Propositional Logic

Syntax

Definition

- The set of *formulas* of propositional logic is given by the abstract syntax:

$$\mathbf{Form} \quad \ni \quad A, B, C \quad ::= \quad P \mid \bot \mid (\neg A) \mid (A \wedge B) \mid (A \vee B) \mid (A \rightarrow B)$$

  where P ranges over a countable set **Prop** , whose elements are called *propositional symbols* or *propositional variables*. (We also let Q, R range over **Prop** .)
- Formulas of the form $\bot$ or P are called *atomic*.
- $\top$ abbreviates $(\neg\bot)$ and $(A \leftrightarrow B)$ abbreviates $((A \rightarrow B) \wedge (B \rightarrow A))$.

Remark

- *Conventions to omit parentheses are:*
  - *outermost parentheses can be dropped;*
  - *the order of precedence (from the highest to the lowest) of connectives is:* $\neg$, $\wedge$, $\vee$ *and* $\rightarrow$*;*
  - *binary connectives are right-associative.*
- *There are recursion and induction principles (e.g. structural ones) for* **Form** *.*

Definition

A is a *subformula* of B when A "occurs in" B.

Semantics

Definition

- $\mathbf{T}$ (*true*) and $\mathbf{F}$ (*false*) form the set of *truth values*.
- A *valuation* is a function $\rho : \mathbf{Prop} \ -> \{\mathbf{F}, \mathbf{T}\}$ that assigns truth values to propositional symbols.
- Given a valuation ρ, the *interpretation function* $[\![\cdot]\!]_\rho : \mathbf{Form} \ -> \{\mathbf{F}, \mathbf{T}\}$ is defined recursively as follows:

$$
\begin{aligned}
[\![\bot]\!]_\rho &= \mathbf{F} \\
[\![P]\!]_\rho &= \mathbf{T} \quad \text{iff} \quad \rho(P) = \mathbf{T} \\
[\![\neg A]\!]_\rho &= \mathbf{T} \quad \text{iff} \quad [\![A]\!]_\rho = \mathbf{F} \\
[\![A \wedge B]\!]_\rho &= \mathbf{T} \quad \text{iff} \quad [\![A]\!]_\rho = \mathbf{T} \text{ and } [\![B]\!]_\rho = \mathbf{T} \\
[\![A \vee B]\!]_\rho &= \mathbf{T} \quad \text{iff} \quad [\![A]\!]_\rho = \mathbf{T} \text{ or } [\![B]\!]_\rho = \mathbf{T} \\
[\![A \rightarrow B]\!]_\rho &= \mathbf{T} \quad \text{iff} \quad [\![A]\!]_\rho = \mathbf{F} \text{ or } [\![B]\!]_\rho = \mathbf{T}
\end{aligned}
$$

Semantics

Definition

A *propositional model* $\mathcal{M}$ is a set of proposition symbols, i.e. $\mathcal{M} \subseteq \mathbf{Prop}$. The *validity relation* $\models\; \subseteq \mathcal{P}(\mathbf{Prop}) \times \mathbf{Form}$ is defined inductively by:

$$\begin{array}{lll}
\mathcal{M} \models P & \text{iff} & P \in \mathcal{M} \\
\mathcal{M} \models \neg A & \text{iff} & \mathcal{M} \not\models A \\
\mathcal{M} \models A \wedge B & \text{iff} & \mathcal{M} \models A \text{ and } \mathcal{M} \models B \\
\mathcal{M} \models A \vee B & \text{iff} & \mathcal{M} \models A \text{ or } \mathcal{M} \models B \\
\mathcal{M} \models A \rightarrow B & \text{iff} & \mathcal{M} \not\models A \text{ or } \mathcal{M} \models B
\end{array}$$

Remark

The two semantics are equivalent. In fact, valuations are in bijection with propositional models. In particular, each valuation ρ determines a model $\mathcal{M}_\rho = \{P \in \mathbf{Prop} \mid \rho(P) = \mathbf{T}\}$ s.t.

$$\mathcal{M}_\rho \models A \quad \textit{iff} \quad [\![A]\!]_\rho = \mathbf{T},$$

which can be proved by induction on A. Henceforth, we adopt the latter semantics.

Definition

- A formula A is *valid in a model* $\mathcal{M}$ (or $\mathcal{M}$ *satisfies* A), iff $\mathcal{M} \models A$. When $\mathcal{M} \not\models A$, A is said *refuted* by $\mathcal{M}$.
- A formula A is *satisfiable* iff there exists some model $\mathcal{M}$ such that $\mathcal{M} \models A$. It is *refutable* iff some model refutes A.
- A formula A is *valid* (also called a *tautology*) iff every model satisfies A. A formula A is a *contradiction* iff every model refutes A.

Semantics

Proposition

Let $\mathcal{M}$ and $\mathcal{M}'$ be two propositional models and let A be a formula. If for any propositional symbol P occuring in A, $\mathcal{M} \models P$ iff $\mathcal{M}' \models P$, then $\mathcal{M} \models A$ iff $\mathcal{M}' \models A$.

Proof.

By induction on A. □

Remark

The previous proposition justifies that the truth table method suffices for deciding weather or not a formula is valid, which in turn guarantees that the validity problem of PL is decidable

Definition

A is *logically equivalent* to B, (denoted by $A \equiv B$) iff A and B are valid exactly in the same models.

