2 Predicate Logic

Why We Need Predicate Logic

▶ Propositional logic allows us to establish some basic inference from one proposition to another. For example, we have $P \wedge (P \to Q) \to Q$. However, it has certain limitations. For example, it cannot even describe the following simple "inference".

Examples: The Limitation of Propositional Logic

- **Premises:** ① Any man will die. ② Bob is a man.
- **Conclusion:** Bob will die.

It is reasonable for us to believe that the above two premises together "imply" the conclusion. However, there is no way to express it in the context of propositional logic. Note that the following inference is not correct

$$(\mathsf{Man} \to \mathsf{Die}) \wedge (\mathsf{Bob} \to \mathsf{Man}) \to (\mathsf{Bob} \to \mathsf{Die})$$

This is because "*Bob is a man*" is already atomic proposition; in the context of propositional logic, it has no meaning to divide it anymore!

▶ The problem of the above example is as follows. The reason why we believe the above inference is correct is because the property of "*will die*" holds for *any man*, while "*Bob*" is a *specific man*. However, propositional logic does not allow us to analyze each component in the sentence for more detail. For example, let us consider the following two propositions:

$$P: \underbrace{\text{Alice}}_{\substack{\text{subject}\\ \text{主语}}} \quad \underbrace{\text{is a man}}_{\substack{\text{predicate}\\ \text{谓语}}} \qquad\qquad Q: \underbrace{\text{Bob}}_{\substack{\text{subject}\\ \text{主语}}} \quad \underbrace{\text{is a man}}_{\substack{\text{predicate}\\ \text{谓语}}}$$

Clearly, both propositions P and Q are atomic, which are independent in propositional logic. However, them do have the same structure, but we do not utilize this information for inference. To be more specific, we can actually write P : $\mathsf{Man}(\mathsf{Alice})$ and Q : $\mathsf{Man}(\mathsf{Bob})$.

▶ In a more general setting, a predicate is something describing whether or not a group of things have some *relations*. For example, we have

- $\underbrace{\text{Alice}}_{\text{subj.}}$ and $\underbrace{\text{Bob}}_{\text{subj.}}$ $\underbrace{\text{are friends}}_{\text{pred.}}$ can be expressed as $\mathsf{Friends}(\mathsf{Alice},\mathsf{Bob})$
- $\underbrace{5}_{\text{subj.}}$ $\underbrace{\text{is greater than}}_{\text{pred.}}$ $\underbrace{3}_{\text{subj.}}$ can be expressed as $\mathsf{Greater}(5,3)$
- $\underbrace{B}_{\text{subj.}}$ $\underbrace{\text{is in the middle of}}_{\text{pred.}}$ $\underbrace{\text{A}}_{\text{subj.}}$ and $\underbrace{\text{C}}_{\text{subj.}}$ can be expressed as $\mathsf{Middle}(B,A,C)$

Elements in Predicate Logic

- Based on the above discussion, we know that we can further specify a proposition by separating its subjects and predicate. In the context of predicate logic, we usually call a subject an **Object/Individual (个体词语)**. For example, "Alice" or "Bob" are object constant (个体常元) and we usually use "x, y, z" to represent unspecified object variables (个体变元).

- Also, when we talk about "*All man will die*", we are actually talking about "*For any object, if it is a man, then it will die*". Therefore, the domain of the objects under consideration is very important. For example, if we consider objects not on the earth, then this sentence may not be true ☺. Therefore, we call the set of all objects under consideration the **Domain of Discourse (论域)** and usually denote it by D.

- Given a set of objects, essentially we want to determine whether their have some properties/relations. This is actually the role of **Predicate (谓词)**, which is a mapping
$$P : D^n \to \{0, 1\},$$
Clearly, for $n = 1$, P is actually a property for $x \in D$; for $n = 2$, P is a relation between $x, y \in D$; and for $n = 0$, P is just a proposition because it has no variable now.

- Sometimes we need to use sentence like "*The son of Alice and Bob will not die*". Here, "The son of Alice and Bob" is actually a new object, which is not given directly but through a **Function (函数)**. Formally, in the context of predicate logic, a function is a mapping that maps a groups of objects to a new object
$$f : D^n \to D$$
For example, the above sentence can be written as $\neg\mathsf{Die}(\mathsf{Son}(\mathsf{Alice}, \mathsf{Bob}))$.

- We note that $P(x)$ is NOT a proposition if predicate P and object x are unspecified variables. There are two possible ways to make $P(x)$ a proposition:
  - specify both predicate P and object x, e.g., $\mathsf{Man}(\mathsf{Alice})$ is a proposition.
  - just specify predicate P and **quantify (量化)** the object, e.g., "for any object $x \in D$ in the domain, we have $\mathsf{Man}(x)$" is again a proposition.

  In predicate logic, we have two different **Quantifiers (量词)**
  - **Universal Quantifier (全称量词)**: "$\forall$" read as "for all", meaning that
$$(\forall x)(P(x)) \text{ is } \mathsf{true}\text{, if and only if, } P(x) \text{ is } \mathsf{true} \text{ for any } x \in D$$
  - **Existential Quantifier (存在量词)**: "$\exists$" read as "there exists", meaning that
$$(\exists x)(P(x)) \text{ is } \mathsf{true}\text{, if and only if, } P(x) \text{ is } \mathsf{true} \text{ for some } x \in D$$

Well-Formed Formula for Predicate Logic

- Now, we discuss how to write down a predicate logic formula in a meaningful way, which is actually the role of well-formed formula. To this end, we need to introduce the concept of **Free Variables (自由变元)** and **Bound Variables (约束变元)**.

