

6	9	13	7
12		10	5
3	1	4	14
15	8	11	2

The Logic of Propositions

6	9	13	7
12		10	5
3	1	4	14
15	8	11	2

Propositional (Boolean) Logic

A **proposition** is either **True** or **False**

Example:

There are 6 regular solids.

False

Non-examples:

Wake up!

Where am I?

6	9	13	7
12		10	5
3	1	4	14
15	8	11	2

English to Math

Greeks carry Swords or Javelins

$$G \rightarrow (S \vee J)$$

True even if a Greek carries both a Sword and a Javelin

6	9	13	7
12		10	5
3	1	4	14
15	8	11	2

English to Math

Greeks carry Bronze or Copper swords

$$G \rightarrow (B \oplus C)$$

Bronze or Copper but **not both**

6	9	13	7
12		10	5
3	1	4	14
15	8	11	2

Definition of OR

The value of $(P \text{ OR } Q)$ is **T** iff P is **T**, or Q is **T**, or **both** are **T**.

Truth Table for **OR**

P	Q	P OR Q
T	T	T
T	F	T
F	T	T
F	F	F

F iff both P, Q are **F**

6	9	13	7
12		10	5
3	1	4	14
15	8	11	2

Definition of XOR

The value of $(P \text{ XOR } Q)$ is **T** iff **exactly one** of P and Q is **T**.

Truth Table for **XOR**

P	Q	P XOR Q
T	T	F
T	F	T
F	T	T
F	F	F

6	9	13	7
12		10	5
3	1	4	14
15	8	11	2

Definition of AND

The value of $(P \text{ AND } Q)$ is **T** iff both P and Q are **T**.

Truth Table for **AND**

P	Q	P AND Q
T	T	T
T	F	F
F	T	F
F	F	F



Albert R. Meyer

February 10, 2010

lec 2W.7

6	9	13	7
12		10	5
3	1	4	14
15	8	11	2

Definition of NOT

The $\text{NOT}(P)$ is **T** iff P is **F**.

Truth Table for **NOT (P)**

P	NOT(P)
T	F
F	T



Albert R. Meyer

February 10, 2010

lec 2W.8

6	9	13	7
12		10	5
3	1	4	14
15	8	11	2

Truth Assignments

A **truth assignment** assigns a value **T** or **F** to each propositional variable. Computer scientists call assignment of values to variables an **environment**. If we **know the environment**, we can **find the value** of a propositional formula.



Albert R. Meyer

February 10, 2010

lec 2W.9

6	9	13	7
12		10	5
3	1	4	14
15	8	11	2

Evaluation in an Environment

Example: Suppose environment, v , assigns

$$v(P) = T, v(Q) = T, v(R) = F.$$

Truth value of

$$(\neg(P \text{ AND } Q)) \text{ OR } (R \text{ XOR } (\neg Q))$$

F T T T F F F F T



Albert R. Meyer

February 10, 2010

lec 2W.10

6	9	13	7
12		10	5
3	1	4	14
15	8	11	2

Equivalence

Two propositional formulas are **equivalent** iff they have the **same truth value** in all environments.



Albert R. Meyer

February 10, 2010

lec 2W.11

6	9	13	7
12		10	5
3	1	4	14
15	8	11	2

DeMorgan's Law

$P \vee Q$ is equivalent to $\overline{P \wedge Q}$

P	Q	$\neg(P \vee Q)$
T	T	F
T	F	F
F	T	F
F	F	T

\overline{P}	\wedge	\overline{Q}
F	F	F
F	F	T
T	F	F
T	T	T



Albert R. Meyer

February 10, 2010

lec 2W.13

6	9	13	7
12		10	5
3	1	4	14
15	8	11	2

DeMorgan's Law

$\overline{P \vee Q}$ is equivalent to $\overline{P} \wedge \overline{Q}$

P	Q	$\overline{(P \vee Q)}$	
T	T	F	T
T	F	F	T
F	T	F	T
F	F	T	F

\overline{P}	\wedge	\overline{Q}
F	F	F
F	F	T
T	F	F
T	T	T

Same final column, so equivalent
-- proof by Truth Table



Albert R. Meyer

February 10, 2010

lec 2W.14

6	9	13	7
12		10	5
3	1	4	14
15	8	11	2

Definition of IMPLIES

The value of (P IMPLIES Q) is F iff P is T and Q is F.