Some logical equivalences

$$
\begin{array}{ccl}
\multicolumn{2}{c}{\neg\neg A \equiv A} & (\textit{double negation}) \\
\neg(A \wedge B) \equiv \neg A \vee \neg B & \neg(A \vee B) \equiv \neg A \wedge \neg B & (\textit{De Morgan's laws}) \\
A \rightarrow B \equiv \neg A \vee B & \neg A \equiv A \rightarrow \bot & (\textit{interdefinability}) \\
A \wedge (B \vee C) \equiv (A \wedge B) \vee (A \wedge C) & A \vee (B \wedge C) \equiv (A \vee B) \wedge (A \vee C) & (\textit{distributivity})
\end{array}
$$

Semantics

Remark

- *$\equiv$ is an equivalence relation on* **Form** *.*
- *Given $A \equiv B$, the replacement in a formula C of an occurrence of A by B produces a formula equivalent to C.*
- *The two previous results allow for equational reasoning in proving logical equivalence.*

Definition

Given a propositional formula A, we say that it is in:

- *Conjunctive normal form* (CNF), if it is a conjunction of disjunctions of *literals* (atomic formulas or negated atomic formulas), i.e. $A = \bigwedge_i \bigvee_j l_{ij}$, for literals l_{ij};
- *Disjunctive normal form* (DNF), if it is a disjunction of conjunctions of literals, i.e. $A = \bigvee_i \bigwedge_j l_{ij}$, for literals l_{ij}.

Note that in some treatments, $\perp$ is not allowed in literals.

Proposition

Any formula is equivalent to a CNF and to a DNF.

Proof.

The wanted CNF and DNF can be obtained by rewriting of the given formula, using the logical equivalences listed before. $\square$

Semantics

Notation

We let $\Gamma, \Gamma', \ldots$ *range over sets of formulas and use* Γ, A *to abbreviate* $\Gamma \cup \{A\}$.

Definition

Let Γ be a set of formulas.

- Γ is *valid in a model* $\mathcal{M}$ (or $\mathcal{M}$ *satisfies* Γ), iff $\mathcal{M} \models A$ for every formula $A \in \Gamma$. We denote this by $\mathcal{M} \models \Gamma$.
- Γ is *satisfiable* iff there exists a model $\mathcal{M}$ such that $\mathcal{M} \models \Gamma$, and it is *refutable* iff there exists a model $\mathcal{M}$ such that $\mathcal{M} \not\models \Gamma$.
- Γ is *valid*, denoted by $\models \Gamma$, iff $\mathcal{M} \models \Gamma$ for every model $\mathcal{M}$, and it is *unsatisfiable* iff it is not satisfiable.

Definition

Let A be a formula and Γ a set of formulas. If every model that validates Γ also validates A, we say that Γ *entails* A (or A is a *logical consequence* of Γ).
We denote this by $\Gamma \models A$ and call $\models \subseteq \mathcal{P}(\textbf{Form}) \times \textbf{Form}$ the *semantic entailment* or *logical consequence* relation.

Semantics

Proposition

- *A is valid iff $\models A$, where $\models A$ abbreviates $\emptyset \models A$.*
- *A is a contradiction iff $A \models \bot$.*
- *$A \equiv B$ iff $A \models B$ and $B \models A$. (or equivalently, $A \leftrightarrow B$ is valid).*

Proposition

The semantic entailment relation satisfies the following properties (of an abstract consequence relation):

- *For all $A \in \Gamma$, $\Gamma \models A$.* *(inclusion)*
- *If $\Gamma \models A$, then $\Gamma, B \models A$.* *(monotonicity)*
- *If $\Gamma \models A$ and $\Gamma, A \models B$, then $\Gamma \models B$.* *(cut)*

Proposition

Further properties of semantic entailment are:

- $\Gamma \models A \wedge B$ *iff* $\Gamma \models A$ and $\Gamma \models B$
- $\Gamma \models A \vee B$ *iff* $\Gamma \models A$ or $\Gamma \models B$
- $\Gamma \models A \rightarrow B$ *iff* $\Gamma, A \models B$
- $\Gamma \models \neg A$ *iff* $\Gamma, A \models \bot$
- $\Gamma \models A$ *iff* $\Gamma, \neg A \models \bot$

Proof system

The natural deduction system $\mathcal{N}_{\mathrm{PL}}$

- The proof system we will consider is a "natural deduction in sequent style" (not to confuse with a "sequent calculus"), which we name $\mathcal{N}_{\mathrm{PL}}$.
- The "judgments" (or "assertions") of $\mathcal{N}_{\mathrm{PL}}$ are sequents $\Gamma \vdash A$, where Γ is a set of formulas (a.k.a. *context* or LHS) and A a formula (a.k.a. *conclusion* or RHS), informally meaning that "A can be proved from the assumptions in Γ".
- Natural deduction systems typically have "introduction" and "elimination" rules for each connective. The set of rules of $\mathcal{N}_{\mathrm{PL}}$ is below.

