COMP170
Discrete Mathematical Tools for Computer Science

Intro to Logic

Version 2.0: Last updated, May 13, 2007

Discrete Math for Computer Science
K. Bogart, C. Stein and R.L. Drysdale
Section 3.1, pp. 91-101

3.1 Equivalence and Implication

- Equivalence of Statements
- Truth Tables
- DeMorgan's Laws
- Implication
- If and Only If

Equivalence of Statements

Equivalence of Statements

```
(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
```

Equivalence of Statements

```
(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
```

Consider the two pieces of code on the left. They are taken from two different versions of *Mergesort*. Do they do the same thing?

```
(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
```

&& = "and"

|| = "or"

Equivalence of Statements

```
(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
```

Consider the two pieces of code on the left. They are taken from two different versions of *Mergesort*. Do they do the same thing?

```
(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
```

$\&\& \;=\;$ "and"

$|| \;=\;$ "or"

Code is same except for line 1

(1) ((i+j $\leq$ p+q) && (i $\leq$ p) &&
((j > q) || (List1[i] $\leq$ List2[j])))

$\&\& \quad = \quad$ "and"

$|| \quad = \quad$ "or"

(1') (((i+j $\leq$ p+q) && (i $\leq$ p) && (j > q))
|| ((i+j $\leq$ p+q) && (i $\leq$ p)
&& (List1[i] $\leq$ List2[j])))

```
(1) ((i+j ≤ p+q) && (i ≤ p) &&
    ((j > q) || (List1[i] ≤ List2[j])))
```

$\&\& \quad = \quad$ “and”

$|| \quad = \quad$ “or”

```
(1') (((i+j ≤ p+q) && (i ≤ p) && (j > q))
    || ((i+j ≤ p+q) && (i ≤ p)
        && (List1[i] ≤ List2[j])))
```

Are they equivalent?

$\&\& \; = \;$ "and"

$|| \; = \;$ "or"

```
(1) ((i+j ≤ p+q) && (i ≤ p) &&
    ((j > q) || (List1[i] ≤ List2[j])))
```

```
(1') (((i+j ≤ p+q) && (i ≤ p) && (j > q))
    || ((i+j ≤ p+q) && (i ≤ p)
        && (List1[i] ≤ List2[j])))
```

Are they equivalent? Let's rewrite using

$s \sim$ (i+j $\leq$ p+q) $t \sim$ (i$\leq$ p) $u \sim$ (j$>$ q)

$v \sim$ (List[i] $\leq$ List2[j])

```
(1) ((i+j ≤ p+q) && (i ≤ p) &&
    ((j > q) || (List1[i] ≤ List2[j])))
```

$\&\& \quad = \quad$ "and"

$|| \quad = \quad$ "or"

```
(1') (((i+j ≤ p+q) && (i ≤ p) && (j > q))
    || ((i+j ≤ p+q) && (i ≤ p)
        && (List1[i] ≤ List2[j])))
```

Are they equivalent? Let's rewrite using

$s \sim$ `(i+j ≤ p+q)` $t \sim$ `(i≤ p)` $u \sim$ `(j> q)`

$v \sim$ `(List[i] ≤ List2[j])`

(1) s and t and (u or v) (1') (s and t and u) or (s and t and v)

```
(1)  ((i+j ≤ p+q) && (i ≤ p) &&
     ((j > q) || (List1[i] ≤ List2[j])))
```

$\&\& \quad = \quad$ "and"

$|| \quad = \quad$ "or"

```
(1') (((i+j ≤ p+q) && (i ≤ p) && (j > q))
     || ((i+j ≤ p+q) && (i ≤ p)
         && (List1[i] ≤ List2[j])))
```

Are they equivalent? Let's rewrite using

$s \sim$ (i+j $\leq$ p+q) $\quad t \sim$ (i$\leq$ p) $\quad u \sim$ (j$>$ q)

$v \sim$ (List[i] $\leq$ List2[j])

(1) s and t and (u or v) (1') (s and t and u) or (s and t and v)

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

(1) `((i+j` $\leq$ `p+q) && (i` $\leq$ `p) &&`
`((j > q) || (List1[i]` $\leq$ `List2[j])))`

$\&\& \quad = \quad$ "and"

$|| \quad = \quad$ "or"

