CSE 311 Lecture 06: Predicate Logic

Emina Torlak and Sami Davies

Topics

Key notions in predicate logic

A brief review of Lecture 05.

Using quantifiers

From predicate logic to English and back.

Negating quantified formulas

DeMorgan’s laws for quantifiers.

Quantifier scopes

Bound and free variables.

Nested quantifiers

The quantifier order matters.

Key notions in predicate logic

A brief review of Lecture 05.

Syntax and semantics of predicate logic

Syntax

Predicate logic extends propositional logic with two key constructs: *predicates* and *quantifiers* (∃, ∀).

Semantics

We define the meaning of formulas in predicate logic with respect to a *domain of discourse*.

Predicates

Predicate is a function that returns a truth value.

Cat(x) ::= “x is a cat”

Prime(x) ::= “x is prime”

HasTaken(x, y) ::= “student x has taken course y”

LessThan(x, y) ::= “$x < y$”

Sum(x, y, z) ::= “$x + y = z$”

GreaterThan5(x) ::= “$x > 5$”

HasNChars(s, n) ::= “string s has length n”

Predicates can have varying arity (numbers of arguments).

But a given predicate accepts a fixed number of arguments.

Predicates can have any names.

The name does *not* determine the meaning of the predicate.

So we can define Cat(x) ::= “x is an even number”.

Domain of discourse

To give meaning to predicates in a formula, we define a set of objects that those predicates can take as input.

This set of objects is called the *domain of discourse* for a formula.

For each of the following, what might the domain be?

“x is a cat”, “x barks”, “x ruined my couch”

“mammals” or “sentient beings” or “cats and dogs” or …

“x is prime”, “x = 0”, “x < 0”, “x is a power of two”

“numbers” or “integers” or “integers greater than 5” or …

“student x has taken course y” “x is a pre-req for z”

“students and courses” or “university entities” or …

Quantifiers

Quantifiers let us talk about *all* or *some* objects in the domain.

$\forall x.\ P(x)$

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

Read as "for all x, $P(x)$".

Called the **universal quantifier**.

$\exists x.\ P(x)$

There is an x in the domain for which $P(x)$ is true.

Read as "there exists x, $P(x)$".

Called the **existential quantifier**.

Understanding quantifiers

The truth value of a quantified formula depends on the domain.

	$\{-3, 3\}$	Integers	Odd Integers
$\forall x.\, \mathrm{Odd}(x)$	True	False	True
$\forall x.\, \mathrm{LessThan5}(x)$	True	False	False

	$\{-3, 3\}$	Integers	Positive Multiples of 5
$\exists x.\, \mathrm{Odd}(x)$	True	True	True
$\exists x.\, \mathrm{LessThan5}(x)$	True	True	False

You can think of $\forall x.\, P(x)$ as conjunction over all objects in the domain, and $\exists x.\, P(x)$ as disjunction over all objects in the domain.

- $\forall x.\, \mathrm{Odd}(x)$
  - over $\{-3, 3\}$ is the conjunction $\mathrm{Odd}(-3) \wedge \mathrm{Odd}(3)$
  - over integers is the infinite conjunction $\ldots \wedge \mathrm{Odd}(-1) \wedge \mathrm{Odd}(0) \wedge \mathrm{Odd}(1) \wedge \ldots$
- $\exists x.\, \mathrm{Odd}(x)$
  - over $\{-3, 3\}$ is the disjunction $\mathrm{Odd}(-3) \vee \mathrm{Odd}(3)$
  - over integers is the infinite disjunction $\ldots \vee \mathrm{Odd}(-1) \vee \mathrm{Odd}(0) \vee \mathrm{Odd}(1) \vee \ldots$

Using quantifiers

From predicate logic to English and back.

Statements with quantifiers

Just like with propositional logic, we need to define variables (this time predicates). And we must also now define a domain of discourse.

What is the truth value of these statements?

$\exists x.\, \mathrm{Even}(x)$
$\forall x.\, \mathrm{Odd}(x)$
$\forall x.\, \mathrm{Even}(x) \lor \mathrm{Odd}(x)$
$\exists x.\, \mathrm{Even}(x) \land \mathrm{Odd}(x)$
$\forall x.\, \mathrm{Greater}(x + 1, x)$
$\exists x.\, \mathrm{Even}(x) \land \mathrm{Prime}(x)$

Domain of discourse
Positive integers

Predicate definitions
Even(x) := “x is even”
Odd(x) := “x is odd”
Prime(x) := “x is prime”
Greater(x, y) := “x > y”
Equal(x, y) := “x = y”
Sum(x, y, z) := “z = x + y”

Statements with quantifiers

Just like with propositional logic, we need to define variables (this time predicates). And we must also now define a domain of discourse.

