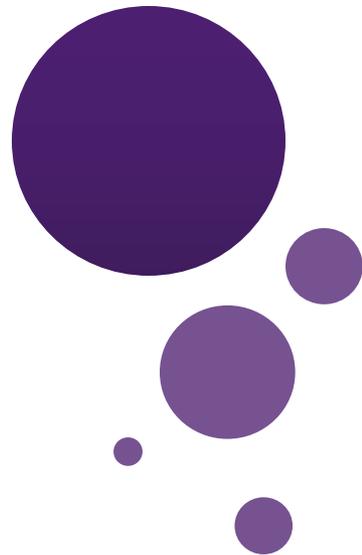




UNIVERSITY
AT ALBANY

State University of New York

Lecture 4: Quantifiers and Rules of Reference



Dr. Chengjiang Long
Computer Vision Researcher at Kitware Inc.
Adjunct Professor at SUNY at Albany.
Email: clong2@albany.edu

Outline

- Quantifiers
- Logic Programming (Prolog)
- Transcribing English to Logic
- Rules of Reference

Outline

- **Quantifiers**
- Logic Programming (Prolog)
- Transcribing English to Logic
- Rules of Reference

Universal Quantifier: Definition

- **Definition:** The universal quantification of a predicate $P(x)$ is the proposition ' $P(x)$ is true for all values of x in the universe of discourse.'

We use the notation: $\forall x P(x)$, which is read 'for all x '.

- If the universe of discourse is finite, say $\{n_1, n_2, \dots, n_k\}$, then the universal quantifier is simply the conjunction of the propositions over all the elements

$$\forall x P(x) \Leftrightarrow P(n_1) \wedge P(n_2) \wedge \dots \wedge P(n_k)$$

Existential Quantifier: Definition

- **Definition:** The existential quantification of a predicate $P(x)$ is the proposition 'There exists a value x in the universe of discourse such that $P(x)$ is true.' We use the notation: $\exists x P(x)$, which is read 'there exists x '.
- If the universe of discourse is finite, say $\{n_1, n_2, \dots, n_k\}$, then the existential quantifier is simply the disjunction of the propositions over all the elements

$$\exists x P(x) \Leftrightarrow P(n_1) \vee P(n_2) \vee \dots \vee P(n_k)$$

Quantifiers: Truth values

- In general, when are quantified statements true or false?

Statement	True when...	False when...
$\forall x P(x)$	$P(x)$ is true for every x	There is an x for which $P(x)$ is false
$\exists x P(x)$	There is an x for which $P(x)$ is true	$P(x)$ is false for every x

Mixing quantifiers (1)

- Existential and universal quantifiers can be used together to quantify a propositional predicate. For example:

$$\forall x \exists y P(x,y)$$

is perfectly valid

- Alert:
 - The quantifiers must be read from left to right
 - The order of the quantifiers is important
 - $\forall x \exists y P(x,y)$ is not equivalent to $\exists y \forall x P(x,y)$

Mixing quantifiers (2)

- Consider
 - $\forall x \exists y \text{ Loves}(x,y)$: Everybody loves somebody
 - $\exists y \forall x \text{ Loves}(x,y)$: There is someone loved by everyone
- The two expressions do not mean the same thing
- $(\exists y \forall x \text{ Loves}(x,y)) \rightarrow (\forall x \exists y \text{ Loves}(x,y))$ but the converse does not hold
- However, you can commute similar quantifiers
 - $\forall x \forall y P(x,y)$ is equivalent to $\forall y \forall x P(x,y)$ (thus, $\forall x,y P(x,y)$)
 - $\exists x \exists y P(x,y)$ is equivalent to $\exists y \exists x P(x,y)$ (thus $\exists x,y P(x,y)$)