(1') `(((i+j` $\leq$ `p+q) && (i` $\leq$ `p) && (j > q))`
`|| ((i+j` $\leq$ `p+q) && (i` $\leq$ `p)`
`&& (List1[i]` $\leq$ `List2[j])))`

Are they equivalent? Let's rewrite using

$s \sim$ (i+j $\leq$ p+q) $\qquad t \sim$ (i$\leq$ p) $\qquad u \sim$ (j$>$ q)

$v \sim$ (List[i] $\leq$ List2[j])

(1) s and t and (u or v) $\qquad$ (1') (s and t and u) or (s and t and v)

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

(1) w and (u or v) $\qquad$ (1') (w and u) or (w and v)

(1) ((i+j $\le$ p+q) && (i $\le$ p) &&
    ((j > q) || (List1[i] $\le$ List2[j])))

$\&\& \quad = \quad$ "and"

$|| \quad = \quad$ "or"

(1') (((i+j $\le$ p+q) && (i $\le$ p) && (j > q))
    || ((i+j $\le$ p+q) && (i $\le$ p)
        && (List1[i] $\le$ List2[j])))

Are they equivalent? Let's rewrite using

$s \sim$ (i+j $\le$ p+q) $\quad t \sim$ (i$\le$ p) $\quad u \sim$ (j$>$ q)

$v \sim$ (List[i] $\le$ List2[j])

(1) s and t and (u or v) (1') (s and t and u) or (s and t and v)

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

(1) w and (u or v) $\xleftrightarrow{\text{equal?}}$ (1') (w and u) or (w and v)

We just transformed our code into symbolic compound statements and now want to develop a theory of how to determine whether two such statements are equal (equivalent)

We just transformed our code into symbolic compound statements and now want to develop a theory of how to determine whether two such statements are equal (equivalent)

Notation for symbolic compound statements:

We just transformed our code into symbolic compound statements and now want to develop a theory of how to determine whether two such statements are equal (equivalent)

Notation for symbolic compound statements:

- Symbols (s, t, etc.), called **variables**, standing for statements

We just transformed our code into symbolic compound statements and now want to develop a theory of how to determine whether two such statements are equal (equivalent)

Notation for symbolic compound statements:

- Symbols (s, t, etc.), called **variables**, standing for statements
- The symbol $\wedge$, denoting and

We just transformed our code into symbolic compound statements and now want to develop a theory of how to determine whether two such statements are equal (equivalent)

Notation for symbolic compound statements:

- Symbols (s, t, etc.), called **variables**, standing for statements
- The symbol $\wedge$, denoting and
- The symbol $\vee$, denoting or

We just transformed our code into symbolic compound statements and now want to develop a theory of how to determine whether two such statements are equal (equivalent)

Notation for symbolic compound statements:

- Symbols (s, t, etc.), called **variables**, standing for statements
- The symbol $\wedge$, denoting and
- The symbol $\vee$, denoting or
- The symbol $\oplus$ denoting exclusive or

We just transformed our code into symbolic compound statements and now want to develop a theory of how to determine whether two such statements are equal (equivalent)

Notation for symbolic compound statements:

- Symbols (s, t, etc.), called **variables**, standing for statements
- The symbol $\wedge$, denoting and
- The symbol $\vee$, denoting or
- The symbol $\oplus$ denoting exclusive or
- The symbol $\neg$, denoting not

We just transformed our code into symbolic compound statements and now want to develop a theory of how to determine whether two such statements are equal (equivalent)

Notation for symbolic compound statements:

- Symbols (s, t, etc.), called **variables**, standing for statements
- The symbol $\wedge$, denoting and
- The symbol $\vee$, denoting or
- The symbol $\oplus$ denoting exclusive or
- The symbol $\neg$, denoting not

logical connectives

We just transformed our code into symbolic compound statements and now want to develop a theory of how to determine whether two such statements are equal (equivalent)

Notation for symbolic compound statements:

- Symbols (s, t, etc.), called **variables**, standing for statements
- The symbol $\wedge$, denoting and
- The symbol $\vee$, denoting or
- The symbol $\oplus$ denoting exclusive or
- The symbol $\neg$, denoting not

logical connectives

- Left and right parentheses (,)

