Predicate Logic: Syntax

Alice Gao

Lecture 12

Outline

Learning goals

By the end of this lecture, you should be able to

- Define the set of terms inductively.
- Define the set of formulas inductively.
- Determine whether a variable in a formula is free or bound.
- Prove properties of terms and formulas by structural induction.
- Draw the parse tree of a formula.

The Language of Predicate Logic

- Domain: a non-empty set of objects.
- Individuals: concrete objects in the domain.
- Variables: placeholders for concrete objects in the domain.
- Functions: takes objects in the domain as arguments and returns an object of the domain.
- Relations: takes objects in the domain as arguments and returns true or false. They describe properties of objects or relationships between objects.
- Quantifiers: for how many objects in the domain is the statement true?

Outline

Predicate Language L

Eight classes of symbols:

- Individual symbols: a, b, c.
- Relation symbols: F, G, H.
  A special equality symbol $\approx$
- Function symbols: f, g, h.
- Free variable symbols: u, v, w.
- Bound variable symbols: x, y, z.
- Connective symbols: $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$.
- Quantifier symbol: $\forall$, $\exists$.
- Punctuation symbols: '(', ')', and ','

Free and Bound Variables

In a formula $\forall x \ A(x)$ or $\exists x \ A(x)$,
the scope of a quantifier is the formula $A(x)$.

A quantifier binds its variable within its scope.

An occurrence of a variable in a formula

- is bound if it lies in the scope of some quantifier of the same variable.
- is free, otherwise.

Outline

Two Kinds of Expressions

Two kinds of expressions:

- A term refers to an object in the domain.
- A formula evaluates to 1 or 0.

Terms

The set of terms $\text{Term}(L)$ is defined below:

1. An individual symbol a standing alone is a term.
2. A free variable symbol u standing alone is a term.
3. If $t_1, \ldots, t_n$ are terms and f is an n-ary function symbol, then $f(t_1, \ldots, t_n)$ is a term.
4. Nothing else is a term.

Examples of Terms

Terms:

- $a, b, c,\ u, v, w$
- $f(b),\ g(a, f(b)),\ g(u, b),\ f(g(f(u), b))$

A term with no free variable symbols is called a closed term.
Which one(s) of the above are closed terms?

CQ: Which expressions are terms?

Which of the following expressions is a term?
If there are multiple correct answers, choose your favourite one.

(A) w

(B) $g(a, u)$

(C) $F(f(u, v), a)$

(D) $f(u, g(v, w), a)$

(E) $g(u, f(v, w), a)$

Individual symbols: a
Relation symbols: F is a binary relation symbol.
Function symbols: f is a binary function symbol and g is a 3-ary function symbol.
Free variable symbols: u, v, w.

Defining the set of terms inductively

The set of terms can be inductively defined as follows:

- The domain set X:
  The set of finite sequences of symbols of L
- The core set C:
  The set of all individual symbols and free variable symbols
- The set of operations P:
  The set of all function symbols

Structural induction on terms

Theorem: Every term has a property P.

Proof by structural induction:

- Base cases:
  The term is an individual symbol.
  The term is a free variable symbol.
- Inductive cases:
  The term is $f(t_1, \ldots, t_n)$ where f is an n-ary function and $t_1, ..., t_n$ are terms.
  Induction hypotheses: Assume that $t_1, \ldots, t_n$ all have the property P.
  We need to show that $f(t_1, \ldots, t_n)$ has the property P.

Outline

Atomic Formulas

The set of atomic formulas Atom(L) is defined below:

- If F is an n-ary relation symbol and $t_1, \ldots, t_n$ $(n \geq 1)$ are terms, then $F(t_1, \ldots, t_n)$ is an atomic formula.
- If t_1, t_2 are terms, then $\approx (t_1, t_2)$ is an atomic formula.
- Nothing else is an atomic formula.

Examples of Atomic Formulas

Terms:

- $a, b, c,\ u, v, w$
- $f(b),\ g(a, f(b)),\ g(u, b),\ f(g(f(u), b))$

Atomic formulas:

- $F(a, u, f(b), f(w), g(v, f(a)))$
- $\approx (b, w)$

Well-Formed Formulas

The set of well-formed formulas Form(L) is defined below:

1. An atomic formula is a well-formed formula.
2. If A is a well-formed formula, then $(\neg A)$ is a well-formed formula.
3. If A and B are well-formed formulas and $\star$ is one of $\wedge$, $\vee$, $\to$, and $\leftrightarrow$, then $(A \star B)$ is a well-formed formula.
4. If $A(u)$ is a well-formed formula and x does not occur in $A(u)$, then $\forall x\, A(x)$ and $\exists x\, A(x)$ are well-formed formulas.
5. Nothing else is a well-formed formula.

Explaining Case 4 of Formulas

If $A(u)$ is a well-formed formula and x does not occur in $A(u)$, then $\forall x\, A(x)$ and $\exists x\, A(x)$ are well-formed formulas.

