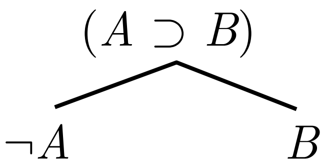
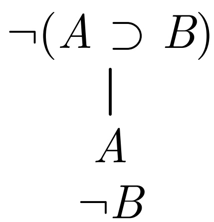
Chapter 18: PLC Trees

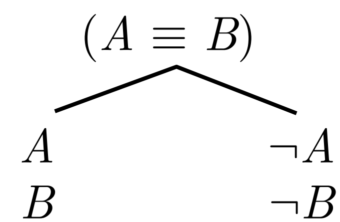
Further PLC Rules:

(f) 

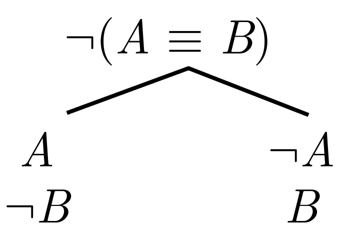
Add a fork with $\neg A$, B as separate branches to each open path containing $(A \supset B)$.

(g) 

Add A, $\neg B$ to each open path containing $\neg(A \supset B)$.

(h) 

Add a fork with A, B and $\neg A$, $\neg B$ as separate branches to each open path containing $(A \equiv B)$.

(e) 

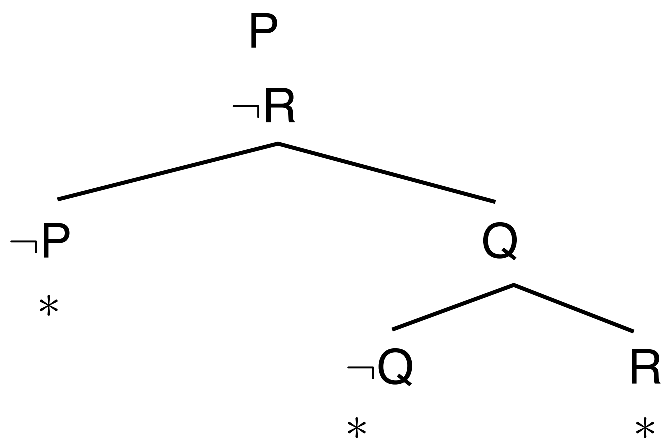
Add a fork with A, $\neg B$ and $\neg A$, B as separate branches to each open path containing $\neg(A \equiv B)$.