Rules of $\mathcal{N}_{\mathrm{PL}}$

$$(\mathrm{Ax})\ \frac{}{\Gamma, A \vdash A} \qquad\qquad (\mathrm{RAA})\ \frac{\Gamma, \neg A \vdash \bot}{\Gamma \vdash A}$$

Introduction Rules:

$$(\mathrm{I}_\wedge)\ \frac{\Gamma \vdash A \qquad \Gamma \vdash B}{\Gamma \vdash A \wedge B} \qquad\qquad (\mathrm{I}_{\vee i})\ \frac{\Gamma \vdash A_i}{\Gamma \vdash A_1 \vee A_2}\ i \in \{1,2\}$$

$$(\mathrm{I}_\rightarrow)\ \frac{\Gamma, A \vdash B}{\Gamma \vdash A \rightarrow B} \qquad\qquad (\mathrm{I}_\neg)\ \frac{\Gamma, A \vdash \bot}{\Gamma \vdash \neg A}$$

Elimination Rules:

$$(\mathrm{E}_{\wedge i})\ \frac{\Gamma \vdash A_1 \wedge A_2}{\Gamma \vdash A_i}\ i \in \{1,2\} \qquad\qquad (\mathrm{E}_\vee)\ \frac{\Gamma \vdash A \vee B \qquad \Gamma, A \vdash C \qquad \Gamma, B \vdash C}{\Gamma \vdash C}$$

$$(\mathrm{E}_\rightarrow)\ \frac{\Gamma \vdash A \qquad \Gamma \vdash A \rightarrow B}{\Gamma \vdash B} \qquad\qquad (\mathrm{E}_\neg)\ \frac{\Gamma \vdash A \qquad \Gamma \vdash \neg A}{\Gamma \vdash B}$$

Proof system

Definition

- A *derivation* of a sequent $\Gamma \vdash A$ is a tree of sequents, built up from *instances of the inference rules* of $\mathcal{N}_{\mathrm{PL}}$, having as root $\Gamma \vdash A$ and as leaves instances of (Ax) . (The set of $\mathcal{N}_{\mathrm{PL}}$-derivations can formally be given as an inductive definition and has associated recursion and inductive principles.)
- Derivations induce a binary relation $\vdash \in \mathcal{P}(\textbf{Form}\ ) \times \textbf{Form}$, called the *derivability/deduction relation*:
  - $(\Gamma, A) \in \vdash$ iff there is a derivation of the sequent $\Gamma \vdash A$ in $\mathcal{N}_{\mathrm{PL}}$;
  - typically we overload notation and abbreviate $(\Gamma, A) \in \vdash$ by $\Gamma \vdash A$, reading "$\Gamma \vdash A$ is derivable", or "A can be derived (or deduced) from Γ", or "Γ infers A";
- A formula that can be derived from the empty context is called a *theorem*.

Definition

An inference rule is *admissible* in $\mathcal{N}_{\mathrm{PL}}$ if every sequent that can be derived making use of that rule can also be derived without it.

Proof system

Proposition

The following rules are admissible in $\mathcal{N}_{\mathsf{PL}}$:

$$\textit{Weakening}\ \frac{\Gamma \vdash A}{\Gamma, B \vdash A} \qquad \textit{Cut}\ \frac{\Gamma \vdash A \qquad \Gamma, A \vdash B}{\Gamma \vdash B} \qquad (\bot)\ \frac{\Gamma \vdash \bot}{\Gamma \vdash A}$$

Proof.

- Admissibility of weakening is proved by induction on the premise's derivation.
- Cut is actually a *derivable rule* in $\mathcal{N}_{\mathsf{PL}}$, i.e. can be obtained through a combination of $\mathcal{N}_{\mathsf{PL}}$ rules.
- Admissibility of $(\bot)$ follows by combining weakening and *RAA*.

□

Definition

Γ is said *inconsistent* if $\Gamma \vdash \bot$ and otherwise is said *consistent*.

Proposition

If Γ is consistent, then either $\Gamma \cup \{A\}$ or $\Gamma \cup \{\neg A\}$ is consistent (but not both).

Proof.

If not, one could build a derivation of $\Gamma \vdash \bot$ (how?), and Γ would be inconsistent. □

Proof system

Remark

Traditional presentations of natural deduction take formulas as judgements and not sequents. In these presentations:

- *derivations are trees of formulas, whose leaves can be either "open" or "closed";*
- *open leaves correspond to the assumptions upon which the conclusion formula (the root of the tree) depends;*
- *some rules allow for the closing of leaves (thus making the conclusion formula not depend on those assumptions).*

For example, introduction and elimination rules for implication look like:

$$(E_{\rightarrow}) \quad \frac{A \rightarrow B \qquad A}{B} \qquad\qquad (I_{\rightarrow}) \quad \frac{\begin{array}{c}[A] \\ \vdots \\ B\end{array}}{A \rightarrow B}$$

In rule $(I_{\rightarrow})$, any number of occurrences of A as a leaf may be closed (signalled by the use of square brackets).

Adequacy of the proof system

Theorem (Soundness)

If $\Gamma \vdash A$*, then* $\Gamma \models A$*.*

Proof.

By induction on the derivation of $\Gamma \vdash A$. Some of the cases are illustrated:

- If the last step is

$$(\mathrm{Ax}) \quad \frac{}{\Gamma', A \vdash A}$$

  We need to prove $\Gamma', A \models A$, which holds by the inclusion property of semantic entailment.

- If the last step is

$$(\mathrm{I}_{\rightarrow}) \quad \frac{\Gamma, B \vdash C}{\Gamma \vdash B \rightarrow C}$$

  By IH, we have $\Gamma, B \models C$, which is equivalent to $\Gamma \models B \rightarrow C$, by one of the properties of semantic entailment.