- When we write a formula as $(\forall x)(\cdots)$ or $(\exists x)(\cdots)$, the "$\cdots$" part after the quantifier is called the **Scope (辖域)** of the quantifier. For example, $(\forall x)(\cdots)$ means that "$\forall$" is applied to each x in its scope "$(\cdots)$"; or we say x is **bound (约束)** by this quantifier. Otherwise, we say a variable x is **free (自由)** if it is not bound by any quantifiers.

- Furthermore, **a quantifier can only bound free variables in its scope**. Hence, if a variable is in the scope of two quantifiers, then it is bound by the closer one. For example,

$$(\forall x)\left(\underbrace{P(x)\wedge(\exists y)\left(\underbrace{Q(x,y)\to R(x)}_{\text{scope of }\exists y}\right)}_{\text{scope of }\forall x}\right)$$

Therefore, y is bound by $\exists y$ and all x are bound by $\forall x$. However, if we write

$$(\forall x)(P(x)\wedge(\exists x)(Q(x)\to R(x)))$$

then variable x in $Q(x)\to R(x)$ are bound by $\exists x$, and since they are not free, $\forall x$ is only applied to $P(x)$. This formula is actually the same as $(\forall x)(P(x)\wedge(\exists y)(Q(y)\to R(y)))$.

- Now we need a meaningful way to put the following elements together:
① proposition variables: p,q,r; ② object variables: x,y,z; ③ object constants: a,b,c; ④ predicate variables: P,Q,R; ⑤ predicate constants: $\mathsf{Man},\mathsf{Die}$; ⑥ functions: f,g; ⑦ logic operators: $\neg,\vee,\wedge,\to,\leftrightarrow$; ⑧ quantifiers: $\forall,\exists$.

Definition: Well-Formed Formula (WFF)

Well-formed formulas for predicate logic are defined inductively as follows:

1. Each atomic proposition variable/constant and atomic predicate variable/constant is a WFF.
2. If α is a formula, then $\neg\alpha$ is a WFF.
3. If α and β are WFFs, and there is no variable x which is free in one but bound in another, then $(\alpha\wedge\beta),(\alpha\vee\beta),(\alpha\to\beta),(\alpha\leftrightarrow\beta)$ are WFFs.
4. If α is WFF and x is a free variable in α, then $(\forall x)\alpha$ and $(\exists x)\alpha$ are WFFs.

- **Example:** According to the definition, the followings are WFFs (with simplifications):

$$\neg p,\quad \neg P(x,y)\wedge Q(x,y),\quad (\forall x)(A(x)\to B(x)),\quad (\exists x)(A(x)\to(\forall y)B(x,y))$$

However, the followings are Not WFFs:

$$((\forall x)(P(x)\wedge Q(x))\vee R(x),\quad (\exists x)((\forall x)F(x)),\quad (\forall x)P(y)$$

Formalize Natural Sentences using Predicate Logic Formula

- **Example 1:** "All rational numbers are real numbers." can be written as

$$(\forall x)(\mathsf{Rational}(x) \to \mathsf{Real}(x))$$

Note that it is incorrect to write $(\forall x)(\mathsf{Rational}(x) \land \mathsf{Real}(x))$ because it requires that any number is both a rational number and a real number, which is stronger than what we want. For example, if we consider an irrational number x, then $\mathsf{Rational}(x) \to \mathsf{Real}(x)$ is true, which is what we need, but $\mathsf{Rational}(x) \land \mathsf{Real}(x)$ is false.

- **Example 2:** "Some real number is a rational number." can be written as

$$(\exists x)(\mathsf{Real}(x) \land \mathsf{Rational}(x))$$

However, it is incorrect to write $(\exists x)(\mathsf{Real}(x) \to \mathsf{Rational}(x))$. To see this, we can consider domain $D = \{e, \pi, \mathsf{Alice}\}$. Clearly, there is no real number that is rational in D, which means that $(\exists x)(\mathsf{Real}(x) \land \mathsf{Rational}(x))$ is false. However, by considering $\mathsf{Alice} \in D$, we have $\mathsf{Real}(\mathsf{Alice}) \to \mathsf{Rational}(\mathsf{Alice}))$ is true because the condition is false. Therefore, $(\exists x)(\mathsf{Real}(x) \to \mathsf{Rational}(x))$ is true, which is not what we want.

- **Example 3:** "There is no irrational number that is rational." can be written as

$$\neg(\exists x)(\mathsf{Irrational}(x) \land \mathsf{Ration}(x))$$

Note that the above sentence is equivalent to "All irrational numbers are not rational.". Therefore, we can also write $(\forall x)(\mathsf{Irrational}(x) \to \neg\mathsf{Ration}(x))$.

