Predicate Logic

Lucia Moura

Winter 2017

Predicates

A Predicate is a declarative sentence whose true/false value depends on one or more variables.

The statement "x is greater than 3" has two parts:

- the subject: x is the subject of the statement
- the predicate: "is greater than 3" (a property that the subject can have).

We denote the statement "x is greater than 3" by $P(x)$, where P is the predicate "is greater than 3" and x is the variable.

The statement $P(x)$ is also called the value of **propositional function** P at x.

Assign a value to x, so $P(x)$ becomes a proposition and has a truth value:
$P(5)$ is the statement "5 is greater than 3", so $P(5)$ is true.
$P(2)$ is the statement "2 is greater than 3", so $P(2)$ is false.

Predicates: Examples

Given each propositional function determine its true/false value when variables are set as below.

- Prime(x) = "x is a prime number."
  Prime(2) is true, since the only numbers that divide 2 are 1 and itself.
  Prime(9) is false, since 3 divides 9.
- $C(x, y)$="x is the capital of y".
  C(Ottawa,Canada) is true.
  C(Buenos Aires,Brazil) is false.
- $E(x, y, z)$ = "$x + y = z$".
  $E(2, 3, 5)$ is ...
  $E(4, 4, 17)$ is ...

Quantifiers

Assign a value to x in $P(x) =$"x is an odd number", so the resulting statement becomes a proposition: $P(7)$ is true, $P(2)$ is false.

Quantification is another way to create propositions from a propositional functions:

- **universal** quantification: $\forall x P(x)$ says
  "the predicate P is true for every element under consideration."
  Under the domain of natural numbers, $\forall x P(x)$ is false.
- **existencial** quantification: $\exists x P(x)$ says
  "there is one or more element under consideration for which the predicate P is true."
  Under the domain of natural numbers, $\exists x P(x)$ is true, since for instance $P(7)$ is true.

Predicate calculus: area of logic dealing with predicates and quantifiers.

Domain, domain of discourse, universe of discourse

Before deciding on the truth value of a quantified predicate, it is mandatory to specify the **domain** (also called domain of discourse or universe of discourse).

$P(x) =$"x is an odd number"

$\forall x P(x)$ is **false** for the domain of **integer numbers**; but
$\forall x P(x)$ is **true** for the domain of **prime numbers greater than 2**.

Universal Quantifier

The **universal quantification** of $P(x)$ is the statement:
"$P(x)$ for all values of x in the domain" denoted $\forall x P(x)$.

$\forall x P(x)$ is **true** when $P(x)$ is true for every x in the domain.

$\forall x P(x)$ is **false** when there is an x for which $P(x)$ is false.
An element for which $P(x)$ is false is called a **counterexample** of $\forall x P(x)$.

If the domain is empty, $\forall x P(x)$ is true for any propositional function $P(x)$, since there are no counterexamples in the domain.

If the domain is finite $\{x_1, x_2, \ldots, x_n\}$, $\forall x P(x)$ is the same as

$$P(x_1) \wedge P(x_2) \wedge \cdots \wedge P(x_n).$$

Universal quantifiers: example

- Let $P(x)$ be "$x^2 > 10$".
  What is the truth value of $\forall x P(x)$ for each of the following domains:

Universal quantifiers: example

- Let $P(x)$ be "$x^2 > 10$".
  What is the truth value of $\forall x P(x)$ for each of the following domains:
  - the set of real numbers: $\mathbb{R}$

Universal quantifiers: example

- Let $P(x)$ be "$x^2 > 10$".
  What is the truth value of $\forall x P(x)$ for each of the following domains:
  - the set of real numbers: $\mathbb{R}$
  - False. 3 is a counterexample.

