CS208 (Semester 1) Topic 3 : Predicate Logic

Dr. Robert Atkey

Computer & Information Sciences

Predicate Logic, Part 1

Introduction

So far:

Propositional Logic

We can say things like:

"If it is raining or sunny, and it is not sunny, then it is raining"

$$((R \vee S) \wedge \neg S) \to R$$

"version 1 is installed, or version 2 is installed, or version 3 is installed"

$$p_1 \vee p_2 \vee p_3$$

What we can't say

"Every day is sunny or rainy, today is not sunny, so today is rainy"

- No way to make *universal* statements ("Every day")

"Some version of the package is installed"

- No way to make *existential* statements ("Some version")

What we can’t say

“Every day is sunny or rainy, today is not sunny, so today is rainy”

- No way to make *universal* statements (“Every day”)

“Some version of the package is installed”

- No way to make *existential* statements (“Some version”)

Best we can do is list the possibilities

$$(S_{\mathrm{monday}} \vee R_{\mathrm{monday}}) \wedge (S_{\mathrm{tuesday}} \vee R_{\mathrm{tuesday}}) \wedge \dots$$

Universal statements

"Classical" examples: (due to Aristole)

1. All human are mortal
2. Socrates is a human
3. Therefore Socrates is mortal

(from the universal to the specific)

1. No bird can fly in space
2. Owls are birds
3. Therefore owls cannot fly in space

Universal and Existential statements are common

Database queries:

"Does there exist a customer that has not paid their invoice?"

"Does there exist a player who is within 10 metres of player 1?"

"Are all players logged off?"

"Do we have any customers?"

Universal and Existential statements are common

The **semantics of Propositional Logic**:

"P is satisfiable if *there exists* a valuation that makes it true."

"P is valid if *all* valuations make it true."

"P entails Q if *for all* valuations, P is true implies Q is true."

Predicate Logic upgrades Propositional Logic

1. Add *individuals*:
   - Specific individuals (e.g., socrates, today, player1, 1, 2, 3)

     (these "name" specific entities in the world)
   - General individuals (x, y, z, ...)

     (like variables in programming, they stand for "some" individual)

2. Add *function symbols*:
   - $x + y$, $\mathsf{dayAfter}(\mathsf{today})$, $\mathsf{dayAfter}(x)$

3. Add *properties* and *relations*:
   - Properties: $\mathrm{canFlyInSpace}(\mathsf{owl})$, $\mathsf{paid}(i)$
   - Relations: $x = y$, $x \leq 10$, $\mathrm{custInvoice}(c, i)$.

4. Add *Quantifiers*:
   - Universal quantification: $\forall x.P$ ("for all" x, it is the case that P)
   - Existential quantification: $\exists x.P$ ("there exists" x, such that P)

Layered Syntax

The syntax of Predicate Logic comes in two layers:

Terms Built from individuals and function symbols:

$$x \qquad \mathsf{socrates} \qquad \mathsf{player1} \qquad \mathsf{dayAfter}(\mathsf{today}) \qquad 2x + 3y$$

$$\mathsf{nameOf}(\mathrm{cust}) \qquad \mathsf{dayAfter}(\mathsf{dayAfter}(d))$$

Formulas Built from properties and relations, connectives and quantifiers.

$$\exists x.\ \mathrm{customer}(x) \land \mathrm{loggedOff}(x)$$

$$\forall x.\ \mathrm{human}(x) \to \mathrm{mortal}(x)$$

Anatomy of a Formula

"All humans are mortal"

$$\forall x.\ \ \mathrm{human}\,(\,x\,) \to \mathrm{mortal}\,(\,x\,)$$

Anatomy of a Formula

"All humans are mortal"

$$\forall x.\ \ \mathrm{human}(x) \rightarrow \mathrm{mortal}(x)$$

1. Variables, standing for general individuals

Anatomy of a Formula

"All humans are mortal"

$$\forall x.\ \mathrm{human}(x) \to \mathrm{mortal}(x)$$

1. Variables, standing for general individuals
2. Properties ("Predicates") of those individuals

Anatomy of a Formula

"All humans are mortal"

$$\forall x.\ \mathrm{human}(x) \to \mathrm{mortal}(x)$$

1. Variables, standing for general individuals
2. Properties ("Predicates") of those individuals
3. Connectives, as in Propositional Logic

