Department of
Computer Science and Engineering
计算机科学与工程系

CS215 DISCRETE MATH

Dr. Qi Wang

Department of Computer Science and Engineering
Office: Room413, CoE South Tower
Email: wangqi@sustech.edu.cn

Compound Propositions

- A *proposition* is a declarative statement that is either true or false. More complex propositions can be built from elementary statements using logical connectives.

- Logical connectives:
  - ◇ $\neg p$ (*Negation*)
  - ◇ $p \wedge q$ (*Conjunction*)
  - ◇ $p \vee q$ (*Disjunction*)
  - ◇ $p \oplus q$ (*Exclusive or*)
  - ◇ $p \rightarrow q$ (*Implication*)
  - ◇ $p \leftrightarrow q$ (*Biconditional*)

Conditional Statement (implication)

- Let p and q be propositions. The *conditional statment* (a.k.a. *implication*) $p \to q$, is the proposition "if p, then q", is false when p is true and q is false, and true otherwise. In $p \to q$, p is called the *hypothesis* and q is called the *conclusion*.

p	q	$p \to q$
T	T	T
T	F	F
F	T	T
F	F	T

Implication

- $p \rightarrow q$ is read in a variety of equivalent ways:
  - ◇ if p then q
  - ◇ p implies q
  - ◇ p is sufficient for q
  - ◇ q is necessary for p
  - ◇ q follows from p
  - ◇ q unless $\neg p$
  - ◇ p only if q

Implication

- $p \rightarrow q$ is read in a variety of equivalent ways:
  - ◇ if p then q
  - ◇ p implies q
  - ◇ p is sufficient for q
  - ◇ q is necessary for p
  - ◇ q follows from p
  - ◇ q unless $\neg p$
  - ◇ p only if q

Example:

- ◇ If you get 100 on the final, then you will get an A.

p q

Implication

- The *converse* of $p \to q$ is $q \to p$.
  The *contrapositive* of $p \to q$ is $\neg q \to \neg p$.
  The *inverse* of $p \to q$ is $\neg p \to \neg q$.

Implication

- The *converse* of $p \rightarrow q$ is $q \rightarrow p$.
  The *contrapositive* of $p \rightarrow q$ is $\neg q \rightarrow \neg p$.
  The *inverse* of $p \rightarrow q$ is $\neg p \rightarrow \neg q$.

  Examples:

  ◇ If you get 100 on the final, then you will get an A. ($p \rightarrow q$)

  ◇ If you get an A, then you get 100 on the final. ($q \rightarrow p$)

  ◇ If you don't get an A, then you don't get 100 on the final. ($\neg q \rightarrow \neg p$)

  ◇ If you don't get 100 on the final, then you don't get an A. ($\neg p \rightarrow \neg q$)

Implication

- The *converse* of $p \to q$ is $q \to p$.
  The *contrapositive* of $p \to q$ is $\neg q \to \neg p$.
  The *inverse* of $p \to q$ is $\neg p \to \neg q$.

  Examples:

  ◇ If you get 100 on the final, then you will get an A. ($p \to q$)

  ◇ If you get an A, then you get 100 on the final. ($q \to p$)
  ◇ If you don't get an A, then you don't get 100 on the final. ($\neg q \to \neg p$)
  ◇ If you don't get 100 on the final, then you don't get an A. ($\neg p \to \neg q$)

  $\neg q \to \neg p$ is *equivalent* to $p \to q$

Constructing the Truth Table

- Construct a truth table for $p \vee q \rightarrow \neg r$

Constructing the Truth Table

- Construct a truth table for $p \vee q \rightarrow \neg r$

p	q	r	¬r	p ∨ q	p ∨ q → ¬r
T	T	T	F	T	F
T	T	F	T	T	T
T	F	T	F	T	F
T	F	F	T	T	T
F	T	T	F	T	F
F	T	F	T	T	T
F	F	T	F	F	T
F	F	F	T	F	T

Computer Representation of True and False

- A *bit* is sufficient to represent two possible values: 0 (false) or 1 (true).
- A varaiable that takes on values 0 and 1 is called a *Boolean variable*.
- A *bit string* is a sequence of zero or more bits. The *length* of this string is the number of bits in the string.

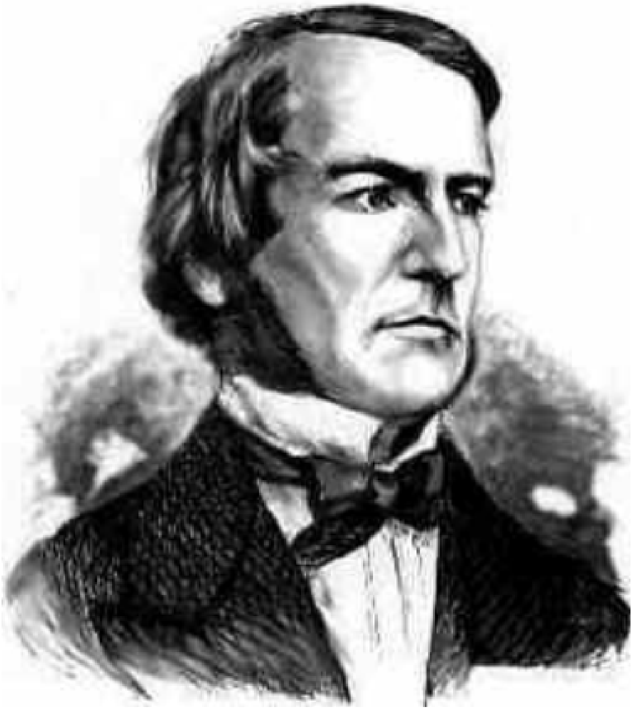
Computer Representation of True and False

- A *bit* is sufficient to represent two possible values: 0 (false) or 1 (true).
- A varaiable that takes on values 0 and 1 is called a *Boolean variable*.
- A *bit string* is a sequence of zero or more bits. The *length* of this string is the number of bits in the string.
- *bitwise operations*: replace "T" and "F" with 1 and 0.

$$\begin{array}{r} 1011\ 0011 \\ \vee\ \underline{0110\ 1010} \\ 1111\ 1011 \end{array}$$

George Boole

- British mathematician (b. 1815, d. 1864)
  - ◇ The inventor of Boolean algebra
  - ◇ Truth tables are an example of B.A.



◇ Although Boole's work was not originally perceived as particularly interesting, even by other mathematicians, he is now seen as one of the founders of the field of Computer Science.