What is the truth value of these statements?

$\exists x.\, \mathrm{Even}(x)$ T
$\forall x.\, \mathrm{Odd}(x)$
$\forall x.\, \mathrm{Even}(x) \vee \mathrm{Odd}(x)$
$\exists x.\, \mathrm{Even}(x) \wedge \mathrm{Odd}(x)$
$\forall x.\, \mathrm{Greater}(x+1, x)$
$\exists x.\, \mathrm{Even}(x) \wedge \mathrm{Prime}(x)$

Domain of discourse

Positive integers

Predicate definitions

Even(x) := “x is even”
Odd(x) := “x is odd”
Prime(x) := “x is prime”
Greater(x, y) := “x > y”
Equal(x, y) := “x = y”
Sum(x, y, z) := “z = x + y”

Statements with quantifiers

Just like with propositional logic, we need to define variables (this time predicates). And we must also now define a domain of discourse.

What is the truth value of these statements?

$\exists x.\ \mathrm{Even}(x)$	T
$\forall x.\ \mathrm{Odd}(x)$	F
$\forall x.\ \mathrm{Even}(x) \vee \mathrm{Odd}(x)$	
$\exists x.\ \mathrm{Even}(x) \wedge \mathrm{Odd}(x)$	
$\forall x.\ \mathrm{Greater}(x + 1, x)$	
$\exists x.\ \mathrm{Even}(x) \wedge \mathrm{Prime}(x)$	

Domain of discourse

Positive integers

Predicate definitions

Even(x) := “x is even”
Odd(x) := “x is odd”
Prime(x) := “x is prime”
Greater(x, y) := “x > y”
Equal(x, y) := “x = y”
Sum(x, y, z) := “z = x + y”

Statements with quantifiers

Just like with propositional logic, we need to define variables (this time predicates). And we must also now define a domain of discourse.

What is the truth value of these statements?

$\exists x.\ \mathrm{Even}(x)$	T
$\forall x.\ \mathrm{Odd}(x)$	F
$\forall x.\ \mathrm{Even}(x) \lor \mathrm{Odd}(x)$	T
$\exists x.\ \mathrm{Even}(x) \land \mathrm{Odd}(x)$	
$\forall x.\ \mathrm{Greater}(x+1, x)$	
$\exists x.\ \mathrm{Even}(x) \land \mathrm{Prime}(x)$	

Domain of discourse

Positive integers

Predicate definitions

Even(x) := "x is even"

Odd(x) := "x is odd"

Prime(x) := "x is prime"

Greater(x, y) := "x > y"

Equal(x, y) := "x = y"

Sum(x, y, z) := "z = x + y"

Statements with quantifiers

Just like with propositional logic, we need to define variables (this time predicates). And we must also now define a domain of discourse.

What is the truth value of these statements?

$\exists x.\ \mathrm{Even}(x)$	T
$\forall x.\ \mathrm{Odd}(x)$	F
$\forall x.\ \mathrm{Even}(x) \vee \mathrm{Odd}(x)$	T
$\exists x.\ \mathrm{Even}(x) \wedge \mathrm{Odd}(x)$	F
$\forall x.\ \mathrm{Greater}(x+1, x)$	
$\exists x.\ \mathrm{Even}(x) \wedge \mathrm{Prime}(x)$	

Domain of discourse

Positive integers

Predicate definitions

Even(x) := “x is even”

Odd(x) := “x is odd”

Prime(x) := “x is prime”

Greater(x, y) := “x > y”

Equal(x, y) := “x = y”

Sum(x, y, z) := “z = x + y”

Statements with quantifiers

Just like with propositional logic, we need to define variables (this time predicates). And we must also now define a domain of discourse.

What is the truth value of these statements?

