Notes by Ioan Steffan Thomas

Limitations of Propositional Logic:

- Is not expressive of the internal structure of propositions, you cannot detach facts/properties from the thing/object it is about.
- There is no way to formally quantify things, which means it cannot explain validity of Aristotelian type arguments are valid, for example:
    All men are mortal.
    Socrates *is a* man.
    Therefore, Socrates *is* mortal.
- This is the motivation for first-order logic, which can be considered an extension on propositional logic.

See "Defining Peano Arithmetic" for examples of the concepts here.

First-Order Logic (FOL)

The first thing to note about first-order logic is that everything from propositional logic applies here, the logical connectives will remain defined the same, however slight variations or generalisations of other aspects may occur.

Propositions are statements, properties, about things. But in predicate logic, the things and properties can be separated, respectively into quantified variables (which represent things) and predicates (which assert properties). Propositional logic allows for common things or properties between propositions to be clearly related.

Term – any expression which can be used to reference an object from the domain of discourse.

Predicate – an assertion (mapping to a truth value) of a property or relation to terms. Also called propositional functions. Often P, Q, R, S . . .. Predicates are often represented by uppercase letters. It is important to note that predicates are interpreted as *relations*, and the argument tuple they appear to apply to is interpreted as whether it is contained within that interpreted relation.

Defining First-Order Languages:

- Assuming there is an infinite alphabet, $A ::= x, y, z, \ldots$, of variables (can be subscripted to create more symbols).
- A first-order **language** Σ has the structure $\Sigma = (\mathcal{F}, \mathcal{P})$.
    - A set of function symbols $\mathcal{F} = \{f_1, f_2, \ldots\}$, each with an arity $k, k \geq 0$. A constant (symbol) is a function with arity 0.
    - A set of predicate (or relation) symbols $\mathcal{P} = \{P_1, P_2, \ldots\}$, each with an arity $k, k \geq 0$, which are relations between k elements, or when $k = 1$ a property about a single element. **A propositional letter can be considered as a predicate with arity 0 (so propositional logic is a subset of first-order logic).**
    - Note that a binary function or predicate can be written in prefix or infix, as long as precedence is clear when in infix.
- Inductive rules for the definition of **terms**:
    (1) Any variable symbol is a term.

   (2) If f is an n-ary, $n \geq 0$, function symbol, and for $0 < k \leq n$, t_k is a term, then $f(t_1, t_2, \ldots, t_n)$ is also a term.
   (3) Nothing else is a term.
- A predicate is not a term, as it is a relation on the set of terms formed by (1) and (2).
- The formal grammar for a term t might be shown $t ::= A \mid f(t_1 \ldots t_n)$.
- To define a first-order language, each symbol must be described by an arity, and a type (constant, function, or predicate).
- The standard grammar for all well-formed formulas ϕ of a first-order language is:

$\phi ::= P(t_1, \ldots, t_n)$	(atomic (or predicate) formula with arity n, over terms $t_k, 0 < k \leq n$, of Σ)
$\mid \neg\phi$	(negation)
$\mid (\phi \wedge \phi)$	(conjunction)
$\mid (\phi \vee \phi)$	(disjunction)
$\mid (\phi \rightarrow \phi)$	(implication)
$\mid (\phi \leftrightarrow \phi)$	(equivalence)
$\mid \forall x(\phi)$	(universal quantification of a variable symbol $x \in A$)
$\mid \exists x(\phi)$	(existential quantification of a variable symbol $x \in A$)

  - Note, the scope of the above quantifiers is the formula they are applied to (the bracketed formulas).
  - **Precedence (highest to lowest):** predicates applied to terms, variables quantified, negation, conjunctions, disjunctions, implications, equivalences. This is a common precedence convention.

Common Shorthand Conventions:

- If the quantifier (or a string of quantifiers), Q, is outermost in a formula, i.e. $Qx(\phi)$, it is often written $Qx.\phi$, which is unambiguous.
- A predicate with arity 1 on a variable $x \in A$ can be written more succinctly as Px.
- A quantifier Q, defined as before, applied to an atomic formula, Px may be written more succinctly as $QxPx$.
- Immediate sequences of quantifiers may omit brackets to contain the next quantifier, so long as it is clear they all end up quantifying over the same formula. For example, for quantifiers $Q_1, Q_2, Q_3, \ldots Q_n$, then $Q_1x_1\left(Q_2x_2\left(\ldots\left(Q_nx_n(\varphi)\right)\right)\right)$ may be written $Q_1x_1Q_2x_2 \ldots Q_nx_n(\varphi)$.
- Immediate sequences of the same quantifier symbol which quantify variables over the same domain, it is often written so that only one symbol need be required. For example for the quantifier Q, $Qx_1Qx_2 \ldots Qx_n(\varphi)$ may be written $Qx_1, x_2, \ldots, x_n(\varphi)$.