Applications of Propositional Logic

- Translation of English sentences
  - ◇ use atomic (elementary) propositions
- Inference and reasoning
  - ◇ new true propositions are inferred from existing ones
  - ◇ used in *Artificial Intelligence*:
    - – rule based (expert) systems, automatic theorem provers, ...
- Design of logic circuit

Translation

- If you are older than 13 or you are with your parents then you can watch this movie.

Translation

- If you are older than 13 or you are with your parents then you can watch this movie.

  If (you are older than 13) **or** (you are with your parents) **then** (you can watch this movie).

Translation

- If you are older than 13 or you are with your parents then you can watch this movie.

  If (you are older than 13) **or** (you are with your parents) **then** (you can watch this movie).

  Atomic (**elementary**) propositions:

  p – you are older than 13
  q – you are with your parents
  r – you can watch this movie

Translation

- If you are older than 13 or you are with your parents then you can watch this movie.

  If (you are older than 13) **or** (you are with your parents) **then** (you can watch this movie).

  Atomic (**elementary**) propositions:

  p – you are older than 13
  q – you are with your parents
  r – you can watch this movie

  Translation: $p \vee q \rightarrow r$

Inference

- **If** (you are older than 13) **or** (you are with your parents) **then** (you can watch this movie).

  Translation: $p \vee q \rightarrow r$

Inference

- **If** (you are older than 13) **or** (you are with your parents) **then** (you can watch this movie).

  Translation: $p \vee q \rightarrow r$

  Given that p is true.

  With the help of the logic, we can infer the following statement:

  – You can watch this movie.

Inference

- *Artificial intelligence*
  - ◇ builds programs that act intelligently
  - ◇ programs often rely on symbolic manipulations

Inference

- *Artificial intelligence*
  - ◇ builds programs that act intelligently
  - ◇ programs often rely on symbolic manipulations

- *Expert systems*
  - ◇ encode knowledge about the world in logic
  - ◇ support inferences where new facts are inferred from existing ones following the semantics of logic

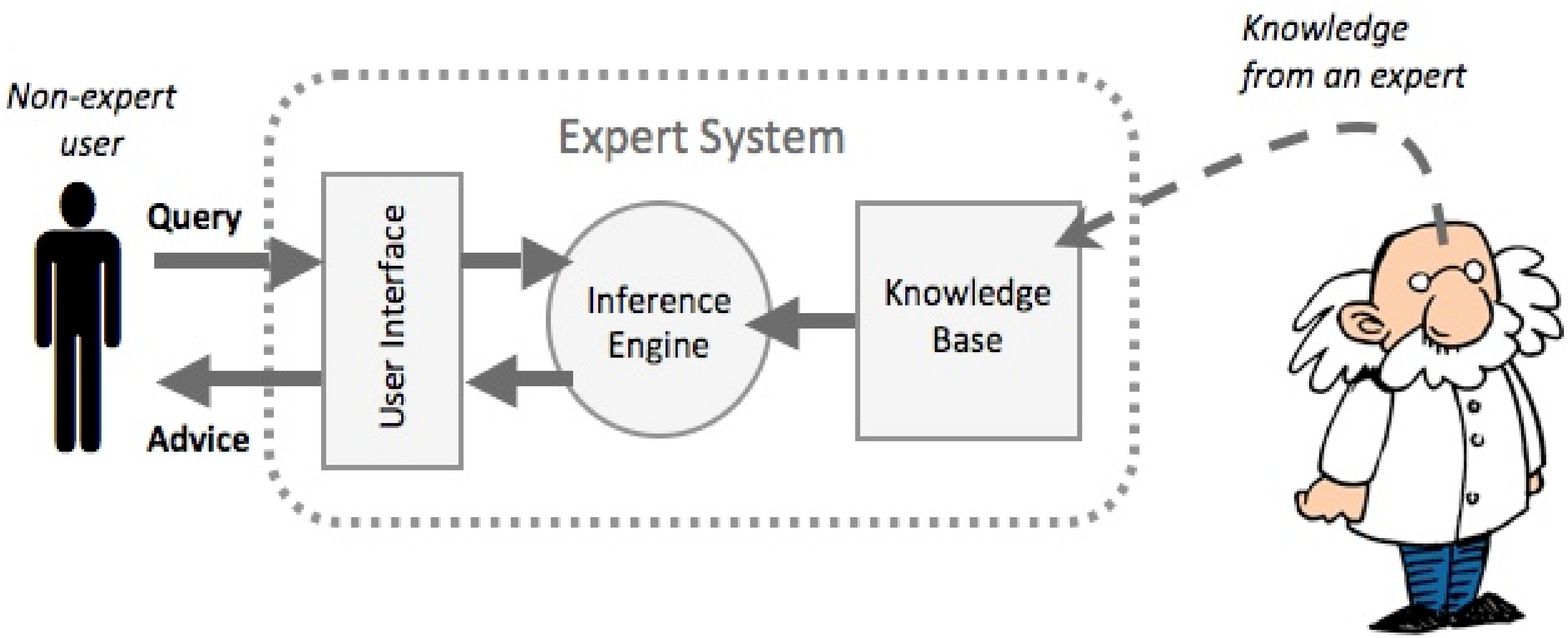
Inference

- *Artificial intelligence*
  - ◇ builds programs that act intelligently
  - ◇ programs often rely on symbolic manipulations