Mixing Quantifiers: Truth values

Statement	True when...	False when...
$\forall x \forall y P(x,y)$	$P(x,y)$ is true for every pair x,y	There is at least one <i>pair</i> x,y for which $P(x,y)$ is false
$\forall x \exists y P(x,y)$	For every x , there is a y for which $P(x,y)$ is true	There is an x for which $P(x,y)$ 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
$\exists x \exists y P(x,y)$	There is at least one pair x,y for which $P(x,y)$ is true	$P(x,y)$ is false for every pair x,y

Mixing Quantifiers: Example (1)

- Express, in predicate logic, the statement that there is an infinite number of integers
- Answer:
 - Let $P(x,y)$ be the statement that $x < y$
 - Let the universe of discourse be the integers, Z
 - The statement can be expressed by the following

$$\forall x \exists y P(x,y)$$

Mixing Quantifiers: Example (2)

- Express the *commutative law of addition* for R
- We want to express that for every pair of reals, x, y , the following holds: $x+y=y+x$
- Answer:

– Let $P(x, y)$ be the statement that $x+y$

– Let the universe of discourse be the reals, R

– The statement can be expressed by the following

$$\forall x \forall y (P(x, y) \Leftrightarrow P(y, x))$$

Alternatively, $\forall x \forall y (x+y = y+x)$

Mixing Quantifiers: Example (3)

- Express the multiplicative *law* for nonzero reals $R \setminus \{0\}$
- We want to express that for every real number x , there exists a real number y such that $xy=1$
- Answer:

$$\forall x \exists y (xy = 1)$$

Mixing Quantifiers: Example (4)

false mathematical statement

- Does commutativity for subtraction hold over the reals?
- That is: does $x-y=y-x$ for all pairs x,y in R ?
- Express using quantifiers

$$\forall x \forall y (x-y = y-x)$$

Mixing Quantifiers: Example (5)

- Express the statement as a logical expression: “There is a number x such that when it is added to any number, the result is that number and if it is multiplied by any number, the result is x ” as a logical expression
- Answer:
 - Let $P(x,y)$ be the expression “ $x+y=y$ ”
 - Let $Q(x,y)$ be the expression “ $xy=x$ ”
 - The universe of discourse is N,Z,R,Q (*but not Z^+*)
 - Then the expression is:

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

Alternatively: $\exists x \forall y (x+y=y) \wedge (xy = x)$

Binding Variables

- When a quantifier is used on a variable x , we say that x is bound
- If no quantifier is used on a variable in a predicate statement, the variable is called free
- Examples
 - In $\exists x \forall y P(x, y)$, both x and y are bound
 - In $\forall x P(x, y)$, x is bound but y is free
- A statement is called a well-formed formula, when all variables are properly quantified

Binding Variables: Scope

- The set of all variables bound by a common quantifier is called the scope of the quantifier
- For example, in the expression $\exists x, y \forall z P(x, y, z, c)$
 - What is the scope of existential quantifier?
 - What is the scope of universal quantifier?
 - What are the bound variables?
 - What are the free variables?
 - Is the expression a well-formed formula?

Negation

- We can use negation with quantified expressions as we used them with propositions
- **Lemma:** Let $P(x)$ be a predicate. Then the followings hold:

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

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

- This is essentially the quantified version of De Morgan's Law (when the universe of discourse is finite, this is exactly De Morgan's Law)

Negation: Truth

Truth Values of Negated Quantifiers

Statement	True when...	False when...
$\neg \exists x P(x) \equiv \forall x \neg P(x)$	$P(x)$ is false for every x	There is an x for which $P(x)$ is true
$\neg \forall x P(x) \equiv \exists x \neg P(x)$	There is an x for which $P(x)$ is false	$P(x)$ is true for every x

Outline

- Quantifiers
- **Logic Programming (Prolog)**
- Transcribing English to Logic
- Rules of Reference

Prolog (1)

- Prolog (Programming in Logic) is a programming language based on (a restricted form of) Predicate Logic (a.k.a. Predicate Calculus and FOL)
- It was developed by the logicians of the Artificial Intelligence community for symbolic reasoning

