lecture 17, trees for MPL

phil1012 introductory logic

overview

this lecture

- an introduction to trees for MPL
- motivation for MPL truth trees
- motivation for new rules
- constructing MPL truth trees
- finished trees, closed paths, and saturation

learning outcomes

- after doing the relevant reading for this lecture, listening to the lecture, and attending the relevant tutorial, you will be able to:
  - explain the general rationale behind trees for MPL
  - explain the particular rationale behind each tree rule for MPL
  - explain what it means to say that a path is saturated
  - construct trees using the tree rules for MPL
  - use trees to test for various logical properties of MPL formulas
  - read off (counter)models from open paths of MPL trees

required reading

- section 10.1 of chapter 10

tables and trees

tables and trees

- recall that in PL we had two methods of proof: truth tables and truth trees
- in MPL we have only one: truth trees
- the tree rules for MPL are similar to those for PL
- but there are additional rules for the quantifiers
- and the rationale appeals to models
- let’s start by running through the rationale for the rules

rationale for tree rules

rationale for tree rules

- the general rationale for the tree rules:
  - the rules prescribe propositions that must be true, given that what we have already written down is true

---

- in the non-branching case:

- if there is a model in which every proposition on some old path is true, then there is a model in which every proposition on the new path is true.

- in the branching case:
- if there is a model in which every proposition on some old path is true, then there is a model in which every proposition on at least one new path is true.

- all of the rules from PL have the relevant property. Take the rules for negated disjunction (a non-branching rule) and disjunction (a branching rule) for example.

the rationale for the rule for negated disjunction

- rule for negated disjunction:

$$\begin{array}{c} \neg(\alpha \vee \beta) \checkmark \\ \neg\alpha \\ \neg\beta \end{array}$$

- rationale:
  - suppose there is some model in which $\lnot (\alpha \lor \beta)$ is true
  - it follows by rule (2) that $(\alpha \lor \beta)$ is false on this model
  - it follows by rule (4) that α and β are both false on this model
  - it follows by rule (2) that $\lnot \alpha$ and $\lnot \beta$ are both true on this model
  - so if there is a model in which every proposition on some old path is true *before* this rule is applied, then there is a model in which every proposition on the new path is true *after* this rule has been applied

the rationale for the rule for disjunction

- rule for disjunction:

$$\begin{array}{c} (\alpha \vee \beta) \checkmark \\ \diagup \quad \diagdown \\ \alpha \qquad \beta \end{array}$$

- rationale:
  - suppose there is a model in which $(\alpha \lor \beta)$ is true
  - it follows by rule (4) that either α is true or β is true in this model
  - so if there is a model in which every proposition on some old path is true before this rule is applied, then there is a model in which every proposition on at least one new path is true after this rule has been applied

rules for quantifiers

rules for quantifiers

- we now introduce four new rules for the quantifiers.
  - the rule for negated existential quantifier.
  - the rule for negated universal quantifier.
  - the rule for existential quantifier.
  - the rule for universal quantifier.
- the application of the rules for the negated quantifiers is the same as for the other rules.
- the unnegated quantifiers require special treatment.

rule for negated existential quantifier

$\lnot\exists\underline{x}\alpha(\underline{x})$ ✓
$\forall\underline{x}\lnot\alpha(\underline{x})$

- rationale:
  - suppose that $\lnot\exists\underline{x}\alpha(\underline{x})$ is true in some model $\mathcal{M}$
  - it follows from this that $\exists\underline{x}\alpha(\underline{x})$ is false in $\mathcal{M}$
  - it follows from this that there is no model just like $\mathcal{M}$ except that it also assigns a referent to $\underline{a}$—where is $\underline{a}$ is some name to which $\mathcal{M}$ assigns no referent—in which $\alpha(\underline{a}/\underline{x})$ is true
  - in other words, $\alpha(\underline{a}/\underline{x})$ is false in every model just like $\mathcal{M}$ except that it also assigns a referent to $\underline{a}$
  - it follows from this that $\lnot\alpha(\underline{a}/\underline{x})$ is true on every model just like $\mathcal{M}$ except that it also assigns a referent to $\underline{a}$
  - it follows from this that $\forall\underline{x}\lnot\alpha(\underline{x})$ is true in $\mathcal{M}$

rule for negated universal quantifier

$\lnot\forall\underline{x}\alpha(\underline{x})$ ✓
$\exists\underline{x}\lnot\alpha(\underline{x})$