Universal quantifiers: example

- Let $P(x)$ be "$x^2 > 10$".
  What is the truth value of $\forall x P(x)$ for each of the following domains:
  - the set of real numbers: $\mathbb{R}$
  - False. 3 is a counterexample.
  - the set of positive integers not exceeding 4: $\{1, 2, 3, 4\}$

Universal quantifiers: example

- Let $P(x)$ be "$x^2 > 10$".
  What is the truth value of $\forall x P(x)$ for each of the following domains:
  - the set of real numbers: $\mathbb{R}$
  - False. 3 is a counterexample.
  - the set of positive integers not exceeding 4: $\{1, 2, 3, 4\}$
  - False. 3 is a counterexample.

Universal quantifiers: example

- Let $P(x)$ be "$x^2 > 10$".
  What is the truth value of $\forall x P(x)$ for each of the following domains:
  - the set of real numbers: $\mathbb{R}$
  - False. 3 is a counterexample.
  - the set of positive integers not exceeding 4: $\{1, 2, 3, 4\}$
  - False. 3 is a counterexample.
  - Also note that here $\forall P(x)$ is $P(1) \wedge P(2) \wedge P(3) \wedge P(4)$, so its enough to observe that $P(3)$ is false.

Universal quantifiers: example

- Let $P(x)$ be "$x^2 > 10''$.
  What is the truth value of $\forall x P(x)$ for each of the following domains:
  - the set of real numbers: $\mathbb{R}$
  - False. 3 is a counterexample.
  - the set of positive integers not exceeding 4: $\{1, 2, 3, 4\}$
  - False. 3 is a counterexample.
  - Also note that here $\forall P(x)$ is $P(1) \wedge P(2) \wedge P(3) \wedge P(4)$, so its enough to observe that $P(3)$ is false.
  - the set of real numbers in the interval $[10, 39.5]$

Universal quantifiers: example

- Let $P(x)$ be "$x^2 > 10$".
  What is the truth value of $\forall x P(x)$ for each of the following domains:
  - the set of real numbers: $\mathbb{R}$
  - False. 3 is a counterexample.
  - the set of positive integers not exceeding 4: $\{1, 2, 3, 4\}$
  - False. 3 is a counterexample.
  - Also note that here $\forall P(x)$ is $P(1) \wedge P(2) \wedge P(3) \wedge P(4)$, so its enough to observe that $P(3)$ is false.
  - the set of real numbers in the interval $[10, 39.5]$
  - True. It takes a bit longer to verify than in false statements.
    Let $x \in [10, 39.5]$. Then $x \geq 10$ which implies $x^2 \geq 10^2 = 100 > 10$, and so $x^2 > 10$.

Existencial Quantifier

The **existential quantification** of $P(x)$ is the statement:
"There exists an element x in the domain such that $P(x)$" denoted $\exists x P(x)$.

$\exists x P(x)$ is **true** when $P(x)$ is true for one or more x in the domain.
An element for which $P(x)$ is true is called a **witness** of $\exists x P(x)$.

$\exists x P(x)$ is **false** when $P(x)$ is false for every x in the domain (if domain nonempty).

If the domain is empty, $\exists x P(x)$ is false for any propositional function $P(x)$, since there are no witnesses in the domain.

If the domain is finite $\{x_1, x_2, \ldots, x_n\}$, $\exists x P(x)$ is the same as

$$P(x_1) \vee P(x_2) \vee \cdots \vee P(x_n).$$

Existencial quantifiers: example

- Let $P(x)$ be "$x^2 > 10$".
  What is the truth value of $\exists x P(x)$ for each of the following domains:

Existencial quantifiers: example

- Let $P(x)$ be "$x^2 > 10$".
  What is the truth value of $\exists x P(x)$ for each of the following domains:
  - the set of real numbers: $\mathbb{R}$

Existencial quantifiers: example

- Let $P(x)$ be "$x^2 > 10$".
  What is the truth value of $\exists x P(x)$ for each of the following domains:
  - the set of real numbers: $\mathbb{R}$
  - True. 10 is a witness.

