First Order Logic (Predicate Calculus)

Syntax

$\mathcal{L}$-terms

1. Every variable is a term: $x, y, z, \ldots, a, b, c, \ldots$.
2. If f is an n-ary function symbol and $t_1, t_2, \ldots, t_n$ are terms, then so is $ft_1t_2\ldots t_n$.

A 0-ary function symbol is a constant (use e as a metasymbol for constants).

Also 0 and 1 are constants (in $\mathcal{L}_A$).

Eg. f is binary and g is unary, then $fgex, fxy, gfege$ are terms.

Ex. Show the Unique Readability Theorem for terms.

Let $r, s, t, \ldots$ denote terms.

$(t_1 \cdot t_2) =_{\text{synt}} \cdot t_1t_2$
$(t_1 + t_2) =_{\text{synt}} +t_1t_2$

Thus, we use **infix** notation instead of the formal **prefix** notation for $\mathcal{L}_A$ function symbols $\cdot, +$.

Let $\mathcal{L}_A = [0, s, +, \cdot; =]$.

Eg. $\mathcal{L}_A$-terms: $sss0, ((x + sy) \cdot (ssz + s0))$

$\mathcal{L}$-formulas

1. $Pt_1t_2\ldots t_n$ is an **atomic** formula, P is an n-ary predicate symbol, $t_1, t_2, \ldots, t_n$ are terms.
2. If α, β are formulas, then so are $\neg\alpha, (\alpha \vee \beta), (\alpha \wedge \beta)$.
3. If α is a formula, and x a variable, then $\forall x\alpha$ and $\exists x\alpha$ are also formulas.

Eg. $(\neg\forall xPx \vee \exists x\neg Px)$, $(\forall x\neg Qxy \wedge \neg\forall zQfyz)$.

Ex. Show that the set of (well-formed) $\mathcal{L}$-formulas can be given by an $G_{\mathcal{L}}$ grammar of type 2 (a CFG).

Let $r = s$ stand for $= rs$ and $r \neq s$ for $\neg = rs$.

An occurrence of x in α is **bound** if it is in a subformula of α of the form $\forall x\beta$ of $\exists x\beta$ (i.e., in the **scope** of a quantifier).

Otherwise, the occurrence is **free**.

Eg. $\exists y(x = y + y)$: x is free, but y is bound.

In $Px \wedge \forall xQx$ the variable x occurs as free and bound.

A term t or formula α are **close** if they contain no free variables. A closed formula is called a **sentence**.

Semantics

A **structure** (or **interpretation**) gives meaning to terms and formulas. An $\mathcal{L}$ structure $\mathcal{M}$ consists of:

1. A nonempty set M, called the **universe of discourse**.
2. For each n-ary f, $f^{\mathcal{M}} : M^n \longrightarrow M$.
3. For each n-ary P, $P^{\mathcal{M}} \subseteq M^n$.

If $\mathcal{L}$ contains $=$, $=^{\mathcal{M}}$ must be the usual $=$. Equality gets a special treatment—it must always be the true equality.

$<^{\mathcal{M}}$ could be anything, not necessarily the order relation we are used to.

Every $\mathcal{L}$-sentence becomes either true or false when interpreted by an $\mathcal{L}$-structure $\mathcal{M}$.

If a sentence α becomes true under $\mathcal{M}$, we say $\mathcal{M}$ **satisfies** α, or $\mathcal{M}$ is a **model** for α, and write $\mathcal{M} \vDash \alpha$.

If α has free variables, then they must get values from M (the universe of discourse), before α can get a truth value under $\mathcal{M}$.

An **object assignment** σ for a structure $\mathcal{M}$ is a mapping from variables to the universe M.

$t^{\mathcal{M}}[\sigma]$ is an element in M—given by the structure $\mathcal{M}$ and the object assignment σ.

$\mathcal{M} \vDash \alpha[\sigma]$ means that $\mathcal{M}$ satisfies α when its free variables are assigned values by σ.