Bigger compound statements are built out of smaller ones

Bigger compound statements are built out of smaller ones

$\boxed{(1)\ w \text{ and } (u \text{ or } v)}$

$\boxed{(1')\ (w \text{ and } u) \text{ or } (w \text{ and } v)}$

Bigger compound statements are built out of smaller ones

(1) w and $(u$ or $v)$

$\downarrow$

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

(1') $(w$ and $u)$ or $(w$ and $v)$

$\downarrow$

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

Bigger compound statements are built out of smaller ones

(1) w and $(u$ or $v)$

$\downarrow$

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

(1') $(w$ and $u)$ or $(w$ and $v)$

$\downarrow$

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

Or something as complicated as

$$(s \oplus t) \wedge (\neg u \vee (s \wedge t)) \wedge \neg(s \oplus (t \vee u))$$

Bigger compound statements are built out of smaller ones

$$\begin{array}{ccc}
\boxed{\text{(1) } w \text{ and } (u \text{ or } v)} & & \boxed{\text{(1') } (w \text{ and } u) \text{ or } (w \text{ and } v)} \\
\downarrow & & \downarrow \\
\boxed{\text{(1) } w \wedge (u \vee v)} & & \boxed{\text{(1') } (w \wedge u) \vee (w \wedge v)}
\end{array}$$

Or something as complicated as

$$(s \oplus t) \wedge (\neg u \vee (s \wedge t)) \wedge \neg(s \oplus (t \vee u))$$

We will always use parentheses to make our statements unambiguous. The one exception will be $\neg$, which we will often write without parentheses.

$\neg$ is always combined with the statement immediately to its right

e.g., $\neg u \vee (s \wedge t)$ is $(\neg u) \vee (s \wedge t)$ and not $\neg(u \vee (s \wedge t))$.

This is same rule used for negative numbers in algebraic expressions.

Variables s, t can be either True (T) or False (F):

Variables s, t can be either True (T) or False (F):

- $s \wedge t$ is True iff *both* s and t are True

Variables s, t can be either True (T) or False (F):

- $s \wedge t$ is True iff *both* s and t are True
- $s \vee t$ is True iff *at least one of* s and t are True

Variables s, t can be either True (T) or False (F):

- $s \wedge t$ is True iff *both* s and t are True
- $s \vee t$ is True iff *at least one of* s and t are True
- $s \oplus t$ is True iff *exactly one of* s and t are True

Variables s, t can be either True (T) or False (F):

- $s \wedge t$ is True iff *both* s and t are True
- $s \vee t$ is True iff *at least one of* s and t are True
- $s \oplus t$ is True iff *exactly one of* s and t are True
- $\neg s$ is True iff s is False

Variables s, t can be either True (T) or False (F):

- $s \wedge t$ is True iff *both* s and t are True
- $s \vee t$ is True iff *at least one of* s and t are True
- $s \oplus t$ is True iff *exactly one of* s and t are True
- $\neg s$ is True iff s is False

How can we calculate whether a statement such as

$$\boxed{(1)\ w \wedge (u \vee v)}$$

is True or False or, even more, whether it is equivalent to another statement such as $\boxed{(1')\ (w \wedge u) \vee (w \wedge v)}$

3.1 Equivalence and Implication

- Equivalence of Statements
- Truth Tables
- DeMorgan's Laws
- Implication
- If and Only If

Truth tables

Truth tables

Gives us a way of deciding when a compound statement is true, based on the truth or falsity of its component statements.

Truth tables

Gives us a way of deciding when a compound statement is true, based on the truth or falsity of its component statements.

We can also use truth tables to determine whether two statements are equivalent.

Truth tables

Gives us a way of deciding when a compound statement is true, based on the truth or falsity of its component statements.

We can also use truth tables to determine whether two statements are equivalent.

- A Truth table works by first listing all of the possible combinations of values of the truth values T/F of the variables used by the compound statement

Truth tables

Gives us a way of deciding when a compound statement is true, based on the truth or falsity of its component statements.

We can also use truth tables to determine whether two statements are equivalent.