$\exists x.\ \mathrm{Even}(x)$	T
$\forall x.\ \mathrm{Odd}(x)$	F
$\forall x.\ \mathrm{Even}(x) \vee \mathrm{Odd}(x)$	T
$\exists x.\ \mathrm{Even}(x) \wedge \mathrm{Odd}(x)$	F
$\forall x.\ \mathrm{Greater}(x+1, x)$	T
$\exists x.\ \mathrm{Even}(x) \wedge \mathrm{Prime}(x)$	

Domain of discourse

Positive integers

Predicate definitions

Even(x) := "x is even"
Odd(x) := "x is odd"
Prime(x) := "x is prime"
Greater(x, y) := "x > y"
Equal(x, y) := "x = y"
Sum(x, y, z) := "z = x + y"

Statements with quantifiers

Just like with propositional logic, we need to define variables (this time predicates). And we must also now define a domain of discourse.

What is the truth value of these statements?

$\exists x.\ \mathrm{Even}(x)$	T
$\forall x.\ \mathrm{Odd}(x)$	F
$\forall x.\ \mathrm{Even}(x) \vee \mathrm{Odd}(x)$	T
$\exists x.\ \mathrm{Even}(x) \wedge \mathrm{Odd}(x)$	F
$\forall x.\ \mathrm{Greater}(x+1, x)$	T
$\exists x.\ \mathrm{Even}(x) \wedge \mathrm{Prime}(x)$	T

Domain of discourse

Positive integers

Predicate definitions

Even(x) := “x is even”
Odd(x) := “x is odd”
Prime(x) := “x is prime”
Greater(x, y) := “x > y”
Equal(x, y) := “x = y”
Sum(x, y, z) := “z = x + y”

Predicate logic to English: literal translations

Translate the following statements to English

$$\forall x.\ \exists y.\ \text{Greater}(y, x) \land \text{Prime}(y)$$

$$\forall x.\ \text{Prime}(x) \to (\text{Equal}(x, 2) \lor \text{Odd}(x))$$

$$\exists x.\ \exists y.\ \text{Sum}(x, 2, y) \land \text{Prime}(x) \land \text{Prime}(y)$$

Domain of discourse

Positive integers

Predicate definitions

Even(x) := "x is even"

Odd(x) := "x is odd"

Prime(x) := "x is prime"

Greater(x, y) := "x > y"

Equal(x, y) := "x = y"

Sum(x, y, z) := "z = x + y"

Predicate logic to English: literal translations

Translate the following statements to English

$$\forall x.\ \exists y.\ \mathrm{Greater}(y, x) \wedge \mathrm{Prime}(y)$$

For every positive integer x, there is a positive integer y, such that $y > x$ and y is prime.

$$\forall x.\ \mathrm{Prime}(x) \rightarrow (\mathrm{Equal}(x, 2) \vee \mathrm{Odd}(x))$$

$$\exists x.\ \exists y.\ \mathrm{Sum}(x, 2, y) \wedge \mathrm{Prime}(x) \wedge \mathrm{Prime}(y)$$

Domain of discourse

Positive integers

Predicate definitions

Even(x) := “x is even”
Odd(x) := “x is odd”
Prime(x) := “x is prime”
Greater(x, y) := “x > y”
Equal(x, y) := “x = y”
Sum(x, y, z) := “z = x + y”

Predicate logic to English: literal translations

Translate the following statements to English

$\forall x.\ \exists y.\ \text{Greater}(y, x) \wedge \text{Prime}(y)$

For every positive integer x, there is a positive integer y, such that $y > x$ and y is prime.

$\forall x.\ \text{Prime}(x) \rightarrow (\text{Equal}(x, 2) \vee \text{Odd}(x))$

For every positive integer x, if x is prime then $x = 2$ or x is odd.

$\exists x.\ \exists y.\ \text{Sum}(x, 2, y) \wedge \text{Prime}(x) \wedge \text{Prime}(y)$

Domain of discourse

Positive integers

Predicate definitions

Even(x) := "x is even"

Odd(x) := "x is odd"

Prime(x) := "x is prime"

Greater(x, y) := "x > y"

Equal(x, y) := "x = y"

Sum(x, y, z) := "z = x + y"

Predicate logic to English: literal translations

Translate the following statements to English

$$\forall x.\ \exists y.\ \mathrm{Greater}(y, x) \wedge \mathrm{Prime}(y)$$

For every positive integer x, there is a positive integer y, such that $y > x$ and y is prime.

$$\forall x.\ \mathrm{Prime}(x) \rightarrow (\mathrm{Equal}(x, 2) \vee \mathrm{Odd}(x))$$

For every positive integer x, if x is prime then $x = 2$ or x is odd.

$$\exists x.\ \exists y.\ \mathrm{Sum}(x, 2, y) \wedge \mathrm{Prime}(x) \wedge \mathrm{Prime}(y)$$

There exist positive integers x and y such that $x + 2 = y$ and x and y are prime.