__Ex1.__ $(P \supset Q), (Q \supset R) \therefore (P \supset R)$

$(P \supset Q)$ ✓

$(Q \supset R)$ ✓

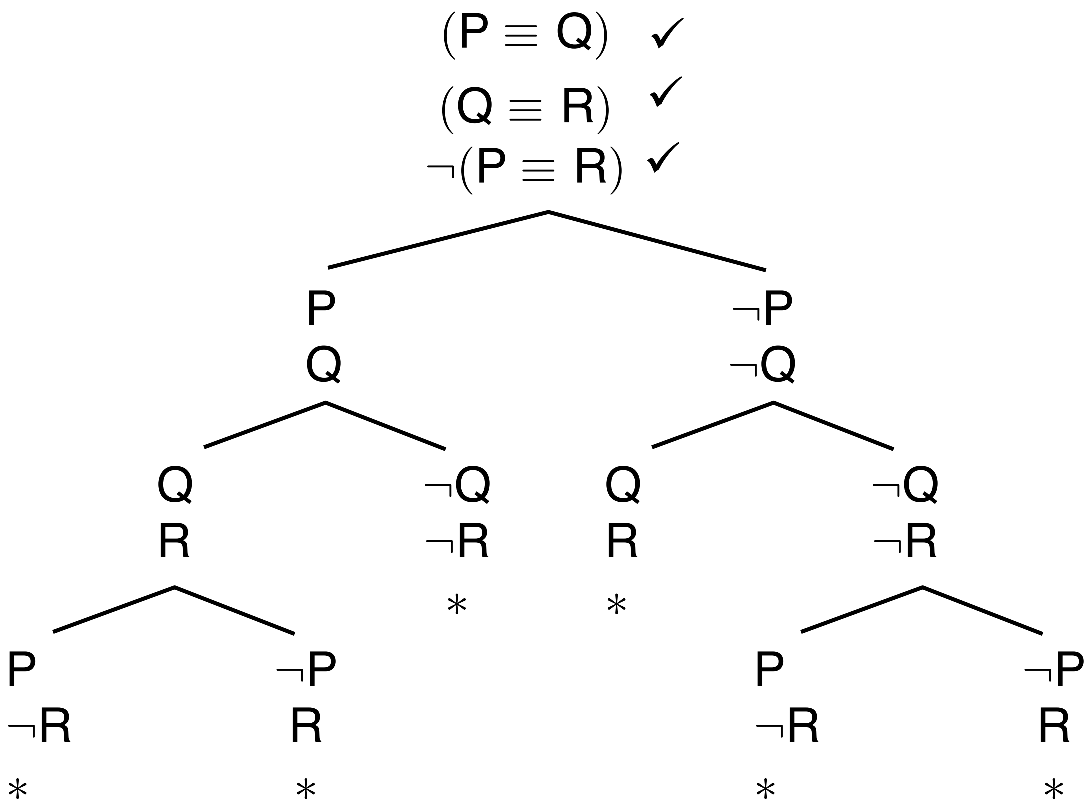
$\neg(P \supset R)$ ✓



All branches close.

__*So*__: **PLC** argument is *tautologically valid.*

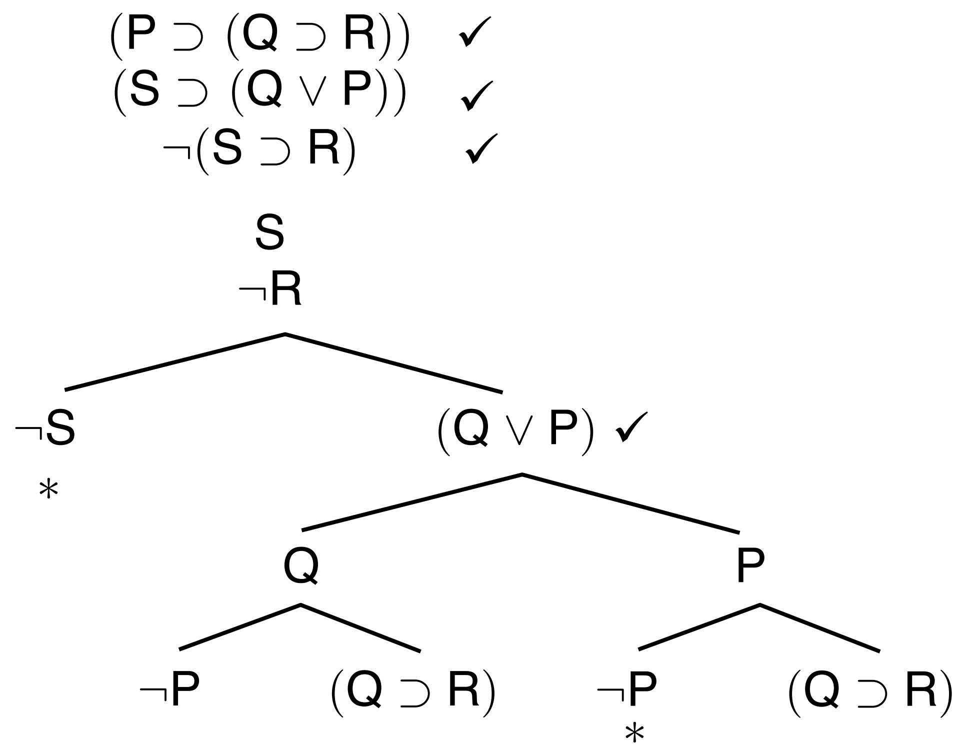
Ex2. $(P \equiv Q), (Q \equiv R) \therefore (P \equiv R)$



All branches close.

