# lecture 11, uses of trees for PL

phil1012 introductory logic

# overview

## this lecture

- in the previous lecture we looked at the motivation for trees for PL
- we looked at the motivation for the particular tree rules
- in this lecture we look at how we construct trees and how we use them to test for various logical properties
- this lecture really only gives a general overview of how to construct trees
- in the live lecture and the tutorials we will do lots of examples

## learning outcomes

- after doing the relevant reading for this lecture, listening to the lecture, and attending the relevant tutorial, you will be able to:
    - construct trees for propositions of PL
    - use trees to determine whether arguments in PL are valid and to find counterexamples to invalid arguments
    - use trees to test for various logical properties of PL propositions and to read off assignments from open paths

# applying the rules

## applying the rules

- let's look at how we apply the rules . . .

## which rule to apply

- before you apply a rule ask yourself: what is the main connective?
    - if it is anything but negation, then apply the relevant rule
    - if it is negation, look for the main connective of the negand and apply the relevant rule
- suppose you have:

| | | |
|---|---|---|
| 1. | $\neg(A \vee B)$ | assumption |

---

- is this right?

| | | |
|---|---|---|
| 1. | $\neg(A \vee B)$ ✓ | assumption |
| 2. | $A \quad B$ | rule for disjunction |

- no, it is wrong

---

- is this right?

| | | |
|---|---|---|
| 1. | $\neg(A \vee B)$ ✓ | assumption |
| 2. | $\neg A$ | |
| 3. | $\neg B$ | rule for negated disjunction |

- yes. here the main connective is negation, and the main connective of the negand is disjunction.

---

- before you apply a rule ask yourself: what is the main connective?
  - if it is anything but negation, then apply the relevant rule
  - if it is negation, look for the main connective of the negand and apply the relevant rule

# testing a single proposition for satisfiability

- suppose we want to test whether the following proposition is satisfiable: $( A \to ( B \to A ) )$
- we begin by writing this proposition on the first line

| | | |
|---|---|---|
| 1. | $(A \to (B \to A))$ | assumption |

- notice that the assumption is not checked

---

- then we apply the rule for the conditional
- remembering to check off the assumption

---

| | | |
|---|---|---|
| 1. | $(A \to (B \to A))$ | assumption |

---

| | | |
|---|---|---|
| 1. | $(A \to (B \to A))$ ✓ | assumption |
| 2. | $\neg A \qquad (B \to A)$ | rule for conditional |

- notice that we have now checked the assumption, since we have applied the rule
- now we ask: have we applied all the rules we can apply?
- no

---

- now we apply the rule for the conditional again

---

| | | |
|---|---|---|
| 1. | $(A \to (B \to A))$ ✓ | assumption |
| 2. | $\neg A \qquad (B \to A)$ | rule for conditional |

---

| | | |
|---|---|---|
| 1. | $(A \to (B \to A))$ ✓ | assumption |
| 2. | $\neg A \qquad (B \to A)$ ✓ | rule for conditional |
| 3. | $\neg B \qquad A$ | rule for conditional |

- now we ask: have we applied all the rules we can apply?
- yes
- this tree is complete

# the notion of a path

- a helpful notion in thinking about trees is the notion of a path
- a **path** through a tree is a complete route from the topmost proposition down until one can go no further
- for example, suppose we have this tree:

| | | | |
|---|---|---|---|
| 1. | $((A \wedge \neg A) \vee B)$ ✓ | | assumption |
| 2. | $(A \wedge \neg A)$ ✓ | $B$ | rule for disjunction |
| 3. | $A$ | | rule for conjunction |
| 4. | $\neg A$ | | rule for conjunction |

---

- here is one path through the tree:

| | | | |
|---|---|---|---|
| 1. | $\boxed{((A \wedge \neg A) \vee B)\ ✓}$ | | assumption |
| 2. | $\boxed{(A \wedge \neg A)\ ✓}$ | $B$ | rule for disjunction |
| 3. | $\boxed{A}$ | | rule for conjunction |
| 4. | $\boxed{\neg A}$ | | rule for conjunction |

