Predicate Logic

We want to able to state things like:

For all numbers x and y, $\quad (x+y)^2 = x^2 + 2xy + y^2$.

For every natural number n, if $n \neq 0$, then there exists a natural number m, s.t. $n = m + 1$.

Every pair of natural numbers has a greatest common divisor: For all numbers n_1 and n_2, there exists a natural number y, s.t. $y|n_1$ and $y|n_2$, and for all y', if $y'|n_1$ and $y'|n_2$, then $y' \leq y$.

Predicate Logic

In order to say such things, we need to be extend the language of propositional logic with the following:

- Object variables, object names and function symbols.
- Relations.
- Equality.
- Quantifiers.

Language of Predicate Logic: Terms

We assume a set of function symbols $\mathcal{F}$. Each function symbol f has an arity ≥ 0 associated to it.

The arity of a function symbol is the number of arguments that it can be applied on.

We call the functions with arity 0 either constants or variables dependent on how we use them now or intend to use them later. For the rest, they are the same symbols. When bound by a quantifier $\exists x$ or $\forall x$, we call them variables. Otherwise, we call them constants.

Terms are recursively defined as follows:

- If f is a function with arity n, $t_1, \ldots, t_n$ are terms, then $f(t_1, \ldots, t_n)$ is a term.

Terms are used to denote objects.

Language of Predicate Logic: Atoms

We assume a set of predicate symbols $\mathcal{P}$. Like the functions, each predicate symbol has an associated arity ≥ 0.

Atoms are defined as follows:

- If p is a function symbol with arity n, $t_1, \ldots, t_n$ are terms, then $p(t_1, \ldots, t_n)$ is an atom.
- If t_1 and t_2 are terms, then $t_1 \approx t_2$ is an atom.

Language of Predicate Logic: Formulas

Formulas are recursively defined as follows:

- If A is an atom, then A is a formula.
- $\bot$ and $\top$ are formulas.
- If F is a formula, then $\neg F$ is a formula.
- If F_1 and F_2 are formulas, then
  $F_1 \wedge F_2, \quad F_1 \vee F_2, \quad F_1 \rightarrow F_2, \quad F_1 \leftrightarrow F_2$ are formulas.
- If x is a variable, F is a formula, then $\forall x{:}X\ F$ and $\exists x{:}X\ F$ are formulas.

Quantifiers

The operators $\exists$ and $\forall$ are called quantifiers:

Examples:

Some birds can fly:

$$\exists x\ [B(x) \wedge CF(x)].$$

Some birds can swim:

$$\exists x\ [B(x) \wedge CS(x)].$$

Some birds can swim and fly:

$$\exists x\ [B(x) \wedge CF(x) \wedge CS(x)].$$

No bird can count:

$$\neg\exists x\ [B(x) \wedge CC(x)].$$

All birds cannot count:

$$\forall x\ [B(x) \rightarrow \neg CC(x)].$$

If something can count, then it is not a bird:

$$\forall x\ [CC(x) \rightarrow \neg B(x)].$$

There exists a bird:

$$\exists x\ B(x).$$

There exist at least two birds:

$$\exists x_1 x_2\ [x_1 \neq x_2 \wedge B(x_1) \wedge B(x_2)].$$

There exist at most one bird:

$$\forall x_1 x_2 [B(x_1) \wedge B(x_2) \rightarrow x_1 \approx x_2].$$

Examples from Mathematics

The atoms $p(t_1, \ldots, t_n)$ can have form
$3 < 4, \;\; 1 < 1 + 1, \; \text{even}(4), \; \text{odd}(5),$
$\text{substring}(\text{"}cde\text{"}, \text{"}abcdefgh\text{"})$.

Examples of formulas are
$\forall n_1, n_2 \; \text{Nat}(n_1) \wedge \text{Nat}(n_2) \rightarrow$
$\quad (\; n_1 | n_2 \leftrightarrow \exists m \; \text{Nat}(m) \wedge n_1.m \approx n_2 \;)$.
This way of using $\rightarrow$ and $\wedge$ for encoding type information is called relativization. Relativization in formulas always occurs with quantifiers. With $\forall$, it is always $\rightarrow$. With $\exists$, it is always $\wedge$.

$\forall n_1, n_2, m \; \text{Nat}(n_1) \wedge \text{Nat}(n_2) \wedge \text{Nat}(m) \rightarrow$
$\quad \text{Gcd}(n_1, n_2, m) \leftrightarrow (m | n_1 \wedge m | n_2 \; \wedge$
$\quad\quad \forall m' \; \text{Nat}(m') \rightarrow (m' | n_1 \wedge m' | n_2 \rightarrow m \leq m')$.

Families of Deduction Systems

The most important types of deduction systems are:

- Natural Deduction: Natural Deduction follows the natural style of reasoning, as it can be found in mathematical textbooks or in spoken arguments. Most of the proof consists of forward reasoning, that is deriving conclusions, deriving new conclusions from these conclusions, etc. Occassionally assumptions are introduced or dropped.