Logical Connectives:

- All logical connectives and laws found in propositional logic are valid here. I.e. all propositional logical equivalences are also first-order logical equivalences.

- **Quantifiers:**
  - Logic connectives for reasoning over a set of values which a variable can take when semantically interpreted.
  - Each quantifier in a formula can be interpreted with a different subset of the domain of discourse.
  - Quantifiers are applied to a variable, and a formula.

  - When applied to an atomic formula, brackets can be omitted.
  - It is important to remember that quantifiers will cover all possible permutations of available values to bind to each variable.

- **Scope:**
  - The scope of a quantifier applies to the formula which it is applied to.
  - A variable is said to be **bound** to a quantifier if the quantifier quantifies values over (is applied to) the variable. Any unquantified variable is said to be **free**,
  - An occurrence of a variable symbol within a formula is said to be a **bound occurrence** if the occurrence is within the scope of a quantifier which binds this symbol, otherwise it is said to be a **free occurrence**.
  - **Nested Quantifiers:**
    - Quantifiers can be **nested**, and have nested scopes, i.e. one variable can be bound, and the formula a quantifier is applied to can contain another quantifier which binds the same or a different variable symbol.
    - I.e. the sets of symbols for free and bound variables need not be disjoint.
    - The binding of a specific variable symbol in nested quantifications gives precedence to the innermost encompassing quantifier of the occurrence of the symbol.
  - A quantified variable is bound within the scope of it's quantifier, and free outside of the scope.
  - A first-order **formula** with **no freely occurring** variable symbols is called a first-order **sentence** or a closed expression.

- **Universal Quantifier:** FOR ALL, symbol ∀.
  - For all x, it is that φ. This states that for all valuations of x in the given domain, it is such that a formula φ is true (i.e. φ is valid).
  - The above statement can be written as '$\forall x(\varphi)$'. This expression is true if φ is true for every value of x in the domain. And x is bound within (φ).
  - Other English interpretations include: "all", "any", "every".

- **Existential Quantifier:** THERE EXISTS, symbol ∃.
  - There exists x such that φ. This states that there is a valuation of x in the given domain for which φ is true (i.e. φ is satisfiable).
  - The above statement can be written as '$\exists x(\varphi)$'. This expression is true if φ is true for at least one value of x in the domain.
  - Other English interpretations include: "there is", "at least", "some".

- **Variable Substitution:**
  - For a formula φ, a variable symbol x and a term t, $\varphi[t/x]$ is the formula obtained by replacing all free occurrences of x in φ with t.
  - It must be checked that no accidental capture of free variables occurs by binders when they are substituted, for example $\big(\forall x(x \neq y)\big)[x/y]$ becomes $\big(\forall x(x \neq x)\big)$, which demonstrates accidental capture.

- Note on domains and quantifications:

- Depending on the universe of a quantifier, more or less predicates may be required to translate a linguistic sentence.
- Additionally, quantifications may not be explicitly seen in sentences to translate.
- For example: “Carobs grow on trees.”:
    - Take the predicates/relations:
        - $C(x)$: "x is a carob"
        - $T(x)$: "x is a tree"
        - $G(x,y)$: "x grows on y"
    - The formal equivalent of this sentence, for the unbounded universe of all things, is $\forall x. C(x) \rightarrow \exists y\big(T(y) \wedge G(x,y)\big)$

- **Quantifier Laws and Logical Equivalences:**
    - In all equivalences below, unless stated otherwise, assume any formula symbol to contain free occurrences of the variable symbol being quantified, although any case where free occurrences occur is fine to use when no free occurrences occur, it is the converse which is not true.

