First-Order Logic

Yu “Tony” Zhang, Ph.D.
Assistant Professor
Arizona State University
Propositional logic

Derive an inescapable conclusion using all of these:

a. All babies are illogical
b. Nobody is despised who can manage a crocodile
c. Illogical persons are despised

- B: it is a baby
- L: it is logical
- M: it can manage a crocodile
- D: it is despised

\[ a) \ B \rightarrow \neg L \]
\[ b) \ M \rightarrow \neg D \]
\[ c) \ \neg L \rightarrow D \]
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 structure. It is a statement about the following properties:
  - Being a student
  - Being an instructor
  - Being younger than somebody else
The Need For a Richer Language

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, ..., then
- Limitations: Cannot deal with modifiers like there exists, all, among, only.
Example: "Every student is younger than some instructor."

- Relationships are expressed by predicates:
  - \(S(\text{andy})\): Andy is a student
  - \(I(\text{paul})\): Paul is an instructor
  - \(Y(\text{andy}, \text{paul})\): Andy is younger than Paul
Example: “Every student is younger than some instructor.”

- **Variables** are placeholders for concrete values
  - $S(x)$: $x$ is a student
  - $I(x)$: $x$ is an instructor

- **Quantifiers** to express “every”, “some”, etc.:
  - Two quantifiers: $\forall$ -- forall, and $\exists$ -- exists

Encoding of the above sentence:

$$\forall x (S(x) \rightarrow (\exists y (I(y) \land Y(x,y))))$$
Dealing with Quantifiers

Formulas under quantifiers:

- $\exists x \Phi$ We try to find some instance of $x$ (some concrete value) such that $\Phi$ holds for that particular instance of $x$. If this succeeds, then $\exists x \Phi$ evaluates to $t$; otherwise (i.e. there is no concrete value of $x$ that realizes $\Phi$) the formula evaluates to $f$.

- $\forall x \Phi$ We try to show that for all possible instances of $x$, $\Phi$ evaluates to $t$. If this is successful, $\forall x \Phi$ evaluates to $t$; otherwise (i.e. if there exists some instance of $x$ that does not realize $\Phi$) the formula evaluates to $f$. 
Not all birds can fly
- \( B(x) \) : \( x \) is a bird
- \( F(x) \): \( x \) can fly

Encoding of the above sentence:
- \( \neg(\forall x (B(x) \rightarrow F(x))) \)
- \( \exists x (B(x) \land \neg F(x)) \)
Example: “Every son of my father is my brother.”

- Predicates $S$, $F$, $B$:
  - $S(x, y)$: $x$ is the son of $y$.
  - $F(x, y)$: $x$ is the father of $y$.
  - $B(x, y)$: $x$ is the brother of $y$.
  - $m$: constant, denoting “myself”.

- Translation:
  - $\forall x \forall y (F(x, m) \land S(y, x) \rightarrow B(y, m))$
Example: “Every son of my father is my brother.”

- Predicates $S, F, B$:
  - $S(x, y) : x$ is the son of $y$.
  - $f(x) :$ father of $x$ -- $f$ is a function
  - $B(x, y): x$ is the brother of $y$.
  - $m$: constant, denoting “myself”.

- Translation:
  - $\forall x (S(x, f(m)) \rightarrow B(x, m))$
Example: “Andy and Paul have the same maternal grandmother.”

- Predicates, $M$,:
  - $M(x, y)$: $x$ is mother of $y$
  - $a$: Andy
  - $p$: Paul

- Translation
  - $\forall x \forall y \forall u \forall v (M(x, y) \land M(y, a) \land M(u, v) \land M(v, p) \rightarrow x = u)$

- Translation with a function: $m(x)$
  - $m(m(a)) = m(m(p))$
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 $\mathcal{C}$
Definitions: **Terms** are defined as follows:

- Any variable is a term;
- Any constant in $\mathcal{C}$ is a term;
- If $t_1, ..., t_n$ are terms and $f \in \mathcal{F}$ has arity $n$, then $f(t_1, ..., t_n)$ is a term;
- Nothing else is a term.
Backus Normal Form (BNF) Definition:

- $t ::= x | c | f(t, ..., t)$ where $x$ represents variables, $c$ represents constants in $\mathcal{C}$, and $f$ represents function

Remarks:

• The first building blocks are constants and variables
• More complex terms are built from function symbols
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) $g(g(f(d, x), g(f(d, x), f(y, d)), f(d, d)), f(g(d, d, x), d), z)$
Formulas

Definition: We define the set of formulas over \((\mathcal{F}, \mathcal{P})\) inductively, using 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 a 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.
Formulas

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.
- Remarks:
  - 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:

\[
\begin{align*}
S(m, x) \\
B(m, f(m)) \\
f(m) \\
B(B(m, x), y) \\
(B(x, y) \rightarrow (\exists z \, S(z, y))) \\
(S(x, y) \rightarrow S(y, f(f(x))))
\end{align*}
\]
Scope of Variables

Parse tree:

$$\forall x ((P(x) \rightarrow Q(x)) \land S(x, y))$$
- Bound and free variables

How about

$$\forall x \text{ in } \forall x (P(x)) \rightarrow \exists x Q(x))$$

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

\[ \rightarrow \]

\[ \forall x \]

\[ \land \]

\[ P \]
\[ x \] \quad \text{bound}

\[ Q \]
\[ x \] \quad \text{bound}

\[ \neg \]

\[ P \]
\[ x \] \quad \text{free}

\[ Q \]
\[ y \] \quad \text{free}

\[ \lor \]
Summary

- Introduction to first-order logic
  - Predicates, variables and quantifiers
  - Functions
  - Terms
  - Formulas

- Parse of first-order logic formulas
  - Scope of variables