- **Example 4:** Let D be a set of numbers and define the following predicate and functions

$$\mathsf{Eq}(x,y) : x = y, \qquad \mathsf{suc}(x) = x + 1, \qquad \mathsf{pre}(x) = x - 1.$$

We can the specify the following properties:

  - "Any number has a unique successor." can be written as

$$(\forall x)(\exists y)(\mathsf{Eq}(y, \mathsf{suc}(x)) \land (\forall z)(\mathsf{Eq}(z, \mathsf{suc}(x)) \to \mathsf{Eq}(y, z)))$$

  - "'**0**' is not a successor of any number." can be written as

$$\neg(\exists x)(\mathsf{Eq}(\mathbf{0}, \mathsf{suc}(x))$$

  - "Any number has a unique predecessor, except '**0**'." can be written as

$$(\forall x)\,(\neg\mathsf{Eq}(x, \mathbf{0}) \to (\exists y)\,(\mathsf{Eq}(y, \mathsf{pre}(x)) \land (\forall z)\,(\mathsf{Eq}(z, \mathsf{pre}(x)) \to \mathsf{Eq}(y, z))))$$

- **Example 5:** Let D be a group of people and define $\mathsf{Love}(x,y)$ be "x loves y". Then

  - $(\forall x)(\forall y)\mathsf{Love}(x,y)$ means "everyone loves each other."
  - $(\exists x)(\exists y)\mathsf{Love}(x,y)$ means "someone loves someone."
  - $(\exists x)(\forall y)\mathsf{Love}(x,y)$ means "someone loves everyone."
  - $(\forall y)(\exists x)\mathsf{Love}(x,y)$ means "everyone has someone that loves him/her."
  - $(\exists y)(\forall x)\mathsf{Love}(x,y)$ means "someone is loved by everyone."
  - $(\forall x)(\exists y)\mathsf{Love}(x,y)$ means "everyone has someone that he/she loves."

Assignments for Predicate Logic Formulas

- Recall that, for any proposition *formula* such as $\alpha = P \wedge Q$, it is not a proposition because variables P and Q are *unspecified*. To make the formula has a truth value, we need to give an assignment determining the values of P and Q.

- This is the same for predicate logic formula. For example, for formula $\alpha = (\forall x)(P(x) \to Q(f(x), a))$, we need to specify the following ingredients to determine the value of α:

Definition: Assigments for Predicate Logic Formulas

An assignment for a predicate logic formula contains the followings

1. *specific domain of discourse* D
2. for each object variables, assign a *specific object* in D
3. for each predicate variables, assign a *specific predicate* in $\{P \mid P : D^n \to \{0, 1\}\}$.
4. for each propositional variables, assign a *specific truth value* in $\{0, 1\}$.
5. for each function variables, assign a *specific function* in $\{f \mid f : D^n \to D\}$.

Let us consider predicate logic formula $(\forall x)(P(x) \to Q(f(x), a))$. Then a possible assignment I can be: $D = \{1, 2\}$, $a = 1$ and

$$f(x) = \begin{cases} 2 & \text{if } x=1 \\ 1 & \text{if } x=2 \end{cases}, P(x) = \begin{cases} \mathsf{F} & \text{if } x=1 \\ \mathsf{T} & \text{if } x=2 \end{cases}, Q(x, y) = \begin{cases} \mathsf{T} & \text{if } x=2 \wedge y=1 \\ \mathsf{F} & \text{otherwise} \end{cases}$$

For the above assignment, we have that

- if $x = 1$, then $P(1) \to Q(f(1), 1) = \mathsf{T}$.
- if $x = 2$, then $P(2) \to Q(f(2), 1) = \mathsf{T}$.

Therefore, we know that $(\forall x)(P(x) \to Q(f(x), a)) = \mathsf{T}$ under this assignment.

- Similar to the case of propositional logic, we say a predicate logic formula is
  - **universally valid (普遍有效的)** if it is true under any assignment;
  - **satisfiable (可满足的)** if it is true under some assignment;
  - a **unsatisfiable (不可满足的)** if it is false under any assignment.

- **Examples:** The followings are universally valid

$$\alpha_1 = (\forall x)P(x) \to P(a), \quad \alpha_2 = ((\forall x)P(x) \vee (\forall x)Q(x)) \to (\forall x)(P(x) \vee Q(x))$$

A satisfiable formula can be, e.g., $\alpha_3 = (\forall x)P(x, a)$.

However, the following formulas are unsatisfiable

$$\alpha_4 = (\forall x)(P(x) \wedge \neg P(x)), \quad \alpha_5 = (\forall x)P(x) \wedge (\exists y)\neg P(y)$$

Equivalence of Predicate Formulas

- Similar to the case of propositional logic, we can also define **equivalence** of two predicate logic formulas as follows:

Definition: Equivalence of Predicate Formulas

Let α and β be two predicate formulas. We say α and β are equivalent, denoted by $\alpha = \beta$ or $\alpha \Leftrightarrow \beta$ if the truth values of α and β are the same under any assignments.

According to the above definition, we also have $\alpha = \beta$ iff $\alpha \leftrightarrow \beta$ is universally valid.

- An direct approach to obtain equivalences are from what we know for proposition logic

① $\neg\neg p = p$	① $\neg\neg(\forall x)P(x) = (\forall x)P(x)$
② $p \to q = \neg p \vee q$	② $P(x) \to Q(x) = \neg P(x) \vee Q(x)$
③ $(p \wedge g) \vee r = (p \vee r) \wedge (q \vee r)$	③ $(P(x) \wedge Q(x)) \vee R(s) = (P(x) \vee R(x)) \wedge (Q(x) \vee R(x))$
④ ...	④ ...