Basic Semantic Definitions (BSDs) – Tarski semantics

Definition of $t^{\mathcal{M}}[\sigma]$ by structural induction:

1. $x^{\mathcal{M}}[\sigma]$ is $\sigma(x)$
2. $(ft_1t_2 \dots t_n)^{\mathcal{M}}[\sigma]$ is $f^{\mathcal{M}}(t_1^{\mathcal{M}}[\sigma], t_2^{\mathcal{M}}[\sigma], \dots, t_n^{\mathcal{M}}[\sigma])$

If x is a variable, and $m \in M$, then $\sigma(m/x)$ is the same object assignment as σ, but x is mapped to m.

Definition of $\mathcal{M} \vDash \alpha[\sigma]$ by structural induction:

1. $\mathcal{M} \vDash (Pt_1 \dots t_n)[\sigma]$ iff $(t_1^{\mathcal{M}}[\sigma], \dots, t_n^{\mathcal{M}}[\sigma]) \in P^{\mathcal{M}}$.
2. $\mathcal{M} \vDash \neg\alpha[\sigma]$ iff $\mathcal{M} \nvDash \alpha[\sigma]$.
3. $\mathcal{M} \vDash (\alpha \overset{(\vee)}{\wedge} \beta)[\sigma]$ iff $\mathcal{M} \vDash \alpha[\sigma]$ $\overset{\text{(or)}}{\text{and}}$ $\mathcal{M} \vDash \beta[\sigma]$.
4. $\mathcal{M} \vDash (\overset{(\exists)}{\forall} x\alpha)[\sigma]$ iff $\mathcal{M} \vDash \alpha[\sigma(m/x)]$ for $\overset{\text{(some)}}{\text{all}}$ $m \in M$.

If t is closed, we write $t^{\mathcal{M}}$. If α is a sentence, we write $\mathcal{M} \vDash \alpha$.

Ex. Let $\mathcal{L} = [; R, =]$ (R binary predicate) and let $\mathcal{M}$ be an $\mathcal{L}$-structure with universe $\mathbb{N}$ and such that $(m, n) \in R^{\mathcal{M}}$ iff $m \leq n$.

Then, $\mathcal{M} \vDash \exists x \forall y Rxy$.

But, $\mathcal{M} \nvDash \exists y \forall x Rxy$.

The **standard structure** $\underline{\mathbb{N}}$ for the language $\mathcal{L}_A$ has universe $M = \mathbb{N} = \{0, 1, 2, \ldots\}$, $s^{\underline{\mathbb{N}}}(n) = n+1$, and $0, +, \cdot, =$ get their usual meaning on the natural numbers.

Ex. $\underline{\mathbb{N}} \vDash \forall x \forall y \exists z(x + z = y \vee y + z = x)$.

But, $\underline{\mathbb{N}} \nvDash \forall x \exists y(y + y = x)$.

Let Φ denote a set of formulas.

α is **satisfiable** iff $\mathcal{M} \vDash \alpha[\sigma]$ for some $\mathcal{M}$ & σ.

$\mathcal{M} \vDash \Phi[\sigma]$ iff $\mathcal{M} \vDash \alpha[\sigma]$ for all $\alpha \in \Phi$.

$\Phi \vDash \alpha$ iff for all $\mathcal{M}$ & σ, if $\mathcal{M} \vDash \Phi[\sigma]$ then $\mathcal{M} \vDash \alpha[\sigma]$. ($\alpha$ is a **logical consequence** of Φ—second use of "$\vDash$")

α is **valid**, $\vDash \alpha$, iff $\mathcal{M} \vDash \alpha[\sigma]$ for all $\mathcal{M}$ & σ.

α & β are **logically equivalent**, $\alpha \iff \beta$, iff for all $\mathcal{M}$ & σ, ($\mathcal{M} \vDash \alpha[\sigma]$ iff $\mathcal{M} \vDash \beta[\sigma]$).

Note that $\vDash$ is a symbol of the "meta language" (English), as opposed to $\wedge, \vee, \exists, \ldots$ which are symbols of first order logic.