Domain of discourse

Positive integers

Predicate definitions

Even(x) := "x is even"

Odd(x) := "x is odd"

Prime(x) := "x is prime"

Greater(x, y) := "x > y"

Equal(x, y) := "x = y"

Sum(x, y, z) := "z = x + y"

Predicate logic to English: natural translations

Translate the following statements to English

$$\forall x.\ \exists y.\ \text{Greater}(y, x) \wedge \text{Prime}(y)$$

$$\forall x.\ \text{Prime}(x) \rightarrow (\text{Equal}(x, 2) \vee \text{Odd}(x))$$

$$\exists x.\ \exists y.\ \text{Sum}(x, 2, y) \wedge \text{Prime}(x) \wedge \text{Prime}(y)$$

Domain of discourse

Positive integers

Predicate definitions

Even(x) := "x is even"
Odd(x) := "x is odd"
Prime(x) := "x is prime"
Greater(x, y) := "x > y"
Equal(x, y) := "x = y"
Sum(x, y, z) := "z = x + y"

Predicate logic to English: natural translations

Translate the following statements to English

$\forall x.\ \exists y.\ \mathrm{Greater}(y, x) \wedge \mathrm{Prime}(y)$

For every positive integer there is a larger number that is prime.

$\forall x.\ \mathrm{Prime}(x) \rightarrow (\mathrm{Equal}(x, 2) \vee \mathrm{Odd}(x))$

$\exists x.\ \exists y.\ \mathrm{Sum}(x, 2, y) \wedge \mathrm{Prime}(x) \wedge \mathrm{Prime}(y)$

Domain of discourse

Positive integers

Predicate definitions

Even(x) := “x is even”

Odd(x) := “x is odd”

Prime(x) := “x is prime”

Greater(x, y) := “x > y”

Equal(x, y) := “x = y”

Sum(x, y, z) := “z = x + y”

Predicate logic to English: natural translations

Translate the following statements to English

$$\forall x.\ \exists y.\ \text{Greater}(y, x) \land \text{Prime}(y)$$

For every positive integer there is a larger number that is prime.

$$\forall x.\ \text{Prime}(x) \rightarrow (\text{Equal}(x, 2) \lor \text{Odd}(x))$$

Every prime number is 2 or odd.

$$\exists x.\ \exists y.\ \text{Sum}(x, 2, y) \land \text{Prime}(x) \land \text{Prime}(y)$$

Domain of discourse

Positive integers

Predicate definitions

Even(x) := “x is even”

Odd(x) := “x is odd”

Prime(x) := “x is prime”

Greater(x, y) := “x > y”

Equal(x, y) := “x = y”

Sum(x, y, z) := “z = x + y”

Predicate logic to English: natural translations

Translate the following statements to English

$$\forall x.\ \exists y.\ \mathrm{Greater}(y, x) \wedge \mathrm{Prime}(y)$$

For every positive integer there is a larger number that is prime.

$$\forall x.\ \mathrm{Prime}(x) \rightarrow (\mathrm{Equal}(x, 2) \vee \mathrm{Odd}(x))$$

Every prime number is 2 or odd.

$$\exists x.\ \exists y.\ \mathrm{Sum}(x, 2, y) \wedge \mathrm{Prime}(x) \wedge \mathrm{Prime}(y)$$

There exist prime numbers that differ by two.

Domain of discourse

Positive integers

Predicate definitions

Even(x) := "x is even"

Odd(x) := "x is odd"

Prime(x) := "x is prime"

Greater(x, y) := "x > y"

Equal(x, y) := "x = y"

Sum(x, y, z) := "z = x + y"

English to predicate logic

“Orange cats like lasagna.”

“Some orange cats don’t like lasagna.”

Domain of discourse

Mammals

Predicate definitions

Cat(x) := “x is a cat”

Orange(x) := “x is orange”

LikesLasagna(x) := “x likes lasagna”

English to predicate logic

"Orange cats like lasagna."

$$\forall x.\ ((\mathrm{Orange}(x) \wedge \mathrm{Cat}(x)) \rightarrow \mathrm{LikesLasagna}(x))$$

"Some orange cats don't like lasagna."