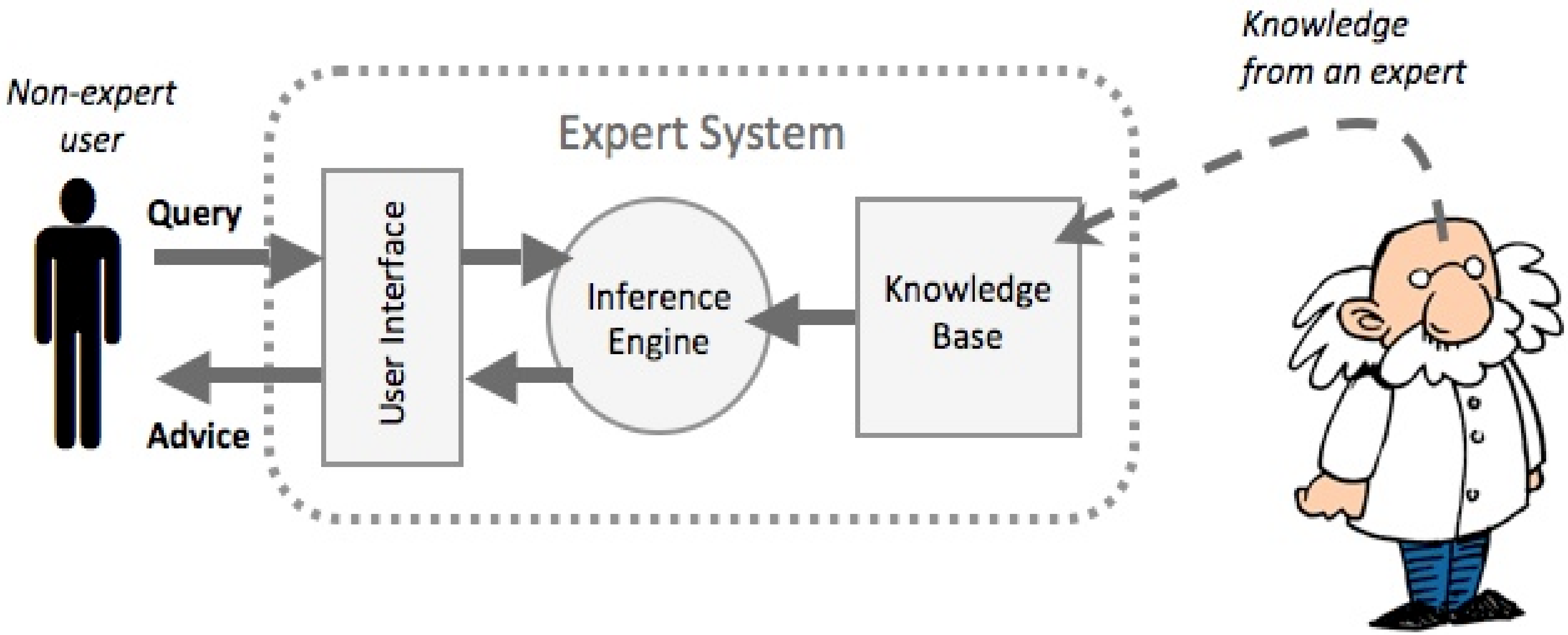
- *Expert systems*
  - ◇ encode knowledge about the world in logic
  - ◇ support inferences where new facts are inferred from existing ones following the semantics of logic

- *Theorem provers*
  - ◇ encode existing knowledge (e.g., math) using logic
  - ◇ show that some hypothesis is true

Expert System



Expert System



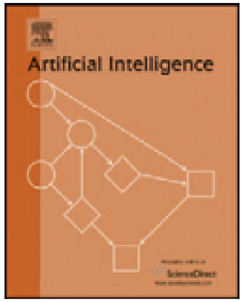
Theorem Prover

Contents lists available at ScienceDirect

Artificial Intelligence

www.elsevier.com/locate/artint

ELSEVIER



Computer-aided proofs of Arrow's and other impossibility theorems ☆

Pingzhong Tang *, Fangzhen Lin

Department of Computer Science, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong

A R T I C L E   I N F O

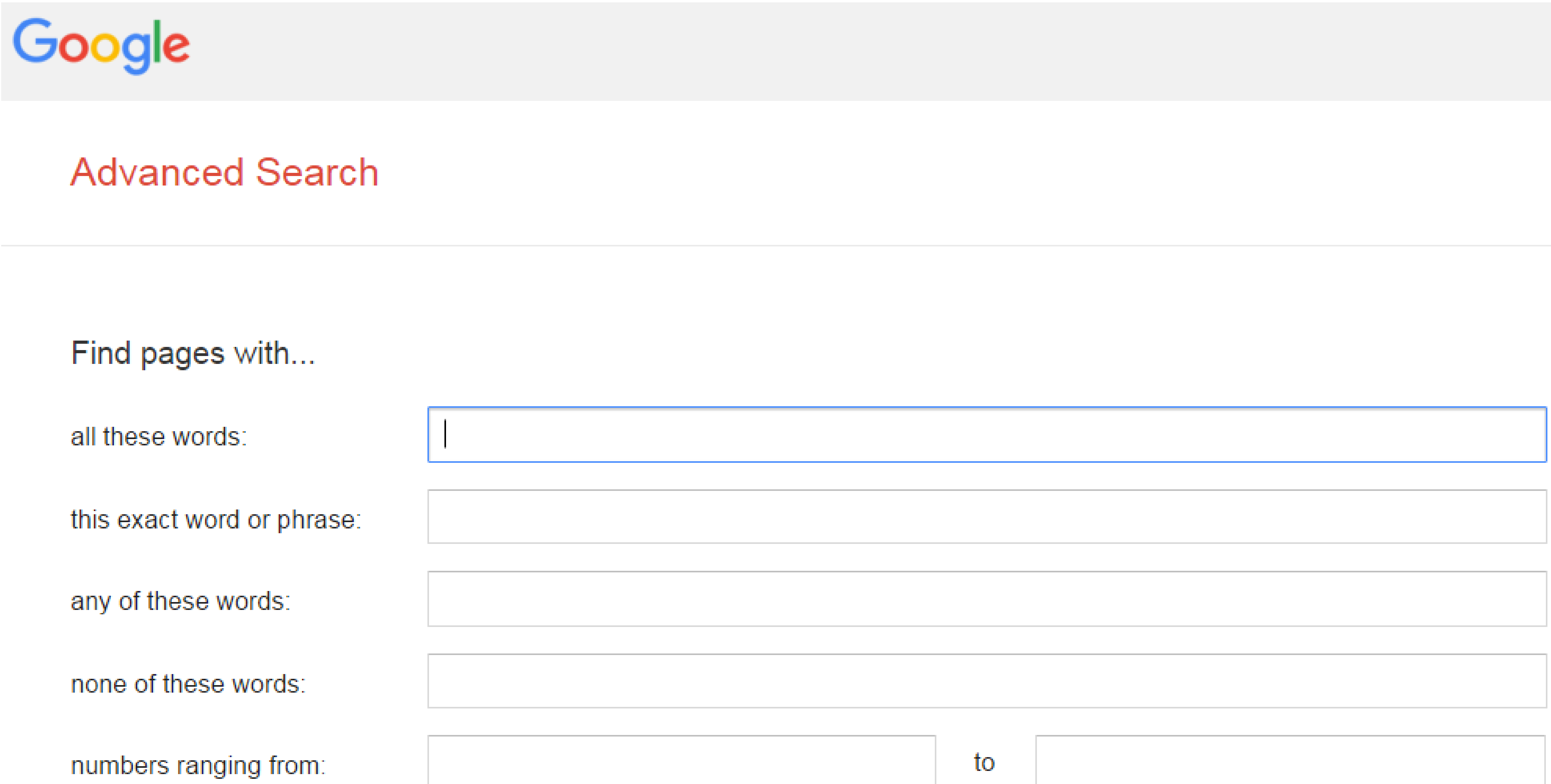
Article history:
Received 20 October 2008
Received in revised form 13 February 2009
Accepted 24 February 2009
Available online 4 March 2009

Keywords:
Social choice theory
Arrow's theorem
Muller–Satterthwaite theorem
Sen's theorem
Knowledge representation
Computer-aided theorem proving

A B S T R A C T

Arrow's impossibility theorem is one of the landmark results in social choice theory. Over the years since the theorem was proved in 1950, quite a few alternative proofs have been put forward. In this paper, we propose yet another alternative proof of the theorem. The basic idea is to use induction to reduce the theorem to the base case with 3 alternatives and 2 agents and then use computers to verify the base case. This turns out to be an effective approach for proving other impossibility theorems such as Muller–Satterthwaite and Sen's theorems as well. Motivated by the insights of the proof, we discover a new theorem with the help of computer programs. We believe this new proof opens an exciting prospect of using computers to discover similar impossibility or even possibility results.