- Another approach is **Change of Quantifiers**, which is essentially the De Morgan's Law

Theorem: Change of Quantifiers

① $(\forall x)\alpha(x) = \neg(\exists x)\neg\alpha(x)$ or $\neg(\forall x)\alpha(x) = (\exists x)\neg\alpha(x)$;
② $(\exists x)\alpha(x) = \neg(\forall x)\neg\alpha(x)$ or $\neg(\exists x)\alpha(x) = (\forall x)\neg\alpha(x)$.

Proof: Let us proof ①. The case of ② is similar.For any assignment I, we have

$$\begin{aligned}(\forall x)\alpha(x) = \mathsf{T} \quad &\text{iff} \quad \text{for any } a \in D, \alpha(a) = \mathsf{T} \quad \text{iff} \quad \text{for any } a \in D, \neg\alpha(a) = \mathsf{F} \\ &\text{iff} \quad (\exists x)\neg\alpha(x) = \mathsf{F} \quad \text{iff} \quad \neg(\exists x)\neg\alpha(x) = \mathsf{T}\end{aligned}$$

- **Example:** Let us consider "天下乌鸦一般黑". We can define predicates $B(x)$ as "x 是乌鸦" and $D(x,y)$ as "x, y 一般黑". Then we can formalize the above by either

$$\alpha_1 = (\forall x)(\forall y)((B(x) \wedge B(y)) \to D(x,y))$$

or

$$\alpha_2 = \neg(\exists x)(\exists y)(B(x) \wedge B(y) \wedge \neg D(x,y))$$

To see how they are equivalent, we have

$$\begin{aligned}\alpha_2 =& (\forall x)\neg(\exists y)(B(x) \wedge B(y) \wedge \neg D(x,y)) = (\forall x)(\forall y)\neg(B(x) \wedge B(y) \wedge \neg D(x,y)) \\ =& (\forall x)(\forall y)(\neg(B(x) \wedge B(y)) \vee D(x,y)) = (\forall x)(\forall y)((B(x) \wedge B(y)) \to D(x,y)) \\ =& \alpha_1\end{aligned}$$

Distributive Laws for Predicate Formulas

- When we say "each one is either a boy or a girl", clearly it is different from "either each one is a boy or each one is a girl", i.e., $(\forall x)(\mathsf{Boy}(x) \vee \mathsf{Girl}(x)) \neq (\forall x)\mathsf{Boy}(x) \vee (\forall x)\mathsf{Girl}(x)$. However, when we say "each one is both smart and lovely", it is the same as "both each one is smart and each one is lovely", i.e., $(\forall x)(\mathsf{Smart}(x) \wedge \mathsf{Lovely}(x)) = (\forall x)\mathsf{Smart}(x) \wedge (\forall x)\mathsf{Lovely}(x)$. Therefore, we may ask when quantifiers can be distributed.

- Suppose that $\alpha(x)$ is a formula in which x is a free variable and β is a formula in which there is no free x (the same as β contains no x). Then we have the following basic results.

  ① $(\forall x)(\alpha(x) \vee \beta) = (\forall x)\alpha(x) \vee \beta$ ② $(\exists x)(\alpha(x) \vee \beta) = (\exists x)\alpha(x) \vee \beta$
  ③ $(\forall x)(\alpha(x) \wedge \beta) = (\forall x)\alpha(x) \wedge \beta$ ④ $(\exists x)(\alpha(x) \wedge \beta) = (\exists x)\alpha(x) \wedge \beta$

  Proof: We prove ①; others can be proved similarly. For any assignment I, then

  $(\Rightarrow)$ If $(\forall x)(\alpha(x) \vee \beta) = \mathsf{T}$, then we know that $\alpha(a) \vee \beta = \mathsf{T}$ for any $a \in D$. We consider the following two cases:

  – if $\beta = \mathsf{T}$, then it gives $(\forall x)\alpha(a) \vee \beta = \mathsf{T}$;
  – if $\beta = \mathsf{F}$, then $\alpha(a) = \mathsf{T}$ for any $a \in D$, i.e., $(\forall x)\alpha(a) = \mathsf{T}$, which again gives us $(\forall x)\alpha(a) \vee \beta = \mathsf{T}$.

  $(\Leftarrow)$ Similarly, if $(\forall x)\alpha(a) \vee \beta = \mathsf{T}$, we know either $(\forall x)\alpha(a) = \mathsf{T}$ or $\beta = \mathsf{T}$. For both cases, we obtain $(\forall x)(\alpha(x) \vee \beta) = \mathsf{T}$.

- The distributive law also apply to $\to$, but we need to be careful

  ① $(\forall x)(\alpha(x) \to \beta) = (\exists x)\alpha(x) \to \beta$ ② $(\exists x)(\alpha(x) \to \beta) = (\forall x)\alpha(x) \to \beta$
  ③ $(\forall x)(\beta \to \alpha(x)) = \beta \to (\forall x)\alpha(x)$ ④ $(\exists x)(\beta \to \alpha(x)) = \beta \to (\exists x)\alpha(x)$

  Proof: We still prove ①; others can be proved similarly.
  $(\forall x)(\alpha(x) \to \beta) = (\forall x)(\neg\alpha(x) \vee \beta) = (\forall x)\neg(\alpha(x)) \vee \beta = \neg(\exists x)\alpha(x) \vee \beta = (\exists x)\alpha(x) \to \beta$.