Domain of discourse

Mammals

Predicate definitions

Cat(x) := "x is a cat"

Orange(x) := "x is orange"

LikesLasagna(x) := "x likes lasagna"

English to predicate logic

“Orange cats like lasagna.”

$$\forall x.\ ((\text{Orange}(x) \land \text{Cat}(x)) \to \text{LikesLasagna}(x))$$

“Some orange cats don’t like lasagna.”

$$\exists x.\ ((\text{Orange}(x) \land \text{Cat}(x)) \land \neg\text{LikesLasagna}(x))$$

Domain of discourse

Mammals

Predicate definitions

Cat(x) := “x is a cat”

Orange(x) := “x is orange”

LikesLasagna(x) := “x likes lasagna”

English to predicate logic: translation hints

"Orange cats like lasagna."

$\forall x.\ ((\mathrm{Orange}(x) \wedge \mathrm{Cat}(x)) \rightarrow \mathrm{LikesLasagna}(x))$

When there's no leading quantification, it means "for all".

When restricting to a smaller domain in a "for all", use implication.

"Some orange cats don't like lasagna."

$\exists x.\ ((\mathrm{Orange}(x) \wedge \mathrm{Cat}(x)) \wedge \neg \mathrm{LikesLasagna}(x))$

"Some" means "there exists".

When restricting to a smaller domain in an "exists", use conjunction.

When putting predicates together, like orange cats, use conjunction.

Domain of discourse

Mammals

Predicate definitions

Cat(x) := "x is a cat"

Orange(x) := "x is orange"

LikesLasagna(x) := "x likes lasagna"

Negating quantified formulas

DeMorgan’s laws for quantifiers.

Negations of quantifiers

Domain of discourse	**Predicate definitions**
{plum, apple, …}	PurpleFruit(x) := "x is a purple fruit"

Let P be the formula $\forall x.\ \mathrm{PurpleFruit}(x)$, "all fruits are purple."

What is the negation of P?

a. "There exists a purple fruit" ($\exists x.\ \mathrm{PurpleFruit}(x)$)
b. "There exists a non-purple fruit" ($\exists x.\ \neg\mathrm{PurpleFruit}(x)$)
c. "All fruits are not purple" ($\forall x.\ \neg\mathrm{PurpleFruit}(x)$)

Negations of quantifiers

Domain of discourse	**Predicate definitions**
{plum, apple, …}	PurpleFruit(x) := “x is a purple fruit”

Let P be the formula $\forall x.\ \mathrm{PurpleFruit}(x)$, “all fruits are purple.”

What is the negation of P?

a. “There exists a purple fruit” ($\exists x.\ \mathrm{PurpleFruit}(x)$)
b. “There exists a non-purple fruit” ($\exists x.\ \neg\mathrm{PurpleFruit}(x)$)
c. “All fruits are not purple” ($\forall x.\ \neg\mathrm{PurpleFruit}(x)$)

Key idea: think of p as a conjunction over all objects in the domain, negate that conjunction, and convert back to a quantified formula.

Negations of quantifiers

Domain of discourse	**Predicate definitions**
{plum, apple, …}	PurpleFruit(x) := "x is a purple fruit"

Let P be the formula $\forall x.\ \mathrm{PurpleFruit}(x)$, "all fruits are purple."

What is the negation of P?

a. "There exists a purple fruit" ($\exists x.\ \mathrm{PurpleFruit}(x)$)
b. "There exists a non-purple fruit" ($\exists x.\ \neg\mathrm{PurpleFruit}(x)$)
c. "All fruits are not purple" ($\forall x.\ \neg\mathrm{PurpleFruit}(x)$)

Key idea: think of p as a conjunction over all objects in the domain, negate that conjunction, and convert back to a quantified formula.

$$p \quad \equiv \mathrm{PurpleFruit}(\mathrm{plum}) \land \mathrm{PurpleFruit}(\mathrm{apple}) \land \ldots$$ **∀ to conjunction**

Negations of quantifiers

Domain of discourse	**Predicate definitions**
{plum, apple, …}	PurpleFruit(x) := “x is a purple fruit”

Let P be the formula $\forall x.\ \mathrm{PurpleFruit}(x)$, “all fruits are purple.”

