5. Propositional Logic (January 22)

Lec 4 Mini Review.

atoms✓ literals✓ conjunctive clauses✓ DNF✓

contrapositive of a conditional statement✓ converse of a conditional statement✓

Using the laws from the Table of Logical Equivalences...

- to prove two propositions are logically equivalent✓
- to show a proposition is a contradiction/tautology✓
- to find a DNF of a given proposition✓
- to find a logically equivalent proposition using specified connectives✓

VALID ARGUMENTS

- An argument $(P_1 \wedge P_2 \wedge \cdots \wedge P_k) \to C$ is called **a valid argument** if and only if the conclusion C is true whenever all the premises $P_1, \ldots, P_k$ are true.
- In other words, $(P_1 \wedge P_2 \wedge \cdots \wedge P_k) \to C$ is a **valid argument** if and only if $(P_1 \wedge P_2 \wedge \cdots \wedge P_k) \to C$ is a tautology.

Questions to Ponder about Arguments.

- What if, for some truth assignment, the premises are all true, but the conclusion is false?

If all premises are T then $P_1 \wedge \cdots \wedge P_k$ will be T. If the conclusion is F, then, for this truth assignment, $(P_1 \wedge \cdots \wedge P_k) \to C$ will be of the form $T \to F$ which is F. Thus, the argument would not be a tautology, hence <u>invalid</u>.

- What happens when any one of the premises is false?

If, for some truth assignment, at least one premise is F, then $P_1 \wedge \cdots \wedge P_k$ will be F. As we know,

$$F \to (\text{anything}) \text{ is } T.$$

Thus, for any truth assignment in which at least one premise is F, the argument $P_1 \wedge \cdots \wedge P_k \to C$ will be T.

<u>Important!</u> If, for some truth assignment, the argument is T, it does <u>not</u> mean the argument is valid.

In order to be valid, $P_1 \wedge \cdots \wedge P_k \to C$ must be a <u>tautology</u>!

◇ What if the set $\{P_1, P_2, \ldots, P_k\}$ of premises is inconsistent?

inconsistent means for every truth assignment, at least one premise is F.

$\therefore$ for every truth assignment, the conjunction $\boxed{(P_1 \wedge \cdots \wedge P_k)}$ of premises will be F.

$\therefore$ for every truth assignment, the argument will be of the form $\boxed{(F) \to C}$ (which is T regardless of C)

$\therefore$ for every truth assignment, the argument $\boxed{(P_1 \wedge \cdots \wedge P_k) \to C}$ will be T.

$\therefore$ the argument $\boxed{(P_1 \wedge \cdots \wedge P_k) \to C}$ will be a tautology, hence a valid argument

◇ What if the conclusion is true?

If, for some truth assignment, the conclusion C is T, then, for that truth assignment, the argument $[(P_1 \wedge \cdots \wedge P_k) \to C]$ will be T because $\boxed{(\text{anything}) \to T}$ is T.

But one such truth assignment does not guarantee that the argument $[(P_1 \wedge \cdots \wedge P_k) \to C]$ is a tautology. (to be valid, the argument must be T for all truth assignments)

◇ An argument with premises $P_1, \ldots, P_k$ and conclusion C is called **invalid** if $(P_1 \wedge \cdots \wedge P_k) \to C$ is not a tautology.

◇ If an argument is invalid, then a **counterexample** is a truth assignment for which $(P_1 \wedge \cdots \wedge P_k) \to C$ is **F**.

◇ In other words, a counterexample is a truth assignment which certifies that $(P_1 \wedge \cdots \wedge P_k) \to C$ is not a tautology.

Example 5.1.

(2 premises)

P_1 $\quad A \vee B$

P_2 $\quad \neg B$

$\therefore C$ $\quad \therefore \neg A$

Is this argument valid? If not, give all counterexamples.

$\boxed{\text{It's a valid argument if and only if } [(A \vee B) \wedge (\neg B)] \to (\neg A) \text{ is a tautology.}}$

A	B	premise P_1 $A \vee B$	premise P_2 $\neg B$	conclusion C $\neg A$	argument $(P_1 \wedge P_2) \to C$ $[(A \vee B) \wedge (\neg B)] \to (\neg A)$
T	T	T	F	F	T
T	F	T	T	F	F
F	T	T	F	T	T
F	F	F	T	T	T

there is one counterexample: when A=T, B=F, both premises are T but conclusion is F.

the argument is not a tautology
$\therefore$ it is invalid.

* These notes are solely for the personal use of students registered in MAT1348.

Truth Trees (a.k.a. Semantic Tableaux)

- A **truth tree** is an alternative structure for examining all the ways that a compound proposition, say X, can be true.
- **Unlike a truth table**, the size of a **truth tree** does not grow exponentially as a function of the number of propositional variables in X; instead, the size of a truth tree varies depending on the number of logical connectives in X, and the order in which we **grow the tree.**
- We place X at the **root** of a truth tree: the **root** is at the top of the tree and the rest of the tree **"grows"** down from there using **branching rules**.

