Logic

Important for
1. understanding mathematical reasoning
2. design of computer circuits
3. construction of computer programs
4. verification of correctness of programs

Def A proposition is a statement which is true or false (but not both)

Letters are used to denote propositions: $p, q, r, s, \ldots$

Def. Truth value of a proposition is either T (true) or F (false).

Def Compound propositions are formed from existing propositions using logical operators.

Def Negation of p, $\neg p$ (other texts use $\sim p$)

A truth table displays the function values of operators

$\neg : \{T,F\} \to \{T,F\}$

Truth table:

p	$\neg p$
T	F
F	T

Def Conjunction of p and q, $p \wedge q$, p "and" q

$\wedge : \{T,F\} \times \{T,F\} \to \{T,F\}$

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

Def Disjunction of p and q, $p \vee q$, p "or" q

$\vee : \{T,F\} \times \{T,F\} \to \{T,F\}$ "Inclusive or"

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

Def. The exclusive or of p and q, $p \oplus q$,

$\oplus : \{T,F\} \times \{T,F\} \to \{T,F\}$

p	q	$p \oplus q$
T	T	F
T	F	T
F	T	T
F	F	F

Def. The implication of p and q, denoted $p \to q$

$\to : \{T,F\} \times \{T,F\} \to \{T,F\}$

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

if p then q
p implies q
if p, q
p only if q
q if p
p is sufficient for q
q is necessary for p
q whenever p

Note: If ... THEN... in computer programming is different from If... then... in logic:

If p then S
if p is true, execute S
if p is false, do not execute S

Def. 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 $\neg p \to \neg q$

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

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

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

<u>Def</u> the biconditional of p and q, $p \leftrightarrow q$, p "iff" q,

$\leftrightarrow : \{T,F\} \times \{T,F\} \to \{T,F\}$

p	q	$p \leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

<u>Def</u> Bit - one or zero

Allow $F = 0$ $T = 1$

<u>Def</u> Bit operators OR, AND, XOR correspond to $\vee$, $\wedge$, $\oplus$

<u>EX</u>

01101	10110	
11000	11101	
11101	11111	bitwise OR
01000	10100	bitwise AND
10101	01011	bitwise XOR

<u>Def</u> A tautology is a compound proposition which is always true: $p \vee \neg p$

A Contradiction is a compound proposition which is always false: $p \wedge \neg p$

A contingency is a compound proposition which is neither a tautology nor a contradiction

* Compound propositions which have the same truth table are logically equivalent.

To prove $P \Leftrightarrow Q$, show P and Q have same truth table.

thm Logical Equivalences.

De Morgans Laws: $\neg(p \vee q) \Leftrightarrow \neg p \wedge \neg q$

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

$\neg p \vee q \Leftrightarrow p \rightarrow q$

Distributive Laws: $p \vee (q \wedge r) \Leftrightarrow (p \vee q) \wedge (p \vee r)$

$p \wedge (q \vee r) \Leftrightarrow (p \wedge q) \vee (p \wedge r)$

Proof $p \wedge (q \vee r) \Leftrightarrow (p \wedge q) \vee (p \wedge r)$

p	q	r	① $q \vee r$	LHS $p \wedge$ ①	② $p \wedge q$	③ $p \wedge r$	RHS ② $\vee$ ③
T	T	T	T	T	T	T	T
T	T	F	T	T	T	F	T
T	F	T	T	T	F	T	T
T	F	F	F	F	F	F	F
F	T	T	T	F	F	F	F
F	T	F	T	F	F	F	F
F	F	T	T	F	F	F	F
F	F	F	F	F	F	F	F

LHS = RHS

Note: A compound proposition with n propositions will have 2^n rows in its truth table.

Rules of Inference

Draw a conclusion or derive a new assertion from old assertions.

<u>Def</u> An argument is a set of premises followed by a conclusion:

$$\begin{array}{l} p_1 \\ p_2 \\ \vdots \\ p_n \\ \hline \therefore C \end{array}$$

p_i = premise i
C = Conclusion

Argument is valid or invalid.