If Φ is just one formula, i.e., $\Phi = \{\beta\}$, then we write $\beta \vDash \alpha$ in place of $\{\beta\} \vDash \alpha$.

Ex. $(\forall x\alpha \vee \forall x\beta) \vDash \forall x(\alpha \vee \beta)$ for all formulas α and β.

We prove this using BSDs:

Let $\mathcal{M}$ be any structure, and σ any object assignment. Suppose $\mathcal{M} \vDash (\forall x\alpha \vee \forall x\beta)[\sigma]$.

Then, $\mathcal{M} \vDash \forall x\alpha[\sigma]$ or $\mathcal{M} \vDash \forall x\beta[\sigma]$.

Case (1): $\mathcal{M} \vDash \forall x\alpha[\sigma]$. Then, $\mathcal{M} \vDash \alpha[\sigma(m/x)]$ for all $m \in M$.

Then, $\mathcal{M} \vDash (\alpha \vee \beta)[\sigma(m/x)]$ for all $m \in M$.

So, $\mathcal{M} \vDash \forall x(\alpha \vee \beta)[\sigma]$.

Case (2): $\mathcal{M} \vDash \forall x\beta[\sigma]$. Same idea.

$\therefore \mathcal{M} \vDash \forall x(\alpha \vee \beta)[\sigma]$

By the def of logical consequence, $(\forall x\alpha \vee \forall x\beta) \vDash \forall x(\alpha \vee \beta)$

Ex. $\forall x(\alpha \vee \beta) \vDash (\forall x\alpha \vee \forall x\beta)$?

No, not necessarily. How to prove this ?

Use the def of logical consequence.

To prove that the RHS is **not** a logical consequence of the LHS, we must:

Exhibit:

1. a model $\mathcal{M}$
2. an object assignment σ
3. formulas α, β

such that:

$\mathcal{M} \vDash \forall x(\alpha \vee \beta)[\sigma]$, but $\mathcal{M} \nvDash (\forall x\alpha \vee \forall x\beta)[\sigma]$.

Let α and β be Px and Qx, respectively (P, Q unary predicates).

Now define $\mathcal{M}$ and σ.

Since the formulas are sentences, no need to define σ.

$\mathcal{M}$: let the universe of discourse be $M = \mathbb{N}$.

We still need to give meaning in $\mathcal{M}$ to P, Q. Let $P^{\mathcal{M}} = \{0, 2, 4, \ldots\}$, and $Q^{\mathcal{M}} = \{1, 3, 5, \ldots\}$.

Then: $\mathcal{M} \vDash \forall x(Px \vee Qx)$
(bec. every number is even or odd).

But, $\mathcal{M} \nvDash (\forall xPx \vee \forall xQx)$
(bec. it is not true that either all numbers are even or all numbers are odd).

Substitution

Syntax: t, u are terms. Then:

$t(u/x)$	result of replacing all occurrences of x in t by u
$\alpha(u/x)$	result of replacing all **free** occurrences of x in α by u

Semantics: $(u(t/x))^{\mathcal{M}}[\sigma] = u^{\mathcal{M}}[\sigma(m/x)]$ where $m = t^{\mathcal{M}}[\sigma]$.

Eg. Let $\mathcal{M}$ be $\underline{\mathbb{N}}$ (the standard structure) for $\mathcal{L}_A$. Suppose $\sigma(x) = 5$ and $\sigma(y) = 7$. Let:

u	be the term $x + y$
t	be the term $ss0$

Then:

$$u(t/x) \text{ is } ss0 + y \text{ and so } (u(t/x))^{\underline{\mathbb{N}}}[\sigma] = 2 + 7 = 9$$

On the other hand, $m = t^{\underline{\mathbb{N}}} = 2$, so $u^{\underline{\mathbb{N}}}[\sigma(m/x)] = 2 + 7 = 9$.

