e.g. **Example:** Symbolize the following propositions:

(1) Rabbits run faster than turtles.
(2) Some rabbits run faster than all turtles.
(3) Not all rabbits run faster than turtles.
(4) There do not exist rabbits and turtles that run at the same speed.

- **Solution:** Using a universal domain, let:
  - $F(x)$: x is a rabbit.
  - $G(y)$: y is a turtle.
  - $H(x,y)$: x runs faster than y.
  - $L(x,y)$: x and y run at the same speed.

(1) $\forall x \forall y(F(x) \wedge G(y) \rightarrow H(x,y))$
(2) $\exists x(F(x) \wedge \forall y(G(y) \rightarrow H(x,y)))$
(3) $\neg \forall x \forall y(F(x) \wedge G(y) \rightarrow H(x,y))$
(4) $\neg \exists x \exists y(F(x) \wedge G(y) \wedge L(x,y))$

↳Notices

- The use of **unary** and ***n*-ary** predicates.
- The **distinction** between universal and **existential** quantifiers.
- When multiple quantifiers appear, their **order** cannot be arbitrarily swapped.

  Such as: in the domain of real numbers R, **let** $H(x,y)$: $x+y=10$
  - $\forall x \exists y H(x,y)$ has a truth value of **1**
  - $\exists y \forall x H(x,y)$ has a truth value of **0**
- The symbolization of propositions is **not unique**.

Such as: based on the previous example, there are the following alternative symbolizations:

(1) $\forall x(F(x) \rightarrow \forall y(G(y) \rightarrow H(x,y)))$ $(\forall x \forall y(F(x) \wedge G(y) \rightarrow H(x,y)))$

(3) $\exists x \exists y(F(x) \wedge G(y) \wedge \neg H(x,y))$ $(\neg \forall x \forall y(F(x) \wedge G(y) \rightarrow H(x,y)))$

(4) $\forall x \forall y(F(x) \wedge G(y) \rightarrow \neg L(x,y))$ $(\neg \exists x \exists y(F(x) \wedge G(y) \wedge L(x,y)))$

- 3.1.1 The Limitations of Propositional Logic
- 3.1.2 Individual Terms, Predicates, and Quantifiers
- 3.1.3 Symbolization of First-Order Logic
- 3.1.4 First-Order Logic Formulas and Classification
  - First-order language (alphabet, terms, atomic formulas, compound formulas)
  - Scope and bound variables, bound occurrences and free occurrences
  - Closed formulas
  - Interpretation of first-order language
  - Tautologies, contradictions, and satisfiable formulas
  - Substitution examples

- A ***first-order language*** is a symbolic language used to express statements in first-order logic. ***Well-formed formulas*** are its constructed expressions, used to precisely express complex statements about objects and their relationships.
- **Definition 3.1:** The alphabet of a first-order language is defined as follows:

(1) **Individual constants:** $a,b,c,\ldots,a_i,b_i,c_i,\ldots$, where $i \geq 1$

(2) **Individual variables:** $x,y,z,\ldots,x_i,y_i,z_i,\ldots$, where $i \geq 1$

(3) **Function symbols:** $f,g,h,\ldots,f_i,g_i,h_i,\ldots$, where $i \geq 1$

(4) **Predicate symbols:** $F,G,H,\ldots,F_i,G_i,H_i,\ldots$, where $i \geq 1$

(5) **Quantifier symbols:** $\forall,\exists$

(6) **Connective symbols:** $\neg,\wedge,\vee,\rightarrow,\leftrightarrow$

(7) **Parentheses and commas:** ( ) , ,

- **Definition 3.2：** The definition of ***terms*** in a first-order language is as follows:

  (1) ***Individual constants*** and ***individual variables*** are terms.

  (2) If $\varphi(x_1,x_2,\ldots,x_n)$ is an arbitrary n-ary function symbol, and $t_1,t_2,\ldots,t_n$ are arbitrary terms, then $\varphi(t_1,t_2,\ldots,t_n)$ is also a term.

  (3) All terms are obtained through a finite number of applications of (1) and (2).

- **Definition 3.3:** Let $R(x_1,x_2,\ldots,x_n)$ be an arbitrary n-ary predicate symbol in a first-order language, and let $t_1,t_2,\ldots,t_n$ be arbitrary terms. Then, $R(t_1,t_2,\ldots,t_n)$ is called an ***atomic formula***.