<u>Def</u> An argument is <u>valid</u> if whenever the premises are true then the conclusion is true.

An argument is invalid if the conclusion can be false when all the premises are true.

<u>thm.</u> To test the validity of an argument, construct the truth table $(p_1 \land p_2 \land \dots \land p_n) \to C$

If it is a tautology, the argument is valid

If it is not a tautology, the argument is invalid.

Note: Validity of the argument should not be confused with the truth or falsity of the premises and/or the conclusion.

Arguments.

① $$\begin{array}{l} p \to q \\ p \\ \hline \therefore q \end{array}$$

p	q	$[(p \to q) \land p] \to q$
T	T	T
T	F	T
F	T	T
F	F	T

Valid.

② $$\begin{array}{l} p \to q \\ q \\ \hline \therefore p \end{array}$$

p	q	$[(p \to q) \land q] \to p$
T	T	T
T	F	T
F	T	F
F	F	T

Invalid

③ $$\begin{array}{l} p \to q \\ \neg p \\ \hline \therefore \neg q \end{array}$$

p	q	$[(p \to q) \land \neg p] \to \neg q$
T	T	T
T	F	T
F	T	F
F	F	T

Invalid

④ $p \to q$
$q \to r$
$\therefore p \to r$

p	q	r	$[(p\to q)\wedge(q\to r)] \to p\to r$
T	T	T	T
T	T	F	T
T	F	T	T
T	F	F	T
F	T	T	T
F	T	F	T
F	F	T	T
F	F	F	T

Valid.

Ex Prove valid or invalid:

① If the client was guilty, then he was at the scene of the crime. the client was not at the scene of the crime. Hence, the client was not guilty:

p = client guilty
q = client at scene

$p \to q$
$\neg q$
$\therefore \neg p$

p	q	$[(p\to q)\wedge \neg q] \to \neg p$
T	T	T
T	F	T
F	T	T
F	F	T

Valid

② If David passes the final exam, then he will pass the course. David passes the course. Hence, he passed the final exam.

p = passes course
q = passes final exam

$p \to q$
q
$\therefore p$

p	q	$[(p\to q)\wedge q] \to p$
T	T	T
T	F	T
F	T	F
F	F	T

Invalid

③ the governor will call a special session if the Senate cannot reach a compromise. If a majority of the cabinet are in agreement then the governor will not call a special session. the Senate can reach a compromise. Therefore, the governor will not call a special session.

p = Senate <u>can</u> reach compromise
q = Governor call special session
r = majority cabinet in agreement

$\neg p \to q$
$r \to \neg q$
p
$\therefore \neg q$

p	q	r	$[(\neg p \to q) \wedge (r \to \neg q) \wedge p] \to \neg q$
T	T	T	T
T	T	F	F
T	F	T	T
T	F	F	T
F	T	T	T
F	T	F	T
F	F	T	T
F	F	F	T

Invalid

Shortcut:

① $\neg r \to (s \to \neg t)$

② $\neg r \vee w$

③ $\neg p \to s$

④ $\neg w$

⑤ $\therefore t \to p$

p	r	s	t	w	$[① \wedge ② \wedge ③ \wedge ④] \to ⑤$
F	F	T	T	F	T

Argument valid unless conclusion = F

$t \to p = F \quad t = T,\ p = F$

Argument Valid unless premises = T

$\neg w = T, \quad w = F$

$\neg r \vee w = T,\ w = F,\ \neg r = T$ or $r = F$

$\neg p \to s = T,\ p = F,\ T \to s = T$ or $s = T$

But $p = F,\ r = F,\ s = T,\ t = T,\ w = F$

makes $\neg r \to (s \to \neg t) = T \to (T \to F)$

$= T \to F$

$= F$

No way to make premises true and conclusion false, so argument is valid

Direct Method of Proof (Formal proof for above)