- rationale:
  - suppose that $\lnot\forall\underline{x}\alpha(\underline{x})$ is true in some model $\mathcal{M}$
  - it follows from this that $\forall\underline{x}\alpha(\underline{x})$ is false in $\mathcal{M}$
  - it follows from this that there is some model just like $\mathcal{M}$ except that it also assigns a referent $\underline{a}$—where $\underline{a}$ is some name to which $\mathcal{M}$ assign no referent—in which α(a_$\underline{a}$/ x_$\underline{x}$) is false
  - it follows from this that $\lnot\alpha$(a_$\underline{a}$/

$\underline{x}$) is true in $\mathcal{M}$. It follows that $\exists\underline{x}\lnot\alpha(\underline{x})$ is true in $\mathcal{M}$

rule for existential quantifier

$$\begin{array}{c} \exists\underline{x}\alpha(\underline{x}) \checkmark \underline{a} \\ \alpha(\underline{a}/\underline{x}) \end{array}$$

- note 1: at the time of applying the rule at the bottom of some path, the name $\underline{a}$ used in applying the rule must be one that has not yet appeared anywhere on the path.
- note 2: when applying the rule and checking off the formula, we write the name we have used in applying the rule next to the check mark.
- let's look at a couple of examples . . .

- suppose we have:

$$\begin{array}{c} Fa \\ Gb \\ \exists xHx \end{array}$$

- the following is a correct application of the rule for the existential quantifier:

$$\begin{array}{c} Fa \\ Gb \\ \exists xHx \checkmark c \\ Hc \end{array}$$

- the name c has not been used on the path

- the following is an incorrect application of the rule for the existential quantifier:

$$\begin{array}{c} Fa \\ Gb \\ \exists xHx \checkmark b \\ Hb \end{array}$$

- the name b has already been used on the path

- okay, now consider another example

- suppose we have:

$$\begin{array}{c} (\exists xFx \lor \exists xGx) \checkmark \\ \begin{array}{cc} \exists xFx & \exists xGx \end{array} \end{array}$$

- we apply the rule for the existential quantifier once:

$(\exists xFx \vee \exists xGx)$ ✓

$\exists xFx$ ✓ a $\quad$ $\exists xGx$

Fa

- we apply the rule again:

  $(\exists xFx \vee \exists xGx)$ ✓

  $\exists xFx$ ✓ a $\quad$ $\exists xGx$ ✓ a

  Fa $\quad\quad$ Ga

- this is okay because the name has not been used *on this path.*

- okay, now let’s look at the rationale for the rule for the existential quantifier

---

- rationale:
  - suppose there is a model 𝓜\mathcal{M} in which ∃x_α(x_)\exists\underline{x}\alpha(\underline{x}) is true.
  - it follows from this that there is at least one object oo in the domain of 𝓜\mathcal{M} such that α(a_/x_)\alpha(\underline{a}/\underline{x}) is true in 𝓜oa_\mathcal{M}^\underline{a}_o, where a_\underline{a} is some name not assigned a referent in 𝓜\mathcal{M}, and 𝓜oa_\mathcal{M}^\underline{a}_o is a model that is just like 𝓜\mathcal{M} except that in it the name a_\underline{a} is assigned the referent oo.
  - so, if there is a model 𝓜\mathcal{M} in which ∃x_α(x_)\exists\underline{x}\alpha(\underline{x}) is true, then there is a different model 𝓜oa_\mathcal{M}^\underline{a}_o in which α(a_/x_)\alpha(\underline{a}/\underline{x}) is true ∃x_α(x_)\exists\underline{x}\alpha(\underline{x}) will be true on 𝓜oa_\mathcal{M}^\underline{a}_o.
  - so, if there is a model 𝓜\mathcal{M} in which ∃x_α(x_)\exists\underline{x}\alpha(\underline{x}) is true, then there is a different model 𝓜oa_\mathcal{M}^\underline{a}_o in which α(a_/x_)\alpha(\underline{a}/\underline{x}) is true, and ∃x_α(x_)\exists\underline{x}\alpha(\underline{x}) is true.

