The need for a richer language

Predicate logic as a formal language
- terms — variables, functions
- formulas — predicates, quantifiers
- free and bound variables
- substitution

Proof theory of predicate logic
- Natural deduction rules
Propositional Logic:

- Study of declarative sentences, statements about the world which can be given a truth value
- Dealt very well with sentence components like: not, and, or, if \( \cdots \) then \( \cdots \)
- **Limitations:** cannot deal with modifiers like there exists, all, among, only.

**Example:** “Every student is younger than some instructor.”

- We could identify the entire phrase with the propositional symbol \( p \).
- However, the phrase has a finer logical structure. It is a statement about the following properties:
  - being a student
  - being an instructor
  - being younger than somebody else
Properties are expressed by predicates. \( S, I, Y \) are *predicates*.

\[ S(andy): \text{Andy is a student.} \]
\[ I(paul): \text{Paul is an instructor.} \]
\[ Y(andy,paul): \text{Andy is younger than Paul.} \]

Variables are placeholders for concrete values.

\[ S(x): x \text{ is a student.} \]
\[ I(x): x \text{ is an instructor.} \]
\[ Y(x,y): x \text{ is younger than } y. \]

Quantifiers make possible encoding the phrase:

\[ \text{“Every student is younger than some instructor.”} \]

Two quantifiers: \( \forall \) — *forall*, and \( \exists \) — *exists*.

Encoding of the above sentence:

\[ \forall x (S(x) \rightarrow (\exists y (I(y) \land Y(x,y)))) \]
“No books are gaseous. Dictionaries are books. Therefore, no dictionary is gaseous.”

We denote: 

\( B(x) \): \( x \) is a book
\( G(x) \): \( x \) is gaseous
\( D(x) \): \( x \) is a dictionary

\[ \neg \exists x (B(x) \land G(x)), \forall x (D(x) \rightarrow B(x)) \]

“Every child is younger than his mother”

We denote: 

\( C(x) \): \( x \) is a child
\( M(x,y) \): \( x \) is \( y \)’s mother

\[ \forall x \forall y (C(x) \land M(x,y) \rightarrow Y(x,y)) \]

Denote \( m(x) \): mother of \( x \)

\[ \forall x (C(x) \rightarrow Y(x, m(x))) \]

Using the function \( m \) to encode the “mother of” relationship is more appropriate, since every person has a unique mother.
“Andy and Paul have the same maternal grandmother”

\[ \forall x \forall y \forall u \forall v (M(x, y) \land M(y, a) \land M(u, v) \land M(v, p) \rightarrow x = u) \]

We have introduced a new, special predicate: *equality*.

Alternative representation:

\[ m(m(a)) = m(m(p)) \]

Consider the relationship \( B(x, y) \): \( x \) is the brother of \( y \). This relationship must be encoded as a predicate, since a person may have more than one brother.
Two sorts of “things” in a predicate formula:

- Objects such as \( a \) (Andy) and \( p \) (Paul). Function symbols also refer to objects. These are modeled by **terms**.

- Expressions that can be given truth values. These are modeled by **formulas**.

A predicate vocabulary consists of 3 sets:

- **Predicate symbols** \( \mathcal{P} \);
- **Function symbols** \( \mathcal{F} \);
- **Constants** \( C \).

Elements of the formal language of predicate logic:

- Terms
- Formulas
- Free and bound variables
- Substitution
**Definition:** *Terms* are defined as follows:

- Any variable is a term;
- Any constant in $C$ is a term;
- If $t_1, \ldots, t_n$ are terms and $f \in F$ has arity $n$, then $f(t_1, \ldots, t_n)$ is a term;
- Nothing else is a term.

**Backus-Naur definition:** $t ::= x | c | f(t, \ldots, t)$ where $x$ represents variables, $c$ represents constants in $C$, and $f$ represents function in $F$ with arity $n$.

**Remarks:**

- The first building blocks of terms are constants and variables.
- More complex terms are built from function symbols using previously built terms.
- The notion of terms is independent on the sets $C$ and $F$. 
**Definition:** We define the set of formulas over $(\mathcal{F}, \mathcal{P})$ inductively, using the already defined set of terms over $\mathcal{F}$.