- If the last step is

$$(\mathrm{E}_{\rightarrow}) \quad \frac{\Gamma \vdash B \qquad \Gamma \vdash B \rightarrow A}{\Gamma \vdash A}$$

  By IH, we have both $\Gamma \models B$ and $\Gamma \models B \rightarrow A$. From these, we can easily get $\Gamma \models A$.

$\square$

Adequacy of the proof system

Definition

Γ is *maximally consistent* iff it is consistent and furthermore, given any formula A, either A or $\neg A$ belongs to Γ (but not both can belong).

Proposition

Maximally consistent sets are closed for derivability, i.e. given a maximally consistent set Γ and given a formula A, $\Gamma \vdash A$ implies $A \in \Gamma$.

Lemma

If Γ is consistent, then there exists $\Gamma' \supseteq \Gamma$ s.t. Γ' is maximally consistent.

Proof.

Let $\Gamma_0 = \Gamma$ and consider an enumeration $A_1, A_2, \ldots$ of the set of formulas **Form** . For each of these formulas, define Γ_i to be $\Gamma_{i-1} \cup \{A_i\}$ if this is consistent, or $\Gamma_{i-1} \cup \{\neg A_i\}$ otherwise. (Note that one of these sets is consistent.) Then, we take $\Gamma' = \bigcup_i \Gamma_i$. Clearly, by construction, $\Gamma' \supseteq \Gamma$ and for each A_i either $A_i \in \Gamma'$ or $\neg A_i \in \Gamma'$. Also, Γ' is consistent (otherwise some Γ_i would be inconsistent). $\square$

Adequacy of the proof system

Proposition

Γ is consistent iff Γ is satisfiable.

Proof.

The "if statement" follows from the soundness theorem. Let us proof the converse.

Let Γ' be a maximally consistent extension of Γ (guaranteed to exist by the previous lemma) and define $\mathcal{M}$ as the set of proposition symbols that belong to Γ'.

Claim: $\mathcal{M} \models A \qquad \text{iff} \qquad A \in \Gamma'$.

As $\Gamma' \supseteq \Gamma$, $\mathcal{M}$ is a model of Γ, hence Γ is satisfiable.

The claim is proved by induction on A. Two cases are illustrated.

Case $A = P$. The claim is immediate by construction of $\mathcal{M}$.

Case $A = B \rightarrow C$. By IH and the fact that Γ' is maximally consistent, $\mathcal{M} \models B \rightarrow C$ is equivalent to $\neg B \in \Gamma'$ or $C \in \Gamma'$, which in turn is equivalent to $B \rightarrow C \in \Gamma'$. The latter equivalence is proved with the help of the fact that Γ', being maximally consistent, is closed for derivability. $\square$

Adequacy of the proof system

Theorem (Completeness)

If $\Gamma \models A$ then $\Gamma \vdash A$.

Proof.

Suppose $\Gamma \vdash A$ does not hold. Then, $\Gamma \cup \{\neg A\}$ is consistent (why?) and thus, by the above proposition, $\Gamma \cup \{\neg A\}$ would have a model, contradicting $\Gamma \models A$. $\square$

Corolary (Compactness)

A (possibly infinite) set of formulas Γ is satisfiable if and only if every finite subset of Γ is satisfiable.

Proof.

The key observation is that, in $\mathcal{N}_{\mathrm{PL}}$, if $\Gamma \vdash A$, then there exists a finite $\Gamma' \subseteq \Gamma$ s.t. $\Gamma' \vdash A$. $\square$

First-Order Logic

Syntax

Definition

The *alphabet of a first-order language* is organised into the following categories.

- *Logical connectives*: $\perp$, $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\forall$ and $\exists$.
- *Auxiliary symbols*: ".", ",", "(" and ")".
- *Variables*: we assume a countable infinite set $\mathcal{X}$ of variables, ranged over by $x, y, z, \ldots$.
- *Constant symbols*: we assume a countable set $\mathcal{C}$ of constant symbols, ranged over by $a, b, c, \ldots$.
- *Function symbols*: we assume a countable set $\mathcal{F}$ of function symbols, ranged over by $f, g, h, \ldots$. Each function symbol f has a fixed arity $\mathrm{ar}(f)$, which is a positive integer.
- *Predicate symbols*: we assume a countable set $\mathcal{P}$ of predicate symbols, ranged over by $P, Q, R, \ldots$. Each predicate symbol P has a fixed arity $\mathrm{ar}(P)$, which is a non-negative integer. (Predicate symbols with arity 0 play the role of propositions.)

The union of the non-logical symbols of the language is called the *vocabulary* and is denoted by $\mathcal{V}$, i.e. $\mathcal{V} = \mathcal{C} \cup \mathcal{F} \cup \mathcal{P}$.

Notation

Throughout, and when not otherwise said, we assume a vocabulary $\mathcal{V} = \mathcal{C} \cup \mathcal{F} \cup \mathcal{P}$.

Syntax

Definition

The set of *terms* of a first-order language over a vocabulary $\mathcal{V}$ is given by:

$$\mathbf{Term}_{\mathcal{V}} \ni t, u ::= x \mid c \mid f(t_1, \ldots, t_{\mathsf{ar}(f)})$$

The set of *variables occurring* in t is denoted by $\mathsf{Vars}(t)$.