- As we can see, when α and β both contain x, the distributive laws cannot be used arbitrarily. However, it works for the following special cases:

  ① $(\forall x)(\alpha(x) \wedge \beta(x)) = (\forall x)\alpha(x) \wedge (\forall x)\beta(x)$
  ② $(\exists x)(\alpha(x) \vee \beta(x)) = (\exists x)\alpha(x) \vee (\exists x)\beta(x)$
  ③ $(\exists x)(\alpha(x) \to \beta(x)) = (\forall x)\alpha(x) \to (\exists x)\beta(x)$

  Proof: For ③, $(\forall x)\alpha(x) \to (\exists x)\beta(x) = \neg(\forall x)\alpha(x) \vee (\exists x)\beta(x) = (\exists x)\neg\alpha(x) \vee (\exists x)\beta(x) = (\exists x)(\neg\alpha(x) \vee \beta(x)) = (\exists x)(\alpha(x) \to \beta(x))$.

  The above laws does not work for "$\forall$-$\vee$" or "$\exists$-$\wedge$", e.g., there exists a smart one and there exists a lovely one do not imply there exists one that is both smart and lovely.

Variable Renamings

- As we mentioned, the names of the object variables essentially just want to connect the quantifiers with the places they apply. Therefore, the names of the objects do not really matter and we can always rename them. For example, we have
$$(\forall x)\alpha(x) = (\forall y)\alpha(y) \quad \text{and} \quad (\exists x)\alpha(x) = (\exists y)\alpha(y)$$
- Furthermore, using the distributive laws discussed above, we also have

① $(\forall x)(\forall y)(\alpha(x) \wedge \beta(y)) = (\forall x)(\alpha(x) \wedge \beta(x))$

② $(\exists x)(\exists y)(\alpha(x) \vee \beta(y)) = (\exists x)(\alpha(x) \vee \beta(x))$

Proof: For ①, we have $(\forall x)(\forall y)(\alpha(x) \vee \beta(y)) = (\forall x)((\forall y)(\beta(y)) \vee \alpha(x)) = (\forall x)\alpha(x) \wedge (\forall y)\beta(y)) = (\forall x)\alpha(x) \wedge (\forall x)\beta(x)) = (\forall x)(\alpha(x) \wedge \beta(x))$.

However, for "∀-∨" or "∃-∧", we only have

③ $(\forall x)(\forall y)(\alpha(x) \vee \beta(y)) = (\forall x)\alpha(x) \vee (\forall y)\beta(y) \neq (\forall x)(\alpha(x) \vee \beta(x))$

④ $(\exists x)(\exists y)(\alpha(x) \wedge \beta(y)) = (\exists x)\alpha(x) \wedge (\exists y)\beta(y) \neq (\exists x)(\alpha(x) \wedge \beta(x))$

Case of Finite Domain

- When the domain D is finite, we can actually list of objects so that ∀ can be expressed by ∧ and ∃ can be expressed by ∨. For example, suppose that $D = \{1, 2, \cdots, k\}$. Then
$$\begin{aligned}(\forall x)P(x) =& P(1) \wedge P(2) \wedge \cdots \wedge P(k) = \bigwedge_{i \in D} P(i)\\ (\exists x)P(x) =& P(1) \vee P(2) \vee \cdots \vee P(k) = \bigvee_{i \in D} P(i)\end{aligned}$$
- Suppose that the domain only has two elements, e.g., $D = \{1, 2\}$. Then we have

① $(\forall x)(\forall y)P(x, y) = (\forall y)P(1, y) \wedge (\forall y)P(2, y) = P(1, 1) \wedge P(1, 2) \wedge P(2, 1) \wedge P(2, 2)$

② $(\exists x)(\forall y)P(x, y) = (\forall y)P(1, y) \vee (\forall y)P(2, y) = (P(1, 1) \wedge P(1, 2)) \vee (P(2, 1) \wedge P(2, 2))$

③ $(\forall y)(\exists x)P(x, y) = (\exists x)P(x, 1) \wedge (\exists x)P(x, 2) = (P(1, 1) \vee P(2, 1)) \wedge (P(1, 2) \vee P(2, 2))$

④ $(\exists x)(\exists y)P(x, y) = (\exists x)P(x, 1) \vee (\exists x)P(x, 2) = P(1, 1) \vee P(1, 2) \vee P(2, 1) \vee P(2, 2)$

By observing the above finite domain case, we have the following conclusions:

  - By comparing ② and ③, we have $(\exists x)(\forall y)P(x, y) \Rightarrow (\forall y)(\exists x)P(x, y)$. This is understandable because if there exists someone who loves everyone, then everyone must have someone that love him/her.
  - However, the converse of the above is not true, i.e., $(\forall y)(\exists x)P(x, y) \not\Rightarrow (\exists x)(\forall y)P(x, y)$. For example, we can consider predicate $P(x, y) : x + y = 0$. Then $(\forall y)(\exists x)P(x, y)$ holds but $(\exists x)(\forall y)P(x, y)$ does not hold. This tells that quantifiers "∃" and "∀" cannot be changed freely.

