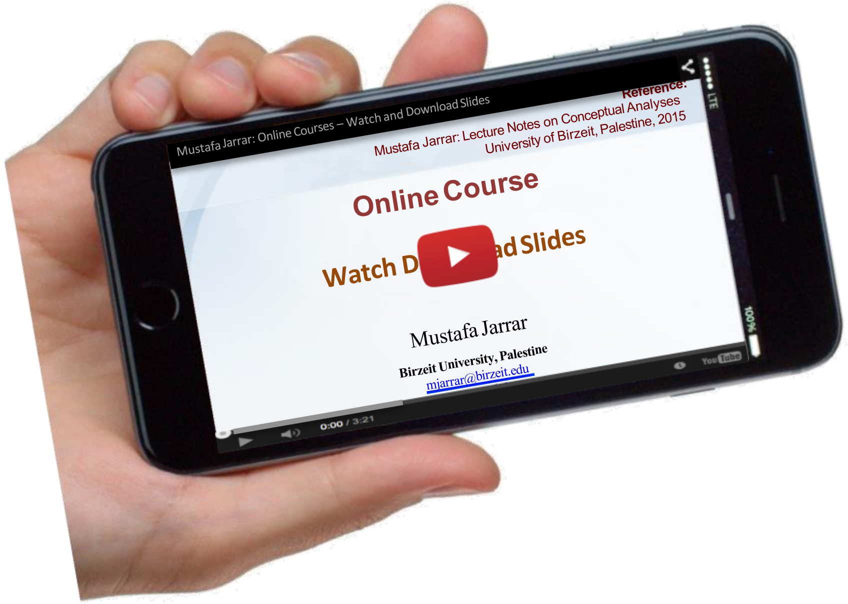
First Order Logic

Mustafa Jarrar

,

Watch this lecture and download the slides



http://jarrar-courses.blogspot.com/2014/03/discrete-mathematics-course.html

More Lectures Courses at: http://www.jarrar.info

Acknowledgement:
This lecture is based on, but not limited to, chapter 3 in "Discrete Mathematics with Applications by Susanna S. Epp (3rd Edition)".

In this Lecture

We will learn

- ❑ **Part1: Negations of Quantified Statements;**
- ❑ Part 2: Contrapositive, Converse and inverse Quantified Statements;
- ❑ Part 3: Necessary and Sufficient Conditions, Only If

Keywords: Predicates, FOL, First Order Logic, Universal Quantifier, Existential Quantifier, Negation, Truth of Universal Statements, Necessary and Sufficient Conditions

Negations of Quantified Statements

How to negate a universal statement:

All Palestinians like Zatar

Some Palestinians do not like Zatar

Theorem 3.2.1 Negation of a Universal Statement

The negation of a statement of the form

$$\forall x \text{ in } D, Q(x)$$

is logically equivalent to a statement of the form

$$\exists x \text{ in } D \text{ such that } \sim Q(x).$$

Symbolically, $\sim(\forall x \in D,\ Q(x)) \equiv \exists x \in D \text{ such that } \sim Q(x).$

,

Negations of Quantified Statements

How to negate an extensional statement:

Some Palestinians Like Zatar

All Palestinians do not like Zatar

Theorem 3.2.2 Negation of an Existential Statement

The negation of a statement of the form

$$\exists x \text{ in } D \text{ such that } Q(x)$$

is logically equivalent to a statement of the form

$$\forall x \text{ in } D, \sim Q(x).$$

Symbolically, $\sim(\exists x \in D \text{ such that } Q(x)) \equiv \forall x \in D, \sim Q(x).$