Definition

The set of *formulas* of a first-order language over a vocabulary $\mathcal{V}$ is given by:

$$\begin{aligned}\mathbf{Form}_{\mathcal{V}} \ni \phi, \psi, \theta \quad ::= \quad & P(t_1, \ldots, t_{\mathsf{ar}(P)}) \mid \bot \mid (\neg\phi) \mid (\phi \wedge \psi) \mid (\phi \vee \psi) \\ & \mid (\phi \rightarrow \psi) \mid (\forall x \,.\, \phi) \mid (\exists x \,.\, \phi)\end{aligned}$$

An *atomic formula* has the form $\bot$ or $P(t_1, \ldots, t_{\mathsf{ar}(P)})$.

Remark

- *We assume the conventions of propositional logic to omit parentheses, and additionally assume that quantifiers have the lowest precedence.*
- *Nested quantifications such as* $\forall x \,.\, \forall y \,.\, \phi$ *are abbreviated to* $\forall x, y \,.\, \phi$.
- *There are recursion and induction principles (e.g. structural ones) for* $\mathbf{Term}_{\mathcal{V}}$ *and* $\mathbf{Form}_{\mathcal{V}}$.

Syntax

Definition

- A formula ψ that occurs in a formula ϕ is called a *subformula* of ϕ.
- In a quantified formula $\forall x \,.\, \phi$ or $\exists x \,.\, \phi$, x is the *quantified variable* and ϕ is the *scope* of the quantification.
- Occurrences of the quantified variable within the respective scope are said to be *bound*. Variable occurrences that are not bound are said to be *free*.
- The set of *free variables* (resp.*bound variables*) of a formula θ, is denoted $\mathsf{FV}(\theta)$ (resp. $\mathsf{BV}(\theta)$).

Definition

- A *sentence* (or *closed* formula) is a formula without free variables.
- If $\mathsf{FV}(\phi) = \{x_1, \ldots, x_n\}$, the *universal closure* of ϕ is the formula $\forall x_1, \ldots, x_n \,.\, \phi$ and the *existential closure* of ϕ is the formula $\exists x_1, \ldots, x_n \,.\, \phi$.

Definition

- A *substitution* is a mapping $\sigma : \mathcal{X} -> \textbf{Term}_{\mathcal{V}}$ s.t. the set $\mathsf{dom}(\sigma) = \{x \in \mathcal{X} \mid \sigma(x) \neq x\}$, called the *substitution domain*, is finite.
- The notation $[t_1/x_1, \ldots, t_n/x_n]$ (for distinct x_i's) denotes the substitution whose domain is contained in $\{x_1, \ldots, x_n\}$ and maps each x_i to t_i.

Syntax

Definition

The *application of a substitution* σ *to a term* t is denoted by $t\,\sigma$ and is defined recursively by:

$$\begin{array}{rcl} x\,\sigma & = & \sigma(x) \\ c\,\sigma & = & c \\ f(t_1,\ldots,t_{\mathrm{ar}(f)})\,\sigma & = & f(t_1\,\sigma,\ldots,t_{\mathrm{ar}(f)}\,\sigma) \end{array}$$

Remark

The result of

$$t\,[t_1/x_1,\ldots,t_n/x_n]$$

corresponds to the simultaneous substitution of $t_1,\ldots,t_n$ *for* $x_1,\ldots,x_n$ *in* t*. This differs from the application of the corresponding singleton substitutions in sequence,*

$$((t\ [t_1/x_1])\ldots)\,[t_n/x_n]\,.$$

Notation

Given a function $f : X \longrightarrow Y$*,* $x \in X$ *and* $y \in Y$*, the notation* $f[x \mapsto y]$ *stands for the function defined as* f *except possibly for* x*, to which* y *is assigned, called the patching of* f *in* x *to* y*.*

Syntax

Definition

The *application of a substitution* σ *to a formula* ϕ, written $\phi\,\sigma$, is given recursively by:

$$
\begin{aligned}
\bot\,\sigma &= \bot \\
P(t_1, \ldots, t_{\mathrm{ar}(P)})\,\sigma &= P(t_1\,\sigma, \ldots, t_{\mathrm{ar}(P)}\,\sigma) \\
(\neg\phi)\,\sigma &= \neg(\phi\,\sigma) \\
(\phi \odot \psi)\,\sigma &= (\phi\,\sigma) \odot (\psi\,\sigma) \\
(Qx\,.\,\phi)\,\sigma &= Qx\,.\,(\phi(\sigma[x \mapsto x]))
\end{aligned}
$$

where $\odot \in \{\wedge, \vee \rightarrow\}$ and $Q \in \{\forall, \exists\}$.

Remark

- *Only free occurrences of variables can change when a substitution is applied to a formula.*
- *Unrestricted application of substitutions to formulas can cause capturing of variables as in:* $(\forall x.\, P(x, y))\,[g(x)/y] = \forall x.\, P(x, g(x))$
- *"Safe substitution" (which we assume throughout) is achieved by imposing that a substitution when applied to a formula should be free for it.*

Definition

- A *term* t *is free for* x *in* θ iff x has no free occurrences in the scope of a quantifier Qy ($y \neq x$) s.t. $y \in \mathrm{Vars}(t)$.
- A *substitution* σ *is free for* θ iff $\sigma(x)$ is free for x in θ, for all $x \in \mathrm{dom}(\sigma)$.