    - Where there are n elements in the domain of discourse, each represented distinctly by $x_i, 0 < i \leq n$, if the quantifiers over a symbol x are over this domain then:

$$\forall x(\varphi) \equiv \bigwedge_{i=1}^{n} \varphi[x_i/x]$$
$$\exists x(\varphi) \equiv \bigvee_{i=1}^{n} \varphi[x_i/x]$$

    - **De Morgan’s Laws for Quantifiers:**
      Following from the above equivalences, describing how quantifiers are to be negated and also showing that only one of the elementary quantifiers is required.
      $\neg\forall x(\varphi) \equiv \exists x(\neg\varphi)$
      $\neg\exists x(\varphi) \equiv \forall x(\neg\varphi)$
    - **Quantifier ‘Commutativity’:**
      $\forall x\forall y(\varphi) \equiv \forall y\forall x(\varphi)$
      $\exists x\exists y(\varphi) \equiv \exists y\exists x(\varphi)$
    - **Renaming Bound Variables:**
      $\forall x(\varphi) \equiv \forall z(\varphi[z/x])$ if and only if z does not occur freely in φ
      $\exists x(\varphi) \equiv \exists z(\varphi[z/x])$ if and only if z does not occur freely in φ
    - **Vacuous Quantifications/No Free Occurrences:**
      $\forall x(\varphi) \equiv \varphi$ if and only if x does not occur freely in φ
      $\exists x(\varphi) \equiv \varphi$ if and only if x does not occur freely in φ
    - **Quantifier Distributivity:**
      Clearly following from the first equivalences.
      $\forall x(\varphi \wedge \psi) \equiv \forall x(\varphi) \wedge \forall x(\psi)$
      $\exists x(\varphi \vee \psi) \equiv \exists x(\varphi) \vee \exists x(\psi)$

      Following from the vacuous quantifications.
      $\forall x(\varphi \vee \psi) \equiv \forall x(\varphi) \vee \psi$ if and only if x does not occur freely in ψ
      $\exists x(\varphi \wedge \psi) \equiv \exists x(\varphi) \wedge \psi$ if and only if x does not occur freely in ψ

- **Important Non-Equivalences:**

  $\forall x \exists y(\varphi) \not\equiv \exists y \forall x(\varphi)$
  $\forall x(\varphi \vee \psi) \not\equiv \forall x(\varphi) \vee \forall x(\psi)$
  $\exists x(\varphi \wedge \psi) \not\equiv \exists x(\varphi) \wedge \exists x(\psi)$

- Notable valid formula: for a relation R, $\exists x \forall y\big(R(x,y)\big) \Rightarrow \forall y \exists x\big(R(x,y)\big)$, however, the converse is not valid. Assume that the hypothesis is true, i.e. there is an x' such that $\forall y\big(R(x',y)\big)$, then for $\forall y \exists x R(x,y)$ is clearly true for $x = x'$. This also holds if y can be considered as any $(n-1)$-tuple for an n-ary relation.

The Humpty Dumpty Theory of Language:

- '"When I use a word," Humpty Dumpty said, in rather a scornful tone, "it means just what I choose it to mean – neither more nor less."' – Lewis Carroll, 'Through the Looking-Glass' (1871).
- This extract, from a notable logician, clearly demonstrates that while a sentence may be constructed following the valid rules of a language, its meaning varies depending on the interpretation (semantics) of the syntax used.
- A further example, in the form of a 'nonsense' poem, "Jabberwocky", also by Lewis Carroll and from the same work of literature:

  'Twas brillig, and the slithy toves
  Did gyre and gimble in the wabe;
  All mimsy were the borogoves,
  And the mome raths outgrabe.

  "Beware the Jabberwock, my son!
  The jaws that bite, the claws that catch!
  Beware the Jubjub bird, and shun
  The frumious Bandersnatch!"
- The Jabberwocky makes use of many words with no meaning, no semantic interpretation, in English.

First-Order Semantics:

- **Interpreting a First-Order Language:**
  - An **interpretation** of a first-order language must define a domain of discourse, which will specify the set of values which quantifiers range over, and a semantic meaning to each predicate, function, or constant symbol in the language.
  - The result of this interpretation is that every term is given an object, or in the case of non-nullary functions a set of mappings from objects to objects, each predicate is given a property or relation between objects, and each sentence is given a truth value.
- **First-Order Structures:**
  - If a **structure** (or '**model'**) M is used to interpret the language $\Sigma = (\mathcal{F}, \mathcal{P})$, then M is the triple such that $M = (D, \mathcal{F}^M, \mathcal{P}^M)$.
  - M is an interpretation for the FO language Σ.
  - It is defined a domain of discourse D which is a nonempty set (of object/things).