Anatomy of a Formula

"All humans are mortal"

$$\forall x.\ \mathrm{human}(x) \rightarrow \mathrm{mortal}(x)$$

1. Variables, standing for general individuals
2. Properties ("Predicates") of those individuals
3. Connectives, as in Propositional Logic
4. Quantifiers, telling us how to interpret the general individual x

Anatomy of a Formula

“Socrates is a human”

$$\mathrm{human}\,(\,\mathsf{socrates}\,)$$

Anatomy of a Formula

"Socrates is a human"

$$\text{human}(\,\texttt{socrates}\,)$$

1. A specific individual

Anatomy of a Formula

"Socrates is a human"

$$\text{human}(\mathsf{socrates})$$

1. A specific individual
2. Property of that individual

Anatomy of a Formula

"No bird can fly in space"

$$\neg (\exists x.\ \mathrm{bird} (x) \wedge \mathrm{canFlyInSpace} (x))$$

Anatomy of a Formula

"No bird can fly in space"

$$\neg(\exists x.\ \mathrm{bird}(x) \wedge \mathrm{canFlyInSpace}(x))$$

1. Variables, standing for general individuals

Anatomy of a Formula

"No bird can fly in space"

$$\neg(\exists x.\ \mathrm{bird}(x) \wedge \mathrm{canFlyInSpace}(x))$$

1. Variables, standing for general individuals
2. Properties ("Predicates") of those individuals

Anatomy of a Formula

"No bird can fly in space"

$$\neg(\exists x.\ \mathrm{bird}(x) \wedge \mathrm{canFlyInSpace}(x))$$

1. Variables, standing for general individuals
2. Properties ("Predicates") of those individuals
3. Connectives, as in Propositional Logic

Anatomy of a Formula

"No bird can fly in space"

$$\neg(\exists x.\ \mathrm{bird}(x) \wedge \mathrm{canFlyInSpace}(x))$$

1. Variables, standing for general individuals
2. Properties ("Predicates") of those individuals
3. Connectives, as in Propositional Logic
4. Quantifiers, telling us how to interpret the general individual x

Anatomy of a Formula

"If it is raining on a day, it is raining the day after"

$$\forall d.\ \mathrm{raining}(\mathrm{d}) \to \mathrm{raining}(\mathsf{dayAfter}(\mathrm{d}))$$

Anatomy of a Formula

"If it is raining on a day, it is raining the day after"

$$\forall d.\ \mathrm{raining}(d) \to \mathrm{raining}(\mathsf{dayAfter}(d))$$

1. Variables, standing for general individuals

Anatomy of a Formula

"If it is raining on a day, it is raining the day after"

$$\forall \mathrm{d}.\ \mathrm{raining}(\mathrm{d}) \rightarrow \mathrm{raining}(\mathrm{dayAfter}(\mathrm{d}))$$

1. Variables, standing for general individuals
2. Function symbols, performing operations on individuals

Anatomy of a Formula

"If it is raining on a day, it is raining the day after"

$$\forall d.\ \mathrm{raining}(d) \rightarrow \mathrm{raining}(\mathsf{dayAfter}(d))$$

1. Variables, standing for general individuals
2. Function symbols, performing operations on individuals
3. Properties ("Predicates") of those individuals

Anatomy of a Formula

"If it is raining on a day, it is raining the day after"

$$\forall d.\ \mathrm{raining}(d) \to \mathrm{raining}(\mathrm{dayAfter}(d))$$

1. Variables, standing for general individuals
2. Function symbols, performing operations on individuals
3. Properties ("Predicates") of those individuals
4. Connectives, as in Propositional Logic

