Lecture 26

Predicate Logic
Predicate Logic as a Formal Language

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** $\mathcal{C}$.

Each predicate and function symbol comes with a fixed arity (i.e., number of arguments).
Terms

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 \( f \in \mathcal{F} \) 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 \( \mathcal{F} \) 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.
Terms

Let $\mathcal{F}$ be $\{d, f, g\}$, where $d$ is a constant, $f$ a function symbol with two arguments and $g$ a function symbol with three arguments.

Which ones are terms over $\mathcal{F}$:

1) $g(d, d)$
2) $g(x, f(y, z), d)$
3) $g(x, h(y, z), d)$
4) $g(g(f(d, x), g(f(d, x), y, f(y, d)), f(d, d)), f(g(d, d, x), d), z)$
Terms

Let $\mathcal{F}$ be $\{d, f, g\}$, where $d$ is a constant, $f$ a function symbol with two arguments and $g$ a function symbol with three arguments.

Which ones are terms over $\mathcal{F}$:

- **no** 1) $g(d, d)$
- **yes** 2) $g(x, f(y, z), d)$
- **no** 3) $g(x, h(y, z), d)$
- **yes** 4) $g(g(f(d, x), g(f(d, x), y, f(y, d)), f(d, d)), f(g(d, d, x), d), z)$
Formulas

**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:**

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

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 \(\neg\).
Formulas

Let $m$ be a constant, $f$ a function symbol with one argument and $S$ and $B$ two predicate symbols, each with two arguments:

Which ones are formulas:

$S(m, x)$
$B(m, f(m))$
$f(m)$
$B(B(m, x), y)$
$(B(x, y) \rightarrow (\exists z \ S(z, y)))$
$(S(x, y) \rightarrow S(y, f(f(x))))$
Formulas

Let $m$ be a constant, $f$ a function symbol with one argument and $S$ and $B$ two predicate symbols, each with two arguments:

Which ones are formulas:

- Yes $S(m, x)$
- Yes $B(m, f(m))$
- No $f(m)$
- No $B(B(m, x), y)$
- Yes $(B(x, y) \rightarrow (\exists z S(z, y)))$
- Yes $(S(x, y) \rightarrow S(y, f(f(x))))$
Parse Tree

\[ \forall x ((P(x) \rightarrow Q(x)) \land S(x, y)) \]
Scope of Variables

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

How about

$$\forall x \text{ in } \forall x (P(x) \rightarrow \exists x Q(x))$$
Bound and Free Variables

\[ \forall x \; ((P(x) \rightarrow Q(x)) \land S(x, y)) \]
Bound and Free Variables

\( \forall x ((P(x) \rightarrow Q(x)) \land S(x, y)) \)

Q: Is it possible to have the same variable being both bound and free?
Bound and Free Variables

\[ \forall x \land P \lor \neg Q \land P \lor y \]

- \(\forall x\) is a bound variable.
- \(P\) and \(\neg Q\) have \(x\) as a bound variable.
- \(P\) and \(y\) have \(x\) as a free variable.
Substitutions

**Definition 2.7** 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 $t$. 
Substitutions -- Example

\[\forall x \phi[f(x, y)/x] \rightarrow \text{Substitute } x \text{ with } f(x, y)\]
Substitutions -- Example

\[ \forall x \]

\[ \land \]

\[ \rightarrow \]

\[ S \]

\[ P \]

\[ Q \]

\[ x \]

\[ y \]

\[ \phi[f(x, y)/x] \quad \rightarrow \quad \text{Substitute } x \text{ with } f(x, y) \]

Q: Any issues with substitutions?
Free Substitutions

**Definition 2.8** 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 any variable \( y \) occurring in \( t \).
Free Substitutions -- Example

The term $f(y, y)$ is not free for $x$ in this formula.
Example

\[ \exists x (P(y, z) \land (\forall y (\neg Q(y, x) \lor P(y, z)))) \]

1. Draw parse tree
2. Identify all bound and free variables
3. What is the scope of \[ \exists x \]
4. What is its scope in \[ \exists x (P(y, z) \land (\forall y (\neg Q(x, x) \lor P(x, z)))) \]
5. Which of \( w \), \( f(x) \) and \( g(y, z) \) are free for \( x \)?
6. Which of \( w \), \( f(x) \) and \( g(y, z) \) are free for \( y \)?
Example

$$\exists x (P(y, z) \land (\forall y (\neg Q(y, x) \lor P(y, z))))$$

1. Draw parse tree
2. Identify all bound and free variables
3. What is the scope of $$\exists x$$
4. What is its scope in $$\exists x (P(y, z) \land (\forall y (\neg Q(x, x) \lor P(x, z))))$$
5. Which of w, f(x) and g(y, z) are free for x?
6. Which of w, f(x) and g(y, z) are free for y?

3. Entire tree
4. P(y, z)
5. All of them are free for x
6. w and g(y, z) are free for y