---

- here is another path through the tree:

| | | | |
|---|---|---|---|
| 1. | $\boxed{((A \wedge \neg A) \vee B)\ ✓}$ | | assumption |
| 2. | $(A \wedge \neg A)$ ✓ | $\boxed{B}$ | rule for disjunction |
| 3. | $A$ | | rule for conjunction |
| 4. | $\neg A$ | | rule for conjunction |

---

- if at any point we find that a path contains both a formula and its negation, we **close** the path with a **cross**
- so, for example, the path above contains both $A$ and $\neg A$.

| | | | |
|---|---|---|---|
| 1. | $((A \wedge \neg A) \vee B)$ ✓ | | assumption |
| 2. | $(A \wedge \neg A)$ ✓ | $B$ | rule for disjunction |
| 3. | $A$ | | rule for conjunction |
| 4. | $\neg A$ | | rule for conjunction |
| | $\otimes$ | | |

# applying rules on every open path

- suppose you have the following unfinished tree:

| | | | |
|---|---|---|---|
| 1. | $((A \to B) \wedge (B \to A))$ ✓ | | assumption |
| 2. | $(A \to B)$ ✓ | | rule for conjunction |
| 3. | $(B \to A)$ | | rule for conjunction |
| 4. | $\neg A$ | $B$ | rule for conditional |

- the rule for conditional has not been applied to the conditional on the third line
- there are two open paths through this tree
- the tree rules must be applied on all open paths on which the formula

it is being applied to is on.

- if you apply the rule correctly this is the tree you get:

| | | |
|---|---|---|
| 1. | $((A \to B) \wedge (B \to A))$ ✓ | assumption |
| 2. | $(A \to B)$ ✓ | rule for conjunction |
| 3. | $(B \to A)$ ✓ | rule for conjunction |
| 4. | $\neg A \qquad B$ | rule for conditional |
| 5. | $\neg B \quad A \quad \neg B \quad A$ | rule for conditional |
| | $\quad\;\; \otimes \quad \otimes$ | |

- it is important, however, to apply the rule only on the path on which the wff is on
- suppose you have this tree

| | | |
|---|---|---|
| 1. | $((A \vee B) \vee (C \vee D))$ ✓ | assumption |
| 2. | $(A \vee B) \quad (C \vee D)$ | rule for disjunction |

- the next step is not:

| | | |
|---|---|---|
| 1. | $((A \vee B) \vee (C \vee D))$ ✓ | assumption |
| 2. | $(A \vee B) \quad (C \vee D)$ ✓ | rule for disjunction |
| 3. | | |
| 4. | $C \quad D \qquad C \quad D$ | rule for disjunction |

- these are not on the same path:

| | | |
|---|---|---|
| 1. | $((A \vee B) \vee (C \vee D))$ ✓ | assumption |
| 2. | $(A \vee B) \quad \boxed{(C \vee D)\ ✓}$ | rule for disjunction |
| 3. | | |
| 4. | $\boxed{C} \quad \boxed{D} \qquad C \quad D$ | rule for disjunction |

- what is wrong with the following tree?

| | | |
|---|---|---|
| 1. | $((A \to B) \wedge (B \to A))$ ✓ | assumption |
| 2. | $\neg B$ | assumption |
| 3. | $(A \to B)$ ✓ | rule for conjunction |
| 4. | $(B \to A)$ ✓ | rule for conjunction |
| 5. | $\neg A \qquad B$ | rule for conditional |
| 6. | $\neg B \quad A \quad \neg B \quad A$ | rule for conditional |
| | $\quad\;\; \otimes \quad \otimes$ | |

- the path with B and ¬B should have been closed.

| | | |
|---|---|---|
| 1. | $((A \to B) \wedge (B \to A))$ ✓ | assumption |
| 2. | $\boxed{\neg B}$ | assumption |
| 3. | $(A \to B)$ ✓ | rule for conjunction |
| 4. | $(B \to A)$ ✓ | rule for conjunction |
| 5. | $\neg A \qquad \boxed{B}$ | rule for conditional |
| 6. | $\neg B \quad A \quad \neg B \quad A$ | rule for conditional |
| | $\quad\; \otimes \quad \otimes$ | |

