First-Order Logic

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

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*.

The Need For a Richer Language

- **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

Predicates, Variables, and Quantifiers

- **Example:** ***"Every student is younger than some instructor."***
  - Relationships are expressed by **predicates**:
    - S(andy): Andy is a student
    - I(paul): Paul is an instructor
    - Y(andy, paul): Andy is younger than Paul

Predicates, Variables, and Quantifiers

- **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, and ∃ -- exists

Encoding of the above sentence:

$$\forall x(S(x) \rightarrow (\exists y(I(y) \wedge Y(x,y))))$$

Dealing with Quantifiers

- **Formulas under quantifiers:**
  - $\exists x \Phi$ We try to find some instance of x (some concrete value) such that Φ 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 Φ) the formula evaluates to f.
  - $\forall x \Phi$ We try to show that for all possible instances of x, Φ 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 Φ) the formula evaluates to f.

Predicates, Variables, and Quantifiers

- **Not all birds can fly**
  - *B(x,y)* : *x* is a bird
  - *F(x): x* can fly

- **Encoding of the above sentence:**
  - $\neg(\forall x(B(x) \rightarrow F(x)))$
  - $\exists x(B(x) \wedge \neg F(x))$

Function

- **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:
    - $\forall x \forall y (F(x, m) \land S(y, x) \rightarrow B(y, m))$

Function

- **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:
    - $\forall x(S(x, f(m)) \rightarrow B(x, m))$

Function

- **Example:** ***"Every child is younger than its mother."***
  - Predicates, *C, M, Y:*
    - $C(x)$: *x* is a child
    - $M(x, y)$: *x* is mother of *y*
    - $Y(x, y)$: *x* is younger than *y*
  - Translation
    - $\forall x \forall y (C(x) \wedge M(y, x) \rightarrow Y(x, y))$
  - Translation with a function: ***m(x)***
    - $\forall x (C(x) \rightarrow Y(x, m(x)))$

Function

- Example: ***"Andy and Paul have the same maternal grandmother."***
  - Predicates, *M,:*
    - $M(x, y)$: *x* is mother of *y*
    - *a: Andy*
    - *p: Paul*
  - Translation
    - $\forall x \forall y \forall u \forall v (M(x,y) \wedge M(y,a) \wedge M(u,v) \wedge M(v,p) \rightarrow x = u)$
  - 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 $\mathcal{C}$

Terms

- **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.

Terms

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

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 $t_1, ..., t_n$ are terms over $\mathcal{F}$, then $P(t_1, ..., t_n)$ is a formula.
  - If Φ is a formula, then so is $\neg\Phi$
  - If Φ and Ψ are a formulas, then so are $\Phi \wedge \Psi, \Phi \vee \Psi, \Phi \rightarrow \Psi$
  - If Φ 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:**

  $$\Phi ::= P(t_1, ..., t_n) | (\neg\Phi) | (\Phi \wedge \Phi) | (\Phi \vee \Phi) | (\Phi \rightarrow \Phi) | (\forall x \Phi) | (\exists x \Phi)$$

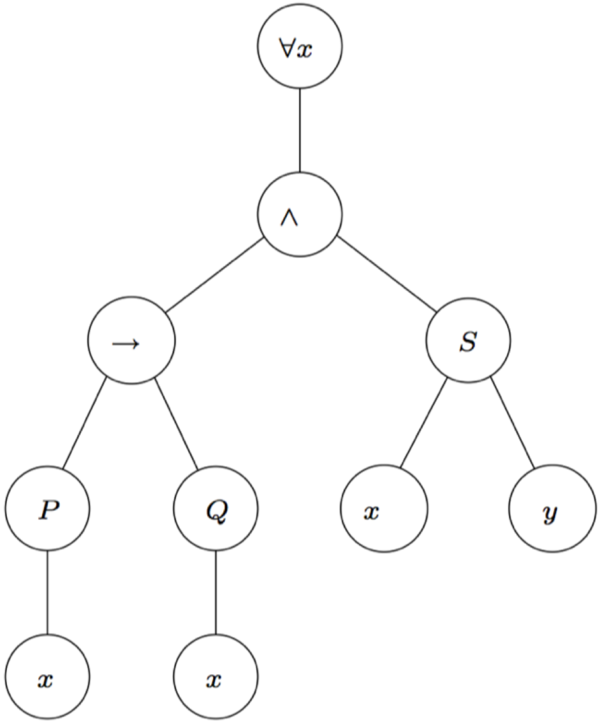
  - where P is a predicate of arity n, t_i are terms, $i \in \{1, ..., n\}, x$ is a variable.
  - Remarks:
    - Convention: We retain the usual binding priorities of the connectives $\neg, \wedge, \vee, \rightarrow$
    - We add that $\forall x$ and $\exists x$ bind like $\neg$

Scope of Variables

- **Parse tree:**

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

  – Bound and free variables



First-Order logic

- **Reasoning with first-order logic:**
  - In addition to proof rules in propositional logic, also have proof rules with quantifiers
  - Sound and complete $\Gamma \vdash \psi$ = $\Gamma \vDash \psi$
  - Undecidability of first-order logic

Summary

- **Introduction to first-order logic**
  - Predicates, variables and quantifiers
  - Functions
  - Terms
  - Formulas
- **Parse of first-order logic formulas**
  - Scope of variables