5. Truth Trees

- 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
$a \vee b$ branches to a and b	$a \wedge b$, then a, then b

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

Splitting Rule	Non-Splitting Rule
$\neg(a \wedge b)$ branches to $\neg a$ and $\neg b$	$\neg(a \vee b)$, then $\neg a$, then $\neg b$

Splitting Rule	Non-Splitting Rule
$a \to b$ branches to $\neg a$ and b	$\neg(a \to b)$, then a, then $\neg b$

(Implication Law)

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

One More Non-Splitting Rule

$$\begin{array}{c} \neg\neg a \\ | \\ a \end{array}$$

(Double Negation Law)

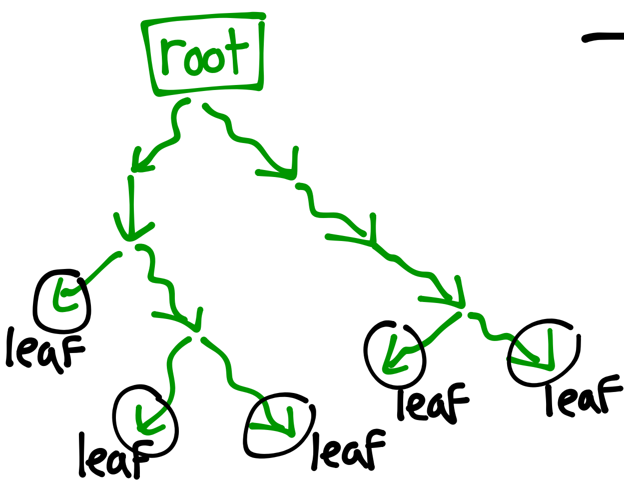
Three More Splitting Rules

$$\begin{array}{ccc} & a \leftrightarrow b & \\ / & & \backslash \\ a & & \neg a \\ b & & \neg b \end{array} \qquad \begin{array}{ccc} & \neg(a \leftrightarrow b) & \\ / & & \backslash \\ a & & \neg a \\ \neg b & & b \end{array} \qquad \begin{array}{ccc} & a \oplus b & \\ / & & \backslash \\ a & & \neg a \\ \neg b & & b \end{array}$$

(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).

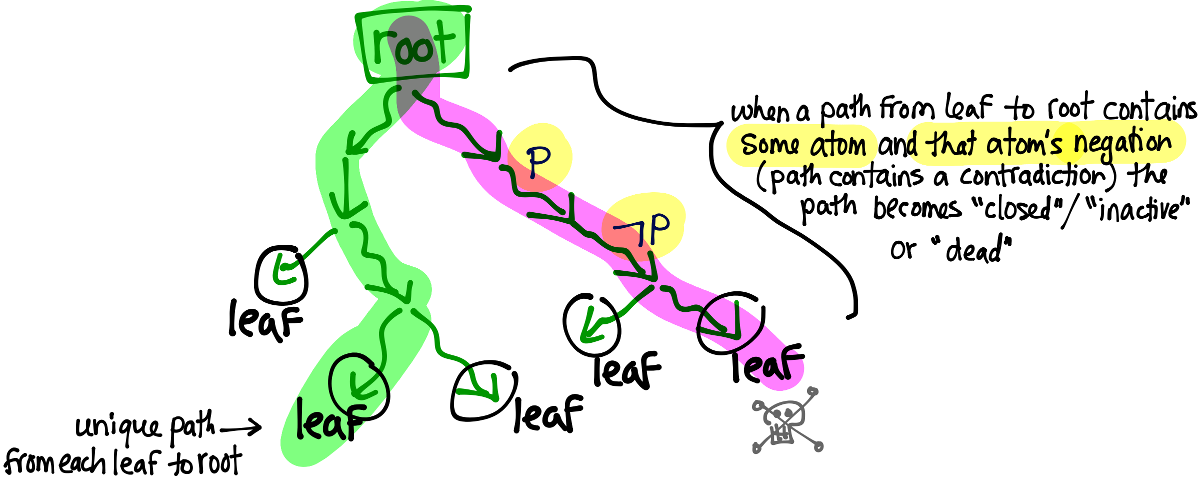


grows down from root by branching rules

"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.