同济经管 TONGJI SEM

- **Definition 3.4**: ***Well-formed formulas*** in a first-order language are defined as follows:

(1) **Atomic formulas** are well-formed formulas.

(2) If *A* is a well-formed formula, then ¬*A* is also a well-formed formula.

(3) If *A* and *B* are well-formed formulas, then (*A*∧*B*), (*A*∨*B*), (*A*→*B*), and (*A*↔*B*) are also well-formed formulas.

(4) If *A* is a well-formed formula, then ∀*xA* and ∃*xA* are also well-formed formulas.

(5) Only those expressions formed by a finite application of rules (1) through (4) are considered well-formed formulas.

- Well-formed formulas are also referred to as ***predicate formulas*** or simply ***formulas***.

↳Bound Variable and Bound Occurrences

- **Definition 3.5:** In the formulas $\forall x A$ and $\exists A$, ***x*** is called the ***bound variable*** (or binder), and A is the scope of the respective quantifier. In the scope of $\forall x$ and $\exists x$, all occurrences of ***x*** are called ***bound occurrences***. Variables in *A* that are not bound occurrences are called ***free occurrences***.

e.g. **Example**: The formula $\forall x(F(x,y)\to\exists yG(x,y,z))$

- The scope of $\forall x$ is $(F(x,y)\to\exists yG(x,y,z))$, with ***x*** as the **bound variable**. Both occurrences of ***x*** are **bound occurrences.**
- The scope of $\exists y$ is $G(x,y,z)$, with ***y*** as the **bound variable**.
- The first occurrence of ***y*** is a **free occurrence**, and the second occurrence is **a bound occurrence.**
- ***z*** is a **free occurrence**.

e.g. »» **Example:** The formula $\forall x(F(x)\rightarrow\exists xG(x))$

- The scope of $\forall x$ is $(F(x)\rightarrow\exists xG(x))$, with x as the bound variable.
- The scope of $\exists x$ is $G(x)$, with x as the bound variable.
- Both occurrences of x are bound occurrences: the first in $\forall x$, and the second in $\exists x$.

■ **Closed formula:** A formula that contains **no free occurrences** of individual variables is called a ***closed well-formed formula***, abbreviated as ***closed formula***.

- A formula is merely a framework of logical expressions. **To evaluate its truth value**, the following tasks must be completed:

  (1) **Interpretation**: Assign specific (semantic) meaning to this framework.

  (2) **Assignment**: Under a given interpretation, assign values to the free variables in the formula.

  (3) **Quantification**: Eliminate the free variables so that the truth value of the formula no longer depends on specific assignments but is determined by the overall situation of all possible assignments.

e.g. Example: Formula $\forall x(F(x) \rightarrow G(x))$

- **Specification1: Domain:** All individuals. ***F*(*x*): *x*** is a person. ***G*(*x*): *x*** is Asian. This formula translates to "For all x, if x is a person, then x is Asian." This is a **false statement** because not all people are Asian.
- **Specification 2: Domain:** The set of real numbers. ***F*(*x*): *x* > 10**. ***G*(*x*): *x*** > 0. This formula translates to "For all x, if x > 10, then x > 0." This is a **true statement** because any number greater than 10 is also greater than 0.

e.g. Example: Formula $\exists xF(x,y)$

- **Specification:Domain:** The set of natural numbers. ***F*(*x*,*y*):** *x* = ***y***. ***y* = 0.** Since ***x***,***y*** both belong to the set of natural numbers N, for any ***y*∈*N***, there exists an ***x***=***y*** such that the equation holds. Therefore, the statement is **always true.**

e.g.>>>**Example:** Given the interpretation *I* and assignment *σ* as follows :

① **Definition** D=N ; ② $\bar{a}=0$; ③ $\bar{f}(x,y)=x+y, \bar{g}(x,y)=xy$) ;

④ $\bar{F}(x,y): x=y$. **Value σ:** $\sigma(x)=0$, $\sigma(y)=1$, $\sigma(z)=2$.

Explain the meaning of the following formula under interpretation *I* and assignment *σ*, and discuss its truth value.