Other Applications


Google

Advanced Search

Find pages with...

all these words:

this exact word or phrase:

any of these words:

none of these words:

numbers ranging from: to

Tautology and Contradiction

- A compound proposition that is always true for all possible truth values is called a *tautology*.
- A compound proposition that is always false for all possible truth values is called a *contradiction*.
- A compound proposition that is neither a tautology nor a contradiction is called a *contingency*.

Tautology and Contradiction

- A compound proposition that is always true for all possible truth values is called a *tautology*.
- A compound proposition that is always false for all possible truth values is called a *contradiction*.
- A compound proposition that is neither a tautology nor a contradiction is called a *contingency*.

p	$\neg p$	$p \vee \neg p$	$p \wedge \neg p$
T	F	T	F
F	T	T	F

Equivalent Propositions

- Two propositions are *equivalent* if they always have the same truth value.

Equivalent Propositions

- Two propositions are *equivalent* if they always have the same truth value.

Examples:

```
(1) if ((i+j ≤ p+q) && (i ≤ p) &&
    ((j > q) || (List1[i] ≤ List2[j])))
(2)   List3[k] = List1[i]
(3)   i = i+1
(4) else
(5)   List3[k] = List2[j]
(6)   j = j+1
(7) k = k+1
```

```
(1) if (((i+j ≤ p+q) && (i ≤ p) && (j > q))
    || ((i+j ≤ p+q) && (i ≤ p)
         && (List1[i] ≤ List2[j])))
(2)   List3[k] = List1[i]
(3)   i = i+1
(4) else
(5)   List3[k] = List2[j]
(6)   j = j+1
(7) k = k+1
```

Consider the two pieces of codes taken from two different versions of *Mergesort*. Do they do the same thing?

Equivalent Propositions

```
(1) if ((i+j ≤ p+q) && (i ≤ p) &&
       ((j > q) || (List1[i] ≤ List2[j])))
(2)    List3[k] = List1[i]
(3)    i = i+1
(4) else
(5)    List3[k] = List2[j]
(6)    j = j+1
(7) k = k+1
```

```
(1) if (((i+j ≤ p+q) && (i ≤ p) && (j > q))
       || ((i+j ≤ p+q) && (i ≤ p)
          && (List1[i] ≤ List2[j])))
(2)    List3[k] = List1[i]
(3)    i = i+1
(4) else
(5)    List3[k] = List2[j]
(6)    j = j+1
(7) k = k+1
```

- Let's rewrite using
  $s \sim (i + j \leq p + q)$, $t \sim (i \leq p)$, $u \sim (j > q)$
  $v \sim (List1[i] \leq List2[j])$

Equivalent Propositions

```
(1) if ((i+j ≤ p+q) && (i ≤ p) &&
    ((j > q) || (List1[i] ≤ List2[j])))
(2)   List3[k] = List1[i]
(3)   i = i+1
(4) else
(5)   List3[k] = List2[j]
(6)   j = j+1
(7) k = k+1
```

```
(1) if (((i+j ≤ p+q) && (i ≤ p) && (j > q))
    || ((i+j ≤ p+q) && (i ≤ p)
       && (List1[i] ≤ List2[j])))
(2)   List3[k] = List1[i]
(3)   i = i+1
(4) else
(5)   List3[k] = List2[j]
(6)   j = j+1
(7) k = k+1
```

- Let's rewrite using
  $s \sim (i + j \leq p + q)$, $t \sim (i \leq p)$, $u \sim (j > q)$
  $v \sim (List1[i] \leq List2[j])$

(1) $s \wedge t \wedge (u \vee v)$

(1') $(s \wedge t \wedge u) \vee (s \wedge t \wedge v)$

Equivalent Propositions

```
(1) if ((i+j ≤ p+q) && (i ≤ p) &&
       ((j > q) || (List1[i] ≤ List2[j])))
(2)    List3[k] = List1[i]
(3)    i = i+1
(4) else
(5)    List3[k] = List2[j]
(6)    j = j+1
(7) k = k+1
```

```
(1) if (((i+j ≤ p+q) && (i ≤ p) && (j > q))
       || ((i+j ≤ p+q) && (i ≤ p)
           && (List1[i] ≤ List2[j])))
(2)    List3[k] = List1[i]
(3)    i = i+1
(4) else
(5)    List3[k] = List2[j]
(6)    j = j+1
(7) k = k+1
```

- Let's rewrite using
  $s \sim (i + j \leq p + q)$, $t \sim (i \leq p)$, $u \sim (j > q)$
  $v \sim (List1[i] \leq List2[j])$

  (1) $s \wedge t \wedge (u \vee v)$ (1') $(s \wedge t \wedge u) \vee (s \wedge t \wedge v)$

  Now set $w \sim (s \wedge t)$

  (1) $w \wedge (u \vee v)$ (1') $(w \wedge u) \vee (w \wedge v)$

Truth Tables

(1) $w \wedge (u \vee v)$

w	u	v	$u \vee v$	$w \wedge (u \vee v)$
T	T	T	T	T
T	T	F	T	T
T	F	T	T	T
T	F	F	F	F
F	T	T	T	F
F	T	F	T	F
F	F	T	T	F
F	F	F	F	F

(1') $(w \wedge u) \vee (w \wedge v)$

w	u	v	$w \wedge u$	$w \wedge v$	$(w \wedge u) \vee (w \wedge v)$
T	T	T	T	T	T
T	T	F	T	F	T
T	F	T	F	T	T
T	F	F	F	F	F
F	T	T	F	F	F
F	T	F	F	F	F
F	F	T	F	F	F
F	F	F	F	F	F

Distributive Laws

- $(w \wedge (u \vee v))$ is *equivalent* to $(w \wedge u) \vee (w \wedge v)$

Distributive Laws

- $(w \wedge (u \vee v))$ is *equivalent* to $(w \wedge u) \vee (w \wedge v)$
  $(w \vee (u \wedge v))$ is *equivalent* to $(w \vee u) \wedge (w \vee v)$

Distributive Laws

- $(w \wedge (u \vee v))$ is *equivalent* to $(w \wedge u) \vee (w \wedge v)$

  $(w \vee (u \wedge v))$ is *equivalent* to $(w \vee u) \wedge (w \vee v)$

- The propositions p and q are called *logically equivalent* if $p \leftrightarrow q$ is a tautology, denoted by $p \equiv q$ or $p \Leftrightarrow q$.

