First-Order Predicate Logic
Basics

Syntax of predicate logic: terms

A variable is a symbol of the form x_i where $i = 1, 2, 3 \ldots$.

A function symbol is of the form f_i^k where $i = 1, 2, 3 \ldots$ und $k = 0, 1, 2 \ldots$.

A predicate symbol is of the form P_i^k where $i = 1, 2, 3 \ldots$ and $k = 0, 1, 2 \ldots$.

We call i the index and k the arity of the symbol.

Terms are inductively defined as follows:

1. Variables are terms.
2. If f is a function symbol of arity k and $t_1, \ldots, t_k$ are terms then $f(t_1, \ldots, t_k)$ is a term.

Function symbols of arity 0 are called constant symbols.
Instead of $f_i^0()$ we write f_i^0.

Syntax of predicate logic: formulas

If P is a predicate symbol of arity k and $t_1, \ldots, t_k$ are terms then $P(t_1, \ldots, t_k)$ is an atomic formula.
If $k = 0$ we write P instead of $P()$.

Formulas (of predicate logic) are inductively defined as follows:

- Every atomic formula is a formula.
- If F is a formula, then $\neg F$ is also a formula.
- If F and G are formulas, then $F \wedge G$, $F \vee G$ and $F \rightarrow G$ are also formulas.
- If x is a variable and F is a formula, then $\forall x\ F$ and $\exists x\ F$ are also formulas. The symbols $\forall$ and $\exists$ are called the universal and the existential quantifier.

Syntax trees and subformulas

Syntax trees are defined as before,
extended with the following trees for $\forall xF$ and $\exists xF$:

$$\begin{array}{ccc} \forall x & \qquad & \exists x \\ | & & | \\ F & & F \end{array}$$

Subformulas again correspond to subtrees.

Sructural induction of formulas

Like for propositional logic but

- Different base case: $\mathcal{P}(P(t_1, \ldots, t_k))$
- Two new induction steps:
  prove $\mathcal{P}(\forall x\ F)$ under the induction hypothesis $\mathcal{P}(F)$
  prove $\mathcal{P}(\exists x\ F)$ under the induction hypothesis $\mathcal{P}(F)$

Naming conventions

$x, y, z, \ldots$	instead of	$x_1, x_2, x_3, \ldots$
$a, b, c, \ldots$	for	constant symbols
$f, g, h, \ldots$	for	function symbols of arity > 0
$P, Q, R, \ldots$	instead of	P_i^k

Precedence of quantifiers

Quantifiers have the same precedence as $\neg$

Example

$\forall x\ P(x) \wedge Q(x)$	abbreviates	$(\forall x\ P(x)) \wedge Q(x)$
	not	$\forall x\ (P(x) \wedge Q(x))$

Similarly for $\vee$ etc.

[This convention is not universal]

Free and bound variables, closed formulas

A variable x occurs in a formula F if it occurs in some atomic subformula of F.

An occurrence of a variable in a formula is either free or bound.

An occurrence of x in F is bound if it occurs in some subformula of F of the form $\exists x G$ or $\forall x G$; the smallest such subformula is the scope of the occurrence. Otherwise the occurrence is free.

A formula without any free occurrence of any variable is closed.

Example

$\forall x \; P(x) \rightarrow \exists y \; Q(a, x, y)$

Exercise

	Closed?
$\forall x\ P(a)$	
$\forall x \exists y\ (Q(x,y) \vee R(x,y))$	Y
$\forall x\ Q(x,x) \rightarrow \exists x\ Q(x,y)$	N
$\forall x\ P(x) \vee \forall x\ Q(x,x)$	Y
$\forall x\ (P(y) \wedge \forall y\ P(x))$	N
$P(x) \rightarrow \exists x\ Q(x, P(x))$	N

	Formula?
$\exists x\ P(f(x))$	
$\exists f\ P(f(x))$	

Semantics of predicate logic: structures

A structure is a pair $\mathcal{A} = (U_\mathcal{A}, I_\mathcal{A})$
where $U_\mathcal{A}$ is an arbitrary, nonempty set called the universe of $\mathcal{A}$,
and the interpretation $I_\mathcal{A}$ is a partial function that maps

- variables to elements of the universe $U_\mathcal{A}$,
- function symbols of arity k to functions of type $U_\mathcal{A}^k \to U_\mathcal{A}$,
- predicate symbols of arity k to functions of type $U_\mathcal{A}^k \to \{0, 1\}$ (predicates) [or equivalently to subsets of $U_\mathcal{A}^k$ (relations)]

$I_\mathcal{A}$ maps syntax (variables, functions and predicate symbols) to their meaning (elements, functions and predicates)

The special case of arity 0 can be written more simply:

- constant symbols are mapped to elements of $U_\mathcal{A}$,
- predicate symbols of arity 0 are mapped to $\{0, 1\}$.