- A Truth table works by first listing all of the possible combinations of values of the truth values T/F of the variables used by the compound statement
- It then evaluates the truth values of the smaller compound statements, building up to evaluating the truth values of the *topmost* compound statement

- $s \wedge t$ is True iff *both* s and t are True

AND

s	t	$s \wedge t$
T	T	T
T	F	F
F	T	F
F	F	F

- $s \wedge t$ is True iff *both* s and t are True

AND

s	t	$s \wedge t$
T	T	T
T	F	F
F	T	F
F	F	F

- $s \wedge t$ is True iff *both* s and t are True

- $s \vee t$ is True iff *at least one of* s and t are True

AND

s	t	$s \wedge t$
T	T	T
T	F	F
F	T	F
F	F	F

OR

s	t	$s \vee t$
T	T	T
T	F	T
F	T	T
F	F	F

- $s \wedge t$ is True iff *both* s and t are True
- $s \vee t$ is True iff *at least one of* s and t are True

AND

s	t	$s \wedge t$
T	T	T
T	F	F
F	T	F
F	F	F

OR

s	t	$s \vee t$
T	T	T
T	F	T
F	T	T
F	F	F

- $s \wedge t$ is True iff *both* s and t are True
- $s \vee t$ is True iff *at least one of* s and t are True
- $s \oplus t$ is True iff *exactly one of* s and t are True

AND

s	t	$s \wedge t$
T	T	T
T	F	F
F	T	F
F	F	F

OR

s	t	$s \vee t$
T	T	T
T	F	T
F	T	T
F	F	F

XOR

s	t	$s \oplus t$
T	T	F
T	F	T
F	T	T
F	F	F

- $s \wedge t$ is True iff *both* s and t are True

- $s \vee t$ is True iff *at least one of* s and t are True

- $s \oplus t$ is True iff *exactly one of* s and t are True

AND

s	t	$s \wedge t$
T	T	T
T	F	F
F	T	F
F	F	F

OR

s	t	$s \vee t$
T	T	T
T	F	T
F	T	T
F	F	F

XOR

s	t	$s \oplus t$
T	T	F
T	F	T
F	T	T
F	F	F

- $s \wedge t$ is True iff *both* s and t are True
- $s \vee t$ is True iff *at least one of* s and t are True
- $s \oplus t$ is True iff *exactly one of* s and t are True
- $\neg s$ is True iff s is False

AND		
s	t	$s \wedge t$
T	T	T
T	F	F
F	T	F
F	F	F

OR		
s	t	$s \vee t$
T	T	T
T	F	T
F	T	T
F	F	F

XOR		
s	t	$s \oplus t$
T	T	F
T	F	T
F	T	T
F	F	F

NOT	
s	$\neg s$
T	F
F	T

- $s \wedge t$ is True iff *both* s and t are True
- $s \vee t$ is True iff *at least one of* s and t are True
- $s \oplus t$ is True iff *exactly one of* s and t are True
- $\neg s$ is True iff s is False

Truth tables for our original programs

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

w	u	v
T	T	T
T	T	F
T	F	T
T	F	F
F	T	T
F	T	F
F	F	T
F	F	F

Truth tables for our original programs

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

w	u	v	$u \vee v$
T	T	T	T
T	T	F	T
T	F	T	T
T	F	F	F
F	T	T	T
F	T	F	T
F	F	T	T
F	F	F	F

Truth tables for our original programs

(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

Truth tables for our original programs

(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
T	T	T
T	T	F
T	F	T
T	F	F
F	T	T
F	T	F
F	F	T
F	F	F

Truth tables for our original programs

(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	$w \wedge u$
T	T	T	T
T	T	F	T
T	F	T	F
T	F	F	F
F	T	T	F
F	T	F	F
F	F	T	F
F	F	F	F

Truth tables for our original programs

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

Truth tables for our original programs

(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

Truth tables for our original programs

(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

The Same!

We will say that two statements are *equivalent* if they have the same truth value for all possible truth settings of their underlying variables.

We will say that two statements are *equivalent* if they have the same truth value for all possible truth settings of their underlying variables.

Examples:

a) $w \wedge (u \vee v)$ and $(w \wedge u) \vee (w \wedge v)$ are equivalent.

We showed this on the previous page using truth tables

We will say that two statements are *equivalent* if they have the same truth value for all possible truth settings of their underlying variables.

Examples:

a) $w \wedge (u \vee v)$ and $(w \wedge u) \vee (w \wedge v)$ are equivalent.

We showed this on the previous page using truth tables

b) $(w \wedge v) \vee u$ and $(w \vee v) \wedge u$ are not equivalent

Set $w = T$, $v = T$, $u = F$.

The left statement is True and the right one is False

Lemma 3.1: "Distributive Law"

The statements

$$w \wedge (u \vee v) \quad \text{and} \quad (w \wedge u) \vee (w \wedge v)$$

are equivalent.