Branching Rules for Truth Trees (a.k.a. Semantic Tableaux)

Splitting Rule | **Non-Splitting Rule**

$$\begin{array}{c} a \vee b \\ \swarrow \quad \searrow \\ a \qquad b \end{array} \qquad\qquad \begin{array}{c} a \wedge b \\ | \\ a \\ b \end{array}$$

Splitting Rule | (De Morgan's Laws) | **Non-Splitting Rule**

$$\begin{array}{c} \neg(a \wedge b) \\ \swarrow \quad \searrow \\ \neg a \qquad \neg b \end{array} \qquad\qquad \begin{array}{c} \neg(a \vee b) \\ | \\ \neg a \\ \neg b \end{array}$$

Splitting Rule | **Non-Splitting Rule**

$$\begin{array}{c} a \rightarrow b \\ \swarrow \quad \searrow \\ \neg a \qquad b \end{array} \qquad\qquad \begin{array}{c} \neg(a \rightarrow b) \\ | \\ a \\ \neg b \end{array}$$

(Implication Law)

One More Non-Splitting Rule

$$\neg\neg a$$
$$|$$
$$a$$

(Double Negation Law)

Two More Splitting Rules

$a \leftrightarrow b$

/	\
a	$\neg a$
b	$\neg b$

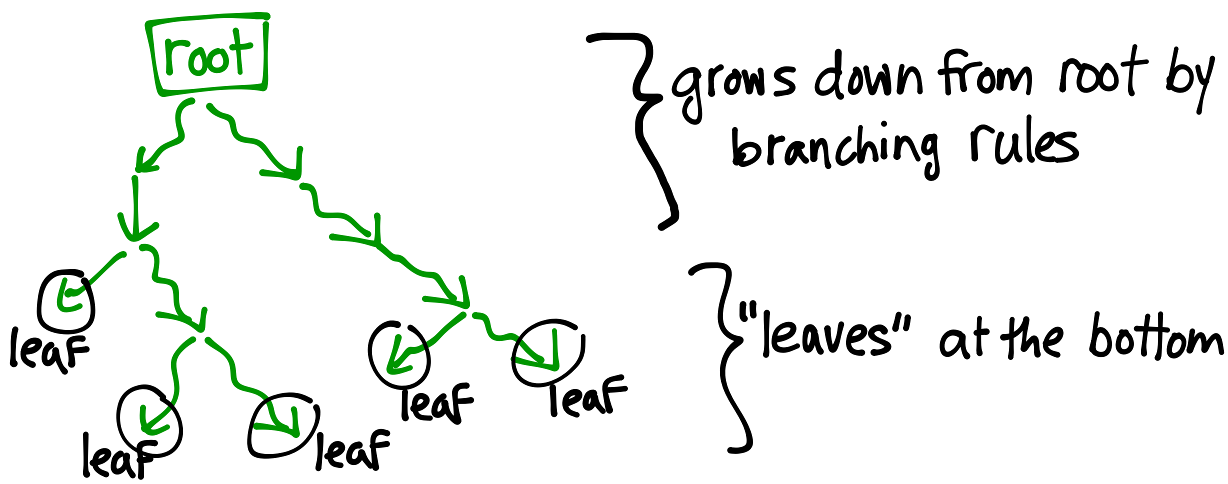
$\neg(a \leftrightarrow b)$

/	\
a	$\neg a$
$\neg b$	b

(Biconditional Law $a \leftrightarrow b \equiv (a \wedge b) \vee (\neg a \wedge \neg b)$)

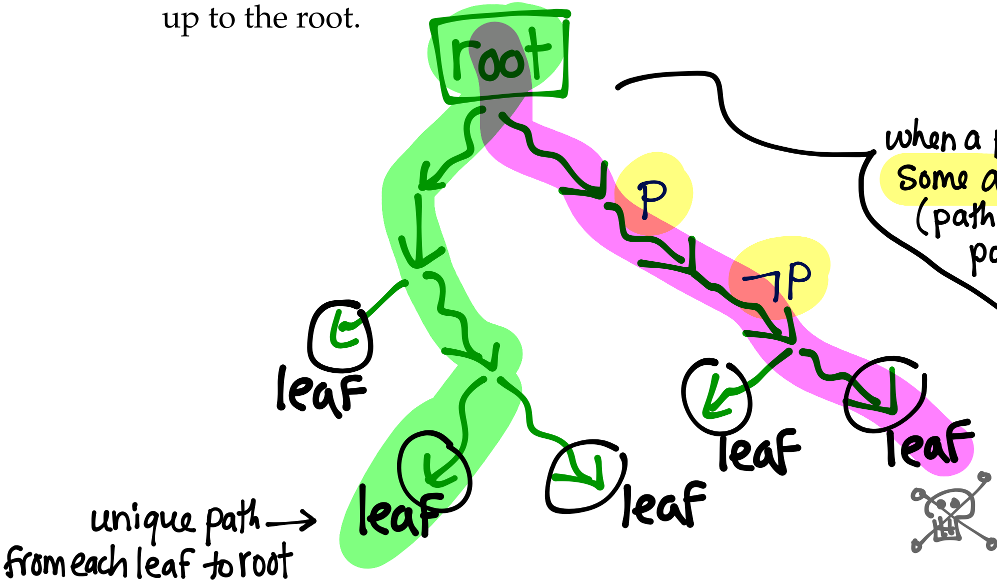
How to Grow a Truth Tree

- ▷ Truth trees grow **from the root** (at the top), **by branching rules**, **down to the leaves** (at the bottom).