,

Negations of Quantified Statements

$\forall\, p \in \text{Prime} \,.\, \text{Odd}(p)$

Some computer hackers are over 40

All computer programs are finite

,

Negations of Quantified Statements

$\forall\, p \in \text{Prime} . \text{Odd}(p)$

$\exists p \in \text{Prime} . \sim\text{Odd}(p)$

Some computer hackers are over 40

All computer programs are finite

,

Negations of Quantified Statements

$\forall p \in \text{Prime} . \text{Odd}(p)$

$\exists p \in \text{Prime} . \sim\text{Odd}(p)$

Some computer hackers are over 40

All computer hackers are not over 40

All computer programs are finite

,

Negations of Quantified Statements

$\forall\, p \in$ Prime . Odd(p)

$\exists p \in$ Prime . ~Odd(p)

Some computer hackers are over 40

All computer hackers are not over 40

All computer programs are finite

Some computer programs are not finite

,

Negations of Quantified Statements

No politicians are honest

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

$\forall p \in \text{Person} \,.\, \text{Blond}(p) \rightarrow \text{BlueEyes}(p)$

If a computer program has more than 10000 lines then it contains a bug

,

Negations of Quantified Statements

No politicians are honest

Some politicians are honest

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

$\forall p \in \text{Person} \,.\, \text{Blond}(p) \rightarrow \text{BlueEyes}(p)$

If a computer program has more than 10000 lines then it contains a bug

,

Negations of Quantified Statements

No politicians are honest

Some politicians are honest

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

$\exists x \,.\, P(x) \wedge {\sim}Q(x)$

$\forall p \in \text{Person} \,.\, \text{Blond}(p) \rightarrow \text{BlueEyes}(p)$

If a computer program has more than 10000 lines then it contains a bug

,

Negations of Quantified Statements

No politicians are honest

Some politicians are honest

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

$\exists x \,.\, P(x) \wedge {\sim}Q(x)$

$\forall p \in \text{Person} \,.\, \text{Blond}(p) \rightarrow \text{BlueEyes}(p)$

$\exists p \in \text{Person} \,.\, \text{Blond}(p) \wedge {\sim}\text{BlueEyes}(p)$

If a computer program has more than 10000 lines then it contains a bug

,

Negations of Quantified Statements

No politicians are honest

Some politicians are honest

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

$\exists x \,.\, P(x) \land {\sim}Q(x)$

$\forall p \in \text{Person} \,.\, \text{Blond}(p) \rightarrow \text{BlueEyes}(p)$

$\exists p \in \text{Person} \,.\, \text{Blond}(p) \land {\sim}\text{BlueEyes}(p)$

If a computer program has more than 10000 lines then it contains a bug

A computer program has more than 10000 and does not contains a bug

,

In this Lecture

We will learn

- ❑ Part1: Negations of Quantified Statements;
- ❑ **Part 2: Contrapositive, Converse and Inverse Quantified Statements;**
- ❑ Part 3: Necessary and Sufficient Conditions, Only If

Keywords: Predicates, FOL, First Order Logic, Universal Quantifier, Existential Quantifier, Negation, Truth of Universal Statements, Necessary and Sufficient Conditions

,

Variants of Universal Conditional Statements

• Definition

Consider a statement of the form: $\forall x \in D,\ \text{if } P(x) \text{ then } Q(x).$

1. Its **contrapositive** is the statement: $\forall x \in D,\ \text{if } \sim Q(x) \text{ then } \sim P(x).$
2. Its **converse** is the statement: $\forall x \in D,\ \text{if } Q(x) \text{ then } P(x).$
3. Its **inverse** is the statement: $\forall x \in D,\ \text{if } \sim P(x) \text{ then } \sim Q(x).$

$$\forall x \in \text{Person} \ .\ \text{Palestinian}(x) \rightarrow \text{Smart}(x)$$

Contrapositive: $\forall x \in \text{Person} \ .\ \sim \text{Smart}(x) \rightarrow \sim\text{Palestinian}(x)$

Converse: $\forall x \in \text{Person} \ .\ \text{Smart}(x) \rightarrow \text{Palestinian}(x)$