What is the negation of P?

a. “There exists a purple fruit” ($\exists x.\ \mathrm{PurpleFruit}(x)$)
b. “There exists a non-purple fruit” ($\exists x.\ \neg\mathrm{PurpleFruit}(x)$)
c. “All fruits are not purple” ($\forall x.\ \neg\mathrm{PurpleFruit}(x)$)

Key idea: think of p as a conjunction over all objects in the domain, negate that conjunction, and convert back to a quantified formula.

$$
\begin{aligned}
p &\equiv \mathrm{PurpleFruit}(\mathrm{plum}) \wedge \mathrm{PurpleFruit}(\mathrm{apple}) \wedge \ldots && \textbf{$\forall$ to conjunction} \\
\neg p &\equiv \neg(\mathrm{PurpleFruit}(\mathrm{plum}) \wedge \mathrm{PurpleFruit}(\mathrm{apple}) \wedge \ldots) && \textbf{Negate both sides}
\end{aligned}
$$

Negations of quantifiers

Domain of discourse	Predicate definitions
{plum, apple, …}	PurpleFruit(x) := “x is a purple fruit”

Let P be the formula $\forall x.\ \mathrm{PurpleFruit}(x)$, “all fruits are purple.”

What is the negation of P?

a. “There exists a purple fruit” ($\exists x.\ \mathrm{PurpleFruit}(x)$)
b. “There exists a non-purple fruit” ($\exists x.\ \neg\mathrm{PurpleFruit}(x)$)
c. “All fruits are not purple” ($\forall x.\ \neg\mathrm{PurpleFruit}(x)$)

Key idea: think of p as a conjunction over all objects in the domain, negate that conjunction, and convert back to a quantified formula.

$$
\begin{aligned}
p &\equiv \mathrm{PurpleFruit}(\mathrm{plum}) \land \mathrm{PurpleFruit}(\mathrm{apple}) \land \ldots && \textbf{$\forall$ to conjunction} \\
\neg p &\equiv \neg(\mathrm{PurpleFruit}(\mathrm{plum}) \land \mathrm{PurpleFruit}(\mathrm{apple}) \land \ldots) && \textbf{Negate both sides} \\
&\equiv \neg\mathrm{PurpleFruit}(\mathrm{plum}) \lor \neg\mathrm{PurpleFruit}(\mathrm{apple}) \lor \ldots && \textbf{DeMorgan}
\end{aligned}
$$

Negations of quantifiers

Domain of discourse	Predicate definitions
{plum, apple, …}	PurpleFruit(x) := "x is a purple fruit"

Let P be the formula $\forall x.\ \mathrm{PurpleFruit}(x)$, "all fruits are purple."

What is the negation of P?

a. "There exists a purple fruit" ($\exists x.\ \mathrm{PurpleFruit}(x)$)
b. "There exists a non-purple fruit" ($\exists x.\ \neg\mathrm{PurpleFruit}(x)$)
c. "All fruits are not purple" ($\forall x.\ \neg\mathrm{PurpleFruit}(x)$)

Key idea: think of p as a conjunction over all objects in the domain, negate that conjunction, and convert back to a quantified formula.

$$
\begin{aligned}
p &\equiv \mathrm{PurpleFruit}(\mathrm{plum}) \wedge \mathrm{PurpleFruit}(\mathrm{apple}) \wedge \ldots && \textbf{∀ to conjunction} \\
\neg p &\equiv \neg(\mathrm{PurpleFruit}(\mathrm{plum}) \wedge \mathrm{PurpleFruit}(\mathrm{apple}) \wedge \ldots) && \textbf{Negate both sides} \\
&\equiv \neg\mathrm{PurpleFruit}(\mathrm{plum}) \vee \neg\mathrm{PurpleFruit}(\mathrm{apple}) \vee \ldots && \textbf{DeMorgan} \\
&\equiv \exists x.\ \neg\mathrm{PurpleFruit}(x) && \textbf{Disjunction to ∃}
\end{aligned}
$$

DeMorgan's laws for quantifiers

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

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

DeMorgan's laws for quantifiers

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

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

"There is no largest integer."

$$\begin{aligned} \neg\exists x.\ \forall y.\ (x \geq y) &\equiv \\ &\equiv \\ &\equiv \forall x.\ \exists y.\ (x < y) \end{aligned}$$

"For every integer there is a larger integer."

DeMorgan's laws for quantifiers

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

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

"There is no largest integer."

$$\begin{aligned} \neg \exists x.\ \forall y.\ (x \geq y) &\equiv \forall x.\ \neg \forall y.\ (x \geq y) && \textbf{DeMorgan} \\ &\equiv \\ &\equiv \forall x.\ \exists y.\ (x < y) \end{aligned}$$

"For every integer there is a larger integer."