Existencial quantifiers: example

- Let $P(x)$ be "$x^2 > 10$".
  What is the truth value of $\exists x P(x)$ for each of the following domains:
  - the set of real numbers: $\mathbb{R}$
  - True. 10 is a witness.
  - the set of positive integers not exceeding 4: $\{1, 2, 3, 4\}$

Existencial quantifiers: example

- Let $P(x)$ be "$x^2 > 10''$.
  What is the truth value of $\exists x P(x)$ for each of the following domains:
  - the set of real numbers: $\mathbb{R}$
  - True. 10 is a witness.
  - the set of positive integers not exceeding 4: $\{1, 2, 3, 4\}$
  - True. 4 is a witness.

Existencial quantifiers: example

- Let $P(x)$ be "$x^2 > 10''$.
  What is the truth value of $\exists x P(x)$ for each of the following domains:
  - the set of real numbers: $\mathbb{R}$
  - True. 10 is a witness.
  - the set of positive integers not exceeding 4: $\{1, 2, 3, 4\}$
  - True. 4 is a witness.
  - Also note that here $\exists P(x)$ is $P(1) \vee P(2) \vee P(3) \vee P(4)$, so its enough to observe that $P(4)$ is true.

Existencial quantifiers: example

- Let $P(x)$ be "$x^2 > 10''$.
  What is the truth value of $\exists x P(x)$ for each of the following domains:
  - the set of real numbers: $\mathbb{R}$
  - True. 10 is a witness.
  - the set of positive integers not exceeding 4: $\{1, 2, 3, 4\}$
  - True. 4 is a witness.
  - Also note that here $\exists P(x)$ is $P(1) \vee P(2) \vee P(3) \vee P(4)$, so its enough to observe that $P(4)$ is true.
  - the set of real numbers in the interval $[0, \sqrt{9.8}]$

Existencial quantifiers: example

- Let $P(x)$ be "$x^2 > 10$".
  What is the truth value of $\exists x P(x)$ for each of the following domains:
  - the set of real numbers: $\mathbb{R}$
  - True. 10 is a witness.
  - the set of positive integers not exceeding 4: $\{1, 2, 3, 4\}$
  - True. 4 is a witness.
  - Also note that here $\exists P(x)$ is $P(1) \vee P(2) \vee P(3) \vee P(4)$, so its enough to observe that $P(4)$ is true.
  - the set of real numbers in the interval $[0, \sqrt{9.8}]$
  - False. It takes a bit longer to conclude than in true statements.
    Let $x \in [0, 9.8]$. Then $0 \leq x \leq \sqrt{9.8}$ which implies
    $x^2 \leq (\sqrt{9.8})^2 = 9.8 < 10$, and so $x^2 < 10$.
    What we have shown is that $\forall x \neg P(x)$, which (we will see) is equivalent to $\neg \exists x P(x)$

Other forms of quantification

- **Other Quantifiers**
  The most important quantifiers are $\forall$ and $\exists$, but we could define many different quantifiers: "there is a unique", "there are exactly two", "there are no more than three", "there are at least 100", etc.
  A common one is the **uniqueness quantifier**, denoted by $\exists!$.
  $\exists! x P(x)$ states "There exists a unique x such that $P(x)$ is true."
  Advice: stick to the basic quantifiers. We can write $\exists! x P(x)$ as
  $\exists x(P(x) \wedge \forall y(P(y) \rightarrow y = x))$ or more compactly
  $\exists x \forall y(P(y) \leftrightarrow y = x)$
- **Restricting the domain of a quantifier**
  Abbreviated notation is allowed, in order to restrict the domain of certain quantifiers.
  - $\forall x > 0(x^2 > 0)$ is the same as $\forall x(x > 0 \rightarrow x^2 > 0)$.
  - $\forall y \neq 0(y^3 \neq 0)$ is the same as $\forall y(y \neq 0 \rightarrow y^3 \neq 0)$.
  - $\exists z > 0(z^2 = 2)$ is the same as $\exists z(z > 0 \wedge z^2 = 2)$