(1) $\forall xF(g(x,a),y)$: $\forall x(0x=1)$ **False**

(2) $\forall x\forall y(F(f(x,a),y)\rightarrow F(f(y,a),x))$ **Truth**

$f(x,a)=x, f(y,a)=y$, $\forall x\forall y(F(x,y)\rightarrow F(y,x))$, $(x=y)\rightarrow(y=x)$

(3) $\forall x\forall y\exists zF(f(x,y),z)$ $x\forall y\exists z\ (x+y=z)$ **Truth**

(4) $\exists xF(f(x,y),g(x,z))$ $\exists x(x+1=2x)$ **Truth**

(5) $F(f(x,a), g(y,a))$ $x+0=1\times 0$ **Truth**

(6) $\forall x(F(x,y)\rightarrow\exists yF(f(x,a), g(y,a)))$ $\forall x(x=1\rightarrow\exists y(x=0))$ **False**

↳Classification of first-order Logic formulas

- ***Tautology*** (logically valid formula): no false interpretation and assignment.
- ***Contradiction*** (contravalid formula): no true interpretation and assignment.
- ***Satisfiable formula***: at least one true interpretation and assignment.

i

- A tautology is a satisfiable formula, but the converse is not true.
- In first-order logic, the satisfiability (tautology, contradiction) of a formula is undecidable, meaning there is no algorithm that can determine in finite steps whether a given formula is satisfiable (a tautology, a contradiction).

↳Substitution instance of a propositional formula

- **Definition 3.6：** Let A_0 be a propositional formula containing propositional variables $p_1,p_2,\ldots,p_n$, and let $A_1,A_2,\ldots,A_n$ be predicate formulas. The formula A, obtained by uniformly replacing each p_i (for $1\le i\le n$) in A_0 with A_i, is called a ***substitution instance*** of A_0.
  - **Such as:** $F(x)\rightarrow G(x)$ and $\forall xF(x)\rightarrow\exists yG(y)$ are substitution instances of $p\rightarrow q$.
- **Theorem 3.2:** All substitution instances of a tautology are **logically valid**, and all substitution instances of a contradiction are **contradictions.**

↳Classification of first-order Logic formulas(e.g.)

e.g.»»**Example:** Determine the type of the following formula:

(1) $\forall x(F(x) \to G(x))$

I_1: D_1=R, $\bar{F}(x)$: x **integer**, $\bar{G}(x)$: x is **rational**. (1) **is a true proposition.**

I_2: D_2=R, $\bar{F}(x)$: x **integer**, $\bar{G}(x)$: x is **natural number**. (1) **is a false proposition.** (1) is a **satisfiable formula** (not logically valid formula).

(2) $\neg(\forall xF(x,y)) \vee (\forall xF(x,y))$,

$\neg p \vee p$ substitution instances , $\neg p \vee p$ tautology, (2) is a **tautology.**

(3) $\neg (\forall xF(x) \to \exists yG(y)) \wedge \exists yG(y)$,

$\neg(p \to q) \wedge q$ substitution instances, $\neg(p \to q) \wedge q$ contradiction, (3) is a **contradiction.**

↳Classification of first-order Logic formulas(e.g.)

e.g.⋙**Example:** Determine the type of the following formula:

(4) $\forall x F(x,y)$

I_1: D_1=N, $\overline{F}(x,y)$: $x \geq y$, **Assign** $\sigma(y)$=0.

(4) is a true proposition.

I_2: D_2=N, $\overline{F}(x,y)$: $x \geq y$, **Assign** $\sigma(y)$=1.

(4) is a false proposition.

(4) is a **satisfiable formula** (not logically valid formula).

Objective :

Key Concepts :