    - The set of all function symbols $\mathcal{F}$ in Σ is interpreted as $\mathcal{F}^M$. For every $f \in \mathcal{F}$ with arity k, there is $f^M \in \mathcal{F}^M$, such that $f^M: D^k \to D$, note $k = 0, f^M: D$.
    - The set of all predicate symbols $\mathcal{P}$ in Σ is interpreted as $\mathcal{P}^M$. For every $P \in \mathcal{P}$ with arity k, there is $P^M \in \mathcal{P}^M$, such that $P^M \subseteq D^k$ (a relation on D).
- **Interpretating Formulas:**
    - For a formula in the language $\Sigma = (\mathcal{F}, \mathcal{P})$, interpreting under a model $M = (D, \mathcal{F}^M, \mathcal{P}^M)$.
    - **Valuation** – a first-order valuation is a function ρ for M which maps the variable symbols to members of D, i.e. $\rho: A \to D$, for all variable symbols $x \in A, \rho: x \mapsto \rho(x), d \in D, \rho(x) = d$.
    - Valuations are then extended to all terms formed with Σ and act as an interpretation for $f \in \mathcal{F}, \rho\big(f(t_1, \dots, t_k)\big) \equiv f^M\big(\rho(t_1), \dots, \rho(t_k)\big)$.
    - **Updating Valuations:**
        - For any totally-defined valuation $\rho: A \to D$, for any variable symbol $x \in A$, for a new valuation which maps all A to the same D as ρ, except that for a $d \in D$ when $\rho(x) \neq d$, $\rho[x \mapsto d]$ represents the 'updated' or new valuation such that:
$$\rho[x \mapsto d](v) = \begin{cases} d & if\ v = x \\ \rho(v) & otherwise \end{cases}$$
    - Given a FO language, a model M to interpret the language, and a totally defined valuation for every variable symbol for M, every well-formed formula in the language can be given a truth value.
- Valuations of well-formed formulas under the valuation ρ for a model M (as satisfaction relations**):
$$M \vDash_\rho P(t_1, \dots, t_n) \equiv \big(\rho(t_1), \dots, \rho(t_n)\big) \in P^M$$
$$M \nvDash_\rho P(t_1, \dots, t_n) \equiv \big(\rho(t_1), \dots, \rho(t_n)\big) \notin P^M$$
$$M \vDash_\rho \neg\varphi \equiv M \nvDash_\rho \varphi \equiv not\ M \vDash_\rho \varphi$$
$$M \vDash_\rho (\varphi \land \psi) \equiv \big(M \vDash_\rho \varphi\big)\ and\ \big(M \vDash_\rho \psi\big)$$
$$M \vDash_\rho (\varphi \lor \psi) \equiv \big(M \vDash_\rho \varphi\big)\ or\ \big(M \vDash_\rho \psi\big)$$
$$M \vDash_\rho (\varphi \Rightarrow \psi) \equiv \big(M \nvDash_\rho \varphi\big)\ or\ \big(M \vDash_\rho \psi\big)$$
$$M \vDash_\rho (\varphi \Leftrightarrow \psi) \equiv \Big(\big(M \vDash_\rho \varphi\big)\ and\ \big(M \vDash_\rho \psi\big)\Big)\ or\ \Big(\big(M \nvDash_\rho \varphi\big)\ and\ \big(M \nvDash_\rho \psi\big)\Big)$$
$$M \vDash_\rho \forall x(\varphi) \equiv M \vDash_{\rho[x \mapsto d]} \varphi, for\ all\ d \in D$$
$$M \vDash_\rho \exists x(\varphi) \equiv M \vDash_{\rho[x \mapsto d]} \varphi, for\ some\ d \in D$$
- **Satisfiability** (in a specified model) – a formula φ is satisfiable in a model M if and only if there exists some valuation ρ for M under which φ is true, this can be denoted $M \vDash_\rho \varphi$. If the formula is not satisfied by ρ, it can be denoted $M \nvDash_\rho \varphi$.
- **Validity** (in a specified model) – a formula φ is valid in a model M if and only if it is satisfied by all possible valuations ρ for M, which can be denoted $M \vDash \varphi$, and if unsatisfiable in M is denoted $M \nvDash \varphi$.
- **Satisfiability** (in general) – a formula φ is satisfiable if and only if it is satisfiable in some model M, i.e. $M \vDash_\rho \varphi$ for some M and some ρ for M.
- **Validity** (in all structures) – a formula φ is valid if and only if it is valid in all structures. This is denoted $\vDash \varphi$.
- Two first-order formulas φ and ψ are **logically equivalent** (and hence interchangeable), $\varphi \equiv \psi$, if and only if for any structure M and any valuation ρ for M, $M \vDash_\rho \varphi \Leftrightarrow M \vDash_\rho \psi$.
- Consequences for satisfiability and validity:

  - **If** a formula is **valid**, **then** it is **valid** in any **arbitrary structure** ($\vDash \varphi \Rightarrow M \vDash \varphi$ for any M). **If** a formula is **valid in a structure**, **then** it is **satisfiable in that structure**. **If** a formula is **satisfiable in some structure**, **then** it is **satisfiable**.
  - Note, that for a given formula φ and a structure M, **if** φ is a **sentence** (has no freely occurring variables) **and** is **satisfiable** in M, **then** φ is **valid** in M. This is a result of the updating of any valuation occurs for all bound variables, so the valuation used when 'removing' the quantifiers is the same for every variable symbol in φ.
  - Also note, to **interpret** the **truth value** of a formula in a structure under a valuation, **all variable** symbols in the formula **must be either bound or interpreted** by the valuation.
- Same as in propositional logic:
  - A formula φ is valid in a structure M if and only if $\neg\varphi$ is unsatisfiable.
  - A formula φ is satisfiable in a structure M if and only if $\neg\varphi$ is invalid in M.
  - Note: contradiction, tautology, and contingency are defined the same, except that they are specified under a specific model or not.

Prenex Normal Form:

- **Prefix** – a string of quantifiers over variables.
- **Matrix** – a first-order formula which contains no quantifiers.
- A formula ϕ is in prenex normal form if and only if it has the form of a string with a prefix followed by a matrix.
- That is, for $Q_k, k \geq 0$ where every Q_k is either $\forall$ or $\exists$, and a quantifier-free formula φ, then $Q_0 x_0 Q_1 x_1 \ldots Q_k x_k . \varphi$ is in prenex form.
- Every first-order formula has an equivalent prenex normal form.
- The prenex normal form of any formula can be found by applying logical equivalences for quantifiers:
  - Represent all connectives by equivalent forms using only negations, conjunctions, and disjunctions.
  - Use De Morgan's Laws to move negations into the innermost quantifiers.
  - Rename reused variable symbols.
  - Use the distributive equivalences for quantifiers.
- Note that the order of the quantifiers will depend on the order in which they occur in the scope of the formula being converted, and any reused variables (ones which are bound multiple times in different scopes, including being both bound and free in occurrence within a formula) from scopes should be substituted to accommodate for this.

Restricted/Bounded Quantifiers:

- The domains of quantifiers can be restricted, this is done by applying a precondition(s) to the variable being quantified, over the universal domain, which it must satisfy.
- **Quantifiers with Set Notation:**
  - Restricting the domain of quantifiers using set membership notation.
  - Under the model M with a domain of discourse D, when the domain a variable is quantified over is some subset of D.
  - For some set $S \subseteq D$ and some formula φ, restricing the quantifier's domain to S:
    $\forall x \in S(\varphi) \equiv \forall x(x \in S \Rightarrow \varphi)$
    $\exists x \in S(\varphi) \equiv \exists x(x \in S \land \varphi)$

- While it has been shown here with set memberships as the most typical method of restricting domains, any allowed predicates on the quantified variable(s) can be used.
- Other common predicates include equalities, or strict inequalities (for ordered domains).
- It is useful to consider how the restrictions apply to the form of quantifiers as chains of conjunctions or disjunctions for universal and existential respectively.

Non-'Elementary' Quantification:

- It is clear how to achieve the quantifications for 'all', 'at least one', or 'no' members of a domain, however, it can also be useful to express a distinct number of elements, 'at least n', or an exact number, 'exactly n'. The below methods require an equality (and hence negated equality) of elements to be defined.
- 'there exists at least n' can be expressed in the way below:
$$\exists x_1 \exists x_2 \ldots \exists x_n \left(\bigwedge_{i=1}^{n} \bigwedge_{j=i+1}^{n} \neg\left(x_i = x_j\right) \right)$$
- The 'there is exactly one' (existential uniqueness) quantifier is commonly represented by $\exists!$, which is able to be considered in two equivalent ways, the latter being preferred.
$$\exists!\, xPx \equiv \exists x\big(Px \wedge \neg\exists y(Py \wedge y \neq x)\big) \equiv \exists x\big(Px \wedge \forall y(Py \Rightarrow y = x)\big)$$
$$equivalently$$
$$\exists!\, xPx \equiv \exists x \forall y(Py \Leftrightarrow y = x)$$
- And further, 'there exist exactly two':
$$\exists x \exists y\big(y \neq x \wedge \forall z(Pz \Leftrightarrow z = x \vee z = y)\big)$$
the left side of the conjunction here begins by stating that there exists at least 2 distinct elements ($x \neq y$), and the right side of the conjunction states that there are at most 2 distinct elements (for every z, if Pz then it must be that $z = x$ or $z = y$, and also the converse). The conjunction of at least n and at most n logically becomes exactly n.
- This carries on for 'there exists exactly n', where n is a positive integer.