- Sequent Calculus: In sequent calculus, conclusions and premisses are treated in the same way. The reasoning proceeds by deriving relations between formulas, instead of deriving only conclusions. This is different from the style found in textbooks, but the resulting calculus is easier to use interactively.

- Axiomatic Method: Axiomatic Methods are historically the oldest proof systems, but they are not important anymore. Their distinguishing feature is that logical operators are defined by axioms. There are usually three deduction rules, modus ponens:

  If A and $A \to B$ are provable, then so is B,

  generalization

  If A is provable, then so is $\forall x\ A$

  and formula instantiation:

  If A is provable, then so is $A[X := F]$.

Sequent Calculus

A sequent is an object of form $\Gamma \vdash \Delta$, in which Γ and Δ are sets of formulas.

The meaning is: Whenever all of the Γ are true, then at least one of the Δ is true.

First-order logic is very symmetric. Because of this, it makes no sense to distinguish between Γ and Δ.

A one-sided sequent is an object of form $\Gamma \vdash \perp$.

Intuitive meaning: Not all of Γ can be simultaneously true (or Γ is unsatisfiable).

A two-sided sequent $A_1, \ldots, A_p \vdash B_1, \ldots, B_q$ can be replaced by one-sided sequent $A_1, \ldots, A_p, \neg B_1, \ldots, \neg B_q \vdash$.

Negation Rules

We want a small calculus. We define rewrite rules for negation, so that we do not need special rules for it:

$$
\begin{array}{lcl}
\neg\top & \Rightarrow & \bot \\
\neg\bot & \Rightarrow & \top \\
\neg\neg A & \Rightarrow & A \\
\neg(A \wedge B) & \Rightarrow & \neg A \vee \neg B \\
\neg(A \vee B) & \Rightarrow & \neg A \wedge \neg B \\
\neg(A \rightarrow B) & \Rightarrow & A \wedge \neg B \\
\neg(A \leftrightarrow B) & \Rightarrow & (A \vee B) \wedge (\neg A \vee \neg B) \\
\neg(\forall x{:}X\ F) & \Rightarrow & \exists x{:}X\ \neg F \\
\neg(\exists x{:}X\ F) & \Rightarrow & \forall x{:}X\ \neg F
\end{array}
$$

Implication Rules

We also define rewrite rules for $\rightarrow$ and $\leftrightarrow$, so that we won't need rules for those:

$$\begin{aligned} A \rightarrow B &\Rightarrow \neg A \vee B \\ A \leftrightarrow B &\Rightarrow (\neg A \vee B) \wedge (A \vee \neg B) \end{aligned}$$

The rewrite rules can be as preprocessing before the proof is constructed, or only on top-level just before a rule is applied.

If they are used a preprocessing, the resulting formulas are called in negation normal form.

One-Sided Sequent Calculus

$$(\text{cut})\ \frac{\Gamma, A \vdash \qquad \Gamma, \neg A \vdash}{\Gamma \vdash}$$

$$(\wedge)\ \frac{\Gamma, A, B \vdash}{\Gamma, A \wedge B \vdash} \qquad (\vee)\ \frac{\Gamma, A \vdash \qquad \Gamma, B \vdash}{\Gamma,\ A \vee B \vdash}$$

$$(\forall)\ \frac{\Gamma,\ P[x := t] \vdash}{\Gamma,\ \forall x{:}X\ P \vdash} \qquad (\exists)\ \frac{\Gamma, P \vdash}{\Gamma,\ \exists x{:}X\ P \vdash}$$

t must be an arbitrary term (of the right type, when we consider types). It must be the case that x is not free in Γ or Δ.

Axioms

How to define axioms?

- There must be enough axioms.
- Being an axiom must be robust against small changes.
- Needs to support equality.
- Must be decidable.
- Closed under cut.

Axioms (2)

- Every sequent of form $\Gamma, \bot \vdash$, $\Gamma, \neg(t \approx t) \vdash$, or $\Gamma, A, \neg A \vdash$ is an axiom.
- $\Gamma,\ t \approx t \vdash$ and $\Gamma, \top \vdash$ are axioms if $\Gamma \vdash$ is an axiom.
- $\Gamma,\ A \wedge B \vdash$ is an axiom if $\Gamma, A, B \vdash$ is an axiom. (I assume that equalities can swapped freely.)
- $\Gamma,\ t \approx u,\ A[t] \vdash$ is an axiom if $t > u$ and $\Gamma,\ t \approx u,\ A[u] \vdash$ is an axiom.
- $\Gamma,\ A \vee B \vdash$ is an axiom if $\Gamma, A \vdash$ and $\Gamma, B \vdash$ are axioms.
- $\Gamma,\ \forall x{:}X\ P_1[x], \ldots, \forall x{:}X\ P_n[x],\ \exists x{:}X\ Q[x] \vdash$ is an axiom if $\Gamma,\ P_1[x], \ldots, P_n[x],\ Q[x] \vdash$ is an axiom.

