formula, there are 2^n possible valuations of that formula, since the cardinality of the domain of discourse for propositional logic is 2.
- The valuation of a proposition P under v can be represented $v(P)$.
- A valuation of a formula is the result of applying the valuation to all propositional letters within the formula and then evaluating the connectives applied to the truth values, i.e. the interpretation of each propositional letter in the formula.
- A valuation can be considered as one row in a truth table, and every valuation of a formula can be thought of as the entire truth table.
- **Satisfiability** – a formula φ is satisfiable if and only if there exists some valuation v under which φ is true, this can be denoted $v \vDash \varphi$. A formula which is not satisfied under a specific valuation can be denoted $v \nvDash \varphi$, i.e. it is not satisfied under v. An unsatisfiable formula is always false, i.e. a **contradiction**, such that no valuation exists for which the formula is satisfied.
- **Validity** – a formula φ is logically valid (a **tautology**) if and only if it is satisfied by every valuation over its propositional letters, i.e. every combination of truth assignments to letters, this is denoted $\vDash \varphi$. A formula is invalid if and only if it is not satisfied by at least one valuation.
- Two formulas φ and ψ are **logically equivalent**, i.e. $\varphi \equiv \psi$ if and only if $\vDash \varphi \Leftrightarrow \psi$, i.e. $\varphi \Leftrightarrow \psi$ is valid for all valuations. This means that the two formulas can be interchanged.
- The valuation of formulas can be represented by relations:

$$v \vDash \varphi \equiv v(\varphi) = \top$$

$$v \vDash \neg\varphi \equiv v \nvDash \varphi \equiv v(\varphi) = \bot$$

$$v \vDash (\varphi_1 \wedge \varphi_2) \equiv (v \vDash \varphi_1) \; and \; (v \vDash \varphi_2)$$

$$v \vDash (\varphi_1 \vee \varphi_2) \equiv (v \vDash \varphi_1) \; or \; (v \vDash \varphi_2)$$

$$v \vDash (\varphi_1 \Rightarrow \varphi_2) \equiv (v \nvDash \varphi_1) \; or \; (v \vDash \varphi_2)$$

$$v \vDash (\varphi_1 \Leftrightarrow \varphi_2) \equiv \big((v \vDash \varphi_1) \; and \; (v \vDash \varphi_2)\big) \; or \; \big((v \nvDash \varphi_1) \; and \; (v \nvDash \varphi_2)\big)$$

- If two valuations v and w agree on all propositions, then for all propositional letters P, $v(P) = w(P)$, and hence for all formulas φ, $v(\varphi) = w(\varphi)$.
- If a formula is valid, then it is clearly satisfiable. If a formula is unsatisfiable, it is clearly invalid.
- A formula φ is unsatisfiable when $\neg\varphi$ is valid.
- A formula φ is invalid when $v \vDash \neg\varphi$ for some valuation v ($\neg\varphi$ is satisfiable).
- **Contingency** – a formula which is neither a tautology nor a contradiction, i.e. it is satisfiable but not valid.

Paradoxes of Material Implication:

- A family of formulas containing the implication which are valid, but do not translate intuitively to natural language.
- **Vacuous Implication:** $Q \Rightarrow (P \Rightarrow Q)$, i.e. whenever the conclusion is true, the implication is always true
- **Paradox of Entailment:** $\neg P \Rightarrow (P \Rightarrow Q)$, i.e. whenever the hypothesis is false, the implication is always true
- Another potentially unintuitive valid formula: $(P \Rightarrow Q) \vee (Q \Rightarrow R)$

Duality:

- The **dual** of a formula φ is represented as φ^*, and given by negating all propositional letters in φ (and truth values), and replacing all conjunctions with disjunctions and all disjunction with conjunctions.
- **Duality Principle:** for all formulas φ, $\varphi \equiv \neg\varphi^*$.
- Also note for formulas φ and ψ, $(\varphi \Leftrightarrow \psi) \equiv (\varphi^* \Leftrightarrow \psi^*)$.

Normal Form – a logical formula which follows a specific structure.

Literal – any atomic proposition, or its negation.

Clause – a logical formula formed of a finite set of literals and logical connectives.

Disjunctive Normal Form (DNF):

- A formula D is in disjunctive normal form if and only if it follows the grammar:
  $L ::= P \mid \neg P$ (literal)
  $C ::= L \mid L \wedge C$ (conjunctive clause)
  $D ::= (C) \mid (C) \vee D$
- Every formula has an equivalent DNF.
- Constructing the DNF of a formula:
    - Syntactically, by using syntactic equivalences/laws:
        - Represent all connectives by equivalent forms using only negations, conjunctions, and disjunctions.
        - Move the negations inwards (into the conjunctive clauses), using De Morgan's Laws and the law of Double Negation.
        - Distribute the conjunctions over disjunctions to move the conjunctions inwards.
    - Semantically, by reading a truth table:
        - Construct a truth table for the formula.

        - Every case where the formula is satisfied, construct a conjunctive clause between all True propositional letters, and the negations of all False letters.
    - Combine the conjunctive clauses with disjunctions.
- DNF is very useful for checking the satisfiability of a formula, as it only needs to be shown that one of its conjunctive clauses is satisfiable. Similarly, if a conjunctive clause can be shown valid, the entire DNF is valid.
- Additionally, it is clear a conjunctive clause is satisfiable if for every literal within the clause, the clause does not also contain the negation of the literal (otherwise it would be unsatisfiable).
- This also means the DNF of the negation of a formula is easily used to check the formula's validity, by determining if every conjunctive clause is unsatisfiable or not.
- The use of DNF to check satisfiability can have a much better computational complexity than checking the entire formula.

Conjunctive Normal Form (CNF):

- A formula C is in conjunctive normal form if and only if it follows the grammar:

  $L ::= P \mid \neg P$ (literal)

  $D ::= L \mid L \vee D$ (disjunctive clause)

  $C ::= (D) \mid (D) \wedge C$
- Constructing CNF:
    - Syntactically:
        - Represent all connectives by equivalent forms using only negations, conjunctions, and disjunctions.
        - Move the negations inwards (into the disjunctive clauses).
        - Distribute the disjunctions over conjunctions to move the disjunctions inwards
    - By using DNF:
        - Find the DNF of the negation of a formula by some method.
        - The dual of the DNF of the negated formula is the CNF of the formula, by the duality principle.
        - Informal proof: $\neg\varphi \equiv \bigvee \bigwedge L$, $\neg\neg\varphi \equiv \neg \bigvee \bigwedge L \equiv \bigwedge \neg \bigwedge L \equiv \bigwedge \bigvee \neg L$.
- Every formula has an equivalent CNF, as a CNF can be derived from a DNF of the formulas negation.
- CNF is useful for proving theorems, as the formula is valid if and only if each of its disjunctive clauses are valid. Also note each disjunctive clause is valid if and only if one of its literals is valid.