Anatomy of a Formula

"If it is raining on a day, it is raining the day after"

$$\forall d.\ \mathrm{raining}(d) \to \mathrm{raining}(\mathrm{dayAfter}(d))$$

1. Variables, standing for general individuals
2. Function symbols, performing operations on individuals
3. Properties ("Predicates") of those individuals
4. Connectives, as in Propositional Logic
5. Quantifiers,telling us how to interpret the general individual d

Anatomy of a Formula

"Every number is even or odd"

$$\forall n.\ \exists k.\ (n = k + k) \vee (n = k + k + 1)$$

Anatomy of a Formula

"Every number is even or odd"

$$\forall n.\ \exists k.\ (n = k + k) \vee (n = k + k + 1)$$

1. General (n, k) and specific (1) individuals

Anatomy of a Formula

"Every number is even or odd"

$$\forall n.\ \exists k.\ (n = k + k) \vee (n = k + k + 1)$$

1. General (n, k) and specific (1) individuals
2. Function symbols, performing operations on individuals

Anatomy of a Formula

"Every number is even or odd"

$$\forall n.\ \exists k.\ (n = k + k) \vee (n = k + k + 1)$$

1. General (n, k) and specific (1) individuals
2. Function symbols, performing operations on individuals
3. Relations between individuals (here: equality)

Anatomy of a Formula

"Every number is even or odd"

$$\forall n.\ \exists k.\ (n = k + k) \vee (n = k + k + 1)$$

1. General (n, k) and specific (1) individuals
2. Function symbols, performing operations on individuals
3. Relations between individuals (here: equality)
4. Connectives, as in Propositional Logic

Anatomy of a Formula

"Every number is even or odd"

$$\forall n.\ \exists k.\ (n = k + k) \vee (n = k + k + 1)$$

1. General (n, k) and specific (1) individuals
2. Function symbols, performing operations on individuals
3. Relations between individuals (here: equality)
4. Connectives, as in Propositional Logic
5. Quantifiers, telling us how to interpret the general individuals n and k

More examples

"Every day is raining or sunny"

$$\forall d. \mathrm{raining}(d) \lor \mathrm{sunny}(d)$$

"Does there exist a player within 10 metres of player 1?"

$$\exists p. \mathrm{player}(p) \land \mathsf{distance}(\mathsf{locationOf}(p), \mathsf{locationOf}(\mathsf{player1})) \leq 10$$

Examples from Mathematics

Fermat's Last Theorem

$$\forall n. n > 2 \to \neg(\exists a. \exists b. \exists c. a^n + b^n = c^n)$$

(stated in 1637, not proved until 1994)

Goldbach's Conjecture

(Every even number greater than 2 is the sum of two primes)

$$\forall n. n > 2 \to \mathrm{even}(n) \to \exists p. \exists q. \mathrm{prime}(p) \wedge \mathrm{prime}(q) \wedge p + q = n$$

Summary

Predicate Logic upgrades Propositional Logic, adding:

- Individuals x, y, z
- Functions $+$, dayAfter
- Predicates $=$, even, odd
- Quantifiers $\forall$, $\exists$

Predicate Logic, Part 2

Saying what you mean

How to say “x is a P”

$$P(x)$$

For example:

$$\mathrm{human}(x)$$
$$\mathrm{mortal}(x)$$
$$\mathrm{swan}(x)$$
$$\mathrm{golden}(x)$$

How to say "x and y are related by R"

$$R(x, y)$$

for example:

$$\mathrm{colour}(x, \mathsf{gold})$$
$$\mathrm{species}(x, \mathsf{swan})$$
$$\mathrm{connected}(x, y)$$
$$\mathrm{knows}(\mathsf{pooh}, \mathsf{piglet})$$

“Everything is P”

everything is boring
everything is wet

$$\forall x.\mathrm{boring}(x)$$
$$\forall x.\mathrm{wet}(x)$$

$$\forall x.P(x)$$

Usually not very *useful* if P is atomic, but things like

$$\forall x.\mathrm{even}(x) \lor \mathrm{odd}(x)$$

are useful.