Prolog (2)

- Prolog allows the users to express facts and rules
- Facts are propositional functions: `student(mia)`, `enrolled(mia,cse235)`, `instructor(patel,cse235)`, etc.
- Rules are implications with conjunctions:
`teaches(X,Y) :- instructor(X,Z), enrolled(Y,Z)`
- Prolog answers queries such as:
`?enrolled(mia,cse235)`
`?enrolled(X,cse476)`
`?teaches(X,mia)`
by binding variables and doing theorem proving (i.e., applying inference rules) as we will see in Section 1.5

Outline

- Quantifiers
- Logic Programming (Prolog)
- **Transcribing English to Logic**
- Rules of Reference

English into Logic

- Logic is more precise than English
- Transcribing English into Logic and vice versa can be tricky
- When writing statements with quantifiers, usually the correct meaning is conveyed with the following combinations:

Use \forall with \Rightarrow

$\forall x \text{ Lion}(x) \Rightarrow \text{Fierce}(x)$: Every lion is fierce

$\forall x \text{ Lion}(x) \wedge \text{Fierce}(x)$: Everyone is a lion and everyone is fierce

Use \exists with \wedge

$\exists x \text{ Lion}(x) \wedge \text{Vegan}(x)$: Holds when you have at least one vegan lion

$\exists x \text{ Lion}(x) \Rightarrow \text{Vegan}(x)$: Holds when you have vegan people in the universe of discourse (even though there is no vegan lion in the universe of discourse)

More exercises (1)

- Rewrite the following expression, pushing negation inward:

$$\neg \forall x (\exists y \forall z P(x,y,z) \wedge \exists z \forall y P(x,y,z))$$

- Answer:

$$\exists x (\forall y \exists z \neg P(x,y,z) \vee \forall z \exists y \neg P(x,y,z))$$

More Exercises (2)

- Let $P(x,y)$ denote 'x is a factor of y' where
 - $x \in \{1,2,3,\dots\}$ and $y \in \{2,3,4,\dots\}$
- Let $Q(x,y)$ denote:
 - $\forall x,y [P(x,y) \rightarrow (x=y) \vee (x=1)]$
- Question: When is $Q(x,y)$ true?

Alert...

- Some students wonder if:

$$\forall x,y P(x,y) \equiv (\forall x P(x,y)) \wedge (\forall y P(x,y))$$

- This is certainly not true.
 - In the left-hand side, both x,y are bound.
 - In the right-hand side,
 - In the first predicate, x is bound and y is free
 - In the second predicate, y is bound and x is free
 - Thus, the left-hand side is a proposition, but the right-hand side is not. They cannot be equivalent
- All variables that occur in a propositional function must be bound to turn it into a proposition

Outline

- Quantifiers
- Logic Programming (Prolog)
- Transcribing English to Logic
- **Rules of Reference**

Motivation (1)

- “Mathematical proofs, like diamonds, are hard and clear, and will be touched with nothing but strict reasoning.” *-John Locke*
- Mathematical proofs are, in a sense, the only true knowledge we have
- They provide us with a guarantee as well as an explanation (and hopefully some insight)

Motivation (2)

- Mathematical proofs are necessary in CS
 - You must always (try to) prove that your algorithm
 - terminates
 - is sound, complete, optimal
 - finds optimal solution
 - You may also want to show that it is more efficient than another method
 - Proving certain properties of data structures may lead to new, more efficient or simpler algorithms
 - Arguments may entail assumptions. You may want to prove that the assumptions are valid

Terminology

- A theorem is a statement that can be shown to be true (via a proof)
- A proof is a sequence of statements that form an argument
- Axioms or postulates are statements taken to be self evident or assumed to be true
- A lemma (plural lemmas or lemmata) is a theorem useful within the proof of a theorem
- A corollary is a theorem that can be established from theorem that has just been proven
- A proposition is usually a ‘less’ important theorem
- A conjecture is a statement whose truth value is unknown
- The rules of inference are the means used to draw conclusions from other assertions, and to derive an argument or a proof