Inference in Predicate Logic

- Now we discuss inference in predicate logic. The definition is exactly the same as the case of propositional logic, i.e.,

  $$\alpha \Rightarrow \beta \quad \text{iff} \quad \beta \text{ is } \mathsf{true} \text{ whenever } \alpha \text{ is } \mathsf{true} \quad \text{iff} \quad \alpha \to \beta \text{ is universally valid}$$

  For example, the following inferences are all correct

  1. $(\forall x)(\mathsf{Int}(x) \to \mathsf{Rat}(x)) \wedge (\forall x)(\mathsf{Rat}(x) \to \mathsf{Real}(x)) \Rightarrow (\forall x)(\mathsf{Int}(x) \to \mathsf{Real}((x))$
  2. $(\forall x)(\mathsf{Man}(x) \to \mathsf{Diel}(x)) \wedge \mathsf{Man}(\mathsf{Alice}) \Rightarrow \mathsf{Die}(\mathsf{Alice})$
  3. $(\exists x)(\mathsf{Tall}(x) \wedge \mathsf{Handsome}(x) \wedge \mathsf{Rich}(x)) \Rightarrow (\exists x)\mathsf{Tall}(x) \wedge (\exists x)\mathsf{Rich}(x) \wedge (\exists x)\mathsf{Handsome}(x)$

- In the followings, we summarize some **Basic Inference Rules**.

  ① $(\forall x)P(x) \vee (\forall x)Q(x) \Rightarrow (\forall x)(P(x) \vee Q(x))$
  ② $(\exists x)(P(x) \wedge Q(x)) \Rightarrow (\exists x)P(x) \wedge (\exists x)Q(x)$
  ③ $(\forall x)(P(x) \to Q(x)) \Rightarrow (\forall x)P(x) \to (\forall x)Q(x)$
  ④ $(\forall x)(P(x) \to Q(x)) \Rightarrow (\exists x)P(x) \to (\exists x)Q(x)$
  ⑤ $(\forall x)(P(x) \leftrightarrow Q(x)) \Rightarrow (\forall x)P(x) \leftrightarrow (\forall x)Q(x)$
  ⑥ $(\forall x)(P(x) \leftrightarrow Q(x)) \Rightarrow (\exists x)P(x) \leftrightarrow (\exists x)Q(x)$
  ⑦ $(\forall x)(P(x) \to Q(x)) \wedge (\forall x)(Q(x) \to R(x)) \Rightarrow (\forall x)(P(x) \to R(x))$
  ⑧ $(\forall x)(P(x) \to Q(x)) \wedge P(a) \Rightarrow Q(a))$
  ⑨ $(\forall x)(\forall y)P(x,y) \Rightarrow (\exists x)(\forall y)P(x,y)$
  ⑩ $(\exists x)(\forall y)P(x,y) \Rightarrow (\forall y)(\exists x)P(x,y)$

- **Proof of ③**: It suffices to prove $(\forall x)(P(x) \to Q(x)) \wedge (\forall x)P(x) \Rightarrow (\forall x)Q(x)$. We assume that the LHS of the above is true. Then we know that

  – $(\forall x)P(x) = \mathsf{T}$, which means that $P(a)$ for any $a \in D$; and

  – $(\forall x)(P(x) \to Q(x)) = \mathsf{T}$, which means that $P(a) \to Q(a)$ for any $a \in D$.

  Then the above together imply that $Q(a)$ for any $a \in D$, i.e., $(\forall x)Q(x) = \mathsf{T}$.

- **Proof of ④**: It suffices to prove $(\forall x)(P(x) \to Q(x)) \wedge (\exists x)P(x) \Rightarrow (\exists x)Q(x)$. We assume that the LHS of the above is true. Then we know that

  – $(\exists x)P(x) = \mathsf{T}$, which means that $P(a)$ for some specific $a \in D$; and

  – $(\forall x)(P(x) \to Q(x)) = \mathsf{T}$, which means that $P(a) \to Q(a)$ for any $a \in D$.

  Then the above together imply that $Q(a)$ for the same $a \in D$, i.e., $(\exists x)Q(x) = \mathsf{T}$.

Inference Calculus

- The inference procedures for predicate formulas are similar to those in propositional formulas. The only difference is that we should use the definitions of the quantifiers, which are summarized as the following rules.

① **Universal Instatiation:** from $(\forall x)\alpha(x)$, we have $\alpha(x)$ (for any $x \in D$)
② **Universal Generation:** from $\alpha(x)$, we have $(\forall x)\alpha(x)$
③ **Existential Instatiation** from $(\exists x)\alpha(x)$, we have $\alpha(a)$ for some $a \in D$
④ **Existential Generation:** from $\alpha(a)$, we have $(\exists x)\alpha(x)$

Example: Predicate Inference

Let us consider the following inference:
"Some students like all teachers and no student likes bad teacher. Therefore, there is no bad teacher."
We define the following predicates:

$S(x)$: "x is a stud." $T(x)$: "x is a teac." $B(x)$: "x is bad" $L(x,y)$: "x likes y"

Then we need to prove the following inference:

- **Premises:** ① $(\exists x)(S(x) \wedge (\forall y)(T(y) \to L(x,y))$
  ② $(\forall x)(S(x) \to (\forall y)((T(y) \wedge B(y)) \to \neg L(x,y)))$.
- **Conclusion:** $(\forall x)(T(x) \to \neg B(x))$.