DeMorgan's laws for quantifiers

$$\boxed{\begin{aligned} \neg\forall x.\, P(x) &\equiv \exists x.\, \neg P(x) \\ \neg\exists x.\, P(x) &\equiv \forall x.\, \neg P(x) \end{aligned}}$$

"There is no largest integer."

$$\begin{aligned} \neg\exists x.\, \forall y.\, (x \geq y) &\equiv \forall x.\, \neg\forall y.\, (x \geq y) && \textbf{DeMorgan} \\ &\equiv \forall x.\, \exists y.\, \neg(x \geq y) && \textbf{DeMorgan} \\ &\equiv \forall x.\, \exists y.\, (x < y) \end{aligned}$$

"For every integer there is a larger integer."

DeMorgan's laws for quantifiers

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

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

"There is no largest integer."

$$\begin{aligned}
\neg\exists x.\, \forall y.\, (x \geq y) &\equiv \forall x.\, \neg\forall y.\, (x \geq y) && \text{DeMorgan} \\
&\equiv \forall x.\, \exists y.\, \neg(x \geq y) && \text{DeMorgan} \\
&\equiv \forall x.\, \exists y.\, (x < y) && \text{Semantics of } >
\end{aligned}$$

"For every integer there is a larger integer."

Quantifier scopes

Bound and free variables.

Scope of quantifiers

$\exists x. \, (P(x) \land Q(x))$ **vs** $(\exists x. \, P(x)) \land (\exists x. \, Q(x))$

Scope of quantifiers

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

There is an object in the domain for which both P and Q are true.

vs

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

There is an object for which P is true and an object for which Q is true, and they may be different objects.

Understanding scope of quantifiers

The formula inside of a quantifier is called its *scope*.

A variable is *bound* if it is in the scope of some quantifier.

A variable is *free* if it isn't in the scope of any quantifier.

Example: $\forall y.\ ((\exists x.\ P(x)) \rightarrow\ Q(x,y))$

Is y in $Q(x, y)$ bound or free?
Is x in $P(x)$ bound or free?
Is x in $Q(x, y)$ bound or free?

Understanding scope of quantifiers

The formula inside of a quantifier is called its *scope*.

A variable is *bound* if it is in the scope of some quantifier.

A variable is *free* if it isn't in the scope of any quantifier.

Example: $\forall y.\ ((\exists x.\ P(x)) \rightarrow\ Q(x,y))$

Is y in $Q(x, y)$ bound or free? Bound.

Is x in $P(x)$ bound or free?

Is x in $Q(x, y)$ bound or free?

Understanding scope of quantifiers

The formula inside of a quantifier is called its *scope*.

A variable is *bound* if it is in the scope of some quantifier.

A variable is *free* if it isn't in the scope of any quantifier.

Example: $\forall y.\ ((\exists x.\ P(x)) \rightarrow\ Q(x,y))$

Is y in $Q(x, y)$ bound or free? Bound.

Is x in $P(x)$ bound or free? Bound.

Is x in $Q(x, y)$ bound or free?

Understanding scope of quantifiers

The formula inside of a quantifier is called its *scope*.

A variable is *bound* if it is in the scope of some quantifier.

A variable is *free* if it isn't in the scope of any quantifier.

Example: $\forall y.\ ((\exists x.\ P(x)) \rightarrow\ Q(x,y))$

Is y in $Q(x, y)$ bound or free? Bound.

Is x in $P(x)$ bound or free? Bound.

Is x in $Q(x, y)$ bound or free? Free.

Quantifier “style”

$$\forall x.\ (\exists y.\ (P(x, y) \rightarrow \forall x.\ Q(y, x)))$$

This isn’t wrong, but it’s confusing. Help your reader by using unique names for quantified variables. Names are cheap :)

Nested quantifiers

The quantifier order matters.

When nesting quantifiers …

Bound variable names don’t matter.

Quantifiers can sometimes move within the enclosing formula.

But the *order* of quantifiers is important.

When nesting quantifiers …

Bound variable names don't matter.

$$\forall x.\, \exists y.\, P(x, y) \equiv \forall a.\, \exists b.\, P(a, b)$$

Quantifiers can sometimes move within the enclosing formula.

But the *order* of quantifiers is important.

When nesting quantifiers …

Bound variable names don't matter.

$$\forall x.\ \exists y.\ P(x, y) \equiv \forall a.\ \exists b.\ P(a, b)$$

Quantifiers can sometimes move within the enclosing formula.