Semantics

Definition

Given a vocabulary $\mathcal{V}$, a $\mathcal{V}$*-structure* is a pair $\mathcal{M} = (D, I)$ where D is a nonempty set, called the *interpretation domain*, and I is called the *interpretation function*, and assigns constants, functions and predicates over D to the symbols of $\mathcal{V}$ as follows:

- for each $c \in \mathcal{C}$, the interpretation of c is a constant $I(c) \in D$;
- for each $f \in \mathcal{F}$, the interpretation of f is a function $I(f) : D^{\mathrm{ar}(f)} \longrightarrow D$;
- for each $P \in \mathcal{P}$, the interpretation of P is a function $I(P) : D^{\mathrm{ar}(P)} \longrightarrow \{\mathbf{F}, \mathbf{T}\}$. In particular, 0-ary predicate symbols are interpreted as truth values.

$\mathcal{V}$-structures are also called *models* for $\mathcal{V}$.

Definition

Let D be the interpretation domain of a structure. An *assignment* for D is a function $\alpha : \mathcal{X} \longrightarrow D$ from the set of variables to the domain D.

Notation

In what follows, we let $\mathcal{M}, \mathcal{M}', \ldots$ range over the structures of an intended vocabulary, and $\alpha, \alpha', \ldots$ range over the assignments for the interpretation domain of an intended structure.

Semantics

Definition

Let $\mathcal{M} = (D, I)$ be a $\mathcal{V}$ -structure and α an assignment for D.

- The *value of a term t w.r.t. $\mathcal{M}$ and α* is an element of D, denoted by $[\![t]\!]_{\mathcal{M},\alpha}$, and recursively given by:

$$\begin{array}{lcl} [\![x]\!]_{\mathcal{M},\alpha} & = & \alpha(x) \\ [\![c]\!]_{\mathcal{M},\alpha} & = & I(c) \\ [\![f(t_1, \ldots, t_{\mathrm{ar}(f)})]\!]_{\mathcal{M},\alpha} & = & I(f)([\![t_1]\!]_{\mathcal{M},\alpha}, \ldots, [\![t_{\mathrm{ar}(f)}]\!]_{\mathcal{M},\alpha}) \end{array}$$

- The *(truth) value of a formula ϕ w.r.t. $\mathcal{M}$ and α*, is denoted by $[\![\phi]\!]_{\mathcal{M},\alpha}$, and recursively given by:

$$\begin{array}{lcl} [\![\bot]\!]_{\mathcal{M},\alpha} = \mathbf{F} & & \\ [\![P(t_1, \ldots, t_{\mathrm{ar}(P)})]\!]_{\mathcal{M},\alpha} & = & I(P)([\![t_1]\!]_{\mathcal{M},\alpha}, \ldots, [\![t_{\mathrm{ar}(P)}]\!]_{\mathcal{M},\alpha}) \\ [\![\neg\phi]\!]_{\mathcal{M},\alpha} = \mathbf{T} & \text{iff} & [\![\phi]\!]_{\mathcal{M},\alpha} = \mathbf{F} \\ [\![\phi \wedge \psi]\!]_{\mathcal{M},\alpha} = \mathbf{T} & \text{iff} & [\![\phi]\!]_{\mathcal{M},\alpha} = \mathbf{T} \text{ and } [\![\psi]\!]_{\mathcal{M},\alpha} = \mathbf{T} \\ [\![\phi \vee \psi]\!]_{\mathcal{M},\alpha} = \mathbf{T} & \text{iff} & [\![\phi]\!]_{\mathcal{M},\alpha} = \mathbf{T} \text{ or } [\![\psi]\!]_{\mathcal{M},\alpha} = \mathbf{T} \\ [\![\phi \rightarrow \psi]\!]_{\mathcal{M},\alpha} = \mathbf{T} & \text{iff} & [\![\phi]\!]_{\mathcal{M},\alpha} = \mathbf{F} \text{ or } [\![\psi]\!]_{\mathcal{M},\alpha} = \mathbf{T} \\ [\![\forall x.\, \phi]\!]_{\mathcal{M},\alpha} = \mathbf{T} & \text{iff} & [\![\phi]\!]_{\mathcal{M},\alpha[x\mapsto a]} = \mathbf{T} \text{ for all } a \in D \\ [\![\exists x.\, \phi]\!]_{\mathcal{M},\alpha} = \mathbf{T} & \text{iff} & [\![\phi]\!]_{\mathcal{M},\alpha[x\mapsto a]} = \mathbf{T} \text{ for some } a \in D \end{array}$$

Semantics

Remark

Universal and existential quantifications are indeed a gain over PL. They can be read (resp.) as generalised conjunction and disjunction (possibly infinite):

$$[\![\forall x.\,\phi]\!]_{\mathcal{M},\alpha} = \bigwedge_{a\in D} [\![\phi]\!]_{\mathcal{M},\alpha[x\mapsto a]} \qquad [\![\exists x.\,\phi]\!]_{\mathcal{M},\alpha} = \bigvee_{a\in D} [\![\phi]\!]_{\mathcal{M},\alpha[x\mapsto a]}$$

Definition

Let $\mathcal{V}$ be a vocabulary and $\mathcal{M}$ a $\mathcal{V}$ -structure.

- $\mathcal{M}$ *satisfies* ϕ *with* α, denoted by $\mathcal{M}, \alpha \models \phi$, iff $[\![\phi]\!]_{\mathcal{M},\alpha} = \mathbf{T}$.
- $\mathcal{M}$ *satisfies* ϕ (or that ϕ *is valid in* $\mathcal{M}$, or $\mathcal{M}$ is a *model* of ϕ), denoted by $\mathcal{M} \models \phi$, iff for every assignment α, $\mathcal{M}, \alpha \models \phi$.
- ϕ is *satisfiable* if exists $\mathcal{M}$ s.t. $\mathcal{M} \models \phi$, and it is *valid*, denoted by $\models \phi$, if $\mathcal{M} \models \phi$ for every $\mathcal{M}$. ϕ is *unsatisfiable* (or a *contradiction*) if it is not satisfiable, and refutable if it is not valid.