rule for universal quantifier:

$\forall \underline{x}\alpha(\underline{x})$ \\ $\underline{a}$

$\alpha(\underline{a}/\underline{x})$

- note 1: the name a_\underline{a} does not have to be new on the path
- note 2: when applying the rule, we write a backslash, not a check mark—and we write the name used in applying the rule next to the backslash

---

- note 3: we can apply the rule multiple times

$$\begin{array}{l}\forall \underline{x}\alpha(\underline{x}) \setminus \underline{a}, \underline{b}, \underline{c} \\ \quad \alpha(\underline{a}/\underline{x}) \\ \quad \alpha(\underline{b}/\underline{x}) \\ \quad \alpha(\underline{c}/\underline{x})\end{array}$$

- okay, let’s go over the rationale for the rule for the universal quantifier

---

- rationale (case 1):
  - suppose that the name a_\underline{a} used in applying the rule is new to the path.
  - if there is a model ℳ\mathcal{M} in which ∀x_α(x_)\forall \underline{x}\alpha(\underline{x}) is true, then for every object oo in the domain of ℳ\mathcal{M}, α(a_/x_)\alpha(\underline{a}/\underline{x}) is true in ℳoa_\mathcal{M}^\underline{a}_o where a_\underline{a} is our new name (which is not assigned a referent in ℳ\mathcal{M}), and ℳoa_\mathcal{M}^\underline{a}_o is a model that is just like ℳ\mathcal{M} except that in it the name a_\underline{a} is assigned the referent oo
  - if ∀x_α(x_)\forall \underline{x}\alpha(\underline{x}) is true in ℳ\mathcal{M}, then ∀x_α(x_)\forall \underline{x}\alpha(\underline{x}) is true in ℳoa_\mathcal{M}^\underline{a}_o.
  - it follows that if there is a model in which ∀x_α(x_)\forall \underline{x}\alpha(\underline{x}) is true, then there is a model on which ∀x_α(x_)\forall \underline{x}\alpha(\underline{x}) is true and α(a_/x_)\alpha(\underline{a}/\underline{x}) is true.

- rationale (case 2):
  - suppose that the name a_\underline{a} used in applying the rule is not new to the path.
  - if there is a model ℳ\mathcal{M} in which ∀x_α(x_)\forall \underline{x}\alpha(\underline{x}) is true, then for every object oo in the domain of ℳ\mathcal{M}, α(d_/x_)\alpha(\underline{d}/\underline{x}) is true in ℳod_\mathcal{M}^\underline{d}_o where d_\underline{d} is a new name (which is not assigned a referent in ℳ\mathcal{M}), and ℳod_\mathcal{M}^\underline{d}_o is a model that is just like ℳ\mathcal{M} except that in it the name d_\underline{d} is assigned the referent oo
  - in this case, a_\underline{a} is (already) assigned to a referent in ℳ\mathcal{M}—suppose it is assigned to the object k
  - we have just seen that α(d_/x_)\alpha(\underline{d}/\underline{x}) is true in every model just like ℳ\mathcal{M} except that it assigns a referent to d_\underline{d}
  - so α(d_/x_)\alpha(\underline{d}/\underline{x}) is true in ℳkd_\mathcal{M}^\underline{d}_k, the model that assigns d_\underline{d}’s referent to k
  - but then α(a_/x_)\alpha(\underline{a}/\underline{x}) must be true in ℳ\mathcal{M}, because a_\underline{a}’s referent is the same object k