- ▷ Each proposition in a truth tree is called **unchecked** until
  - ◦ it is a literal (just an atom or just the negation of an atom), or
  - ◦ its branching rule has been applied to all paths stemming down from the proposition's location in the tree.
- ▷ Starting from the top, one unchecked proposition at a time, apply the branching rule of each unchecked proposition to all paths that stem down from the unchecked proposition.
- ▷ Once the branching rule has been applied, the proposition becomes **checked**✓

▷ **Fact.** From each **leaf** (at the bottom of the tree so far), there is a unique path going back up to the root.



when a path from leaf to root contains some atom and that atom's negation (path contains a contradiction) the path becomes "closed"/"inactive" or "dead"

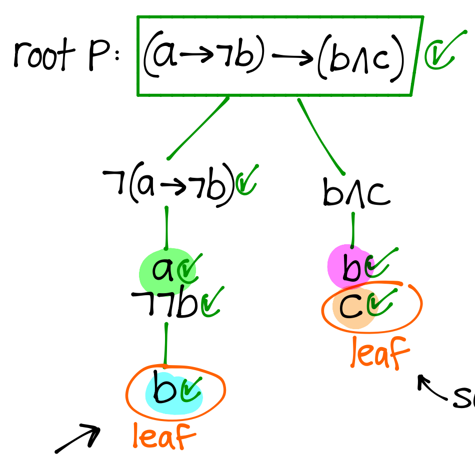
▷ A path from a leaf back to the root is called **inactive** or **closed** if it contains an atom as well as that atom's negation; otherwise the path is called **active** or **open**.

▷ A path from a leaf back to the root is called **complete** if there are no unchecked propositions on that path.

▷ **Each complete open path tells us one way that will make the root true.**

▷ The tree is **complete** (i.e. **done growing**) when each path from leaf to root is **closed** or **complete**.

Example 5.2. Grow a complete truth tree for the compound proposition $P: \quad (a \to \neg b) \to (b \wedge c)$.



the path from this leaf to the root does not contain any contradictions and all propositions have been checked. ∴ it's a complete active path ⇝ when a is T and b is T the root P is true

same here ∴ it's a complete active path ⇝ when b is T and c is T the root P is T.

In summary, the complete active paths tell us that P is T when

$a \wedge b$ is T or when $b \wedge c$ is T ie $P \equiv (a \wedge b) \vee (b \wedge c)$ ← what is this?! a DNF for P!

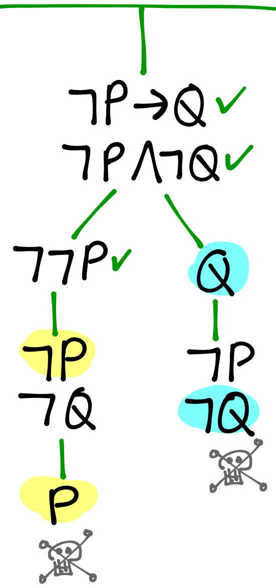
Determining Whether the Root of a Truth Tree is a Contradiction

Question to Ponder. Suppse we grow a complete truth tree with X at its root. What does it mean if **all paths are closed/inactive** ("dead") ?

It means the root can never be true.
i.e. the root is a contradiction.

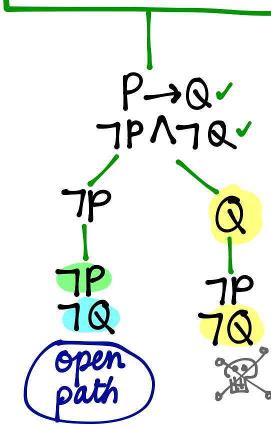
Example 5.3. Use a truth tree to determine whether each of the following compound propositions is a contradiction. If it is not a contraction, give all counterexamples (that is, give all truth assignments that make the proposition true, thereby certifying that the proposition is not a contradiction).

i. $(\neg P \to Q) \wedge (\neg P \wedge \neg Q)$ root ✓



Since all paths are "dead" the root is never True. ∴ the root is a contradiction

ii. $(P \to Q) \wedge (\neg P \wedge \neg Q)$ root ✓



Since there is an open path, the root can be True. ∴ the root is not a contradiction.

Counterexample (to prove root is not a contradiction)
- When P=F and Q=F, the root is True

Study Guide

Important terms and concepts:

- ⋄ valid argument   invalid argument   counterexamples
- ⋄ truth trees (semantic tableaux)   branching rules   open vs. closed paths
- ⋄ using a truth tree to check whether a proposition is a contradiction

Exercises	Supp. Ex. List §1 # 2, 5, 6, 7c