Lemma 3.1: "Distributive Law"
The statements

$$w \wedge (u \vee v) \quad \text{and} \quad (w \wedge u) \vee (w \wedge v)$$

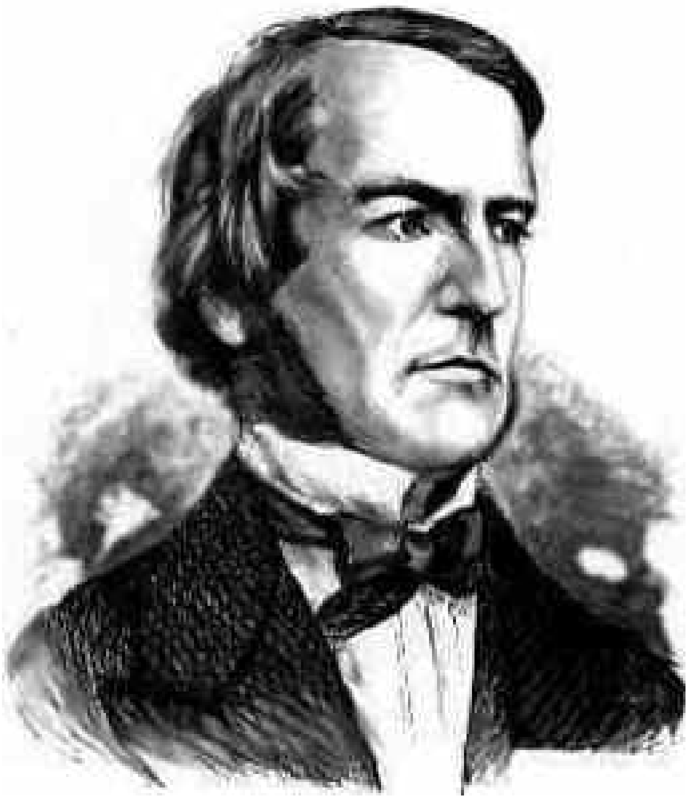
are equivalent.

Lemma 3.X1 "Associative Laws"
$(w \wedge u) \wedge v$ is equivalent to $w \wedge (u \wedge v)$
and
$(w \vee u) \vee v$ is equivalent to $w \vee (u \vee v)$

George Boole

English 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.

George Boole

English 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.

See *http://en.wikipedia.org/wiki/George_Boole* for more details

3.1 Equivalence and Implication

- Equivalence of Statements
- Truth Tables
- DeMorgan's Laws
- Implication
- If and Only If

DeMorgan's Laws

DeMorgan's Laws

DeMorgan's Laws say that

(i) $\neg(p \vee q)$ is equivalent to $\neg p \wedge \neg q$,

and that (ii) $\neg(p \wedge q)$ is equivalent to $\neg p \vee \neg q$.

DeMorgan's Laws

DeMorgan's Laws say that

(i) $\neg(p \vee q)$ is equivalent to $\neg p \wedge \neg q$,

and that (ii) $\neg(p \wedge q)$ is equivalent to $\neg p \vee \neg q$.

We will use truth tables to prove (i)
(and leave (ii) for the homework)

DeMorgan's Laws

DeMorgan's Laws say that

(i) $\neg(p \vee q)$ is equivalent to $\neg p \wedge \neg q$,

and that (ii) $\neg(p \wedge q)$ is equivalent to $\neg p \vee \neg q$.

We will use truth tables to prove (i)
(and leave (ii) for the homework)

p	q
T	T
T	F
F	T
F	F

DeMorgan's Laws

DeMorgan's Laws say that

(i) $\neg(p \vee q)$ is equivalent to $\neg p \wedge \neg q$,

and that (ii) $\neg(p \wedge q)$ is equivalent to $\neg p \vee \neg q$.

We will use truth tables to prove (i)
(and leave (ii) for the homework)

p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

DeMorgan's Laws

DeMorgan's Laws say that

(i) $\neg(p \vee q)$ is equivalent to $\neg p \wedge \neg q$,

and that (ii) $\neg(p \wedge q)$ is equivalent to $\neg p \vee \neg q$.

We will use truth tables to prove (i)
(and leave (ii) for the homework)

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

DeMorgan's Laws

DeMorgan's Laws say that

(i) $\neg(p \vee q)$ is equivalent to $\neg p \wedge \neg q$,

and that (ii) $\neg(p \wedge q)$ is equivalent to $\neg p \vee \neg q$.