- If $P$ is a predicate with $n \geq 1$ arguments, and $t_1, \ldots, t_n$ are terms over $\mathcal{F}$, then $P(t_1, \ldots, t_n)$ is a formula.
- If $\Phi$ is a formula, then so is $\neg \Phi$.
- If $\Phi$ and $\Psi$ are formulas, then so are $\Phi \land \Psi$, $\Phi \lor \Psi$, $\Phi \rightarrow \Psi$.
- If $\Phi$ is a formula and $x$ is a variable, then $\forall x \Phi$ and $\exists x \Phi$ are formulas.
- Nothing else is a formula.

**BNF definition:**

\[
\Phi ::= P(t_1, \ldots, t_n) | (\neg \Phi) | (\Phi \land \Phi) | (\Phi \lor \Phi) | (\Phi \rightarrow \Phi) | (\forall x \Phi) | (\exists x \Phi)
\]

where $P$ is a predicate of arity $n$, $t_i$ are terms, $i \in \{1, \ldots, n\}$, $x$ is a variable.

**Convention:** We retain the usual binding priorities of the connectives $\neg, \land, \lor, \rightarrow$. We add that $\forall x$ and $\exists x$ bind like $\neg$. 
Consider translating the sentence:

“Every son of my father is my brother”

Two alternatives:

- **“Father of”** relationship encoded as a predicate.
  
  \[ S(x,y): x \text{ is the son of } y. \]
  \[ F(x,y): x \text{ is the father of } y. \]
  \[ B(x,y): x \text{ is the brother of } y. \]
  
  \[ m: \text{ constant, denoting “myself”}. \]

  **Translation:** \[ \forall x \forall y (F(x,m) \land S(y,x) \rightarrow B(y,m)) \]

- **“Father of”** relationship encoded as a function.

  \[ f(x): \text{ father of } x. \]

  **Translation:** \[ \forall x (S(x,f(m)) \rightarrow B(x,m)) \]
**Definition:** Let $\Phi$ be a formula in predicate logic. An occurrence of $x$ in $\Phi$ is **free in $\Phi$** if it is a leaf node in the parse tree of $\Phi$ such that there is no path upwards from that node $x$ to a node $\forall x$ or $\exists x$. Otherwise, that occurrence $x$ is called **bound**. For $\forall x \Phi$, we say that $\Phi$—minus any of its sub-formulas $\exists x \Psi$, or $\forall x \Psi$—is the scope of $\forall x$, respectively $\exists x$.

**Formula:**

$$\forall x ((P(x) \to Q(x)) \land S(x, y))$$

$x$ is bound.

$y$ is free.
Examples of Free and Bound Variables

Formula: \((\forall x (P(x) \land Q(x))) \rightarrow (\neg P(x) \lor Q(y))\)

Parse tree:
Variables are placeholders, so we must have means of replacing them with more concrete information.

**Definition:** Given a variable \( x \), a term \( t \), and a formula \( \Phi \), we define \( \Phi[t/x] \) to be the formula obtained by replacing each free occurrence of variable \( x \) in \( \Phi \) with \( t \).

\[
((\forall x (P(x) \land Q(x)))) \rightarrow (\neg P(x) \lor Q(y))[f(x,y)/x]
\]

is

\[
(\forall x (P(x) \land Q(x))) \rightarrow (\neg P(f(x,y)) \lor Q(y))
\]
**Definition:** Given a term $t$, a variable $x$, and a formula $\Phi$, we say that $t$ is free for $x$ in $\Phi$ if no free $x$ leaf in $\Phi$ occurs in the scope of $\forall y$ or $\exists y$, for every variable $y$ occurring in $t$.