Inverse: $\forall x \in \text{Person} \ .\ \sim\text{Palestinian}(x) \rightarrow \sim\text{Smart}(x)$

,

Variants of Universal Conditional Statements

$$\forall x \in \mathbf{R}.\quad x > 2 \rightarrow x^2 > 4.$$

$$\forall x \in \mathbf{R}.\ \text{MoreThan}(x,2) \rightarrow \text{MoreThan}(x^2,4)$$

Contrapostive: $\forall x \in \mathbf{R}\ .\ x^2 \leq 4 \rightarrow x \leq 2$

Converse: $\forall x \in \mathbf{R}\ .\ x^2 > 4 \rightarrow x > 2$

Inverse: $\forall x \in \mathbf{R}\ .\ x \leq 2 \rightarrow x^2 \leq 4$

,

Variants of Universal Conditional Statements

- **Definition**

Consider a statement of the form: $\forall x \in D,$ if $P(x)$ then $Q(x)$.

1. Its **contrapositive** is the statement: $\forall x \in D,$ if $\sim Q(x)$ then $\sim P(x)$.
2. Its **converse** is the statement: $\forall x \in D,$ if $Q(x)$ then $P(x)$.
3. Its **inverse** is the statement: $\forall x \in D,$ if $\sim P(x)$ then $\sim Q(x)$.

Logically equivalent

$$\forall x \in D, \text{ if } P(x) \text{ then } Q(x) \equiv \forall x \in D, \text{ if } \sim Q(x) \text{ then } \sim P(x)$$

$$\forall x \in D, \text{ if } P(x) \text{ then } Q(x) \not\equiv \forall x \in D, \text{ if } Q(x) \text{ then } P(x).$$

$$\forall x \in D, \text{ if } P(x) \text{ then } Q(x) \not\equiv \forall x \in D, \text{ if } \sim P(x) \text{ then } \sim Q(x).$$

,

In this Lecture

We will learn

- ❑ Part1: Negations of Quantified Statements;
- ❑ Part 2: Contrapositive, Converse and inverse Quantified Statements;
- ❑ **Part 3: Necessary and Sufficient Conditions, Only If**

Keywords: Predicates, FOL, First Order Logic, Universal Quantifier, Existential Quantifier, Negation, Truth of Universal Statements, Necessary and Sufficient Conditions

Necessary and Sufficient Conditions

- **Definition**
  - "$\forall x, r(x)$ is a **sufficient condition** for $s(x)$" means "$\forall x, r(x) \rightarrow s(x)$."
  - "$\forall x, r(x)$ is a **necessary condition** for $s(x)$" means "$\forall x, \sim r(x) \rightarrow \sim s(x)$" or, equivalently, "$\forall x, s(x) \rightarrow r(x)$."

Example:

Squareness is a sufficient condition for rectangularity.

If something is a square, then it is a rectangle.

$\forall$ x . Square(x) → Rectangular(x)

To get a job it is sufficient to be loyal.

If one is loyal (s)he will get a job

$\forall$ x . Loyal(x) → GotaJob(x)

Necessary and Sufficient Conditions

- **Definition**
  - "$\forall x, r(x)$ is a **sufficient condition** for $s(x)$" means "$\forall x, r(x) \rightarrow s(x)$."
  - "$\forall x, r(x)$ is a **necessary condition** for $s(x)$" means "$\forall x, \sim r(x) \rightarrow \sim s(x)$" or, equivalently, "$\forall x, s(x) \rightarrow r(x)$."

Example:

Being smart is necessary to get a job.

If you are not smart you don't get a job

If you got a job then you are smart

∀x . ~Smart(x) → ~GotaJob(x)

∀x . GotaJob(x) → Smart(x)

Being above 40 years is necessary for being president of Palestine

∀x . ~Above(x, 40) → ~CanBePresidentOfPalestine(x)

∀x . CanBePresidentOfPalestine(x) → Above(x, 40)