---

- it should have looked like this:

| | | |
|---|---|---|
| 1. | $((A \to B) \wedge (B \to A))$ ✓ | assumption |
| 2. | $\neg B$ | assumption |
| 3. | $(A \to B)$ ✓ | rule for conjunction |
| 4. | $(B \to A)$ ✓ | rule for conjunction |
| 5. | $\neg A \qquad B$ | rule for conditional |
| 6. | $\neg B \quad A \qquad \otimes$ | rule for conditional |
| | $\qquad \otimes$ | |

- every time you add wffs to the tree you *must* check for closure
- this applies when you first write down your assumptions, and applies again whenever you apply a rule

# the order of the rules

## the order of the rules

- it doesn't matter which order you apply the rules
- however, it is often convenient to apply non-branching rules first whenever you have a choice between non-branching and branching rules
- to see why, suppose we have:

| | | |
|---|---|---|
| 1. | $(A \vee B)$ | assumption |
| 2. | $(A \to B)$ | assumption |

---

- branching first:

| | | |
|---|---|---|
| 1. | $(A \wedge B)$ ✓ | assumption |
| 2. | $(A \to B)$ ✓ | assumption |
| 3. | $\neg A \quad B$ | rule for conditional |
| 4. | $A \quad A$ | |
| 5. | $B \quad B$ | rule for conjunction |
| | $\otimes$ | |

- non-branching first:

| | | |
|---|---|---|
| 1. | $(A \land B)$ ✓ | assumption |
| 2. | $(A \to B)$ ✓ | assumption |
| 3. | $A$ | |
| 4. | $B$ | rule for conjunction |
| 5. | $\neg A \qquad B$ | rule for conditional |
| | $\otimes$ | |

- there's only a small difference here (2 wffs to be exact)
- but the difference can be significant with bigger trees
- another good idea is to apply rules which will give you a closed branch first
- but this requires some foresight

---

- want to build trees which look exactly like the trees which the program I use to generate answers look?
- mechanically follow these steps
  - **step 1.** from the top down, apply double negation rule and check for contradictions, until you can't. go to step 2.
  - **step 2.** from the top down, apply any non-branching rule, check for contradictions, and do step 1, until you can't. (conjunction first, negated disjunction next, then negated conditional). go to step 3.
  - **step 3.** from the top down, apply a branching rule if you can, check for contradictions, and return to step 1. (disjunction first, conditional next, biconditional, then negated biconditional). if you can't apply a branching rule (or any other rules for that matter), you are done!

---

- if you do this then . . .
  - the input and output of the double negation rule will always be together
  - you will never miss a contradiction (think of checking for contradictions as part of what you do when you apply a rule)
  - you will never apply a branching rule before you could have applied a non-branching rule

# what trees do

## what trees do

- the rules specify things to write which must be true assuming that the propositions at the top of the tree are true
- the rules are constructed so that if the propositions at the top of the tree are true, then there is at least one path such that everything on that path is true

---

- each path represents an alleged way for the propositions at the top all to be true together: an alleged assignment of truth values to basic propositions that makes the propositions at the top true
- this is only an alleged way for them to be true together, since some paths close
- a closed path does not represent a possible assignment of values at all, while an open path represents a possible assignment of values on which the propositions at the top are all true

- suppose that all paths close
    - then the propositions at the top are unsatisfiable: there is no way of making them all true
- but suppose a path remains open: then the propositions at the top are satisfiable: there is an assignment of truth values to basic propositions that makes them all true
- this is what allows us to “read off’ an assignment of values to basic propositions on which our initial propositions are all true

# testing for satisfiability

## testing for satisfiability

- in the first instance, trees allow us to test whether a given set of propositions is satisfiable or not
- if we know how to set things up properly, we can use trees to test for other properties too

---

To prove: whether $\{P, \neg P\}$ is satisfiable.