We will use truth tables to prove (i)
(and leave (ii) for the homework)

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

DeMorgan's Laws

DeMorgan's Laws say that

(i) $\boxed{\neg(p \vee q)}$ is equivalent to $\boxed{\neg p \wedge \neg q}$,

and that (ii) $\neg(p \wedge q)$ is equivalent to $\neg p \vee \neg q$.

We will use truth tables to prove (i)

(and leave (ii) for the homework)

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

Use truth tables to show that

$p \oplus q$ (the exclusive or of p and q)

is equivalent to

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

Use truth tables to show that

$p \oplus q$ (the exclusive or of p and q)

is equivalent to

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

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

Use truth tables to show that

$p \oplus q$ (the exclusive or of p and q)

is equivalent to

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

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

We just saw that

$$p \oplus q = (p \vee q) \wedge \neg(p \wedge q)$$

We just saw that

$$p \oplus q = (p \vee q) \wedge \neg(p \wedge q)$$

Since $\neg(\neg(p \vee q)) = p \vee q$ this gives

We just saw that

$$p \oplus q = (p \vee q) \wedge \neg(p \wedge q)$$

Since $\neg(\neg(p \vee q)) = p \vee q$ this gives

$$p \oplus q = \neg(\neg(p \vee q)) \wedge \neg(p \wedge q)$$

We just saw that

$$p \oplus q = (p \vee q) \wedge \neg(p \wedge q)$$

Since $\neg(\neg(p \vee q)) = p \vee q$ this gives

$$p \oplus q = \neg(\neg(p \vee q)) \wedge \neg(p \wedge q)$$

We now apply DeMorgan's law (i) to the RHS to get

We just saw that

$$p \oplus q = (p \vee q) \wedge \neg(p \wedge q)$$

Since $\neg(\neg(p \vee q)) = p \vee q$ this gives

$$p \oplus q = \neg(\neg(p \vee q)) \wedge \neg(p \wedge q)$$

We now apply DeMorgan's law (i) to the RHS to get

$$p \oplus q = \neg(\neg(p \vee q) \vee (p \wedge q))$$

3.1 Equivalence and Implication

- Equivalence of Statements
- Truth Tables
- DeMorgan's Laws
- Implication
- If and Only If

Implication: "$\Longrightarrow$"

Implication: "$\Rightarrow$"

Recall Fermat's Little Theorem (Theorem 2.21):

If p is a prime, then $a^{p-1} \bmod p = 1$ for each nonzero $a \in Z_p$.

Implication: "$\Rightarrow$"

Recall Fermat's Little Theorem (Theorem 2.21):
If p is a prime, then $a^{p-1} \bmod p = 1$ for each nonzero $a \in Z_p$.

It combines two different statements:

- $s \sim$ (p is a prime), and
- $t \sim$ ($a^{p-1} \bmod p = 1$ for each nonzero $a \in Z_p$).

Implication: "$\Rightarrow$"

Recall Fermat's Little Theorem (Theorem 2.21):
If p is a prime, then a^{p-1} mod $p = 1$ for each nonzero $a \in Z_p$.

It combines two different statements:

- $s \sim$ (p is a prime), and
- $t \sim$ (a^{p-1} mod $p = 1$ for each nonzero $a \in Z_p$).

We use $p \Rightarrow q$ to denote the *implication*
If p then q

Implication: "$\Rightarrow$"

Recall Fermat's Little Theorem (Theorem 2.21):
If p is a prime, then $a^{p-1} \bmod p = 1$ for each nonzero $a \in Z_p$.

It combines two different statements:

- $s \sim$ (p is a prime), and
- $t \sim$ ($a^{p-1} \bmod p = 1$ for each nonzero $a \in Z_p$).

We use $p \Rightarrow q$ to denote the *implication*
If p then q

Fermat's Little Theorem then becomes

$$s \Rightarrow t$$

In $s \Rightarrow t$, statement s is the **hypothesis** of the implication statement t is the **conclusion** of the implication.

In $s \Rightarrow t$, statement s is the **hypothesis** of the implication statement t is the **conclusion** of the implication.

Note that English is not a very precise language.
In English, the following four phrases all usually mean the same thing. In other words, they are all defined by the same truth table:

- s implies t.
- t if s.
- if s then t.
- s only if t.