**Remark:** If $t$ is not free for $x$ in $\Phi$, then the substitution $\Phi[t/x]$ has unwanted effects.

**Example:**

$$(S(x) \land (\forall y (P(x) \rightarrow Q(y))))[y/x] \text{ is } S(y) \land (\forall y (P(y) \rightarrow Q(y)))$$

Avoid this by renaming $\forall y$ into $\forall z$.

$$(S(x) \land (\forall z (P(x) \rightarrow Q(z))))[y/x] \text{ is } S(y) \land (\forall z (P(y) \rightarrow Q(z)))$$
Natural deduction rules for propositional logic are still valid

Natural deduction rules for predicate logic:

– proof rules from propositional logic;
– proof rules for equality;
– proof rules for universal quantification;
– proof rules for existential quantification.

Quantifier equivalences
Proof Rules for Equality

\[
\begin{align*}
\text{Convention: } & \text{ When we write a substitution in the form } \Phi[t/x], \text{ we implicitly assume that } t \text{ is free for } x \text{ in } \Phi. \\
\text{Proof example: } & \quad x + 1 = 1 + x, (x + 1 > 1) \rightarrow (x + 1 > 0) \vdash (1 + x > 1) \rightarrow (1 + x > 0) \\
1 & \quad x + 1 = 1 + x \quad \text{premise} \\
2 & \quad (x + 1 > 1) \rightarrow (x + 1 > 0) \quad \text{premise} \\
3 & \quad (1 + x > 1) \rightarrow (1 + x > 0) \quad =\text{e } 1,2
\end{align*}
\]
Proof Rules for Universal Quantification

\[
\begin{align*}
\forall x \Phi & \quad \forall x e \\
\Phi[t/x] & \quad \Phi[x_0/x] \\
\end{align*}
\]

\[
\begin{align*}
\forall x \Phi & \quad \forall x i \\
\end{align*}
\]

Proof examples:

\[
\begin{align*}
\forall x (P(x) \rightarrow Q(x)), \forall x P(x) & \vdash \forall x Q(x) & P(t), \forall x (P(x) \rightarrow \neg Q(x)) & \vdash \neg Q(t) \\
1 & \forall x (P(x) \rightarrow Q(x)) & \text{premise} & 1 & P(t) & \text{premise} \\
2 & \forall x P(x) & \text{premise} & 2 & \forall x (P(x) \rightarrow \neg Q(x)) & \text{premise} \\
3 & x_0 & P(x_0) \rightarrow Q(x_0) & \forall x e 1 & p(t) \rightarrow \neg Q(t) & \forall x e 2 \\
4 & P(x_0) & \forall x e 2 & 4 & \neg Q(t) & \rightarrow e 3,1 \\
5 & Q(x_0) & \rightarrow e 3,4 \\
6 & \forall x Q(x) & \forall x i 3-5 \\
\end{align*}
\]
Proof Rules for Existential Quantification

\[ \Phi[t/x] \quad \exists x \Phi \]
\[ \exists x \Phi \]
\[ \exists x \Phi \]
\[ \exists x \Phi \]

\[ \exists x \Phi \]
\[ \exists x \Phi \]

Proof examples:

\[ \forall x (P(x) \rightarrow Q(x)), \exists x P(x) \vdash \exists x Q(x) \]

1. \[ \forall x \Phi \] \quad premise
2. \[ \Phi[x/x] \quad \forall x e 1 \]
3. \[ \exists x \Phi \quad \exists x i 2 \]
4. \[ \forall x (P(x) \rightarrow Q(x)) \]
5. \[ \exists x P(x) \] \quad premise
6. \[ \exists x Q(x) \]
7. \[ \exists x Q(x) \]

Side condition: \( x_0 \) doesn’t occur in \( \chi \)
Another Example

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

1. $\exists x P(x)$  
   premise
2. $\forall x \forall y (P(x) \rightarrow Q(y))$  
   premise
3. $y_0$
4. $x_0$  
   $P(x_0)$  
   assumption
5. $\forall y (P(x_0) \rightarrow Q(y))$  
   $\forall x \in 2$
6. $P(x_0) \rightarrow Q(y_0)$  
   $\forall y \in 2$
7. $Q(y_0)$  
   $\rightarrow 6,4$
8. $Q(y_0)$  
   $\exists x \in 1,4-7$
9. $\forall y Q(y)$  
   $\forall y \in 3-8$