- $A(u)$ is a well-formed formula where u is a free variable in the formula. We want to quantify u.
- In order to do so, we need to choose a symbol for a bound variable, e.g. x. We need to make sure that our choice of the bound variable symbol does not already occur in $A(u)$.

Examples for Case 4

- We are allowed to generate the formula $\forall y F(y, y)$.
  Start with $F(u, u)$. If we quantify u by replacing it with y, we get $\forall y F(y, y)$.

- We are not allowed to generate the formula $\exists y \forall y F(y, y)$.
  Start with $\forall y F(y, y)$. If we want to add the $\exists$ quantifier, we will need to choose a bound variable symbol that is not y because y already appears in $\forall y\, F(y, y)$. So, there is no way for us to generate $\exists y \forall y F(y, y)$.

- We are allowed to generate the formula $\exists x G(x) \vee \forall x H(x)$.
  Start with $G(u)$ and $H(v)$ separately. We can quantify u by replacing it with x since x does not appear in $G(u)$). We get $\exists x G(x)$. We can quantify v by replacing it with x since x does not appear in $H(v)$. We get $\forall x H(x)$. Connecting the two formulas using $\vee$, we get $\exists x G(x) \vee \forall x H(x)$.

Examples of Formulas

Well-Formed Formulas:

- $F(a,b)$, $\forall y\, F(a,y)$, $\exists x \forall y\, F(x,y)$
- $F(u,v)$, $\exists y\, F(u,y)$

A formula with no free variable symbols is called a closed formula or a sentence.
Which formulas above are closed formulas?

Determine whether a formula is well-formed

Which of the following is a well-formed formula?

(A) $f(u) \to F(u, v)$

(B) $\forall x \; F(m, f(x))$

(C) $F(u, v) \to G(G(u))$

(D) $G(m, f(m))$

(E) $F(m, f(G(u, v)))$

Individual symbols: m.
Free Variable Symbols: u, v.
Bound Variable symbols: x.
Relation symbols: F and G are binary relation symbols.
Function symbols: f is a unary function.

Defining the set of formulas inductively

The set of formulas can be inductively defined as follows:

- The domain set X:
  The set of finite sequences of symbols of L
- The core set C:
  The set of all atomic formulas.
- The set of operations P:
  $f_1(x) = (\neg x)$
  $f_2(x, y) = (x * y)$ where $*$ is one of $\wedge$, $\vee$, $\rightarrow$, and $\leftrightarrow$.
  $f_3(A(u)) = \forall x\, A(x), f_4(A(u)) = \exists x\, A(x)$ where x does not occur in $A(u)$.

Structural induction on formulas

Theorem: Every formula has a property P.

Proof by structural induction:

- Base cases:
  The formula is an atomic formula.
- Inductive cases:
  The formula is $(\neg A)$ where A is a formula.
  The formula is $(A * B)$ where A and B are formulas and $*$ is a binary connective.
  The formula is $\forall x\, A(x)$ and $\exists x\, A(x)$ where $A(u)$ is a formula and x does not occur in $A(u)$.

Comparing the Definitions of Well-Formed Formulas

Let's compare the set of predicate formulas to the set of propositional formulas.

Questions to think about:

- Which parts of the two definitions are the same?
  The cases for negation and binary connectives are the same.
- Which parts of the two definitions are different?
  Atomic formulas are different.
  - Atomic propositional formulas are propositional variables.
  - Atomic predicate formulas are relations applied to terms.

  Predicate formulas have one additional case for quantifiers.

Outline

Parse Trees of Predicate Formulas

- The leaves are atomic formulas.
- Every quantifier has exactly one child (namely the formula which is its scope).

Example: $\forall x(F(b) \to \exists y(\forall z\, G(y,z) \vee H(f(u),x,y)))$

Parse tree

Example: $\forall x(F(b) \to \exists y(\forall z\, G(y,z) \lor H(f(u),x,y)))$

Parse tree:

- $\forall x(F(b) \to \exists y(\forall z\, G(y,z) \lor H(f(u),x,y)))$
  - $F(b) \to \exists y(\forall z\, G(y,z) \lor H(f(u),v,y))$
    - $F(b)$
    - $\exists y(\forall z\, G(y,z) \lor H(f(u),v,y))$
      - $\forall z\, G(w,z) \lor H(f(u),v,w)$
        - $\forall z\, G(w,z)$
          - $G(w,u_1)$
        - $H(f(u),v,w)$

A few notes on parse trees

1. While constructing the parse tree, when removing a quantifier, we change the bound variable symbol to one of the free variable symbols that hasn't appeared in the parse tree.
   For example, When removing $\forall x$, we changed x to v. When removing $\exists y$, we changed y to w.
2. Th quantifiers have higher precedence than any other connective. Each quantifier modifies the formula that is immediately after it.
   For example, $\forall z$ modifies $G(w, z)$ instead of $G(w, z) \vee H(f(u), v, w)$.

Revisiting the learning goals

By the end of this lecture, you should be able to

- Define the set of terms inductively.
- Define the set of formulas inductively.
- Determine whether a variable in a formula is free or bound.
- Prove properties of terms and formulas by structural induction.
- Draw the parse tree of a formula.