| | | |
|---|---|---|
| 1. | $P$ | assumption |
| 2. | $\neg P$ | assumption |

- if all of the paths close, then the propositions are not jointly satisfiable
- if a path remains open, then the propositions are jointly satisfiable

---

To prove: whether $\{P, \neg P\}$ is satisfiable.

| | | |
|---|---|---|
| 1. | $P$ | assumption |
| 2. | $\neg P$ | assumption |
| | $\otimes$ | |

- all paths close, so the propositions are not jointly satisfiable

---

To prove: whether $\{P, P \to Q, \neg Q\}$ is satisfiable.

| | | |
|---|---|---|
| 1. | $P$ | assumption |
| 2. | $P \to Q$ | assumption |
| 3. | $\neg Q$ | assumption |

---

To prove: whether $\{P, P \to Q, \neg Q\}$ is satisfiable.

| | | | |
|---|---|---|---|
| 1. | $P$ | | assumption |
| 2. | $P \to Q$ ✓ | | assumption |
| 3. | $\neg Q$ | | assumption |
| 4. | $\neg P$ | $Q$ | |

---

To prove: whether $\{P, P \to Q, \neg Q\}$ is satisfiable.

| | | |
|---|---|---|
| 1. | $P$ | assumption |
| 2. | $P \to Q$ ✓ | assumption |
| 3. | $\neg Q$ | assumption |
| 4. | $\neg P$ $\quad$ $Q$ | |
| | $\otimes$ $\quad$ $\otimes$ | |

- all paths close, so the propositions are not jointly satisfiable

---

To prove: whether $\{P \to Q, P \wedge Q\}$ is satisfiable.

| | | |
|---|---|---|
| 1. | $P \to Q$ | assumption |
| 2. | $P \wedge Q$ | assumption |

---

To prove: whether $\{P \to Q, P \wedge Q\}$ is satisfiable.

| | | |
|---|---|---|
| 1. | $P \to Q$ ✓ | assumption |
| 2. | $P \wedge Q$ ✓ | assumption |
| 3. | $P$ | |
| 4. | $Q$ | |
| 5. | $\neg P$ $\quad$ $Q$ | |
| | $\otimes$ | |

- the tree is finished and there is an open path

---

To prove: whether $\{P \to Q, P \wedge Q\}$ is satisfiable.

| | | |
|---|---|---|
| 1. | $P \to Q$ ✓ | assumption |
| 2. | $P \wedge Q$ ✓ | assumption |
| 3. | $\boxed{P}$ | |
| 4. | $\boxed{Q}$ | |
| 5. | $\neg P$ $\quad$ $\boxed{Q}$ | |
| | $\otimes$ | |

- we read off an assignment on which the propositions are true. $P$: T, $Q$: T

---

To prove: whether $\{A \vee B, B \wedge C\}$ is satisfiable.

| | |
|---|---|
| 1. | $A \vee B$ |
| 2. | $B \wedge C$ |

---

To prove: whether $\{A \vee B, B \wedge C\}$ is satisfiable.

| | |
|---|---|
| 1. | $A \vee B$ |
| 2. | $B \wedge C$ ✓ |
| 3. | $B$ |
| 4. | $C$ |

---

To prove: whether $\{A \vee B, B \wedge C\}$ is satisfiable.

1. $A \vee B$ ✓
2. $B \wedge C$ ✓
3. $B$
4. $C$
5. $A \quad B$

---

To prove: whether $\{A \vee B, B \wedge C\}$ is satisfiable.

1. $A \vee B$ ✓
2. $B \wedge C$ ✓
3. $\boxed{B}$
4. $\boxed{C}$
5. $\boxed{A} \quad B$

- `left path yields:` $A$`: T,` $B$`: T,` $C$`: T`

---

To prove: whether $\{A \vee B, B \wedge C\}$ is satisfiable.