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

But the *order* of quantifiers is important.

When nesting quantifiers …

Bound variable names don't matter.

$$\forall x.\ \exists y.\ P(x, y) \equiv \forall a.\ \exists b.\ P(a, b)$$

Quantifiers can sometimes move within the enclosing formula.

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

$$\forall x.\ (Q(x) \land (\neg \exists y.\ P(x, y))) \not\equiv \forall x.\ \exists y.\ (Q(x) \land \neg P(x, y))$$

But the *order* of quantifiers is important.

When nesting quantifiers …

Bound variable names don't matter.

$$\forall x.\ \exists y.\ P(x, y) \equiv \forall a.\ \exists b.\ P(a, b)$$

Quantifiers can sometimes move within the enclosing formula.

$$\forall x.\ (Q(x) \wedge \exists y.\ P(x, y)) \equiv \forall x.\ \exists y.\ (Q(x) \wedge P(x, y))$$

$$\forall x.\ (Q(x) \wedge (\neg \exists y.\ P(x, y))) \not\equiv \forall x.\ \exists y.\ (Q(x) \wedge \neg P(x, y))$$

But the *order* of quantifiers is important.

Example: are these formulas true or false?

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

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

Domain of discourse

Integers

Predicate definitions

GreaterEq(x, y) := "$x \geq y$"

When nesting quantifiers …

Bound variable names don't matter.

$$\forall x.\ \exists y.\ P(x, y) \equiv \forall a.\ \exists b.\ P(a, b)$$

Quantifiers can sometimes move within the enclosing formula.

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

$$\forall x.\ (Q(x) \land (\neg\exists y.\ P(x, y))) \not\equiv \forall x.\ \exists y.\ (Q(x) \land \neg P(x, y))$$

But the *order* of quantifiers is important.

Example: are these formulas true or false?

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

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

Domain of discourse

Integers

Predicate definitions

GreaterEq(x, y) := "x $\geq$ y"

When nesting quantifiers …

Bound variable names don't matter.

$$\forall x.\ \exists y.\ P(x, y) \equiv \forall a.\ \exists b.\ P(a, b)$$

Quantifiers can sometimes move within the enclosing formula.

$$\forall x.\ (Q(x) \wedge \exists y.\ P(x, y)) \equiv \forall x.\ \exists y.\ (Q(x) \wedge P(x, y))$$

$$\forall x.\ (Q(x) \wedge (\neg \exists y.\ P(x, y))) \not\equiv \forall x.\ \exists y.\ (Q(x) \wedge \neg P(x, y))$$

But the *order* of quantifiers is important.

Example: are these formulas true or false?

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

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

Domain of discourse

Integers

Predicate definitions

GreaterEq(x, y) := "x $\geq$ y"

Quantification with two variables

Formula	When true	When false
$\forall x.\ \forall y.\ P(x, y)$	Every pair is true.	At least one pair is false.
$\exists x.\ \exists y.\ P(x, y)$	At least one pair is true.	All pairs are false.
$\forall x.\ \exists y.\ P(x, y)$	Every x has a corresponding y.	Some x doesn't have a y.
$\exists y.\ \forall x.\ P(x, y)$	A particular y works for every x.	Every y has an x that makes $P(x, y)$ false.

Quantification with two variables

Formula	When true	When false
$\forall x.\ \forall y.\ P(x, y)$	Every pair is true.	At least one pair is false.
$\exists x.\ \exists y.\ P(x, y)$	At least one pair is true.	All pairs are false.
$\forall x.\ \exists y.\ P(x, y)$	Every x has a corresponding y.	Some x doesn't have a y.
$\exists y.\ \forall x.\ P(x, y)$	A particular y works for every x.	Every y has an x that makes $P(x, y)$ false.

This is the form of the program synthesis query!

$$\exists P.\ \forall x.\ S(x, P(x))$$

"There is a program P that satisfies the spec S on every input x."

Summary

Predicate logic adds predicates and quantifiers to propositional logic.

Predicate is a function that returns a truth value.

Quantifiers let us talk about *all* ($\forall$) or *some* ($\exists$) objects in the domain.

The domain of discourse is the set of objects over which the predicates and quantifiers in a formula are evaluated.

When using quantifiers, keep in mind

the DeMorgan's laws for negating quantified formulas,

which variables are free and bound, and

the order of quantifiers.