Abbreviations:

$x^{\mathcal{A}}$	abbreviates	$I_{\mathcal{A}}(x)$
$f^{\mathcal{A}}$	abbreviates	$I_{\mathcal{A}}(f)$
$P^{\mathcal{A}}$	abbreviates	$I_{\mathcal{A}}(P)$

Example

$U_{\mathcal{A}} = \mathbb{N}$

$I_{\mathcal{A}}(P) = P^{\mathcal{A}} = \{(m, n) \mid m, n \in \mathbb{N} \text{ and } m < n\}$

$I_{\mathcal{A}}(Q) = Q^{\mathcal{A}} = \{m \mid m \in \mathbb{N} \text{ and } m \text{ is prime}\}$

$I_{\mathcal{A}}(f)$ is the successor function: $f^{\mathcal{A}}(n) = n + 1$

$I_{\mathcal{A}}(g)$ is the addition function: $g^{\mathcal{A}}(m, n) = m + n$

$I_{\mathcal{A}}(a) = a^{\mathcal{A}} = 2$

$I_{\mathcal{A}}(z) = z^{\mathcal{A}} = 3$

Intuition: is $\forall x\ P(x, f(x)) \land Q(g(a, z))$ true in this structure?

Evaluation of a term in a structure

Definition

Let t be a term and let $\mathcal{A} = (U_{\mathcal{A}}, I_{\mathcal{A}})$ be a structure.
$\mathcal{A}$ is suitable for t if $I_{\mathcal{A}}$ is defined for all variables and function symbols occurring in t.

The value of a term t in a suitable structure $\mathcal{A}$, denoted by $\mathcal{A}(t)$, is defined recursively:

$$\begin{array}{rcl} \mathcal{A}(x) & = & x^{\mathcal{A}} \\ \mathcal{A}(c) & = & c^{\mathcal{A}} \\ \mathcal{A}(f(t_1, \dots, t_k) & = & f^{\mathcal{A}}(\mathcal{A}(t_1), \dots, \mathcal{A}(t_k)) \end{array}$$

Example

$\mathcal{A}(f(g(a, z))) =$

Definition

Let F be a formula and let $\mathcal{A} = (U_{\mathcal{A}}, I_{\mathcal{A}})$ be a structure.
$\mathcal{A}$ is suitable for F if $I_{\mathcal{A}}$ is defined for all predicate and function symbols occurring in F and for all variables occurring free in F.

Evaluation of a formula in a structure

Let $\mathcal{A}$ be suitable for F. The (truth)value of F in $\mathcal{A}$, denoted by $\mathcal{A}(F)$, is defined recursively:

$$\mathcal{A}(\neg F),\ \mathcal{A}(F \wedge G),\ \mathcal{A}(F \vee G),\ \mathcal{A}(F \rightarrow G)$$

as for propositional logic

$$\mathcal{A}(P(t_1, \ldots, t_k)) = \begin{cases} 1 & \text{if } (\mathcal{A}(t_1), \ldots, \mathcal{A}(t_k)) \in P^{\mathcal{A}} \\ 0 & \text{otherwise} \end{cases}$$

$$\mathcal{A}(\forall x\ F) = \begin{cases} 1 & \text{if for every } d \in U_{\mathcal{A}},\ (\mathcal{A}[d/x])(F) = 1 \\ 0 & \text{otherwise} \end{cases}$$

$$\mathcal{A}(\exists x\ F) = \begin{cases} 1 & \text{if for some } d \in U_{\mathcal{A}},\ (\mathcal{A}[d/x])(F) = 1 \\ 0 & \text{otherwise} \end{cases}$$

$\mathcal{A}[d/x]$ coincides with $\mathcal{A}$ everywhere except that $x^{\mathcal{A}[d/x]} = d$.

Notes

- During the evaluation of a formulas in a structure,
  the structure stays unchanged
  except for the interpretation of the variables.
- If the formula is closed,
  the initial interpretation of the variables is irrelevant.

Example

$\mathcal{A}(\forall x\ P(x, f(x)) \land Q(g(a, z))) =$

Relation to propositional logic

- Every propositional formula can be seen as a formula of predicate logic where the atom A_i is replaced by the atom P_i^0.
- Conversely, every formula of predicate logic
  that does not contain quantifiers and variables
  can be seen as a formula of propositional logic
  by replacing atomic formulas by propositional atoms.