Truth Table for IMPLIES (\rightarrow)

P	Q	$P \rightarrow Q$
T	T	T
T	F	F
F	T	T
F	F	T



Albert R. Meyer

February 10, 2010

lec 2W.15

6	9	13	7
12		10	5
3	1	4	14
15	8	11	2

A True Implication

(1=-1) IMPLIES (I am Pope)
We reasoned *correctly* to reach the false conclusion



Albert R. Meyer

February 10, 2010

lec 2W.16

6	9	13	7
12		10	5
3	1	4	14
15	8	11	2

A True Implication

(1=-1) IMPLIES (I am Pope)
We reasoned *correctly* to reach the false *conclusion*



Albert R. Meyer

February 10, 2010

lec 2W.17

6	9	13	7
12		10	5
3	1	4	14
15	8	11	2

A True Implication

(1=-1) IMPLIES (I am Pope)
We reasoned *correctly* to reach the false conclusion from the false hypothesis.



Albert R. Meyer

February 10, 2010

lec 2W.18

6	9	13	7
12		10	5
3	1	4	14
15	8	11	2

A True Implication

(1=-1) IMPLIES (I am Pope)
We reasoned *correctly* to reach the false conclusion from the false *hypothesis*.



Albert R. Meyer

February 10, 2010

lec 2W.19

6	9	13	7
12		10	5
3	1	4	14
15	8	11	2

A True Implication

(1=-1) IMPLIES (I am Pope)
 The *whole* implication is **true**,
 even though both conclusion
 & hypothesis are **false**.



Albert R. Meyer

February 10, 2010

lec 2W.20

6	9	13	7
12		10	5
3	1	4	14
15	8	11	2

Satisfiability & Validity

A formula is **satisfiable** iff it
 is **true** in **some** environment.

A formula is **valid** iff it is
true in **all** environments.



Albert R. Meyer

February 10, 2010

lec 2W.21

6	9	13	7
12		10	5
3	1	4	14
15	8	11	2

Verifying Valid, Satisfiable

Truth table size **doubles** with
 each additional variable
 --**exponential growth**. Makes
 truth tables impossible when
 there are hundreds of variables.
 (In current digital circuits,
 there are millions of variables.)



Albert R. Meyer

February 10, 2010

6	9	13	7
12		10	5
3	1	4	14
15	8	11	2

Efficient Test for Satisfiability?

The **P=NP?** question is equivalent
 to asking if there is an efficient
 (polynomial rather than
 exponential time) procedure
 to check satisfiability.



Albert R. Meyer

February 10, 2010

6	9	13	7
12		10	5
3	1	4	14
15	8	11	2

Java Logical Expression

```

    OR                AND
if ((x>0) || (x <= 0 && y>100))
    :
    (more code)
  
```



Albert R. Meyer

February 10, 2010

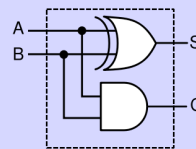
lec 2W.38

6	9	13	7
12		10	5
3	1	4	14
15	8	11	2

Digital Logic

$s ::= A \text{ XOR } B$

$c ::= A \text{ AND } B$



half adder

from [http://en.wikipedia.org/wiki/Adder_\(electronics\)](http://en.wikipedia.org/wiki/Adder_(electronics))



Albert R. Meyer

February 10, 2010

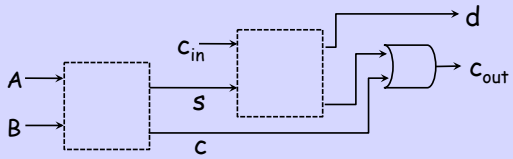
lec 2W.40

6	9	13	7
12		10	5
3	1	4	14
15	8	11	2

Digital Logic

$$d ::= c_{in} \text{ XOR } s$$

$$c_{out} ::= (c_{in} \text{ AND } s) \text{ OR } c$$



full adder



Albert R Meyer

February 10, 2010

lec 2W.41

6	9	13	7
12		10	5
3	1	4	14
15	8	11	2

Team Problems

Problems 1-4



Albert R Meyer

February 10, 2010

lec 2W.42

MIT OpenCourseWare
<http://ocw.mit.edu>

6.042J / 18.062J Mathematics for Computer Science
Spring 2010

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.