Precedence and scope of quantifiers

- $\forall$ and $\exists$ have higher precedence than logical operators.
  Example: $\forall x P(x) \vee Q(x)$ means $(\forall x P(x)) \vee Q(x)$, it doesn't mean $\forall x (P(x) \vee Q(x))$.
  (Note: This statement is not a proposition since there is a free variable!)
- **Binding variables and scope**
  When a quantifier is used on the variable x we say that this occurrence of x is **bound**. When the occurrence of a variable is not bound by a quantifier or set to a particular value, the variable is said to be **free**.
  The part of a logical expression to which a quantifier is applied is the **scope** of the quantifier. A variable is free if it is outside the scope of all quantifiers.
  In the example above, $(\forall x \underline{P(x)}) \vee Q(x)$, the x in $P(x)$ is bound by the existencial quantifier, while the x in $Q(x)$ is free. The scope of the universal quantifier is underlined.

Logical Equivalences Involving Quantifiers

Definition

Two statements S and T involving predicates and quantifiers are *logically equivalent* if and only if they have the same truth value regardless of the **interpretation**, i.e. regardless of

- the meaning that is attributed to each propositional function,
- the domain of discourse.

We denote $S \equiv T$.

Is $\forall x(P(x) \land Q(x))$ **logically equivalent** to $\forall x P(x) \land \forall x Q(x)$?
Is $\forall x(P(x) \lor Q(x))$ **logically equivalent** to $\forall x P(x) \lor \forall x Q(x)$?

- Prove that $\forall x(P(x) \wedge Q(x))$ is logically equivalent to $\forall x P(x) \wedge \forall x Q(x)$ (where the same domain is used throughout).

- Prove that $\forall x(P(x) \land Q(x))$ is logically equivalent to $\forall x P(x) \land \forall x Q(x)$ (where the same domain is used throughout).
- Use two steps:

- Prove that $\forall x(P(x) \wedge Q(x))$ is logically equivalent to $\forall x P(x) \wedge \forall x Q(x)$ (where the same domain is used throughout).
- Use two steps:
  - If $\forall x(P(x) \wedge Q(x))$ is true, then $\forall x P(x) \wedge \forall x Q(x)$ is true.

- Prove that $\forall x(P(x) \land Q(x))$ is logically equivalent to $\forall x P(x) \land \forall x Q(x)$ (where the same domain is used throughout).
- Use two steps:
  - If $\forall x(P(x) \land Q(x))$ is true, then $\forall x P(x) \land \forall x Q(x)$ is true.
  - Proof: Suppose $\forall x(P(x) \land Q(x))$ is true.
    Then if a is in the domain, $P(a) \land Q(a)$ is true, and so $P(a)$ is true and $Q(a)$ is true.
    So, if a in in the domain $P(a)$ is true, which is the same as $\forall x P(x)$ is true; and similarly, we get that $\forall x Q(x)$ is true.
    This means that $\forall x P(x) \land \forall x Q(x)$ is true.

- Prove that $\forall x(P(x) \wedge Q(x))$ is logically equivalent to $\forall x P(x) \wedge \forall x Q(x)$ (where the same domain is used throughout).
- Use two steps:
  - If $\forall x(P(x) \wedge Q(x))$ is true, then $\forall x P(x) \wedge \forall x Q(x)$ is true.
  - Proof: Suppose $\forall x(P(x) \wedge Q(x))$ is true.
    Then if a is in the domain, $P(a) \wedge Q(a)$ is true, and so $P(a)$ is true and $Q(a)$ is true.
    So, if a in in the domain $P(a)$ is true, which is the same as $\forall x P(x)$ is true; and similarly, we get that $\forall x Q(x)$ is true.
    This means that $\forall x P(x) \wedge \forall x Q(x)$ is true.
  - If $\forall x P(x) \wedge \forall x Q(x)$ is true, then $\forall x(P(x) \wedge Q(x))$ is true.