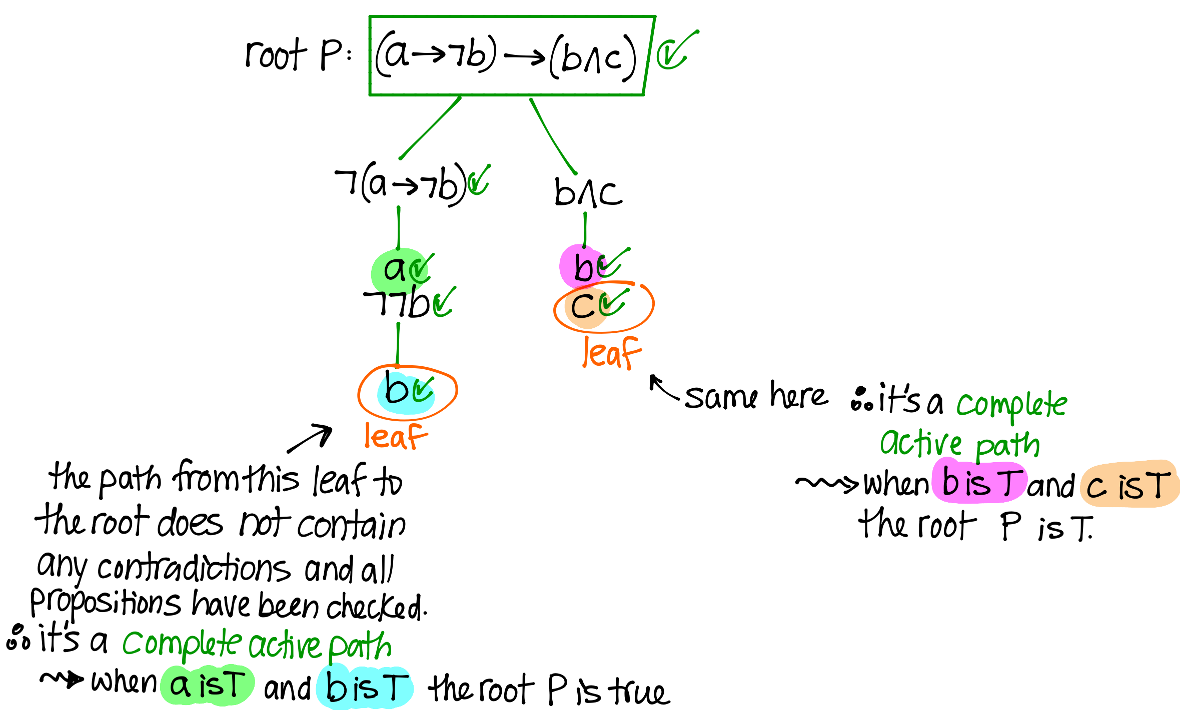
▹ A path from a leaf back to the root is called **inactive** or **closed** if it contains an atom as well as that atom's negation; otherwise the path is called **active** or **open**.

▹ 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**.

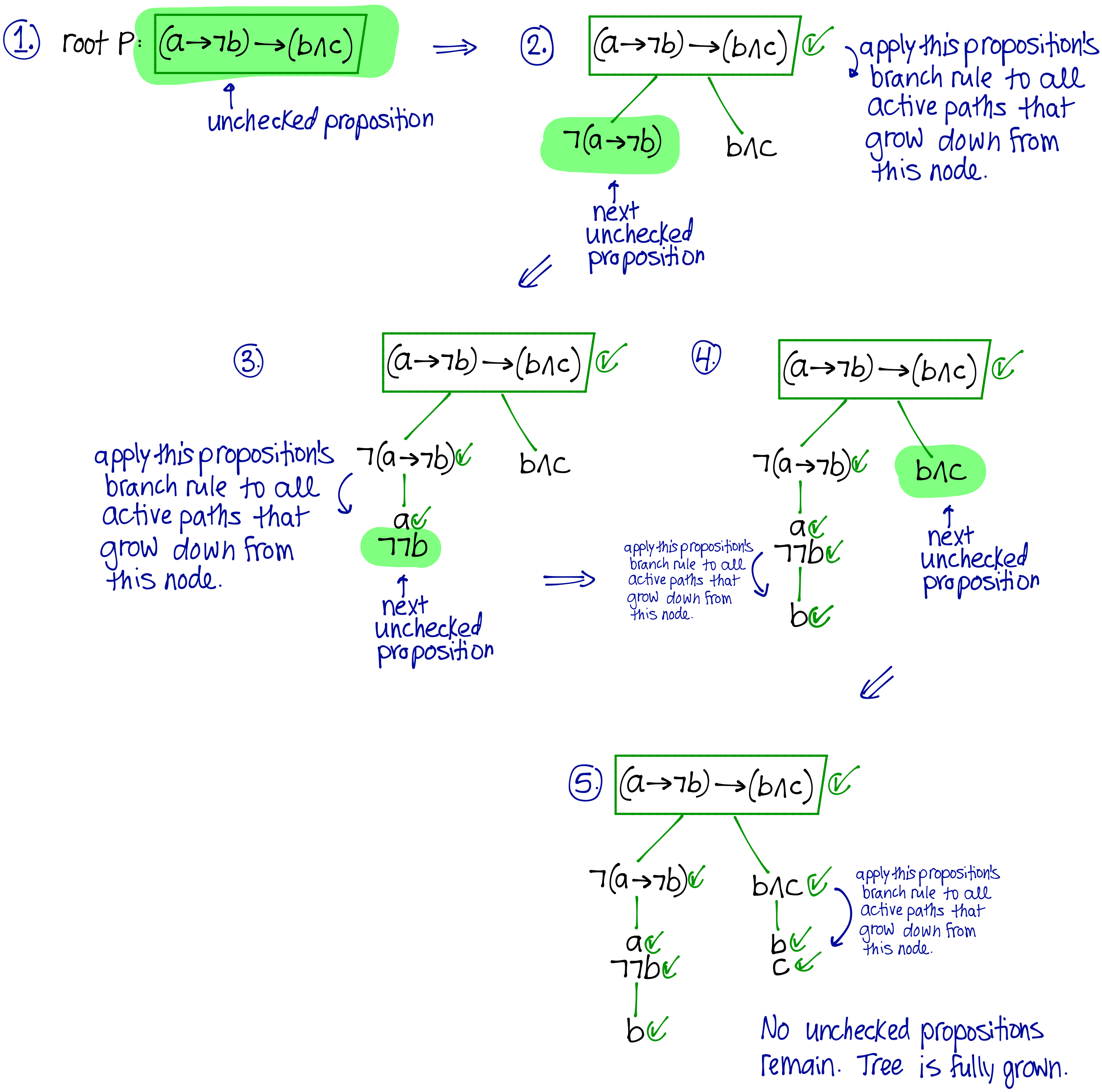
Example 5.1. Grow a complete truth tree for the compound proposition $P: \quad (a \to \neg b) \to (b \wedge c)$.