Ex. Prove $(u(t/x))^{\mathcal{M}}[\sigma] = u^{\mathcal{M}}[\sigma(m/x)]$ where $m = t^{\mathcal{M}}[\sigma]$, by structural induction on u.

Does the above apply to formulas α ?
That is, is it true that $\mathcal{M} \vDash \alpha(t/x)[\sigma]$ iff $\mathcal{M} \vDash \alpha[\sigma(m/x)]$, where $m = t^{\mathcal{M}}[\sigma]$?

No.

Eg. Suppose α is $\forall y \neg(x = y + y)$. This says "$x$ is odd". But $\alpha(x + y/x)$ is $\forall y \neg(x + y = y + y)$ which is always false, regardless of the value of $\sigma(x)$. The problem is that y in the term $x + y$ got "caught" by the quantifier $\forall y$.

Def. A term t is **freely substitutable for x in α** iff no free occurrence of x in α is in a subformula of α of the form $\forall y\beta$ or $\exists y\beta$, where y occurs in t.

Substitution Theorem: If t is freely substitutable for x in α then for all structures $\mathcal{M}$ and all object assignments σ, $\mathcal{M} \models \alpha(t/x)[\sigma]$ iff $\mathcal{M} \models \alpha[\sigma(m/x)]$, where $m = t^{\mathcal{M}}[\sigma]$.

Ex. Prove the Substitution Theorem. (Hint. Use structural induction on α and the BSDs.)

If a term t is not freely substitutable for x in α, it is because some variable y in t gets caught by a quantifier $\forall y$ or $\exists y$ in α.

One way to fix this is simply rename the bound variable y in α to some new variable z. This renaming does not change the meaning of α.

LK

Let $a, b, c, \ldots$ denote free and let $x, y, z, \ldots$ to denote bound var.

A first order formula α is called a **proper formula** if it satisfies the restriction that it has no free occurrence of any "bound" variable and no bound occurrence of any "free" variable.

Similarly a **proper term** has no "bound" variable.

Notice that a subformula of a proper formula is not necessarily proper, and a proper formula may contain terms which are not proper.

The sequent system LK is an extension of the propositional system PK where now all formulas in the sequent $\alpha_1, \ldots, \alpha_k \to \beta_1, \ldots, \beta_l$ must be proper formulas.

The system LK is PK together with the following four rules for introducing quantifiers:

$$\forall \text{ introduction:} \qquad \frac{\alpha(t), \Gamma \to \Delta}{\forall x\alpha(x), \Gamma \to \Delta} \qquad \frac{\Gamma \to \Delta, \alpha(b)}{\Gamma \to \Delta, \forall x\alpha(x)}$$

$$\exists \text{ introduction:} \qquad \frac{\alpha(b), \Gamma \to \Delta}{\exists x\alpha(x), \Gamma \to \Delta} \qquad \frac{\Gamma \to \Delta, \alpha(t)}{\Gamma \to \Delta, \exists x\alpha(x)}$$

Restrictions: t is a proper term.
$\alpha(t)$ (resp. $\alpha(b)$) is the result of substituting t (resp. b) for all free occurrences of x in $\alpha(x)$.
Note that t, b can be freely substituted for x in $\alpha(x)$ because $\forall x\alpha(x), \exists x\alpha(x)$ are proper formulas.
The free variable b must not occur in the conclusion in $\forall$ right and $\exists$ left.

Ex. Show that the four new rules are sound.

Ex. Give a specific example of a sequent $\Gamma \to \Delta, \alpha(b)$ which is valid, but the bottom sequent $\Gamma \to \Delta, \forall x\alpha(x)$ is not valid, because the restriction on b is violated (b occurs in Γ or Δ or $\forall x\alpha(x)$). Do the same for $\exists$ left.

An LK proof of a valid first order sequent can be obtained using the same method as in the propositional case. Write the goal sequent at the bottom, and move up using the introduction rules in reverse.

If there is a choice about which quantifier to remove next, choose $\forall$ right or $\exists$ left (working backward), since these rules carry a restriction.