Distributive Laws

- $(w \wedge (u \vee v))$ is *equivalent* to $(w \wedge u) \vee (w \wedge v)$

  $(w \vee (u \wedge v))$ is *equivalent* to $(w \vee u) \wedge (w \vee v)$

- The propositions p and q are called *logically equivalent* if $p \leftrightarrow q$ is a tautology, denoted by $p \equiv q$ or $p \Leftrightarrow q$.

- Equivalent statements are important for logical reasoning since they can be substituted and can help us to:
  – make a logical argument
  – infer new propositions

Distributive Laws

- $(w \wedge (u \vee v))$ is *equivalent* to $(w \wedge u) \vee (w \wedge v)$

  $(w \vee (u \wedge v))$ is *equivalent* to $(w \vee u) \wedge (w \vee v)$

- The propositions p and q are called *logically equivalent* if $p \leftrightarrow q$ is a tautology, denoted by $p \equiv q$ or $p \Leftrightarrow q$.

- Equivalent statements are important for logical reasoning since they can be substituted and can help us to:
  – make a logical argument
  – infer new propositions

  Example $p \to q \equiv \neg q \to \neg p$

De Morgan's Laws

- $\neg(p \vee q) \equiv \neg p \wedge \neg q$

  $\neg(p \wedge q) \equiv \neg p \vee \neg q$

De Morgan's Laws

- $\neg(p \vee q) \equiv \neg p \wedge \neg q$

  $\neg(p \wedge q) \equiv \neg p \vee \neg q$

p	q	$\neg p$	$\neg q$	$(p \vee q)$	$\neg(p \vee q)$	$\neg p \wedge \neg q$
T	T	F	F	T	F	F
T	F	F	T	T	F	F
F	T	T	F	T	F	F
F	F	T	T	F	T	T

Important Logical Equivalences

- *Identity laws*
  - $p \wedge T \equiv p$
  - $p \vee F \equiv p$
- *Domination laws*
  - $p \vee T \equiv T$
  - $p \wedge F \equiv F$
- *Idempotent laws*
  - $p \vee p \equiv p$
  - $p \wedge p \equiv p$

Important Logical Equivalences

- *Double negation laws*
  - $\neg(\neg p) \equiv p$

- *Commutative laws*
  - $p \vee q \equiv q \vee p$
  - $p \wedge q \equiv q \wedge p$

- *Associative laws*
  - $(p \vee q) \vee r \equiv p \vee (q \vee r)$
  - $(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$

Important Logical Equivalences

- *Distributive laws*
  - $p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$
  - $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$
- *De Morgan's laws*
  - $\neg(p \vee q) \equiv \neg p \wedge \neg q$
  - $\neg(p \wedge q) \equiv \neg p \vee \neg q$
- *Others*
  - $p \vee (p \wedge q) \equiv p$
  - $p \wedge (p \vee q) \equiv p$

    Absorption laws
  - $p \vee \neg p \equiv T$
  - $p \wedge \neg p \equiv F$

    Negation laws
  - $p \rightarrow q \equiv \neg p \vee q$

    Useful law

Using Logical Equivalences

- Equivalences can be used in proofs. A proposition or its part can be transformed using equivalences.

Using Logical Equivalences

- Equivalences can be used in proofs. A proposition or its part can be transformed using equivalences.

- **Example**: Show that $(p \land q) \rightarrow p$ is a tautology.

Using Logical Equivalences

- Equivalences can be used in proofs. A proposition or its part can be transformed using equivalences.

- **Example**: Show that $(p \wedge q) \to p$ is a tautology.

  Proof:

$$
\begin{aligned}
(p \wedge q) \to p &\equiv \neg(p \wedge q) \vee p && \text{Useful} \\
&\equiv (\neg p \vee \neg q) \vee p && \text{De Morgan's} \\
&\equiv (\neg q \vee \neg p) \vee p && \text{Commutative} \\
&\equiv \neg q \vee (\neg p \vee p) && \text{Associative} \\
&\equiv \neg q \vee T && \text{Negation} \\
&\equiv T && \text{Domination}
\end{aligned}
$$

Using Logical Equivalences

- Equivalences can be used in proofs. A proposition or its part can be transformed using equivalences.

- **Example**: Show that $(p \wedge q) \rightarrow p$ is a tautology.

  Proof (alternatively):

p	q	$p \wedge q$	$(p \wedge q) \rightarrow p$
T	T	T	**T**
T	F	F	**T**
F	T	F	**T**
F	F	F	**T**

Using Logical Equivalences

- Equivalences can be used in proofs. A proposition or its part can be transformed using equivalences.

- **Example**: Show that $p \to q \equiv \neg q \to \neg p$.

Using Logical Equivalences

- Equivalences can be used in proofs. A proposition or its part can be transformed using equivalences.

- **Example**: Show that $p \to q \equiv \neg q \to \neg p$.

Proof:

$\neg q \to \neg p \equiv \neg(\neg q) \vee (\neg p)$	Useful
$\equiv q \vee (\neg p)$	Double negation
$\equiv (\neg p) \vee q$	Communtative
$\equiv p \to q$	Useful

Limitations of Propositional Logic

- Propositional logic: the world is described in terms of elementary propositions and their logical combinations. Elementary statements typically refer to objects, their properties and relations.

Limitations of Propositional Logic

- Propositional logic: the world is described in terms of elementary propositions and their logical combinations. Elementary statements typically refer to objects, their properties and relations.

  Example 1: (repeated statements for many objects)
  - ◇ "$3^2 \geq 0$"

Limitations of Propositional Logic

- Propositional logic: the world is described in terms of elementary propositions and their logical combinations. Elementary statements typically refer to objects, their properties and relations.

  Example 1: (repeated statements for many objects)

  - "$3^2 \geq 0$"

    object property

- Propositional logic: the world is described in terms of elementary propositions and their logical combinations. Elementary statements typically refer to objects, their properties and relations.

Example 1: (repeated statements for many objects)

- "$3^2 \geq 0$"

  object property

- "$2^2 \geq 0$"
- "$1^2 \geq 0$"
- "$0^2 \geq 0$"
- "$(-1)^2 \geq 0$"
- ...