Lemma

Let $\mathcal{M}$ be a structure, t and u terms, ϕ a formula, and α, α' assignments.

- *If for all* $x \in \mathrm{Vars}(t)$, $\alpha(x) = \alpha'(x)$, *then* $[\![t]\!]_{\mathcal{M},\alpha} = [\![t]\!]_{\mathcal{M},\alpha'}$
- *If for all* $x \in \mathrm{FV}(\phi)$, $\alpha(x) = \alpha'(x)$, *then* $\mathcal{M}, \alpha \models \phi$ *iff* $\mathcal{M}, \alpha' \models \phi$.
- $[\![t\,[u/x]]\!]_{\mathcal{M},\alpha} = [\![t]\!]_{\mathcal{M},\alpha[x\mapsto [\![u]\!]_{\mathcal{M},\alpha}]}$
- *If t is free for x in ϕ, then* $\mathcal{M}, \alpha \models \phi\,[t/x]$ *iff* $\mathcal{M}, \alpha[x \mapsto [\![\mathcal{M}]\!]_{\alpha,t}] \models \phi$.

Semantics

Proposition (Lifting validity of PL)

*Let $\lceil\cdot\rceil$: **Prop** $->$ **Form**$_{\mathcal{V}}$, be a mapping from the set of proposition symbols to first-order formulas and denote also by $\lceil\cdot\rceil$ its homomorphic extension to all propositional formulas. Then, for all propositional formulas A and B:*

- $\mathcal{M}, \alpha \models \lceil A \rceil$ *iff* $\overline{\mathcal{M}_\alpha} \models_{PL} A$, *where* $\overline{\mathcal{M}_\alpha} = \{P \mid \mathcal{M}, \alpha \models \lceil P \rceil\}$.
- *If* $\models_{PL} A$, *then* $\models_{FOL} \lceil A \rceil$.
- *If* $A \equiv_{PL} B$, *then* $\lceil A \rceil \equiv_{FOL} \lceil B \rceil$.

Some properties of logical equivalence

- The properties of logical equivalence listed for PL hold for FOL.
- The following equivalences hold:

$$\neg\forall x\,.\,\phi \equiv \exists x\,.\,\neg\phi \qquad\qquad \neg\exists x\,.\,\phi \equiv \forall x\,.\,\neg\phi$$

$$\forall x\,.\,\phi \wedge \psi \equiv (\forall x\,.\,\phi) \wedge (\forall x\,.\,\psi) \qquad\qquad \exists x\,.\,\phi \vee \psi \equiv (\exists x\,.\,\phi) \vee (\exists x\,.\,\psi)$$

- For $Q \in \{\forall, \exists\}$, if y is free for x in ϕ and $y \notin \mathrm{FV}(\phi)$, then $Qx\,.\,\phi \equiv Qy\,.\,\phi\,[y/x]$.
- For $Q \in \{\forall, \exists\}$, if $x \notin \mathrm{FV}(\phi)$, then $Qx\,.\,\phi \equiv \phi$.
- For $Q \in \{\forall, \exists\}$ and $\odot \in \{\wedge, \vee\}$, if $x \notin \mathrm{FV}(\psi)$, then $Qx\,.\,\phi \odot \psi \equiv (Qx\,.\,\phi) \odot \psi$.

Semantics

Definition

A formula is in *prenex form* if it is of the form $Q_1 x_1 . Q_2 x_2 . \dots Q_n x_n . \psi$ (possibly with $n = 0$) where each Q_i is a quantifier (either $\forall$ or $\exists$) and ψ is a quantifier-free formula .

Proposition

For any formula of first-order logic, there exists an equivalent formula in prenex form.

Proof.

Such a prenex form can be obtained by rewriting, using the logical equivalences listed before. $\square$

Remark

Unlike PL, the validity problem of FOL is not decidable, but it is semi-decidable, i.e. there are procedures s.t., given a formula ϕ, they terminate with "yes" if ϕ is valid but may fail to terminate if ϕ is not valid.

Semantics

Definition

- $\mathcal{M}$ satisfies Γ with α, denoted by $\mathcal{M}, \alpha \models \Gamma$, if $\mathcal{M}, \alpha \models \phi$ for every $\phi \in \Gamma$.
- The notions of satisfiable, valid, unsatisfiable and refutable set of formulas are defined in the expected way.
- Γ *entails* ϕ (or ϕ is a *logical consequence* of Γ), denoted by $\Gamma \models \phi$, iff for every structure $\mathcal{M}$ and assignment α, if $\mathcal{M}, \alpha \models \Gamma$ then $\mathcal{M}, \alpha \models \phi$.
- ϕ is *logically equivalent* to ψ , denoted by $\phi \equiv \psi$, iff $[\![\phi]\!]_{\mathcal{M},\alpha} = [\![\psi]\!]_{\mathcal{M},\alpha}$ for every structure $\mathcal{M}$ and assignment α.