---

- okay, so much for the motivation for the rules for the existential quantifier and the universal quantifier, now let's take a look at trees for MPL in their entirety

tree rules for MPL

tree rules for MPL

- the tree rules for MPL consist of:
  - the tree rules from PL.
  - tree rules for disjunction, negated disjunction, conjunction, negated conjunction, conditional, negated conditional, biconditional, negated biconditional, and double negation are the same as in PL
- and the tree rules for the quantifiers:

- rule for existential quantifier

$$\begin{array}{ll} \exists \underline{x}\alpha(\underline{x}) \checkmark \underline{a} & \quad (\text{new } \underline{a}) \\ \quad \alpha(\underline{a}/\underline{x}) & \end{array}$$

- rule for negated existential quantifier

$$\begin{array}{l} \neg\exists \underline{x}\alpha(\underline{x}) \checkmark \\ \quad \forall \underline{x}\neg\alpha(\underline{x}) \end{array}$$

- rule for universal quantifier

$$\begin{array}{ll} \forall \underline{x}\alpha(\underline{x}) \setminus \underline{a} & \quad (\text{any } \underline{a}) \\ \quad \alpha(\underline{a}/\underline{x}) & \end{array}$$

- rule for negated universal quantifier

$$\begin{array}{l} \neg\forall \underline{x}\alpha(\underline{x}) \checkmark \\ \quad \exists \underline{x}\neg\alpha(\underline{x}) \end{array}$$

recommended order of application

- there's a recommended order of application for the rules:
  - rules from PL. non-branching first
  - rules for negated quantifiers
  - rule for (unnegated) existential quantifier
  - rule for (unnegated) universal quantifier

- that's it. those are the four new rules, and notes on their application.
- now we've got to say something about finished trees

finished trees, saturated paths

finished trees, saturated paths

- because we can go on applying the rule for the universal quantifier as many times as we like, we need to say something about when our trees are finished
- to do so we appel to the idea of a saturated path
- a tree is **finished** if each of its paths is either **closed** or **saturated**.

- a path is **saturated** if and only if
  - every formula on it—apart from atomic formulas, negations of atomic formulas, and formulas whose main operator is a universal quantifier—has had the relevant rule applied; and
  - every formula on it whose main operator is a universal quantifier
    - has had the universal quantifier rule applied to it at least once, and
    - has had the rule applied to it once for each name that appears on the path.

- let's consider an example

- suppose we have:

$$\begin{array}{c} \exists xFx \\ \forall xHx \\ Gb \end{array}$$

- we apply the existential quantifier rule to get:

$$\begin{array}{c} \exists xFx \checkmark a \\ \forall xHx \\ Gb \\ Fa \end{array}$$

- then we apply the universal quantifier rule to get:

$$\begin{array}{c} \exists xFx \checkmark a \\ \forall xHx \backslash a \\ Gb \\ Fa \\ Ha \end{array}$$

- this tree is unfinished!
- why? because there is a name no the path, bb in this case, which we have not yet used in applying the universal quantifier rule

---

- so we apply the universal quanfitier rule using the name bb to get:

$$\begin{array}{c} \exists xFx \checkmark a \\ \forall xHx \backslash a,b \\ Gb \\ Fa \\ Ha \\ Hb \end{array}$$

- this tree is finished!
- why? because we’ve applied the universal quantifier rule at least once, and we have applied it using every name on the path

---

- don’t worry if you haven’t fully understood the idea of saturating a path
- we’ll do plenty of practice in the live lecture and in the tutorials

wrapping up

this lecture

- introducting the tree rules for MPL
- the rationale for the rules
- finished trees, saturated paths

next lecture

- lecture 18, uses of trees for MPL