- Prove that $\forall x(P(x) \wedge Q(x))$ is logically equivalent to $\forall x P(x) \wedge \forall x Q(x)$ (where the same domain is used throughout).
- Use two steps:
  - If $\forall x(P(x) \wedge Q(x))$ is true, then $\forall x P(x) \wedge \forall x Q(x)$ is true.
  - Proof: Suppose $\forall x(P(x) \wedge Q(x))$ is true.
    Then if a is in the domain, $P(a) \wedge Q(a)$ is true, and so $P(a)$ is true and $Q(a)$ is true.
    So, if a in in the domain $P(a)$ is true, which is the same as $\forall x P(x)$ is true; and similarly, we get that $\forall x Q(x)$ is true.
    This means that $\forall x P(x) \wedge \forall x Q(x)$ is true.
  - If $\forall x P(x) \wedge \forall x Q(x)$ is true, then $\forall x(P(x) \wedge Q(x))$ is true.
  - Proof: Suppose that $\forall x P(x) \wedge \forall x Q(x)$ is true.
    It follows that $\forall x P(x)$ is true and $\forall x Q(x)$ is true.
    So, if a is in the domain, then $P(a)$ is true and $Q(a)$ is true.
    It follows that if a is in the domain $P(a) \wedge Q(a)$ is true.
    This means that $\forall x(P(x) \wedge Q(x))$ is true.

- Prove that $\forall x(P(x) \vee Q(x))$ **is not logically equivalent** to $\forall x P(x) \vee \forall x Q(x)$.

- Prove that $\forall x(P(x) \vee Q(x))$ **is not logically equivalent** to $\forall x P(x) \vee \forall x Q(x)$.
- It is enough to give a counterexample to the assertion that they have the same truth value for all possible interpretations.

- Prove that $\forall x(P(x) \vee Q(x))$ **is not logically equivalent** to $\forall x P(x) \vee \forall x Q(x)$.
- It is enough to give a counterexample to the assertion that they have the same truth value for all possible interpretations.
- Under the following interpretation:
  domain: set of people in the world
  $P(x) =$"x is male".
  $Q(x) =$"x is female".

  We have:
  $\forall x(P(x) \vee Q(x))$ (every person is a male or a female) is true;
  while $\forall x P(x) \vee \forall x Q(x)$ (every person is a male or every person is a female) is false.

Negating Quantified Expressions: De Morgan Laws

$$\neg\forall x P(x) \equiv \exists x \neg P(x)$$

Proof:
$\neg\forall x P(x)$ is true if and only if $\forall x P(x)$ is false.
Note that $\forall x P(x)$ is false if and only if there exists an element a in the domain for which $P(a)$ is false.
But this holds if and only if there exists an element a in the domain for which $\neg P(a)$ is true.
The latter holds if and only if $\exists x \neg P(x)$ is true.

De Morgan Laws for quantifiers (continued)

$$\neg\exists x P(x) \equiv \forall x \neg P(x)$$

Proof:
$\neg\exists P(x)$ is true if and only if $\exists x P(x)$ is false.
Note that $\exists x P(x)$ is false if and only there exists no element a in the domain for which $P(a)$ is true.
But this holds if and only if for all elements a in the domain we have $P(a)$ is false;
which is the same as for all elements a in the domain we have $\neg P(a)$ is true.
The latter holds if and only if $\forall x \neg P(x)$ is true.

Practice Exercises

1. What are the negations of the following statements:
   "There is an honest politician."
   "All americans eat cheeseburgers."
2. What are the negations of $\forall x(x^2 > x)$ and $\exists x(x^2 = 2)$?
3. Show that $\neg\forall x(P(x) \to Q(x))$ and $\exists x(P(x) \land \neg Q(x))$ are logically equivalent.

Example from Lewis Caroll's book *Symbolic Logic*

Consider these statements (two premises followed by a conclusion):
"All lions are fierce."
"Some lions do not drink coffee."
"Some fierce creatures do not drink coffee."

Assume that the domain is the set of all creatures and $P(x) =$"x is a lion", $Q(x) =$"x is fierce", $R(x) =$"x drinks coffee".

Exercise: Express the above statements using $P(x)$, $Q(x)$ and $R(x)$, under the domain of all creatures.