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!

Here is Example 5.1 again, showing one step at a time:



Finding a DNF for the Root of a Truth Tree

To summarize:

- To find a DNF for a compound proposition X, we grow a complete truth tree with X at its root.
- Each complete open path gives one **conjunctive clause** consisting of the conjunction of the literals found on that path going from the leaf back up to the root.
- The disjunction of all such conjunctive clauses gives a DNF for P.

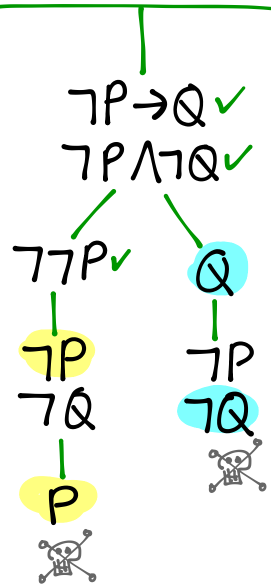
Determining Whether the Root of a Truth Tree is a Contradiction

Question to Ponder. Suppose 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.

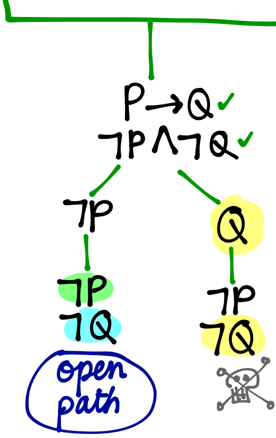
Example 5.2. Use a truth tree to determine whether each of the following compound propositions is a contradiction. If it is not a contraction, give all counterexamples (that is, give all truth assignments that make the proposition true, thereby certifying that the proposition is not a contradiction).

i. $\boxed{(\neg P \to Q) \land (\neg P \land \neg Q)}$ root ✓



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

ii. $\boxed{(P \to Q) \land (\neg P \land \neg Q)}$ 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

Determining Whether a Set of Compound Propositions is Consistent

- A set $\{P_1, P_2, \ldots, P_n\}$ of compound propositions is called **consistent** if there exists at least one truth assignment that makes all the propositions $P_1, \ldots, P_n$ true at the same time.
- If $\{P_1, P_2, \ldots, P_n\}$ is not consistent, then, for each possible truth assignment, at least one of the propositions P_i is false; in this case, the set $\{P_1, P_2, \ldots, P_n\}$ is called **inconsistent**. In this case, the conjunction $P_1 \land \cdots \land P_n$ is a contradiction.

  Equivalently, a set $\{P_1, P_2, \ldots, P_n\}$ is...

  ▷ **consistent** if the conjunction $P_1 \land \cdots \land P_n$ is *not* a contradiction.

  ▷ **inconsistent** if the conjunction $P_1 \land \cdots \land P_n$ *is* a contradiction.