“There exists an P”

there is a human
there is a swan
there is an insect

$$\exists x.\mathrm{human}(x)$$
$$\exists x.\mathrm{swan}(x)$$
$$\exists x.\mathrm{class}(x, \mathsf{insecta})$$

$$\exists x.P(x)$$

there is at least one thing with property P

"All P are Q"

all humans are mortal
all swans are white
all insects have 6 legs

$$\forall x.\mathrm{human}(x) \to \mathrm{mortal}(x)$$
$$\forall x.\mathrm{swan}(x) \to \mathrm{white}(x)$$
$$\forall x.\mathrm{insect}(x) \to \mathrm{numLegs}(x, 6)$$

$$\forall x.P(x) \to Q(x)$$

for all x, *if* x *is* P, *then* x *is* Q

"Some P is Q"

some human is mortal
some swan is black
some insect has 6 legs

$$\exists x.\mathrm{human}(x) \land \mathrm{mortal}(x)$$

$$\exists x.\mathrm{swan}(x) \land \mathrm{colour}(x, \mathsf{black})$$

$$\exists x.\mathrm{insect}(x) \land \mathrm{numLegs}(x, 6)$$

$$\exists x.P(x) \land Q(x)$$

exists x*, such that* x *is a* P *and* x *is a* Q

“All P are Q” vs “Some P are Q”

$$\forall x.P(x) \to Q(x)$$

uses $\to$, but

$$\exists x.P(x) \land Q(x)$$

uses $\land$.

“All P are Q” vs “Some P are Q”

$$\forall x. P(x) \to Q(x)$$

uses $\to$, but

$$\exists x. P(x) \wedge Q(x)$$

uses $\wedge$.

Tempting to write:

$$\forall x. P(x) \wedge Q(x) \quad \text{everything is both P and Q}$$

or

$$\exists x. P(x) \to Q(x) \quad \text{there is some } x\text{, such that if P then Q}$$

but **almost always not what you want**.

“No P is Q”

no swans are blue
no bird can fly in space
no program works

$$\forall x.\mathrm{swan}(x) \to \neg\mathrm{blue}(x)$$
$$\neg(\exists x.\mathrm{bird}(x) \land \mathrm{canFlyInSpace}(x))$$
$$\forall x.\mathrm{program}(x) \to \neg\mathrm{works}(x)$$

$\neg(\exists x.P(x) \land Q(x))$ or $\forall x.P(x) \to \neg Q(x)$

The two statements are equivalent.

“For every P, there exists a related Q”

every farmer owns a donkey
every day has a next day
every list has a sorted version
every position has a nearby safe position

$$\forall f. \mathrm{farmer}(f) \to (\exists d. \mathrm{donkey}(d) \land \mathrm{owns}(f, d))$$

$$\forall d. \mathrm{day}(d) \to (\exists d'. \mathrm{day}(d') \land \mathrm{next}(d, d'))$$

$$\forall x. \mathrm{list}(x) \to (\exists y. \mathrm{list}(y) \land \mathrm{sorted}(y) \land \mathrm{sameElements}(x, y))$$

$$\forall p_1. \exists p_2. \mathrm{nearby}(p_1, p_2) \land \mathrm{safe}(p_2)$$

In steps:

1. For every x (they choose),
2. There is a y (we choose),
3. such that x and y are related.

“There exists an P such that every Q is related”

every farmer owns a donkey (!!!)
there is someone that everyone loves
there is someone that loves everyone

$$\exists d.\mathrm{donkey}(d) \wedge (\forall f.\mathrm{farmer}(f) \to \mathrm{owns}(f, d))$$

$$\exists x.\forall y.\mathrm{loves}(y, x)$$

$$\exists x.\forall y.\mathrm{loves}(x, y)$$

In steps:

1. there exists an x (we choose), such that
2. forall y (they choose),
3. it is the case that x and y are related.