Is the conclusion a valid consequence of the premises?
In this case, yes. (See more on this type of derivation, in a future lecture on Rules of Inference).

Nested Quantifiers

Two quantifiers are nested if one is in the scope of the other.
Everything within the scope of a quantifier can be thought of as a propositional function.
For instance,

"$\forall x \exists y(x + y = 0)$" is the same as
"$\forall x Q(x)$", where $Q(x)$ is "$\exists y(x + y = 0)$".

The order of quantifiers

Let $P(x, y)$ be the statement "$x + y = y + x$".
Consider the following:
$\forall x \forall y P(x, y)$ and $\forall y \forall x P(x, y)$.

- What is the meaning of each of these statements?
- What is the truth value of each of these statements?
- Are they equivalent?

Let $Q(x, y)$ be the statement "$x + y = 0$".
Consider the following:
$\exists y \forall x Q(x, y)$ and $\forall x \exists y Q(x, y)$.

- What is the meaning of each of these statements?
- What is the truth value of each of these statements?
- Are they equivalent?

Summary of quantification of two variables

statement	when true ?	when false ?
$\forall x \forall y P(x,y)$ $\forall y \forall x P(x,y)$	$P(x,y)$ is true for every pair x, y.	There is a pair x, y for which $P(x,y)$ is false
$\forall x \exists y P(x,y)$	For every x there is y for which $P(x,y)$ is true	There is an x such that $P(x,y)$ is false for every y
$\exists x \forall y P(x,y)$	There is an x for which $P(x,y)$ is true for every y	For every x there is a y for which $P(x,y)$ is false.
$\exists x \exists y P(x,y)$ $\exists y \exists x P(x,y)$	There is a pair x, y for which $P(x,y)$ is true	$P(x,y)$ is false for every pair x, y.

Translating Math Statements into Nested quantifiers

Translate the following statements:

1. "The sum of two positive integers is always positive."
2. "Every real number except zero has a multiplicative inverse."
   (a multiplicative inverse of x is y such that $xy = 1$).
3. "Every positive integer is the sum of the squares of four integers."

Translating from Nested Quantifiers into English

Let $C(x)$ denote "x has a computer" and $F(x, y)$ be "x and y are friends.", and the domain be all students in your school.
Translate:

1. $\forall x(C(x) \vee \exists y(C(y) \wedge F(x, y))$.
2. $\exists x \forall y \forall z((F(x, y) \wedge F(x, z) \wedge (y \neq z)) \rightarrow \neg F(y, z))$

Translating from English into Nested Quantifiers

1. "If a person is female and is a parent, then this person is someone's mother."
2. "Everyone has exactly one best friend."
3. "There is a woman who has taken a flight on every airline of the world."

Negating Nested Quantifiers

Express the negation of the following statements, so that no negation precedes a quantifier (apply DeMorgan successively):

- $\forall x \exists y (xy = 1)$
- $\forall x \exists y P(x, y) \vee \forall x \exists y Q(x, y)$
- $\forall x \exists y (P(x, y) \wedge \exists z R(x, y, z))$

Predicate calculus in Mathematical Reasoning

Using predicates to express definitions.
$D(x) =$ "x is <u>a prime number</u>"
(defined term)
$P(x) =$ "<u>$x \geq 2$ and the only divisors of x are 1 and x</u>"
(defining property about x)
Definition of prime number: $\forall x(D(x) \leftrightarrow P(x))$

Note that definitions in English form use *if* instead of *if and only if*, but we really mean *if and only if*.

Predicate calculus in Mathematical Reasoning (cont'd)

Let $P(n, x, y, z)$ be the predicate $x^n + y^n = z^n$.

1. Write the following statements in predicate logic, using the domain of positive integers:
   "For every integer $n > 2$, there does not exist positive integers x, y and z such that $x^n + y^n = z^n$."
2. Negate the previous statement, and simplify it so that no negation precedes a quantifier.
3. What needs to be found in order to give a counter example to 1 ?
4. Which famous theorem is expressed in 1, who proved and when?