Peano Arithmetic

Recall the language of arithmetic, $\mathcal{L}_A = [0, s, +, \cdot; =]$.

Axioms for PA

P1 $\forall x(sx \neq 0)$
P2 $\forall x \forall y(sx = sy \supset x = y)$
P3 $\forall x(x + 0 = x)$
P4 $\forall x \forall y(x + sy = s(x + y))$
P5 $\forall x(x \cdot 0 = 0)$
P6 $\forall x \forall y(x \cdot sy = x \cdot y + x)$

And the **Induction Scheme:**

$$\forall y_1 \dots \forall y_k[(\alpha(0) \wedge \forall x(\alpha(x) \supset \alpha(sx))) \supset \forall x \alpha(x)] \tag{1}$$

where α is any $\mathcal{L}_A$-formula, and (1) is a sentence.

Equality Axioms

E1 $\forall x(x = x)$
E2 $\forall x \forall y(x = y \supset y = x)$
E3 $\forall x \forall y \forall z((x = y \wedge y = z) \supset x = z)$
E4 $\forall x_1 \dots \forall x_n \forall y_1 \dots \forall y_n(x_1 = y_1 \wedge \cdots \wedge x_n = y_n) \supset fx_1 \dots x_n = fy_1 \dots y_n$
E5 $\forall x_1 \dots \forall x_n \forall y_1 \dots \forall y_n(x_1 = y_1 \wedge \cdots \wedge x_n = y_n) \supset Px_1 \dots x_n \supset Py_1 \dots y_n$

where E4 and E5 hold for all n-ary function and predicate symbols.
In $\mathcal{L}_A$ s is unary, $+, \cdot$ are binary, and $=$ is binary.

Let LK-PA be LK where the leaves are allowed to be P1-6 and E1-5, besides the usual axioms $\alpha \to \alpha$.

For example, $\to \forall x(x = x)$ would be a valid leaf.

Ex. Show that LK-PA proves that all nonzero elements have predecessor.

Let $\alpha(x)$ be $(x = 0 \vee \exists y(x = sy))$.

We outline the proof informally, but the proof can of course be formalized in LK-PA.

Basis: $x = 0$, and LK-PK proves $\alpha(0)$ easily:

$$
\begin{prooftree}
\AxiomC{$\to \forall x(x = x)$}
\doubleLine
\RightLabel{weak \& exch}
\UnaryInfC{$\to 0 = 0, \forall x(x = x)$}
\AxiomC{$0 = 0 \to 0 = 0$}
\RightLabel{$\forall$-left}
\UnaryInfC{$\forall x(x = x) \to 0 = 0$}
\RightLabel{Cut}
\BinaryInfC{$\to 0 = 0$}
\RightLabel{weak}
\UnaryInfC{$\to 0 = 0, \exists y(0 = sy)$}
\RightLabel{$\vee$-right}
\UnaryInfC{$\to 0 = 0 \vee \exists y(0 = sy)$}
\end{prooftree}
$$

Induction Step: Show that LK-PA proves $\forall x(\alpha(x) \supset \alpha(sx))$, i.e., we must give an LK-PA proof of the sequent:

$$\to \forall x(\neg(x = 0 \vee \exists y(x = sy)) \vee (sx = 0 \vee \exists y(sx = sy)))$$

This is not difficult, and it is left as an exercise. From $\alpha(0)$ and $\forall x(\alpha(x) \supset \alpha(sx))$, and using the axiom:

$$\rightarrow (\alpha(0) \wedge \forall x(\alpha(x) \supset \alpha(sx))) \supset \forall x \alpha(x)$$

we can now conclude (in just a few steps): $\rightarrow \forall x \alpha(x)$ which is what we wanted to prove. Thus, LK-PA proves $\forall x \alpha(x)$.

Ex. Show that LK-PA proves the following:
–the associative and commutative law of addition
–the associative and commutative laws of multiplication
–that multiplication distributes over addition
Specify carefully which axioms you are using.