“For all P, there is a related Q, related to all R”

everyone knows someone who knows everyone

$$\forall x.\exists y.\mathrm{knows}(x, y) \wedge (\forall z.\mathrm{knows}(y, z))$$

$$\forall x.P(x) \to (\exists y.Q(x, y) \wedge (\forall z.R(x, y, z))$$

In steps:

1. for all x (they choose),
2. there exists a y (we choose),
3. for all z (they choose),
4. such that x, y, z are related.

“There exists exactly one X”

there’s only one moon

“Any other individual with the same property is equal”

$$\exists x.\mathrm{moon}(x) \land (\forall y.\mathrm{moon}(y) \to x = y)$$

not quite the same, but similar:

$$\forall x.\forall y.(\mathrm{moon}(x) \land \mathrm{moon}(y)) \to x = y$$

this says: *at most one moon*, but doesn’t say one exists.

“For every X, there exists exactly one Y”

every train has one driver

$$\forall t.\mathrm{train}(t) \to (\exists d.\mathrm{driver}(d,t) \land (\forall d'.\mathrm{driver}(d',t) \to d = d'))$$

There exists an X such that for all Y there exists a Z

there is a node, such that for all reachable nodes,
there is a safe node in one step

$$\exists a.\forall b.\mathrm{reachable}(a, b) \to (\exists c.\mathrm{safe}(c) \land \mathrm{step}(b, c))$$

Not the same as:

$$\exists a.\exists c.\forall b.\mathrm{reachable}(a, b) \to (\mathrm{safe}(c) \land \mathrm{step}(b, c))$$

1. First one: c can be different for each b.
2. Second: the same c for all b.

Summary

- Many of the things you want to say in Predicate Logic fall into one of several predefined templates.
- It helps to think of quantifiers as a game
  - $\forall$ means “they choose”
  - $\exists$ means “I choose”

  (but they switch places under a negation or on the left of an implication!)

Predicate Logic, Part 3

Syntax Details

Predicate Logic

Predicate Logic upgrades Propositional Logic, adding:

- Individuals x, y, z
- Functions $+$, dayAfter
- Predicates $=$, even, odd
- Quantifiers $\forall$, $\exists$

Predicate Logic is for Modelling

To state properties of some situation we want to model, we choose:

1. Names of specific individuals

   (socrates, 1, 2, 10000, localhost, www.strath.ac.uk)

2. Function symbols

   ($+$, $\times$, nameOf)

3. Relation symbols

   ($\mathrm{human}(x)$, $x = y$, $\mathrm{linksTo}(x, y)$)

4. Some axioms

   (later ...)

Usually, we build a vocabulary based on what we want to do.

Vocabulary for Arithmetic

Individuals:

$$0 \qquad 1 \qquad 2 \qquad 3 \qquad \ldots$$

Functions:

$$t_1 + t_2 \qquad t_1 - t_2 \qquad \ldots$$

Predicates:

$$t_1 = t_2 \qquad t_1 < t_2 \qquad \ldots$$

Vocabulary for Documents

Individuals:

“Frankenstein” “Dracula” “Bram Stoker” “Mary Shelley”

Predicates:

$$\mathrm{linksTo}(\mathrm{doc}_1, \mathrm{doc}_2) \qquad \mathrm{authorOf}(\mathrm{doc}, \mathrm{person})$$

$$\mathrm{ownerOf}(\mathrm{doc}, \mathrm{person})$$

Vocabulary for Programs

Individuals

`java.lang.Object` `j.l.String` `j.l.Runnable`

`String toString()`

Relations

$$\text{extends}(\text{class}_1, \text{class}_2) \qquad \text{implements}(\text{class}, \text{interface})$$

$$\text{hasMethod}(\text{class}, \text{method}) \qquad \ldots$$

Equality

The equality predicate

$$t_1 = t_2$$

is treated specially:

- and in proofs (Topic 4)
- in the semantics (Topic 8)

Formal Grammar

$$
\begin{array}{rcll}
t & ::= & x & \text{variables} \\
  & \mid & c & \text{constants} \\
  & \mid & f(t_1, \ldots, t_n) & \text{function terms} \\
  \\
P & ::= & R(t_1, \ldots, t_n) & \text{predicates} \\
  & \mid & P \land Q \mid P \lor Q \mid P \to Q \mid \neg P & \text{connectives} \\
  & \mid & \forall x.P \mid \exists x.P & \text{quantifiers}
\end{array}
$$

Propositional Logic as special case: all relation symbols have arity 0.