Predicate calculus in Program Verification: a toy example

The following program is designed to exchange the value of x and y:

```
temp := x
x := y
y := temp
```

Find preconditions, postconditions and verify its correctness.

- Precondition: $P(x, y)$ is "$x = a$ and $y = b$", where a and b are the values of x and y before we execute these 3 statements.

Predicate calculus in Program Verification: a toy example

The following program is designed to exchange the value of x and y:

```
temp := x
x := y
y := temp
```

Find preconditions, postconditions and verify its correctness.

- Precondition: $P(x, y)$ is "$x = a$ and $y = b$", where a and b are the values of x and y before we execute these 3 statements.
- Postconditon: $Q(x, y)$ is "$x = b$ and $y = a$".

Predicate calculus in Program Verification: a toy example

The following program is designed to exchange the value of x and y:

```
temp := x
x := y
y := temp
```

Find preconditions, postconditions and verify its correctness.

- Precondition: $P(x, y)$ is "$x = a$ and $y = b$", where a and b are the values of x and y before we execute these 3 statements.
- Postconditon: $Q(x, y)$ is "$x = b$ and $y = a$".
- Assume $P(x, y)$ holds before and show that $Q(x, y)$ holds after.

Predicate calculus in Program Verification: a toy example

The following program is designed to exchange the value of x and y:

```
temp := x
x := y
y := temp
```

Find preconditions, postconditions and verify its correctness.

- Precondition: $P(x, y)$ is "$x = a$ and $y = b$", where a and b are the values of x and y before we execute these 3 statements.
- Postconditon: $Q(x, y)$ is "$x = b$ and $y = a$".
- Assume $P(x, y)$ holds before and show that $Q(x, y)$ holds after.
- Originally $x = a$ and $y = b$, by $P(x, y)$.

Predicate calculus in Program Verification: a toy example

The following program is designed to exchange the value of x and y:

```
temp := x
x := y
y := temp
```

Find preconditions, postconditions and verify its correctness.

- Precondition: $P(x,y)$ is "$x = a$ and $y = b$", where a and b are the values of x and y before we execute these 3 statements.
- Postconditon: $Q(x,y)$ is "$x = b$ and $y = a$".
- Assume $P(x,y)$ holds before and show that $Q(x,y)$ holds after.
- Originally $x = a$ and $y = b$, by $P(x,y)$.
- After step 1, $x = a$, $temp = a$ and $y = b$.

Predicate calculus in Program Verification: a toy example

The following program is designed to exchange the value of x and y:

```
temp := x
x := y
y := temp
```

Find preconditions, postconditions and verify its correctness.

- Precondition: $P(x, y)$ is "$x = a$ and $y = b$", where a and b are the values of x and y before we execute these 3 statements.
- Postconditon: $Q(x, y)$ is "$x = b$ and $y = a$".
- Assume $P(x, y)$ holds before and show that $Q(x, y)$ holds after.
- Originally $x = a$ and $y = b$, by $P(x, y)$.
- After step 1, $x = a$, $temp = a$ and $y = b$.
- After step 2, $x = b$, $temp = a$ and $y = b$.

Predicate calculus in Program Verification: a toy example

The following program is designed to exchange the value of x and y:

```
temp := x
x := y
y := temp
```

Find preconditions, postconditions and verify its correctness.

- Precondition: $P(x, y)$ is "$x = a$ and $y = b$", where a and b are the values of x and y before we execute these 3 statements.
- Postconditon: $Q(x, y)$ is "$x = b$ and $y = a$".
- Assume $P(x, y)$ holds before and show that $Q(x, y)$ holds after.
- Originally $x = a$ and $y = b$, by $P(x, y)$.
- After step 1, $x = a$, $temp = a$ and $y = b$.
- After step 2, $x = b$, $temp = a$ and $y = b$.
- After step 3, $x = b$, $temp = a$ and $y = a$.

Predicate calculus in Program Verification: a toy example

The following program is designed to exchange the value of x and y:

```
temp := x
x := y
y := temp
```

Find preconditions, postconditions and verify its correctness.

