Lesson 2

=> and <=>

$P \Rightarrow Q$ (*)

hypothesis — thesis

antecedent — Consequent

premise — Conclusion

$Q \Rightarrow P$ is the converse of (*)

$\neg Q \Rightarrow \neg P$ is the contrapositive of (*)

An implication and its contrapositive mean the same thing i.e they always have the same truth value.

Ex Seattle is in Oregon $\Rightarrow$ apple is a fruit

Converse apple is a fruit $\Rightarrow$ Seattle is in Oregon

Contrapositive apple is not a fruit $\Rightarrow$ Seattle is not in oregon

A propositional formula is a **Tautology** if it is always true, no matter what truth value we give its variables.

Ex $p \vee \neg p$

A propositional formula is a **contradiction** if it is always false, no matter which truth value we give its variables.

Ex $p \wedge \neg p$

A propositional formula is **satisfiable** if we can find a truth value assignment to its variables that makes it True.

Ex $p \vee q$

Two propositional formulas are **equivalent** if they have the same variables and they have the same truth value for every truth value assignment we give their variables.

$P \Rightarrow Q$ and $Q \Rightarrow P$ are not equivalent.
since when $P = F$ and $Q = T$ $P \Rightarrow Q$ is True
and $Q \Rightarrow P$ is F

$P \Rightarrow Q$ and $\neg Q \Rightarrow \neg P$ are equivalent
(hw problem)

$P \Rightarrow Q$ is equivalent to $Q \vee \neg P$

Proof:

P	Q	$\neg P$	$P \Rightarrow Q$	$Q \vee \neg P$
T	T	F	T	T
T	F	F	F	F
F	T	T	T	T
F	F	T	T	T

$\langle = \rangle$ Can be defined using a truth table

P	Q	P <=> Q
T	T	T
T	F	F
F	T	F
F	F	T

or by saying that $P \langle = \rangle Q$ is equivalent to $(P \Rightarrow Q) \wedge (Q \Rightarrow P)$

How many connectives do we need?

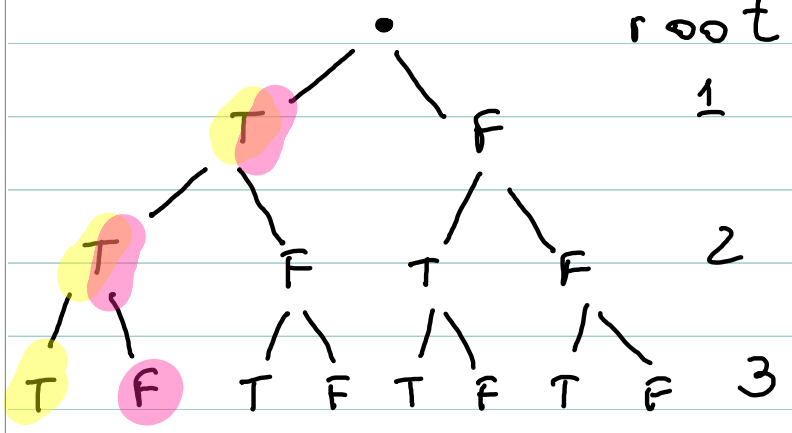
If a propositional formula A has n variables, how many rows are there in a truth table for A? (See Lesson 2 quiz)

Suppose A has 3 variables.

How do we list all possible

Truth value assignments?

Binary tree:



How many leaves? $2^3 = 8$

Paths from root to leaves give you all possible Truth value assignments to 3 variables

Ex TTT, TTF, ...

In hw1 you look at formula with 4 vars

Negation rules (see handout)