Discrete Mathematics 2025 Spring

同济经管
TONGJI SEM

魏可佶 kejiwei@tongji.edu.cn

CAMEA 中国高质量MBA教育认证 AACSB ACCREDITED EQUIS ACCREDITED

- **3.2.1 First-Order Logic Equivalences and Substitution Rules**

  Basic Equivalences Substitution Rules, Renaming Rules

- **3.2.2 Prenex normal form of first-order logic**

TONGJI SEM

2.2.1 Equivalence Expressions and Equivalence Calculus
↳ Basic Equivalence Expressions

- Double Negation Law: $\neg\neg A \Leftrightarrow A$
- Idempotent Law: $A \vee A \Leftrightarrow A$, $A \wedge A \Leftrightarrow A$
- Commutative Law: $A \vee B \Leftrightarrow B \vee A$, $A \wedge B \Leftrightarrow B \wedge A$
- Associative Law: $(A \vee B) \vee C \Leftrightarrow A \vee (B \vee C)$
  $(A \wedge B) \wedge C \Leftrightarrow A \wedge (B \wedge C)$
- Distributive Law: $A \vee (B \wedge C) \Leftrightarrow (A \vee B) \wedge (A \vee C)$
  $A \wedge (B \vee C) \Leftrightarrow (A \wedge B) \vee (A \wedge C)$
- De Morgan's Laws: $\neg(A \vee B) \Leftrightarrow \neg A \wedge \neg B$
  $\neg(A \wedge B) \Leftrightarrow \neg A \vee \neg B$
- Absorption Law: $A \vee (A \wedge B) \Leftrightarrow A$
  $A \wedge (A \vee B) \Leftrightarrow A$

2.2.1 Equivalence Expressions and Equivalence Calculus
↳ Basic Equivalence Expressions (cont.)

- Zero Law: $A \vee 1 \Leftrightarrow 1$, $A \wedge 0 \Leftrightarrow 0$
- Identity Law: $A \vee 0 \Leftrightarrow A$, $A \wedge 1 \Leftrightarrow A$
- Law of the Excluded Middle: $A \vee \neg A \Leftrightarrow 1$
- Law of Contradiction: $A \wedge \neg A \Leftrightarrow 0$
- Implication Equivalence: $A \rightarrow B \Leftrightarrow \neg A \vee B$
- Biconditional Equivalence: $A \leftrightarrow B \Leftrightarrow (A \rightarrow B) \wedge (B \rightarrow A)$
- Contraposition: $A \rightarrow B \Leftrightarrow \neg B \rightarrow \neg A$
- Negation of Equivalence: $A \leftrightarrow B \Leftrightarrow \neg A \leftrightarrow \neg B$
- Reductio ad Absurdum (Proof by Contradiction):

$$(A \rightarrow B) \wedge (A \rightarrow \neg B) \Leftrightarrow \neg A$$

↳ Tautology A ↔ B vs. Equivalence A ⇔ B

- **Definition 3.7:** If $A \leftrightarrow B$ is a **tautology** (a valid formula), then A and B are called equivalent, denoted by $A \Leftrightarrow B$, and A B is referred to as an ***equivalence***.
- There are 24 propositional basic equivalences and their substitution examples, all of which are equivalences in first-order logic.
- For example:

$$\forall xF(x) \rightarrow \exists yG(y) \Leftrightarrow \neg \forall xF(x) \vee \exists yG(y)$$

$$\neg(\forall xF(x) \vee \exists yG(y)) \Leftrightarrow \neg \forall xF(x) \wedge \neg \exists yG(y)$$ **and so on**

- **Quantifier Elimination Equivalences:** To transform logical expressions to remove quantifiers (∃, ∀), yielding an equivalent **quantifier-free** form.
  - Let $D=\{a_1,a_2,\ldots,a_n\}$

$$\forall x A(x) \Leftrightarrow A(a_1) \wedge A(a_2) \wedge \ldots \wedge A(a_n)$$

$$\exists x A(x) \Leftrightarrow A(a_1) \vee A(a_2) \vee \ldots \vee A(a_n)$$

- **Quantifier Negation Equivalences: To** convert quantifiers (**∀ and ∃**) and negation (¬), changing the scope of the negation without altering the logical meaning.
  - Let $A(x)$ be a formula in which x appears freely.

$$\neg \forall x A(x) \Leftrightarrow \exists x \neg A(x)$$

$$\neg \exists x A(x) \Leftrightarrow \forall x \neg A(x)$$

- **Quantifier Distribution Equivalences:** To allocate or restructure quantifiers (∀, ∃) to interact correctly with logical operations (∧, ∨, →) while preserving logical equivalence.
  - $\forall x(A(x) \wedge B(x)) \Leftrightarrow \forall x A(x) \wedge \forall x B(x)$

    $\exists x(A(x) \vee B(x)) \Leftrightarrow \exists x A(x) \vee \exists x B(x)$

Attention： ∀ to ∨， ∃ to ∧ no Quantifier Distribution Equivalences.