When are two formulas the same?

Is there a difference in meaning between these two?

$$\forall x. P(x) \quad \text{and} \quad \forall y. P(y)$$

When are two formulas the same?

Is there a difference in meaning between these two?

$$\forall x. P(x) \qquad \text{and} \qquad \forall y. P(y)$$

No! They both mean the same thing.

When are two formulas the same?

Is there a difference in meaning between these two?

$$\forall x.P(x) \qquad \text{and} \qquad \forall y.P(y)$$

No! They both mean the same thing.

So we treat them as identical formulas.

Free and Bound Variables

In the formula:

$$\exists y. R(x, y)$$

1. The variable x is *free*
2. The variable y is *bound* (by the $\exists$ quantifier)

The quantifiers are *binders*.

Free and Bound Variables

Pay attention to the bracketing:

$$(\forall x.P(x) \to Q(x)) \land (\exists y.R(x, y))$$

The xs to the left of the $\land$ are bound (by the $\forall$)

The x to the right of the $\land$ is free.

When a variable is bound by quantifier, we say that it is in that quantifiers *scope*.

Identical Formulas, again

We can only rename *bound* variables

$\exists y.R(x, y)$ is identical to $\exists z.R(x, z)$

but

$\exists y.R(x, y)$ is not identical to $\exists y.R(z, y)$

because x and z do not have the same "global" meaning.

Summary

Vocabularies define the symbols we can use in our formulas.

The formal syntax of Predicate Logic is more complex than Propositional Logic

- Free and Bound Variables
- Formulas are identical even when renaming bound variables.

Predicate Logic, Part 4

Substitution

From General to Specific

We will have *general* assumptions like:

$$\forall x.\mathrm{human}(x) \to \mathrm{mortal}(x)$$

And we want to *specialise* (or *instantiate*) to:

$$\mathrm{human}(\mathsf{socrates}()) \to \mathrm{mortal}(\mathsf{socrates}())$$

Substitution

The notation

$$P[x := t]$$

means "replace all *free* occurrences of x in P with t".

- x is a *variable*
- P is a *formula*
- t is a *term*

But there is a subtlety...

Substitution Examples

$$
\begin{aligned}
& (\mathrm{mortal}(x))[x := \mathsf{socrates}()] \\
\Longrightarrow\ & \mathrm{mortal}(\mathsf{socrates}())
\end{aligned}
$$

Substitution Examples

$$(\forall y.\mathrm{weatherIs}(\mathsf{d}, y) \to \mathrm{weatherIs}(\mathsf{dayAfter}(\mathsf{d}), y))[\mathsf{d} := \mathsf{tuesday}]$$
$$\implies \forall y.\mathrm{weatherIs}(\mathsf{tuesday}, y) \to \mathrm{weatherIs}(\mathsf{dayAfter}(\mathsf{tuesday}), y)$$

Substitution Examples

$$
\begin{array}{rl}
& (\exists y.\mathrm{sameElements}(x, y) \land \mathrm{sorted}(y))[x := \mathsf{cons}(z_1, \mathsf{cons}(z_2, \mathsf{nil}))] \\
\Longrightarrow & \exists y.\mathrm{sameElements}(\mathsf{cons}(z_1, \mathsf{cons}(z_2, \mathsf{nil})), y) \land \mathrm{sorted}(y)
\end{array}
$$

Substitution Examples

$$
\begin{aligned}
& (\forall y. x + y = y + x)[x := z - z] \\
\Longrightarrow\ & \forall y. (z - z) + y = y + (z - z)
\end{aligned}
$$

Accidental Name Capture

If we substitute naively, then we produce nonsense:

1. $\exists y.\mathrm{sameElements}(x, y)$
   "there exists a y that has the same elements as x"

2. $(\exists y.\mathrm{sameElements}(x, y))[x := \mathsf{append}(y, [1, 2])]$
   "replace x by the list $\mathsf{append}(y, [1, 2])$*"*

3. $\exists y.\mathrm{sameElements}(\mathsf{append}(y, [1, 2]), y)$
   "there exists a y that has the same elements as $y + [1, 2]$?"