1. $A \vee B$ ✓
2. $B \wedge C$ ✓
3. $\boxed{B}$
4. $\boxed{C}$
5. $A \quad \boxed{B}$

- `right path yields:` $B$`: T,` $C$`: T.`
- `either:` $B$`: T,` $C$`: T,` $A$`: T or` $B$`: T,` $C$`: T,` $A$`: F.`
- `so one tree can yield multiple assignments`

---

To prove: whether $\{A \leftrightarrow \neg B, A \vee \neg B\}$ is satisfiable.

1. $A \leftrightarrow \neg B$
2. $A \vee \neg B$

---

To prove: whether $\{A \leftrightarrow \neg B, A \vee \neg B\}$ is satisfiable.

1. $A \leftrightarrow \neg B$ ✓
2. $A \vee \neg B$
3. $A \quad \neg A$
4. $\neg B \quad \neg\neg B$

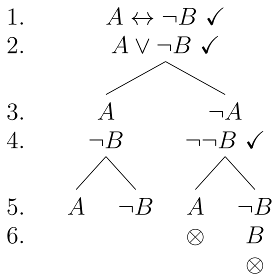
---

To prove: whether $\{A \leftrightarrow \neg B, A \vee \neg B\}$ is satisfiable.

1. $A \leftrightarrow \neg B$ ✓
2. $A \vee \neg B$ ✓
3. $A \quad \neg A$
4. $\neg B \quad \neg\neg B$
5. $A \quad \neg B \quad A \quad \neg B$

---

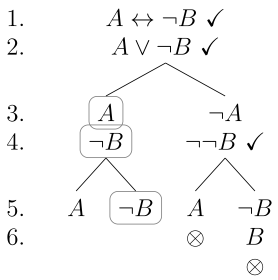
To prove: whether $\{A \leftrightarrow \neg B, A \vee \neg B\}$ is satisfiable.




- there are two open paths

---

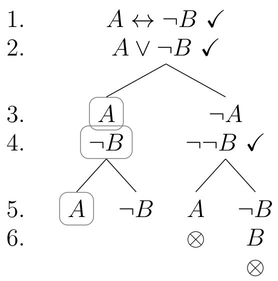
To prove: whether $\{A \leftrightarrow \neg B, A \vee \neg B\}$ is satisfiable.




- this path yields: $A$: T, $B$: F

---

To prove: whether $\{A \leftrightarrow \neg B, A \vee \neg B\}$ is satisfiable.




- this path yields: $A$: T, $B$: F
- so different paths can yield the same assignment

---

- one tree can yield multiple assignments
- different paths can yield the same assignment
- every truth-making assignment, if there are any at all, can be read of some path
  - none are overlooked

# testing for validity

## testing for validity

- validity: is it possible for the premises to be true and the conclusion false?

- is it possible for the premises to be true and the negation of the conclusion to be true?
- is the set containing the premises and the negation of the conclusion satisfiable?

---

To prove: whether $(A \to B), (B \to C) \therefore (A \to C)$ is valid.

| | | |
|---|---|---|
| 1. | $(A \to B)$ | assumption |
| 2. | $(B \to C)$ | assumption |
| 3. | $\neg(A \to C)$ | negation of Conclusion |

- if all of the paths close, then the argument if valid.
- if any paths remain open, then the argument is invalid.

---

- as with truth tables, if an argument is invalid, we can read off a counterexample
- the counterexample is provided by the basic propositions and negations of basic propositions on an open path
- we assign T to the basic propositions on an open path, and F to the basic proposition which is the negand of the negation on an open path

---

To prove: whether $(A \to B), (B \to C) \therefore (B \to A)$ is valid.

| | | |
|---|---|---|
| 1. | $(A \to B)$ | assumption |
| 2. | $(B \to C)$ | assumption |
| 3. | $\neg(B \to A)$ ✓ | negation of Conclusion |
| 4. | $B$ | |
| 5. | $\neg A$ | rule for negated conditional |

---

To prove: whether $(A \to B), (B \to C) \therefore (B \to A)$ is valid.

| | | | |
|---|---|---|---|
| 1. | $(A \to B)$ ✓ | | assumption |
| 2. | $(B \to C)$ | | assumption |
| 3. | $\neg(B \to A)$ ✓ | | negation of Conclusion |
| 4. | $B$ | | |
| 5. | $\neg A$ | | rule for negated conditional |
| 6. | $\neg A$ | $B$ | rule for conditional |