Theorems: Example

- Theorem
 - Let a , b , and c be integers. Then
 - If $a|b$ and $a|c$ then $a|(b+c)$
 - If $a|b$ then $a|bc$ for all integers c
 - If $a|b$ and $b|c$, then $a|c$
- Corrolary:
 - If a , b , and c are integers such that $a|b$ and $a|c$, then $a|mb+nc$ whenever m and n are integers
- What is the assumption? What is the conclusion?

Proofs: A General How to (1)

- An argument is valid
 - If, whenever all the hypotheses are true,
 - Then, the conclusion also holds
- From a sequence of assumptions, p_1, p_2, \dots, p_n , you draw the conclusion p . That is:

$$(p_1 \wedge p_2 \wedge \dots \wedge p_n) \rightarrow q$$

Proofs: A General How to (2)

- Usually a proof involves proving a theorem via intermediate steps
- **Example**
 - Consider the theorem ‘If $x > 0$ and $y > 0$, then $x + y > 0$ ’
 - What are the assumptions?
 - What is the conclusion?
 - What steps should we take?
 - Each intermediate step in the proof must be justified.

Rules of Inference: Modus Ponens

- Intuitively, modus ponens (or law of detachment) can be described as the inference:

p implies q; p is true; therefore q holds

- In logic terminology, modus ponens is the tautology:

$$(p \wedge (p \rightarrow q)) \rightarrow q$$

- Note: 'therefore' is sometimes denoted \therefore , so we have:

$$p \rightarrow q \equiv p \therefore q$$

Rules of Inference: Addition

- Addition involves the tautology

$$p \rightarrow (p \vee q)$$

- Intuitively,
 - if we know that p is true
 - we can conclude that either p or q are true (or both)
- In other words: $p \therefore (p \vee q)$
- Example: I read the newspaper today,
therefore I read the newspaper or I ate custard
 - Note that these are not mutually exclusive

Rules of Inference: Simplification

- Simplification is based on the tautology

$$(p \wedge q) \rightarrow p$$

- So we have: $(p \wedge q) \therefore p$
- Example: Prove that if $0 < x < 10$, then $x \geq 0$

1. $0 < x < 10 \equiv (0 < x) \wedge (x < 10)$

2. $(x > 0) \wedge (x < 10) \rightarrow (x > 0)$ by simplification

3. $(x > 0) \rightarrow (x > 0) \vee (x = 0)$ by addition

4. $(x > 0) \vee (x = 0) \equiv (x \geq 0)$ Q.E.D.

Rules of inference: Conjunction

- The conjunction is almost trivially intuitive. It is based on the following tautology:

$$((p) \wedge (q)) \rightarrow (p \wedge q)$$

- Note the subtle difference though:
 - On the left-hand side, we independently know p and q to be true
 - Therefore, we conclude, on the right-hand side, that a logical conjunction is true

Rules of Inference: Modus Tollens

- Similar to the modus ponens, modus tollens is based on the following tautology

$$(\neg q \wedge (p \rightarrow q)) \rightarrow \neg p$$

- In other words:
 - If we know that q is not true
 - And that p implies q
 - Then we can conclude that p does not hold either
- Example
 - If you are New York State citizen student, then you pay in-state tuition fee at SUNY.
 - Tom pay out-state tuition fee at SUNY.
 - Therefore we can conclude that Tom is not a New York State citizen student.

Rules of Inference: Contrapositive

- The contrapositive is the following tautology

$$(p \rightarrow q) \rightarrow (\neg q \rightarrow \neg p)$$

- Usefulness

- If you are having trouble proving the p implies q in a direct manner
- You can try to prove the contrapositive instead!

Next class

- Topic: Proofs
- Pre-class reading: Chap 1.5-1.8