Capture Avoidance

Solution: Rename bound variables

$$
\begin{aligned}
& (\exists y.\mathrm{sameElements}(x, y))[x := \mathsf{append}(y, [1, 2])] \\
\Longrightarrow\ & (\exists z.\mathrm{sameElements}(x, z))[x := \mathsf{append}(y, [1, 2])] \\
\Longrightarrow\ & \exists z.\mathrm{sameElements}(\mathsf{append}(y, [1, 2]), z)
\end{aligned}
$$

Capture Avoiding Substitution

When working out

$$P[x := t]$$

If any of the variables in t are bound in P then rename them before doing the substitution.

Substitution Examples

1. $P(x, y)[x := y + y]$

Substitution Examples

1. $P(x, y)[x := y + y] \quad = P(y + y, y)$

Substitution Examples

1. $P(x, y)[x := y + y] \quad = P(y + y, y)$
2. $P(x, y)[y := y + y]$

Substitution Examples

1. $P(x, y)[x := y + y] \quad = P(y + y, y)$
2. $P(x, y)[y := y + y] \quad = P(x, y + y)$

Substitution Examples

1. $P(x, y)[x := y + y] \quad = P(y + y, y)$
2. $P(x, y)[y := y + y] \quad = P(x, y + y)$
3. $(\forall x.P(x, y))[x := y + y]$

Substitution Examples

1. $P(x, y)[x := y + y] \quad = P(y + y, y)$
2. $P(x, y)[y := y + y] \quad = P(x, y + y)$
3. $(\forall x. P(x, y))[x := y + y] \quad = \forall x. P(x, y)$

Substitution Examples

1. $(\forall x.P(x, y))[y := x + x]$

Substitution Examples

1. $(\forall x.P(x, y))[y := x + x] \quad = \forall z.P(z, x + x)$

   Renaming!

Substitution Examples

1. $(\forall x.P(x,y))[y := x + x] \quad = \forall z.P(z, x + x)$

   Renaming!

2. $(\forall x.P(x,y) \to (\exists z.Q(y,z)))[y := z + z]$

Substitution Examples

1. $(\forall x.P(x,y))[y := x + x] \quad = \forall z.P(z, x + x)$

   Renaming!

2. $(\forall x.P(x,y) \rightarrow (\exists z.Q(y,z)))[y := z + z]$

   $= \forall x.P(x, z + z) \rightarrow (\exists w.Q(z + z, w))$

   Renaming!

Substitution Examples

1. $(\forall x.P(x,y))[y := x + x] \quad = \forall z.P(z, x + x)$

   Renaming!

2. $(\forall x.P(x,y) \to (\exists z.Q(y,z)))[y := z + z]$
   $= \forall x.P(x, z + z) \to (\exists w.Q(z + z, w))$

   Renaming!

3. $(\forall x.P(x,z) \to (\exists z.Q(y,z)))[z := x + x]$

Substitution Examples

1. $(\forall x.P(x,y))[y := x + x] \quad = \forall z.P(z, x + x)$

   Renaming!

2. $(\forall x.P(x,y) \to (\exists z.Q(y,z)))[y := z + z]$
   $= \forall x.P(x, z + z) \to (\exists w.Q(z + z, w))$

   Renaming!

3. $(\forall x.P(x,z) \to (\exists z.Q(y,z)))[z := x + x]$
   $= \forall w.P(w, x + x) \to (\exists z.Q(y,z))$

   Renaming! and no substitution of the final z

Summary

- Substitution

$$P[x := t]$$

  is how we go from the general x to the specific t.
- We need to be careful to rename bound variables to avoid accidental name capture.