---

To prove: whether $(A \to B), (B \to C) \therefore (B \to A)$ is valid.

| | | | | | |
|---|---|---|---|---|---|
| 1. | $(A \to B)$ ✓ | | | | assumption |
| 2. | $(B \to C)$ ✓ | | | | assumption |
| 3. | $\neg(B \to A)$ ✓ | | | | negation of Conclusion |
| 4. | $B$ | | | | |
| 5. | $\neg A$ | | | | rule for negated conditional |
| 6. | $\neg A$ | | $B$ | | rule for conditional |
| 7. | $\neg B$ | $C$ | $\neg B$ | $C$ | rule for conditional |
| | $\otimes$ | | $\otimes$ | | |

To prove: whether $(A \to B), (B \to C) \therefore (B \to A)$ is valid.

| | | |
|---|---|---|
| 1. | $(A \to B)$ ✓ | assumption |
| 2. | $(B \to C)$ ✓ | assumption |
| 3. | $\neg(B \to A)$ ✓ | negation of Conclusion |
| 4. | $\boxed{B}$ | |
| 5. | $\boxed{\neg A}$ | rule for negated conditional |
| 6. | $\neg A$     $\boxed{B}$ | rule for conditional |
| 7. | $\neg B$   $C$   $\neg B$   $\boxed{C}$ | rule for conditional |
| | $\otimes$      $\otimes$ | |

- a counterexample: $B$: true, $C$: true, $A$: false

---

- the relevant row of the truth table looks like this:

| $A$ | $B$ | $C$ | $(A \to B)$ | $(B \to C)$ | $(B \to A)$ |
|---|---|---|---|---|---|
| F | T | T | T | T | F |

# testing for contraries and contradictories

## testing for contraries and contradictories

To prove: whether $\{\alpha, \beta\}$ are contraries or contradictories. Step 1.

| | | |
|---|---|---|
| 1. | $\alpha$ | assumption |
| 2. | $\beta$ | assumption |

- if the propositions are jointly satisfiable, they are not contraries or contradictories
- if they are not jointly satisfiable, they are either contraries or contradictories, but we need a further test to determine which

---

To prove: whether $\{\alpha, \beta\}$ are contraries or contradictories. Step 2.

| | | |
|---|---|---|
| 1. | $\neg\alpha$ | negation of $\alpha$ |
| 2. | $\neg\beta$ | negation of $\beta$ |

- if the propositions are jointly satisfiable then they are contraries
- if they are not jointly satisfiable, then they are contradictories

# testing for tautologies and logical truths

## testing for tautologies and logical truths

- tautology: is it possible for the proposition to be false?
- is it possible for the negation of the proposition to be true?
- is the set containing the negation of the proposition satisfiable?
  - if it is not, then it is a tautology

---

To prove: whether $\alpha$ is a logical truth.

1. $\neg\alpha$ negation of Proposition to be Tested

- if the negation of the proposition is not satisfiable (if all paths close), then the proposition itself is a logical truth.

# testing for equivalence

## testing for equivalence

- equivalent: is it possible for one proposition to be true while the other is false or vice versa?
- is the biconditional constructed from the propositions a tautology?
- is the set containing the negation of the biconditional constructed from the propositions unsatisfiable? If it is, then they are equivalent

---

To prove: whether $\alpha$ and $\beta$ are equivalent.

1. $\neg(\alpha \leftrightarrow \beta)$ negated biconditional formed from $\alpha$ and $\beta$

- if the negated biconditional is not satisfiable (all paths close) then the propositions are equivalent.

# wrapping up

## this lecture

- trees can be used to test for various properties
- once you know how to build trees, all you need to know is how to set them up the right way to test for various properties, and how to understand the results of your test
- we've looked at a few examples in this lecture
- you will do many more in tutorials

## next lectures

- lecture 12, issues in translation: conditional
- lecture 13, the formal language MPL