♠ **To test whether the set $\{P_1, P_2, \ldots, P_k\}$ is consistent**

- grow a tree with **root**

$$\boxed{\begin{array}{c} P_1 \\ P_2 \\ \vdots \\ P_k \end{array}}$$

← this root corresponds to non-spitting rule for the conjunction $P_1 \wedge \cdots \wedge P_k$

- if all paths are dead, then $(P_1 \wedge P_2 \wedge \cdots \wedge P_k)$ is a contradiction, which means $(P_1 \wedge P_2 \wedge \cdots \wedge P_k)$ can never be true, which means $\{P_1, P_2, \ldots, P_k\}$ is inconsistent.
- if one path (or more) is alive (complete active path), then these paths give the truth assignments that prove that $(P_1 \wedge P_2 \wedge \cdots \wedge P_k)$ can be true, which translates to showing that $(P_1 \wedge P_2 \wedge \cdots \wedge P_k)$ is consistent.

Example 5.3. Use a truth tree to determine whether the following set of four compound propositions is consistent: $\left\{ \underbrace{x \wedge \neg y}_{P_1},\ \underbrace{\neg((p \vee s) \to x)}_{P_2},\ \underbrace{x \leftrightarrow \neg p}_{P_3},\ \underbrace{\neg(s \vee y)}_{P_4} \right\}$

root

P_1 $x \wedge \neg y$ ✓
P_2 $\neg((p \vee s) \to x)$ ✓
P_3 $x \leftrightarrow \neg p$
P_4 $\neg(s \vee y)$

x
$\neg y$

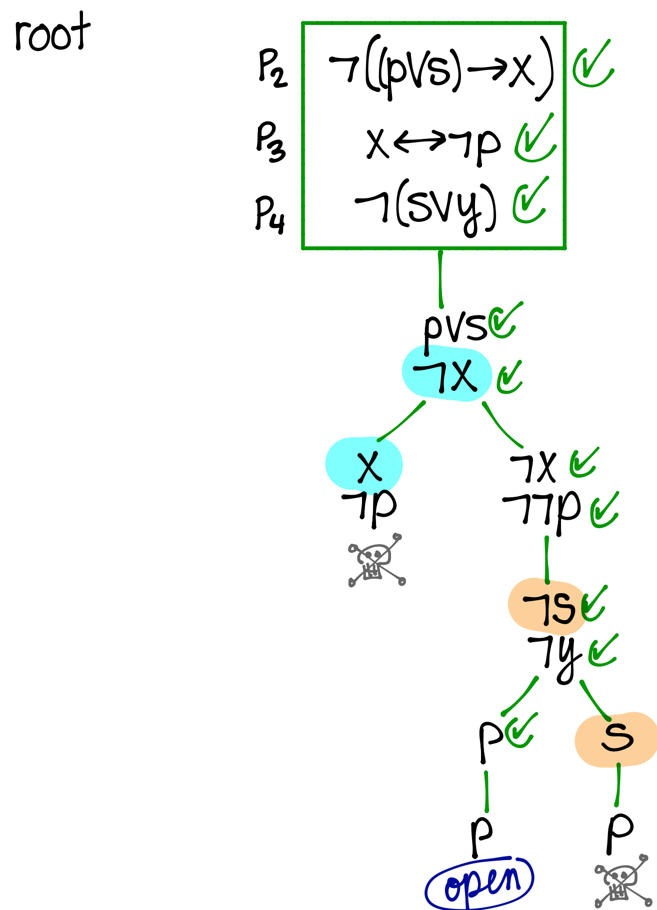
$p \vee s$
$\neg x$

☠

The tree's only path so far is already dead, so the root (which corresponds to the conjunction $P_1 \wedge P_2 \wedge P_3 \wedge P_4$) is a contradiction

∴ the set $\{P_1, P_2, P_3, P_4\}$ is inconsistent.

Exercise 5.4. Consider the same propositions P_1, P_2, P_3, P_4 from Example 5.3. Determine whether the set $\{P_2, P_3, P_4\}$ is consistent, first, using a truth tree, then using a truth table.



Since there exists at least one complete active path the root can be true

$\therefore$ the set $\{P_2, P_3, P_4\}$ is consistent.

The open path tells us a truth assignment for which $P_2 \wedge P_3 \wedge P_4$ is true, namely, when $p = T$, $y = F$, $s = F$, $x = F$

Exercise: Verify the above answer using a truth table.

Study Guide

Important terms and concepts:

- ◇ truth trees (semantic tableaux) branching rules open vs. closed paths
- ◇ using a truth tree to find a DNF for a given proposition
- ◇ using a truth tree to check whether a proposition is a contradiction
- ◇ using a truth tree to determine whether a set of propositions is consistent/inconsistent

Exercises	Sup.Ex. §1 # 4b, 7b
	Sup.Ex. §1 # 1 Is $\{P_1, P_2, P_3, P_4\}$ consistent? Is $\{P_1, P_2, P_3\}$ consistent? Is $\{P_2, P_3\}$ consistent?
	Rosen §1.2 # 9–11