So: **PLC** argument is *tautologically valid*.

Ex3. $(P \supset (Q \supset R)), (S \supset (Q \vee P)) \therefore (S \supset R)$



There is at least one completed open branch.

It corresponds to a valuation in which $P \Rightarrow F$, $Q \Rightarrow T$, $R \Rightarrow F$, $S \Rightarrow T$.

So: **PLC** argument is *tautologically invalid*.

Ex4. $((\neg Q \supset R) \supset (Q \equiv S)), (P \wedge R), \neg(Q \wedge S) \therefore \neg((P \wedge Q) \supset (R \wedge S))$

$((\neg Q \supset R) \supset (Q \equiv S))$ ✓
$(P \wedge R)$ ✓
$\neg(Q \wedge S)$ ✓
$\neg\neg((P \wedge Q) \supset (R \wedge S))$ ✓

$((P \wedge Q) \supset (R \wedge S))$ ✓
P
R

- $\neg(P \wedge Q)$ ✓
  - $\neg P$
    $*$
  - $\neg Q$
    - $\neg Q$
      - $\neg(\neg Q \supset R)$ ✓
        $\neg Q$
        R
        $*$
      - $(Q \equiv S)$ ✓
        - Q
          S
          $*$
        - $\neg Q$
          $\neg S$
    - $\neg S$
- $(R \wedge S)$ ✓
  R
  S
  - $\neg Q$
  - $\neg S$
    $*$

At least one completed open branch:

$S \Rightarrow T$, $Q \Rightarrow T$, $R \Rightarrow T$, $P \Rightarrow T$.

So: **PLC** argument is *tautologically invalid*.

Chapter 19: PL Trees Vindicated

We now have two notions of *derivation*:

(a) Tree derivation: From $A_1, A_2, ..., A_n, \neg C$, derive a closed tree.
Syntactic derivation: "$\vdash$" $A_1, A_2, ..., A_n \vdash C$

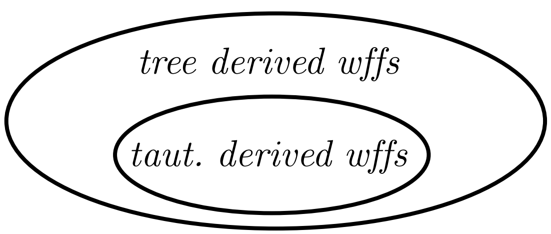
(b) Tautological entailment: From the truth of $A_1, A_2, ..., A_n$, derive the truth of C.
Semantic derivation: "$\vDash$" $A_1, A_2, ..., A_n \vDash C$

Are these equivalent notions?

(1) Is Tree-derivation ***sound***? Does it imply tautological derivation?

Still leaves open possibility of wffs that can be tautologically derived, but not tree-derived.

(2) Is Tree-derivation ***complete***? Does it capture everything that tautological derivation does (*i.e.*, does tautological derivation imply it)?



Still leaves open possibility of wffs that can be tree-derived, but not tautologically derived.

So: The equivalence of tree-derivation and tautological derivation requires us to prove both the *soundness* and *completeness* of tree-derivation.

Soundness of PL Tree-Derivation

Claim: If from $A_1, \ldots, A_n, \neg C$ we can construct a *closed tree*, then $A_1, \ldots, A_n \vDash C$.

(*Or*: If $A_1, \ldots, A_n \vdash C$, then $A_1, \ldots, A_n \vDash C$.)

Proof: We will show the converse: If $A_1, \ldots, A_n \nvDash C$, then the corresponding tree will not close.

Suppose: $A_1, \ldots, A_n \nvDash C$.