  Example
  $F = (Q(a) \vee \neg P(f(b), b) \wedge P(b, f(b)))$
  can be viewed as the propositional formula
  $F' = (A_1 \vee \neg A_2 \wedge A_3)$.

  Exercise
  F is satifiable/valid iff F' is satisfiable/valid

Predicate logic with equality

Predicate logic

$+$

distinguished predicate symbol "=" of arity 2

Semantics: A structure $\mathcal{A}$ of predicate logic with equality always maps the predicate symbol = to the identity relation:

$$\mathcal{A}(=) \ = \ \{(d, d) \mid d \in U_{\mathcal{A}}\}$$

Model, validity, satisfiability

Like in propositional logic

Definition

We write $\mathcal{A} \models F$ to denote that the structure $\mathcal{A}$ is suitable for the formula F and that $\mathcal{A}(F) = 1$.
Then we say that F is true in $\mathcal{A}$ or that $\mathcal{A}$ is a model of F.

If every structure suitable for F is a model of F,
then we write $\models F$ and say that F is valid.

If F has at least one model then we say that F is satisfiable.

Exercise

V: valid   S: satisfiable, but not valid   U: unsatisfiable

	V	S	U
$\forall x\ P(a)$			
$\exists x\ (\neg P(x) \vee P(a))$			
$P(a) \rightarrow \exists x\ P(x)$			
$P(x) \rightarrow \exists x\ P(x)$			
$\forall x\ P(x) \rightarrow \exists x\ P(x)$			
$\forall x\ P(x) \wedge \neg \forall y\ P(y)$			

Consequence and equivalence

Like in propositional logic

Definition

A formula G is a consequence of a set of formulas M
if every structure that is a model of all $F \in M$ and suitable for G
is also model of G. The we write $M \models G$.

Two formulas F and G are (semantically) equivalent
if every structure $\mathcal{A}$ suitable for both F and G satisfies
$\mathcal{A}(F) = \mathcal{A}(G)$. Then we write $F \equiv G$.

Exercise

1. $\forall x\ P(x) \lor \forall x\ Q(x,x)$
2. $\forall x\ (P(x) \lor Q(x,x))$
3. $\forall x\ (\forall z\ P(z) \lor \forall y\ Q(x,y))$

	Y	N
$1 \models 2$		
$2 \models 3$		
$3 \models 1$		

Exercise

1. $\exists y \forall x \; P(x, y)$
2. $\forall x \exists y \; P(x, y)$

	Y	N
$1 \models 2$		
$2 \models 1$		

Exercise

	Y	N
$\forall x \forall y\ F \ \equiv\ \forall y \forall x\ F$		
$\forall x \exists y\ F \ \equiv\ \exists x \forall y\ F$		
$\exists x \exists y\ F \ \equiv\ \exists y \exists x\ F$		
$\forall x\ F \vee \forall x\ G \ \equiv\ \forall x\ (F \vee G)$		
$\forall x\ F \wedge \forall x\ G \ \equiv\ \forall x\ (F \wedge G)$		
$\exists x\ F \vee \exists x\ G \ \equiv\ \exists x\ (F \vee G)$		
$\exists x\ F \wedge \exists x\ G \ \equiv\ \exists x\ (F \wedge G)$		

Equivalences

Theorem

1. $\neg\forall x F \equiv \exists x \neg F$
   $\neg\exists x F \equiv \forall x \neg F$
2. *If x does not occur free in G then:*
   $(\forall x F \wedge G) \equiv \forall x (F \wedge G)$
   $(\forall x F \vee G) \equiv \forall x (F \vee G)$
   $(\exists x F \wedge G) \equiv \exists x (F \wedge G)$
   $(\exists x F \vee G) \equiv \exists x (F \vee G)$
3. $(\forall x F \wedge \forall x G) \equiv \forall x (F \wedge G)$
   $(\exists x F \vee \exists x G) \equiv \exists x (F \vee G)$
4. $\forall x \forall y F \equiv \forall y \forall x F$
   $\exists x \exists y F \equiv \exists y \exists x F$

Replacement theorem

Just like for propositional logic it can be proved:

Theorem

Let $F \equiv G$. Let H be a formula with an occurrence of F as a subformula. Then $H \equiv H'$, where H' is the result of replacing an arbitrary occurrence of F in H by G.