Some properties of semantic entailment

- The properties of semantic entailment listed for PL hold for FOL.
- If t is free for x in ϕ and $\Gamma \models \forall x.\, \phi$, then $\Gamma \models \phi[t/x]$.
- If $x \notin \mathsf{FV}(\Gamma)$ and $\Gamma \models \phi$, then $\Gamma \models \forall x.\, \phi$.
- If t is free for x in ϕ and $\Gamma \models \phi[t/x]$, then $\Gamma \models \exists x.\, \phi$.
- If $x \notin \mathsf{FV}(\Gamma \cup \{\psi\})$, $\Gamma \models \exists x.\, \phi$ and $\Gamma, \phi \models \psi$, then $\Gamma \models \psi$.

Proof system

The natural deduction system $\mathcal{N}_{\mathsf{FOL}}$

- The proof system for FOL we consider is a natural deduction system in sequent style extending $\mathcal{N}_{\mathsf{PL}}$.
- The various definitions made in the context of $\mathcal{N}_{\mathsf{PL}}$ carry over to $\mathcal{N}_{\mathsf{FOL}}$. The difference is that$\mathcal{N}_{\mathsf{FOL}}$ deals with first-order formulas and it has additional introduction and elimination rules to deal with the quantifiers.

Quantifier rules of $\mathcal{N}_{\mathsf{FOL}}$

$$(\mathsf{I}_\forall)\ \frac{\Gamma \vdash \phi\,[y/x]}{\Gamma \vdash \forall x.\,\phi}\ (\mathrm{a}) \qquad\qquad (\mathsf{E}_\forall)\ \frac{\Gamma \vdash \forall x.\,\phi}{\Gamma \vdash \phi\,[t/x]}$$

$$(\mathsf{I}_\exists)\ \frac{\Gamma \vdash \phi\,[t/x]}{\Gamma \vdash \exists x.\,\phi} \qquad\qquad (\mathsf{E}_\exists)\ \frac{\Gamma \vdash \exists x.\,\phi \qquad \Gamma, \phi\,[y/x] \vdash \theta}{\Gamma \vdash \theta}\ (\mathrm{b})$$

(a) $y \notin \mathsf{FV}(\Gamma)$ and either $x = y$ or $y \notin \mathsf{FV}(\phi)$.
(b) $y \notin \mathsf{FV}(\Gamma \cup \{\theta\})$ and either $x = y$ or $y \notin \mathsf{FV}(\phi)$.
(c) Recall that we assume safe substitution, i.e. in a substitution $\phi[t/x]$, we assume that t is free for x in ϕ.

Remark

The properties of $\mathcal{N}_{\mathsf{PL}}$ can be extended to $\mathcal{N}_{\mathsf{FOL}}$, in particular the soundness and completeness theorems.

Theorem (Adequacy)

$\Gamma \models \varphi$ *iff* $\Gamma \vdash \varphi$.

First-order theories

Definition

Let $\mathcal{V}$ be a vocabulary of a first-order language.

- A first-order *theory* $\mathcal{T}$ is a set of $\mathcal{V}$-sentences that is closed under derivability (i.e., $\mathcal{T} \vdash \phi$ implies $\phi \in \mathcal{T}$). A $\mathcal{T}$*-structure* is a $\mathcal{V}$-structure that validates every formula of $\mathcal{T}$.
- A formula ϕ is $\mathcal{T}$*-valid* (resp. $\mathcal{T}$*-satisfiable*) if every (resp. some) $\mathcal{T}$-structure validates ϕ. $\mathcal{T} \models \phi$ denotes the fact that ϕ is $\mathcal{T}$-valid.
- Other concepts regarding validity of first-order formulas are carried over to theories in the obvious way.

Definition

A subset $\mathcal{A} \subseteq \mathcal{T}$ is called an *axiom set* for the theory $\mathcal{T}$ when $\mathcal{T}$ is the deductive closure of $\mathcal{A}$, i.e. $\psi \in \mathcal{T}$ iff $\mathcal{A} \vdash \psi$, or equivalently, iff $\vdash \psi$ can be derived in $\mathcal{N}_{\mathsf{FOL}}$ with an axiom-schema:

$$\frac{}{\Gamma \vdash \phi} \text{ if } \phi \in \mathcal{A}$$

.

First-order theories

Equality theory

The *theory of equality* $\mathcal{T}_{\mathrm{E}}$ for $\mathcal{V}$ (which is assumed to have a binary equality predicate symbol "$=$") has the following axiom set:

- *reflexivity:* $\forall x \,.\, x = x$
- *symmetry:* $\forall x, y \,.\, x = y \to y = x$
- *transitivity:* $\forall x, y, z \,.\, x = y \land y = z \to x = z$
- *congruence for function symbols:* for every $f \in \mathcal{F}$ with $\mathrm{ar}(f) = n$,

$$\forall \overline{x}, \overline{y} \,.\, x_1 = y_1 \land \ldots \land x_n = y_n \to f(x_1, \ldots, x_n) = f(y_1, \ldots, y_n)$$

- *congruence for predicate symbols:* for every $P \in \mathcal{P}$ with $\mathrm{ar}(P) = n$,

$$\forall \overline{x}, \overline{y} \,.\, x_1 = y_1 \land \ldots \land x_n = y_n \to P(x_1, \ldots, x_n) \to P(y_1, \ldots, y_n)$$

Theorem

A sentence ϕ is valid in all normal structures (i.e. structures which interpret $=$ as the equality relation over the interpretation domain) iff $\phi \in \mathcal{T}_{\mathrm{E}}$.