- Propositional logic: the world is described in terms of elementary propositions and their logical combinations. Elementary statements typically refer to objects, their properties and relations.

Example 1: (repeated statements for many objects)

- "$3^2 \geq 0$"

object property

- "$2^2 \geq 0$"
- "$1^2 \geq 0$"
- "$0^2 \geq 0$"
- "$(-1)^2 \geq 0$"
- …

What is a more natural solution to express the knowledge?

- Propositional logic: the world is described in terms of elementary propositions and their logical combinations. Elementary statements typically refer to objects, their properties and relations.

Example 1: (repeated statements for many objects)

- "$3^2 \geq 0$"

object property

- "$2^2 \geq 0$"
- "$1^2 \geq 0$"
- "$0^2 \geq 0$"
- "$(-1)^2 \geq 0$"
- ...

What is a more natural solution to express the knowledge?

Solution: make statements with variables

– If x is an integer, then $x^2 \geq 0$.

– x is an integer $\rightarrow x^2 \geq 0$.

Limitations of Propositional Logic

- Propositional logic: the world is described in terms of elementary propositions and their logical combinations. Elementary statements typically refer to objects, their properties and relations.

  Example 2: (statements that define the property of a group of objects)
  - "The square of every integer is ≥ 0"
  - "Some of the integers are prime."

Limitations of Propositional Logic

- Propositional logic: the world is described in terms of elementary propositions and their logical combinations. Elementary statements typically refer to objects, their properties and relations.

Example 2: (statements that define the property of a group of objects)

- ◇ "The square of every integer is ≥ 0"
- ◇ "Some of the integers are prime."

Solutions: make statements with *quantifiers*

– universal quantifier: the property is satisfied by all members of the group

– existential quantifier: at least one member of the group satisfies the property

Predicate Logic

- Remedies the limitations of propositional logic:
  - ◇ explicitly models objects and their properties
  - ◇ allows to make statements with variables and quantify them

Predicate Logic

- Remedies the limitations of propositional logic:
  - ◇ explicitly models objects and their properties
  - ◇ allows to make statements with variables and quantify them

- Basic building blocks of the predicate logic:
  - ◇ *Constant* – models a specific object

    Examples: "1", "SUSTech", ...
  - ◇ *Variable* – represents object of specific type

    Examples: x, y, ... (universe can be people, numbers ...)
  - ◇ *Predicate* – represents properties or relations among objects

    Examples: Red(car23), student(x), married(A, B)...

Predicates

- A *predicate* $P(x)$ assigns a value T or F to each x depending on whether the property holds or not for x.

Predicates

- A *predicate* $P(x)$ assigns a value T or F to each x depending on whether the property holds or not for x.

  Example: Assume Prime(x) where the universe of discourse are integers

  - Prime(2) ... T
  - Prime(6) ... F
  - ...

Predicates

- A *predicate* $P(x)$ assigns a value T or F to each x depending on whether the property holds or not for x.

  Example: Assume Prime(x) where the universe of discourse are integers

  - Prime(2) ... T
  - Prime(6) ... F
  - ...

  Is Prime(x) a proposition?

Predicates

- A *predicate* $P(x)$ assigns a value T or F to each x depending on whether the property holds or not for x.

  Example: Assume Prime(x) where the universe of discourse are integers

  - Prime(2) ... T
  - Prime(6) ... F
  - ...

  Is Prime(x) a proposition?

  No, but after the substitution it becomes one.

Predicates

- A *predicate* is a statement $P(x_1, x_2, \ldots, x_n)$ that contains n variables $x_1, x_2, \ldots, x_n$ and becomes a proposition when specific values are substituted for the variables x_i.

- The *universe* (*domain*) D of the *predicate variables* $(x_1, x_2, \ldots, x_n)$ is the set of all values that may be substituted in place of the variables.

- The *truth set* of $P(x_1, x_2, \ldots, x_n)$ is the set of all values of the predicate variables $(x_1, x_2, \ldots, x_n)$ such that the proposition $P(x_1, x_2, \ldots, x_n)$ is true.

Examples of Predicates

- **Example 1**: (Predicates with one variable)
  Let $P(x)$ be the predicate "$x^2 > x$" with universe of the real numbers.
  - ◇ What are the truth values of $P(2)$ and $P(1)$?
  - ◇ What is the truth set of $P(x)$?

Examples of Predicates

- **Example 1**: (Predicates with one variable)
  Let $P(x)$ be the predicate "$x^2 > x$" with universe of the real numbers.
  - ◇ What are the truth values of $P(2)$ and $P(1)$?
  - ◇ What is the truth set of $P(x)$?

  - ◇ $P(2) = \mathrm{T}$, $P(1) = \mathrm{F}$
  - ◇ $x > 1$ or $x < 0$

Examples of Predicates

- **Example 2**: (Predicates with two variables)
  Let $Q(x, y)$ be the predicate "$x = y + 3$" with universe of the real numbers.
  - ◇ What are the truth values of $Q(1, 2)$ and $Q(3, 0)$?
  - ◇ What is the truth set of $Q(x, y)$?

Examples of Predicates

- **Example 2**: (Predicates with two variables)
  Let $Q(x, y)$ be the predicate "$x = y + 3$" with universe of the real numbers.
  - ◇ What are the truth values of $Q(1, 2)$ and $Q(3, 0)$?
  - ◇ What is the truth set of $Q(x, y)$?

  - ◇ $Q(1, 2) = F$, $Q(3, 0) = T$
  - ◇ $(a, a - 3)$ for all real numbers a

Compound Statements in Predicate Logic

- Compound statements are obtained via logical connectives.

Compound Statements in Predicate Logic

- Compound statements are obtained via logical connectives.

Example:

- ◇ $\text{Prime}(2) \wedge \text{Prime}(3)$
  – Translation: "Both 2 and 3 are primes." (T)
- ◇ $\text{City}(\text{Shenzhen}) \vee \text{River}(\text{Shenzhen})$
  – Translation: "Shenzhen is a city or a river." (T)
- ◇ $\text{CS-major}(x) \rightarrow \text{Student}(x)$
  – Translation: "If x is CS-major then x is a student." (not a proposition)

Predicates

- The statement $P(x)$ is not a proposition since there are many objects that it can be applied to.

Predicates

- The statement $P(x)$ is not a proposition since there are many objects that it can be applied to.

- But the difference is:

  ◇ predicate logic allows us to explicitly manipulate and substitute for the objects

  ◇ predicate logic permits quantified sentences where variables are substituted for statements about the group of objects

Quantified Statements

- Two types of quantified statements:
  - *Universal*

    Example: "All CS-major graduates have to pass CS201".
    (This is true for all CS-major graduates.)