The proof procedure is as follows:
(1) $(\exists x)(S(x) \wedge (\forall y)(T(y) \to L(x,y))$.
(2) $S(a) \wedge (\forall y)(T(y) \to L(a,y))$ for some $a \in D$.
(3) $(\forall x)(S(x) \to (\forall y)((T(y) \wedge B(y)) \to \neg L(x,y)))$.
(4) $S(x) \to (\forall y)((T(y) \wedge B(y)) \to \neg L(x,y))$.
(5) $S(a) \to (\forall y)((T(y) \wedge B(y)) \to \neg L(a,y))$.
(6) $S(a)$.
(7) $(\forall y)((T(y) \wedge B(y)) \to \neg L(a,y))$.
(8) $(T(y) \wedge B(y)) \to \neg L(a,y)$.
(9) $(\forall y)(T(y) \to L(a,y))$.
(10) $T(y) \to L(a,y)$.
(11) $\neg L(a,y) \to \neg T(y)$.
(12) $(T(y) \wedge B(y)) \to \neg T(y)$.
(13) $T(y) \to \neg T(y) \vee \neg B(y)$.
(14) $T(y) \to \neg B(y)$.
(15) $(\forall y)(T(y) \to \neg B(y))$.

Prenex Normal Form

- In a predicate logic formula, according to the WFF, quantifiers can be put in any feasible places such as $(\forall x)(P(x) \to (\exists y)Q(x,y))$. In some cases, however, it will be very useful if we can put all quantifiers at the very beginning in order. This is called the **Prenex Normal Form (前束范式)** defined as follows.

Definition: Prenex Normal Form

A predicate formula α is said fo be in Prenex Normal Form (PNF) if it is in the form of

$$\alpha = (Q_1 x_1)(Q_2 x_2) \cdots (Q_n x_n) M\left(x_1, x_2, \cdots, x_n\right),$$

where Q_i is a quantifier, either "$\exists$" or "$\forall$" and $M(x_1, x_2, \ldots, x_n)$ is a predicate with no quantifier, which is called the matrix (母基).

- For example, $(\forall x)(P(x) \to (\exists y)Q(x,y))$ is not in PNF. However, we can convert it to an equivalent formula in PNF, e.g., $(\forall x)(\exists y)(P(x) \to Q(x,y))$. In fact, any predicate formula α has an equivalent PNF and we can take the following transformation steps

Procedure: Obtain Equivalent PNF

1. Cancel " $\leftrightarrow$ " and " $\to$ "
2. Move "$\neg$" inner using De Morgan's Law.
3. Move quantifiers to the left using distributive laws.
4. Rename if necessary.

- **Example:** Let us consider the follow transformation

$$\begin{aligned}
&(\forall x)(\forall y)((\exists z)(P(x,z) \wedge P(y,z)) \to (\exists u)Q(x,y,u))\\
=&(\forall x)(\forall y)(\neg(\exists z)(P(x,z) \wedge P(y,z)) \vee (\exists u)Q(x,y,u))\\
=&(\forall x)(\forall y)((\forall z)\underbrace{(\neg P(x,z) \vee \neg P(y,z))}_{\alpha(z)} \vee \underbrace{(\exists u)Q(x,y,u)}_{\beta})\\
=&(\forall x)(\forall y)(\forall z)(\underbrace{(\neg P(x,z) \vee \neg P(y,z))}_{\beta} \vee \underbrace{(\exists u)Q(x,y,u)}_{\alpha(u)})\\
=&(\forall x)(\forall y)(\forall z)(\exists u)(\underbrace{\neg P(x,z) \vee \neg P(y,z) \vee Q(x,y,u)}_{M(x,y,z,u)})
\end{aligned}$$

In fact, if we just need PNF, we do not really need to cancel "$\to$" (but should be careful when using the distributive law). For example, we can also directly have

$$\begin{aligned}
&(\forall x)(\forall y)((\exists z)(P(x,z) \wedge P(y,z)) \to (\exists u)Q(x,y,u))\\
=&(\forall x)(\forall y)(\exists u)((\exists z)(P(x,z) \wedge P(y,z)) \to Q(x,y,u))\\
=&(\forall x)(\forall y)(\exists u)(\forall z)((P(x,z) \wedge P(y,z)) \to Q(x,y,u))
\end{aligned}$$

Skolem Normal Form

- In general, the universal quantifier and the existential quantifier may alternate many times making the analysis of the formula very difficult. Here, we consider a more restricted form.

Definition: Skolem Normal Form

A Skolem normal form is a PNF without existential quantifiers, i.e., $\alpha = (\forall x_1)(\forall x_2)\cdots(\forall x_n)M(x_1, x_2, \cdots, x_n)$.

- Clearly, for any predicate formula α, we may not find a Skolem normal form that is equivalent to α. However, we can always transfer it to a Skolem normal form α' such that α is satisfiable iff α' is satisfiable.

Procedure: Obtain Skolem Normal Form

1. Convert α to PNF $(Q_i x_1)\cdots(Q_n x_n)M(x_1, \cdots, x_n)$.
2. For each "$(\exists x_i)$", remove the quantifier, and replace variable x_i in $M(x_1, \ldots, x_n)$ by a function of those variables bound by universal quantifiers *in front of* $(\exists x_i)$.