Axioms (3)

Deciding whether a sequent is an axiom is PSPACE complete.
That is a bit costly, but it works well.

The order $>$ must be a simplification order:

- If $t > u$, then $A[t] > A[u]$.
- $>$ is well-founded, i.e. there exists no infinite descending chain $t_1 > t_2 > t_3 > \cdots$

Axioms (4)

When defining equality for formulas, one has to take into account that the bound variables may differ?

Equal or Not?

$$\forall x\ P(x) \text{ and } \forall x\ P(x)?$$

$$\forall x\ P(x) \text{ and } \forall y\ P(x)?$$

$$\forall xy\ P(x,y) \text{ and } \forall yx\ P(y,x)?$$

$$\forall z\ P(x,z) \text{ and } \forall z\ P(y,z)?$$

$$\forall xx\ P(x,y) \text{ and } \forall xz\ P(x,y)?$$

The answer is α-equivalence.

Renaming, α-Equivalence

Two formulas F_1 and F_2 are α-equivalent if they have the same skeleton, and for every variable x_1 in F_1 and x_2 occurring at the same position in F_2, either

- x_1 is free in F_1, x_2 is free in F_2, and $x_1 = x_2$, or
- x_1 is bound at the same place in F_1 as on which x_2 is bound in F_2.

Substitution

We write $u[x := t]$ for the substitution that replaces variable x by term t in term u.

- If u is a variable or constant, and $u \neq x$, then $u[x := t] = u$.
- If $u = x$, then $u[x := t]$ is t.
- If $n > 0$, then $f(t_1, \ldots, t_n)[x := t]$ equals $f(t_1[x := t], \ldots, t_n[x := t])$.

For an atom, we define:

- $p(t_1, \ldots, t_n)[x := t] = p(t_1[x := t], \ldots, t_n[x := t])$.
- $(t_1 = t_2)[x := t] = t_1[x := t] = t_2[x := t]$.

Substitution in a Formula

- $\bot[x := t] = \bot, \quad \top[x := t] = \top.$
- $(\neg F)[x := t] = \neg(F[x := t]).$
- $(F_1 \vee F_2)[x := t] = F_1[x := t] \vee F_2[x := t]$. The cases for $\wedge, \rightarrow,$ and $\leftarrow$ are analogous.
- If x is not free in $\forall y\ F$, or y is not free in t, then $(\forall y\ F)[x := t] = \forall y\ (F[x := t]).$
- If x is free in $\forall y\ F$ and y is free in t, then let $\forall y'\ F'$ be an α-variant of $\forall y\ F$, s.t. y' is not free in t. Proceed as in the previous case.
- Cases for $\exists y\ F$ are defined analogeously.

Rules for Equality

The replacement rule (in the definition of axiom) can be made precise by means of substitution:

$\Gamma,\ t \approx u,\ A[t] \vdash$ is an axiom if $t > u$ and $\Gamma,\ t \approx u,\ A[u] \vdash$ is an axiom.

There must exist a formula A, and a variable α that is not free in α, s.t. $A[t] = A[\alpha := t], \quad A[u] = A[\alpha := u]$.

Preconditions in the Quantifier Rules

1. Can you give an example of a wrong derivation, in case that the condition 'y is not free in Γ, Δ' is dropped from $\forall$-right?

2. Suppose that substitution would not take into account the possible capture of variables. Can you give an example of a wrong derivation using rule $\exists$-right?

1. For example:

$$\dfrac{\dfrac{P(x) \vdash P(x)}{\exists x\ P(x) \vdash P(x)}}{\exists x\ P(x) \vdash \forall x\ P(x)}$$

2. For example:

$$\dfrac{\vdash \forall y\ (y \approx y)}{\vdash \exists x \forall y\ (y \approx x)}$$

   If we ignore capture, we can substitute
   $(\ \forall y\ (y \approx x)\ )[x := y] = \forall y\ (y \approx y)$.

Drinker Paradox

A good example of a formula whose proof with cut is more clear than its proof without cut, is the drinker paradox $\exists x\ (D(x) \to \forall x\ D(x))$. Either everyone drinks, then x can be an arbitrary person, or somebody does not drink, then x can be one of the non-drinkers. It is easy to prove:

$$\vdash \exists x\ \neg D(x),\ \forall x\ D(x)$$

$$\exists x\ \neg D(x) \vdash \exists x\ (D(x) \to \forall x\ D(x))$$

$$\forall x\ D(x) \vdash \exists x\ (D(x) \to \forall x\ D(x))$$

Using two applications of cut, one can obtain

$$\vdash \exists x\ (D(x) \to \forall x\ D(x)).$$

As far as I know, there is no intuitive proof without cut.