  - *Existential*

    Example: "Some CS-major students graduate with honor."
    (This is true for some students.)

Universal Quantifier

- The *universal quantification* of $P(x)$ is the proposition: "$P(x)$ is true for all values of x in the universe of discourse." denoted by $\forall x\ P(x)$, and is expressed as for every x, $P(x)$.

Universal Quantifier

- The *universal quantification* of $P(x)$ is the proposition: "$P(x)$ is true for all values of x in the universe of discourse." denoted by $\forall x\ P(x)$, and is expressed as for every x, $P(x)$.

  Example:

  ◇ $P(x)$ – "$x > x - 1$".
  ◇ What is the truth value of $\forall x\ P(x)$?

Universal Quantifier

- The *universal quantification* of $P(x)$ is the proposition: "$P(x)$ is true for all values of x in the universe of discourse." denoted by $\forall x\ P(x)$, and is expressed as for every x, $P(x)$.

  Example:

  - $P(x)$ – "$x > x - 1$".
  - What is the truth value of $\forall x\ P(x)$?
  - Assume that the universe is all real numbers
  - $\forall x\ P(x)$ is true.

Universal Quantifier

- The *universal quantification* of $P(x)$ is the proposition: "$P(x)$ is true for all values of x in the universe of discourse." denoted by $\forall x\ P(x)$, and is expressed as for every x, $P(x)$.

Example:

◇ $P(x)$ – "$x > x - 1$".
◇ What is the truth value of $\forall x\ P(x)$?

◇ Assume that the universe is all real numbers
◇ $\forall x\ P(x)$ is true.

◇ Is $P(x)$ a proposition?
◇ Is $\forall x\ P(x)$ a proposition?

Universal Quantifier

- The *universal quantification* of $P(x)$ is the proposition: "$P(x)$ is true for all values of x in the universe of discourse." denoted by $\forall x\ P(x)$, and is expressed as for every x, $P(x)$.

Example:

◇ $P(x)$ – "$x > x - 1$".
◇ What is the truth value of $\forall x\ P(x)$?

◇ Assume that the universe is all real numbers
◇ $\forall x\ P(x)$ is true.

◇ Is $P(x)$ a proposition? No. Many possible substitutions.
◇ Is $\forall x\ P(x)$ a proposition?
Yes. True if for all x from the universe $P(x)$ is ture.

Existential Quantifier

- The *existential quantification* of $P(x)$ is the proposition: "There exists an element in the universe of discourse such that $P(x)$ is true ." denoted by $\exists x\ P(x)$, and is expressed as there is an x such that $P(x)$.

Existential Quantifier

- The *existential quantification* of $P(x)$ is the proposition: "There exists an element in the universe of discourse such that $P(x)$ is true ." denoted by $\exists x\ P(x)$, and is expressed as there is an x such that $P(x)$.

  Example:

  - $P(x)$ – "$x > 5$".
  - What is the truth value of $\exists x\ P(x)$?

Existential Quantifier

- The *existential quantification* of $P(x)$ is the proposition: "There exists an element in the universe of discourse such that $P(x)$ is true ." denoted by $\exists x \; P(x)$, and is expressed as there is an x such that $P(x)$.

Example:

- ◇ $P(x)$ – "$x > 5$".
- ◇ What is the truth value of $\exists x \; P(x)$?
- ◇ Assume that the universe is all real numbers
- ◇ $\exists x \; P(x)$ is true.

Existential Quantifier

- The *existential quantification* of $P(x)$ is the proposition: "There exists an element in the universe of discourse such that $P(x)$ is true ." denoted by $\exists x\ P(x)$, and is expressed as there is an x such that $P(x)$.

Example:

- ◇ $P(x)$ – "$x > 5$".
- ◇ What is the truth value of $\exists x\ P(x)$?
- ◇ Assume that the universe is all real numbers
- ◇ $\exists x\ P(x)$ is true.
- ◇ Is $P(x)$ a proposition?
- ◇ Is $\exists x\ P(x)$ a proposition?

Existential Quantifier

- The *existential quantification* of $P(x)$ is the proposition: "There exists an element in the universe of discourse such that $P(x)$ is true ." denoted by $\exists x \; P(x)$, and is expressed as there is an x such that $P(x)$.

Example:

- ◇ $P(x)$ – "$x > 5$".
- ◇ What is the truth value of $\exists x \; P(x)$?
- ◇ Assume that the universe is all real numbers
- ◇ $\exists x \; P(x)$ is true.
- ◇ Is $P(x)$ a proposition? No. Many possible substitutions.
- ◇ Is $\exists x \; P(x)$ a proposition?

Yes. True if there is even an x s.t. $P(x)$ is ture (e.g. 10).

Existential Quantifier

- **Example**:
  - $Q(x)$ – "$x = x + 2$ where x is a real number".
  - What is the truth value of $\exists x\ Q(x)$?

Existential Quantifier

- **Example**:
  - $Q(x)$ – "$x = x + 2$ where x is a real number".
  - What is the truth value of $\exists x \; Q(x)$?
  - $\exists x \; Q(x)$ is false.

Existential Quantifier

- **Example**:
  - ◇ $Q(x)$ – "$x = x + 2$ where x is a real number".
  - ◇ What is the truth value of $\exists x\ Q(x)$?
  - ◇ $\exists x\ Q(x)$ is false.

  - ◇ $C(x)$ – CS-major(x) $\wedge$ Honor-student(x).
  - ◇ What is the truth value of $\exists x\ C(x)$?
  - ◇ Translation: "There is a person who is a CS-major student and who also graduated with honor." (T)

Summary of Quantified Statements

- When $\forall x\ P(x)$ and $\exists x\ P(x)$ are true and false?

Statement	When true?	When false?
∀x P(x)	P(x) true for all x	There is an x where P(x) is false.
∃x P(x)	There is some x for which P(x) is true.	P(x) is false for all x.

Summary of Quantified Statements

- When $\forall x\ P(x)$ and $\exists x\ P(x)$ are true and false?

Statement	When true?	When false?
∀x P(x)	P(x) true for all x	There is an x where P(x) is false.
∃x P(x)	There is some x for which P(x) is true.	P(x) is false for all x.

- Suppose that the elements in the universe can be enumerated as $x_1, x_2, \ldots, x_n$ then:
  - ◇ $\forall x\ P(x)$ is true whenever $P(x_1) \wedge P(x_2) \wedge \ldots \wedge P(x_n)$ is true
  - ◇ $\exists x\ P(x)$ is true whenever $P(x_1) \vee P(x_2) \vee \ldots \vee P(x_n)$ is true.