3.1 Equivalence and Implication

- Equivalence of Statements
- Truth Tables
- DeMorgan's Laws
- Implication
- If and Only If

If and Only If "$\Leftrightarrow$"

If and Only If "$\Leftrightarrow$"

s if and only if t.

If and Only If "$\Leftrightarrow$"

s if and only if t.

We parse this as

s if t and s only if t.

If and Only If "$\Leftrightarrow$"

s if and only if t.

We parse this as

s if t and s only if t.

In compound statement notation this is the same as

$t \Rightarrow s$ and $s \Rightarrow t$.

If and Only If "$\Leftrightarrow$"

s if and only if t.

We parse this as

s if t and s only if t.

In compound statement notation this is the same as

$t \Rightarrow s$ and $s \Rightarrow t$.

We denote the statement "s if and only if t" by

$s \Leftrightarrow t$.

If and Only If "$\Leftrightarrow$"

$$s \text{ if and only if } t.$$

We parse this as

$$s \text{ if } t \quad \text{and} \quad s \text{ only if } t.$$

In compound statement notation this is the same as

$$t \Rightarrow s \quad \text{and} \quad s \Rightarrow t.$$

We denote the statement "s if and only if t" by

$$s \Leftrightarrow t.$$

Statements of the form $s \Rightarrow t$ and $s \Leftrightarrow t$ are called **conditional statements**; the connectives $\Rightarrow$ and $\Leftrightarrow$ are called **conditional connectives**.

“Conditional” Truth Tables

IMPLIES

s	t	$s \Rightarrow t$
T	T	T
T	F	F
F	T	T
F	F	T

IF AND ONLY IF

s	t	$s \Leftrightarrow t$
T	T	T
T	F	F
F	T	F
F	F	T

$s \Rightarrow t$

sometimes confusing due to ambiguity in English.

$s \Rightarrow t$

sometimes confusing due to ambiguity in English.

Suppose a classmate holds an ordinary playing card (with its back to you) and says,

> "If this card is a heart, then it is a queen."

$s \Rightarrow t$

sometimes confusing due to ambiguity in English.

Suppose a classmate holds an ordinary playing card (with its back to you) and says,

“If this card is a heart, then it is a queen.”

When is your classmate telling the truth?

$s \Rightarrow t$

sometimes confusing due to ambiguity in English.

Suppose a classmate holds an ordinary playing card (with its back to you) and says,

"If this card is a heart, then it is a queen."

When is your classmate telling the truth?

- The card is a heart and a queen.
- The card is a heart and a king.
- The card is a diamond and a queen.
- The card is a diamond and a king.

“If this card is a heart, then it is a queen.”

When is your classmate telling the truth?

- The card is a heart and a queen.
- The card is a heart and a king.
- The card is a diamond and a queen.
- The card is a diamond and a king.

"If this card is a heart, then it is a queen."

When is your classmate telling the truth?

	Truth	Lie
• The card is a heart and a queen.	$\surd$	
• The card is a heart and a king.		$\surd$
• The card is a diamond and a queen.	?	No
• The card is a diamond and a king.	?	No

"If this card is a heart, then it is a queen."

When is your classmate telling the truth?

	Truth	Lie
• The card is a heart and a queen.	$\surd$	
• The card is a heart and a king.		$\surd$
• The card is a diamond and a queen.	?	No
• The card is a diamond and a king.	?	No

The Principle of the Excluded Middle:
A statement is true exactly when it is not false.

"If this card is a heart, then it is a queen."

When is your classmate telling the truth?

	Truth	Lie
• The card is a heart and a queen.	$\surd$	
• The card is a heart and a king.		$\surd$
• The card is a diamond and a queen.	?	No
• The card is a diamond and a king.	?	No

The Principle of the Excluded Middle:
A statement is true exactly when it is not false.

So the two "?" become $\surd$

"If this card is a heart, then it is a queen."

When is your classmate telling the truth?

	Truth	Lie
• The card is a heart and a queen.	$\surd$	
• The card is a heart and a king.		$\surd$
• The card is a diamond and a queen.	~~?~~ $\surd$	No
• The card is a diamond and a king.	~~?~~ $\surd$	No

The Principle of the Excluded Middle:

A statement is true exactly when it is not false.

So the two "?" become $\surd$