$\neg r \to (s \to \neg t)$	Assumption
$\neg r \vee w$	"
$\neg p \to s$	"
$\neg w$	"
$\neg r$	$(\neg r \vee w) \wedge \neg w$ ①
$s \to \neg t$	$(\neg r \to (s \to \neg t)) \wedge \neg r$ ②
$\neg p \to \neg t$	$(\neg p \to s) \wedge (s \to \neg t)$ ③
$t \to p$	contrapositive

Valid arguments:

① $p \vee q$
$\neg q$
$\therefore p$

② $p \to q$
p
$\therefore q$

③ $p \to q$
$q \to r$
$\therefore p \to r$

Propositional Functions and Quantifiers

Def A propositional function, $P(x)$ is true or false for each $x \in \mathcal{U}$

e.g. $\mathcal{U} = \mathbb{Z}$ $\quad P(x)$: "$x > 3$" $\quad P(4) = T,\ P(2) = F$

In general $P(x_1, x_2, \ldots, x_n)$ is a propositional function of n variables.

Def the universal quantification of $P(x)$:
" $P(x)$ is true for all x in the universe "
$\forall x\, P(x)$ (for every x)

Ex Let $\mathcal{U} = \mathbb{Z}$ Let $P(x)$: "$x+1 > x$"
$\forall x\, P(x) = T$
Let $Q(x)$: "$x > 2$"
$\forall x\, Q(x) = F$

Def the existential quantification of $P(x)$ is
" there exists an element x, such that $P(x)$ is true"
$\exists x\, P(x)$ (there is an x, "there is at least one x, for some x, "

Ex Let $\mathcal{U} = \mathbb{Z}$ Let $P(x)$: "$x > 3$"
$\exists x\, P(x) = T$
Let $Q(x)$: "$x+1 = x$"
$\exists x\, Q(x) = F$

Commutative Law of Addition : $\mathcal{U} = $ Reals
$\forall x \forall y\ x + y = y + x$

Additive Inverse Law : $\mathcal{U} = $ Reals.
$\forall x \exists y\ x + y = 0$

Associative Law of Addition $\mathcal{U} = $ Reals
$\forall x \forall y \forall z\ \ x + (y+z) = (x+y) + z$

Translating Sentences into Logical Expressions

<u>Ex</u> "Some student in this class has visited Mexico"

$\mathcal{U}$ = students in class.

$M(x)$: x has visited Mexico

$$\exists x\, M(x)$$

<u>Ex</u> "Every student in this class has visited either Mexico or Canada"

$C(x)$: x has visited Canada.

$$\forall x\,(M(x) \vee C(x))$$

Negations	Equivalent Statement
$\neg \exists x\, P(x)$	$\forall x\, \neg P(x)$ ($P(x)$ false for every x)
$\neg \forall x\, P(x)$	$\exists x\, \neg P(x)$ (there is one x for which $P(x)$ is false

<u>Ex</u> $\neg \exists y\,(\exists x\, R(x,y) \vee \forall x\, S(x,y))$

$$= \forall y\,(\forall x\, \neg R(x,y) \wedge \exists x\, \neg S(x,y))$$

<u>Ex.</u> Negate the following sentences. Put negation <u>next</u> to the propositional function:

① "No one has lost more than \$1000 playing the lottery"

$L(x)$ = x has lost more than \$1000 playing the lottery

$\forall x\, \neg L(x)$

To negate: $\neg \forall x\, \neg L(x) = \exists x\, \neg\neg L(x) = \exists x\, L(x)$

"Someone has lost more than \$1000 playing the lottery."

(2) there is a student in class who has passed the final and received a grade of C.

$P(x)$: x passed final $C(x)$: x received grade of C

$\exists x\,(P(x) \wedge C(x))$

To negate: $\neg(\exists x\,(P(x) \wedge C(x))) = \forall x\,(\neg P(x) \vee \neg C(x))$

"All students in class have not passed the final or not received a grade of C"