Properties of Quantifiers

- The truth values of $\exists x\ P(x)$ and $\forall x\ P(x)$ depend on both the propositional function $P(x)$ and the universe.

Properties of Quantifiers

- The truth values of $\exists x\ P(x)$ and $\forall x\ P(x)$ depend on both the propositional function $P(x)$ and the universe.

Example: $P(x)$ – "$x < 2$"

◇ universe: the positive integers

$\exists x\ P(x)$ – T, $\forall x\ P(x)$ – F

◇ universe: the negative integers

$\exists x\ P(x)$ – T, $\forall x\ P(x)$ – T

◇ universe: $\{\ 3,\ 4,\ 5\ \}$

$\exists x\ P(x)$ – F, $\forall x\ P(x)$ – F

Precedence of Quantifiers

- The quantifiers $\forall$ and $\exists$ have *higher precedence* than all the logical operators.

  ◇ $\forall x\ P(x) \vee Q(x)$ means $(\forall x\ P(x)) \vee Q(x)$ rather than $\forall x\ (P(x) \vee Q(x))$

Translation with Quantifiers

- Sentence: All SUSTech students are smart.
  - ◇ universe: SUSTech students

    translation: $\forall x \; Smart(x)$

Translation with Quantifiers

- Sentence: All SUSTech students are smart.
  - ◇ universe: SUSTech students
    translation: $\forall x\ Smart(x)$
  - ◇ universe: all students
    translation: $\forall x\ (At(x, SUSTech) \rightarrow Smart(x))$

Translation with Quantifiers

- Sentence: All SUSTech students are smart.

  ◇ universe: SUSTech students

  translation: $\forall x \; Smart(x)$

  ◇ universe: all students

  translation: $\forall x \; (At(x, SUSTech) \rightarrow Smart(x))$

  $\mathcal{Q}$: What about this?

  $\forall x \; (At(x, SUSTech) \land Smart(x))$

Translation with Quantifiers

- Sentence: All SUSTech students are smart.

  ◇ universe: SUSTech students

  translation: $\forall x\ Smart(x)$

  ◇ universe: all students

  translation: $\forall x\ (At(x, SUSTech) \rightarrow Smart(x))$

$\mathcal{Q}$: What about this?

$\forall x\ (At(x, SUSTech) \wedge Smart(x))$

This means every student is at SUSTech and is smart!

Translation with Quantifiers

- Sentence: All SUSTech students are smart.
  - ◇ universe: SUSTech students

    translation: $\forall x\ Smart(x)$
  - ◇ universe: all students

    translation: $\forall x\ (At(x, SUSTech) \rightarrow Smart(x))$

  $\mathcal{Q}$: What about this?

    $\forall x\ (At(x, SUSTech) \wedge Smart(x))$

    This means every student is at SUSTech and is smart!
  - ◇ universe: people

    translation: $\forall x\ (Student(x) \wedge At(x, SUSTech) \rightarrow Smart(x))$

Translation with Quantifiers

- Sentence: Someone at SUSTech is smart.
  - ◇ universe: all SUSTech affiliates

    translation: $\exists x\ Smart(x)$

Translation with Quantifiers

- Sentence: Someone at SUSTech is smart.
  - universe: all SUSTech affiliates

    translation: $\exists x \ Smart(x)$
  - universe: people

    translation: $\exists x \ (At(x, SUSTech) \land Smart(x))$

Translation with Quantifiers

- Sentence: Someone at SUSTech is smart.

  ◇ universe: all SUSTech affiliates

  translation: $\exists x \; Smart(x)$

  ◇ universe: people

  translation: $\exists x \; (At(x, SUSTech) \wedge Smart(x))$

$\mathcal{Q}$: What about this?
$\exists x \; (At(x, SUSTech) \rightarrow Smart(x))$

Translation with Quantifiers

- Sentence: Someone at SUSTech is smart.

  - ◇ universe: all SUSTech affiliates

    translation: $\exists x\ Smart(x)$

  - ◇ universe: people

    translation: $\exists x\ (At(x, SUSTech) \wedge Smart(x))$

$\mathcal{Q}$: What about this?

$\exists x\ (At(x, SUSTech) \rightarrow Smart(x))$

This is even true if there is anyone who is not at SUSTech!

Negation of Quantifiers

- Sentence: Nothing is perfect.
  - translation: $\neg \exists x \; Perfect(x)$

Negation of Quantifiers

- Sentence: Nothing is perfect.
  - ◇ translation: $\neg \exists x \; \mathit{Perfect}(x)$
  - ◇ translation: $\forall x \; \neg \mathit{Perfect}(x)$
    (Everything is imperfect.)

Negation of Quantifiers

- Sentence: Nothing is perfect.
  - ◇ translation: $\neg\exists x\ Perfect(x)$
  - ◇ translation: $\forall x\ \neg Perfect(x)$
    (Everything is imperfect.)

Conclusion: $\neg\exists x\ P(x)$ is equivalent to $\forall x\ \neg P(x)$

Negation of Quantifiers

- Sentence: Not all horses are white.
  - translation: $\neg\forall x\ (Horse(x) \rightarrow White(x))$

Negation of Quantifiers

- Sentence: Not all horses are white.
  - translation: $\neg\forall x\ (Horse(x) \to White(x))$
  - translation: $\exists x\ (Horse(x) \land \neg White(x))$
    (There is a horse that is not white.)

Negation of Quantifiers

- Sentence: Not all horses are white.
  - translation: $\neg\forall x\ (Horse(x) \rightarrow White(x))$
  - translation: $\exists x\ (Horse(x) \wedge \neg White(x))$
    (There is a horse that is not white.)
  - logically equivalent to
    $\exists x\ \neg(Horse(x) \rightarrow White(x))$

Negation of Quantifiers

- Sentence: Not all horses are white.
  - ◇ translation: $\neg \forall x\ (Horse(x) \to White(x))$
  - ◇ translation: $\exists x\ (Horse(x) \wedge \neg White(x))$
    (There is a horse that is not white.)
  - ◇ logically equivalent to
    $\exists x\ \neg(Horse(x) \to White(x))$

Conclusion: $\neg \forall x\ P(x)$ is equivalent to $\exists x\ \neg P(x)$

Negation of Quantified Statements

- a.k.a. De Morgan laws for quantifiers

Negation	Equivalent Statement	When Is Negation True?	When False?
$\neg \exists x\ P(x)$	$\forall x\ \neg P(x)$	For every x, $P(x)$ is false.	There is an x for which $P(x)$ is true.
$\neg \forall x\ P(x)$	$\exists x\ \neg P(x)$	There is an x for which $P(x)$ is false.	$P(x)$ is true for every x.

Next Lecture

- Predicate logic, proofs, sets ...