- For example, for PNF $\alpha = (\forall x)(\exists y)P(x, y)$, its Skolem normal form is $\alpha' = (\forall x)P(x, f(x))$. This is because "*for any x, there exists y*", which means that the value of y depends on the value of x you choose first.

  Clearly, the satisfiabilities of α and α' are the same. To see this more clearly, we assume that $D = \{1, 2\}$. Then we have

$$\begin{aligned} \alpha &= (\forall x)(\exists y)P(x,y) = (P(1,1) \vee P(1,2)) \wedge (P(2,1) \vee P(2,2)) \\ \alpha' &= (\forall x)P(x, f(x)) = \qquad P(1, f(1)) \qquad \wedge \qquad P(2, f(2)) \end{aligned}$$

  If α is satisfiable, then it naturally gives us a function $f : D \to D$ making α' satisfiable. On the other hand, if α' is satisfiable, the existence of function f makes each disjunctive clause in α satisfied.

- Note that, if there exists no universal quantifier before $(\forall x)$, then object variable x can just be replaced by an object constant because a function without varible is just a constant. For example, for PNF $(\exists x)(\forall y)P(x, y)$, its Skolem normal form is $(\forall y)P(a, y)$.

- Let us consider the following more complicated example, we can write the Skolem normal form by

$$\begin{aligned} &(\exists x)(\forall y)(\forall z)(\exists u)(\forall v)(\exists w)P(x,y,z,u,v,w) \\ \rightsquigarrow\ &(\forall y)(\forall z)(\exists u)(\forall v)(\exists w)P(a,y,z,u,v,w) \\ \rightsquigarrow\ &(\forall y)(\forall z)(\forall v)(\exists w)P(a,y,z,f(y,z),v,w) \\ \rightsquigarrow\ &(\forall y)(\forall)z(\forall v)P(a,y,z,f(y,z),v,g(y,z,v)) \end{aligned}$$

Resolution Method for Predicate Logic

- To prove inferences for predicate formulas, we can also use the resolution method. The basic idea is similar to the case of propositional logic, i.e., to prove that $\alpha \Rightarrow \beta$, it suffices to prove that $\alpha \wedge \neg\beta$ unsatisfiable. However, here we need one more thing: "$\alpha \wedge \neg\beta$ is unsatisfiable if and only if its Skolem normal form is unsatisfiable".

Resolution Method

Step 1: Write the Skolem Normal Form for $\alpha \wedge \neg\beta$, e.g., $G^* = (\forall x_1)\ldots(\forall x_n)M(\cdot)$.

Step 2: Write matrix M in CNF, e.g., $M = C_1 \wedge C_2 \wedge \ldots \wedge C_n$.

- note that in clause set $S = \{C_1, C_2, \ldots, C_n\}$, each variable is bound by $\forall$.

Step 3: Resolve for S as follows:

- a **substitution (置换)** $\sigma = \{x/a\}$ is an operator that replaces each symbol x in a formula by a new symbol a.

For $L = P(x) \to Q(x, y)$ and $\sigma = \{x/a\}$, we have $L\sigma = P(a) \to Q(a, y)$.

- for $C_1 = L_1 \vee C_1'$ and $C_2 = \neg L_2 \vee C_2'$. If exists a substitution σ, such that $L_1\sigma = L_2\sigma$, then C_1 and C_2 are resolved by $C_1'\sigma \vee C_2'\sigma$.

For $C_1 = P(x) \vee Q(x)$ and $C_2 = \neg P(a) \vee R(y)$, by taking $\sigma = \{x/a\}$, we have $P(x)\sigma = P(a)$ and $P(a)\sigma = P(a)$. This gives us a new clause $Q(x)\sigma \vee R(y)\sigma = Q(a) \vee R(y)$.

Step 4: Repeat until we find a contradiction.

- **Example:** $(\forall x)(P(x) \to Q(x)) \wedge (\forall x)(Q(x) \to R(x)) \Rightarrow (\forall x)(P(x) \to R(x))$.

First, we write the Skolem normal form of $\alpha \wedge \neg\beta$

$$\begin{aligned}
&(\forall x)(P(x)\to Q(x)) \wedge (\forall x)(Q(x)\to R(x)) \wedge \neg(\forall x)(P(x)\to R(x))\\
=&(\forall x)(\neg P(x) \vee Q(x)) \wedge (\forall x)(\neg Q(x) \vee R(x)) \wedge (\exists x)(P(x) \wedge \neg R(x))\\
=&(\forall x)(\forall y)(\exists z)\,((\neg P(x) \vee Q(x)) \wedge (\neg Q(y) \vee R(y)) \wedge P(z) \wedge \neg R(z))\\
\approx&(\forall x)(\forall y)((\neg P(x) \vee Q(x)) \wedge (\neg Q(y) \vee R(y)) \wedge P(a) \wedge \neg R(a))
\end{aligned}$$

Then the clause set is $S = \{\neg P(x) \vee Q(x), \neg Q(x) \vee R(y), P(a), \neg R(a)\}$.

$$\left.\begin{array}{r}\neg P(x) \vee Q(x)\\ P(a)\end{array}\right\} \xrightarrow{\sigma=\{x/a\}} Q(a), \qquad \left.\begin{array}{r}Q(a)\\ \neg Q(y) \vee R(y)\end{array}\right\} \xrightarrow{\sigma=\{y/a\}} R(a)$$

This gives us a contradiction $R(a)$ and $\neg R(a)$.