- Precondition: $P(x, y)$ is "$x = a$ and $y = b$", where a and b are the values of x and y before we execute these 3 statements.
- Postconditon: $Q(x, y)$ is "$x = b$ and $y = a$".
- Assume $P(x, y)$ holds before and show that $Q(x, y)$ holds after.
- Originally $x = a$ and $y = b$, by $P(x, y)$.
- After step 1, $x = a$, $temp = a$ and $y = b$.
- After step 2, $x = b$, $temp = a$ and $y = b$.
- After step 3, $x = b$, $temp = a$ and $y = a$.
- Therefore, after the program we know $Q(x, y)$ holds.

Predicate calculus in System Specification

Use predicates and quantifiers to express system specifications:

1. “Every mail message larger than one megabyte will be compressed.”

Predicate calculus in System Specification

Use predicates and quantifiers to express system specifications:

1. "Every mail message larger than one megabyte will be compressed."
   - $S(m, y)$: "mail message m is larger than y megabytes"

Predicate calculus in System Specification

Use predicates and quantifiers to express system specifications:

1. "Every mail message larger than one megabyte will be compressed."
   - $S(m, y)$: "mail message m is larger than y megabytes"
   - $C(m)$: "mail message m will be compressed"

Predicate calculus in System Specification

Use predicates and quantifiers to express system specifications:

1. "Every mail message larger than one megabyte will be compressed."
   - $S(m, y)$: "mail message m is larger than y megabytes"
   - $C(m)$: "mail message m will be compressed"
   - $\forall m(S(m, 1) \rightarrow C(m))$

Predicate calculus in System Specification

Use predicates and quantifiers to express system specifications:

1. "Every mail message larger than one megabyte will be compressed."
   - $S(m, y)$: "mail message m is larger than y megabytes"
   - $C(m)$: "mail message m will be compressed"
   - $\forall m(S(m, 1) \rightarrow C(m))$
2. "If a user is active, at least one network link will be available."

Predicate calculus in System Specification

Use predicates and quantifiers to express system specifications:

1. "Every mail message larger than one megabyte will be compressed."
   - $S(m, y)$: "mail message m is larger than y megabytes"
   - $C(m)$: "mail message m will be compressed"
   - $\forall m(S(m, 1) \rightarrow C(m))$
2. "If a user is active, at least one network link will be available."
   - $A(u)$: "User u is active."

Predicate calculus in System Specification

Use predicates and quantifiers to express system specifications:

1. "Every mail message larger than one megabyte will be compressed."
   - $S(m, y)$: "mail message m is larger than y megabytes"
   - $C(m)$: "mail message m will be compressed"
   - $\forall m(S(m, 1) \rightarrow C(m))$
2. "If a user is active, at least one network link will be available."
   - $A(u)$: "User u is active."
   - $S(n, x)$: "Network n is in state x".

Predicate calculus in System Specification

Use predicates and quantifiers to express system specifications:

1. "Every mail message larger than one megabyte will be compressed."
   - $S(m, y)$: "mail message m is larger than y megabytes"
   - $C(m)$: "mail message m will be compressed"
   - $\forall m(S(m, 1) \rightarrow C(m))$
2. "If a user is active, at least one network link will be available."
   - $A(u)$: "User u is active."
   - $S(n, x)$: "Network n is in state x".
   - $\exists u A(u) \rightarrow \exists n S(n, available)$

Predicate calculus in Logic Programming

Prolog is a declarative language based in predicate logic.
The program is expressed as **Prolog facts** and **Prolog rules**.
Execution is triggered by running queries over these relations.

```
mother_child(trude, sally).

father_child(tom, sally).
father_child(tom, erica).
father_child(mike, tom).

sibling(X, Y)      :- parent_child(Z, X), parent_child(Z, Y).

parent_child(X, Y) :- father_child(X, Y).
parent_child(X, Y) :- mother_child(X, Y).
```

The result of the following query is given:

```
?- sibling(sally, erica).
Yes
```