Then: There's a valuation for which $A_1 \Rightarrow$ T, ..., $A_n \Rightarrow$ T, $\neg C \Rightarrow$ T.

So: The initial trunk of the corresponding tree is a *satisfiable* path (a path whose nodes are all true).

Now: Suppose we extend the trunk using any of the non-branching rules (a, b, c, g).

Then: The resulting single path will remain satisfiable.

Now: Extend the trunk using any of the branching rules (d, e, f, h, i).

Then: At least one resulting path will remain satisfiable.

Thus: Given that we started with a satisfiable trunk, there will always be at least one satisfiable path in the tree.

Hence: The tree will not close.

Completeness of PL Tree-Derivation

Claim: If $A_1, \ldots, A_n \vDash C$, then from $A_1, \ldots, A_n, \neg C$ we can construct a *closed tree*.

(Or: If $A_1, \ldots, A_n \vDash C$, then $A_1, \ldots, A_n \vdash C$.)

Proof: We will show the converse: If from $A_1, \ldots, A_n, \neg C$ we can construct a *completed open tree*, then $A_1, \ldots, A_n \nvDash C$.

(C1): For any completed open path in a **PL** tree, there's a valuation that makes the primitive *wffs* true.

(C2): If there's a valuation that makes the primitive *wffs* on a completed open path true, then it makes all the *wffs* on that path true, too.

Now: Suppose (C1) and (C2) hold, and suppose from $A_1, \ldots, A_n, \neg C$ we can construct a *completed open tree*.

Then: From (C1), there's a valuation that makes all primitive *wffs* on any completed open path in the tree true. (There's got to be at least one such completed open path.)

And: From (C2), this valuation makes all *wffs* on that path true.

And: In particular, it makes $A_1, \ldots, A_n, \neg C$ all true!

So: There's a way to make $A_1, \ldots, A_n$ all true and C false.

Thus: $A_1, \ldots, A_n \nvDash C$.

> <u>(C1):</u> For any completed open path in a **PL** tree, there's a valuation that makes the primitive *wffs* true.

***<u>Proof</u>*:** Consider the following valuation, call it V:

(1) Assign T to all bare (free-standing) atomic *wff* on the completed open path.

(2) Assign F to all atomic *wff* that appear inside non-atomic *wff* on the completed open path.

<u>Then</u>: If a *wff* on the completed open path is a bare atomic *wff*, then V assigns it T.

<u>And</u>: If a *wff* on the open path is the negation of an atomic *wff*, then V assigns its atom F, so it is assigned T. (The atom of such a negation cannot appear bare on the path since it is a completed open path.)

<u>So</u>: V makes all primitive *wffs* on the completed open path true.

Definition: The *wffs* S_1, S_2, ... are ***truth makers*** for the *wff* W if the joint truth of S_1, S_2, ... ensures the truth of W.

Ex: A, B are truth makers for $(A \wedge B)$.

$\neg A$ is a truth maker for $\neg(A \wedge B)$.

$\neg B$ is a truth maker for $\neg(A \wedge B)$.

Thus: The tree rules say: "Add the truth makers of W to an open path containing W."

(C2): If there's a valuation that makes the primitive *wffs* on a completed open path true, then it makes all the *wffs* on that path true, too.

Proof (sketch):

Note: On a completed open path,

(i) All non-primitive *wffs* have been unpacked.

(ii) The primitive *wffs* that occur are truth makers for the non-primitive *wffs* immediately above them, which are truth makers for the *wffs* immediately above them, *etc.*; all the way back to the trunk.

So: If a valuation makes all primitive *wffs* on a completed open path true, it will make all the truth makers on that path true.

Thus: It will make all *wffs* on that path true.

*Soundness and Completeness of **PL** Tree-Derivation*

$A_1, \ldots, A_n \vdash C$, if and only if $A_1, \ldots, A_n \vDash C$.