lecture 20, the semantics of GPL and trees for GPL

phil1012 introductory logic

overview

this lecture

- the semantics of GPL and trees for GPL
- the notion of sets of ordered n-tuples as extensions of n-place predicates in GPL

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 how the truth values of atomic formulas are determined on a model in GPL
  - determine the truth value of a GPL proposition on a model
  - find a model on which a GPL proposition is true and/or a model on which a GPL proposition is false
  - construct trees using the tree rules for GPL
  - use trees to test for various logical properties of GPL formulas
  - read off (counter)models from open paths of GPL trees

required reading

- sections 12.3 and 12.3 of chapter 12

the semantics of GPL

the semantics of GPL

- recall that the semantics for MPL proceeded in two stages.
- in the first stage we say what the values of nonlogical symbols are. A *model* is then *any* assignment of values to nonlogical symbols (together with a domain from which these values are drawn.
- in the second stage we specify how the truth values of all propositions (closed wffs) are determined, given values for their nonlogical components.

- in moving from MPL to GPL we have simply added many-place predicates.
- we need to say what the value of a many-place predicate is.
- the rest of the semantics is just like that of MPL.

- in MPL the extensions of one-place predicates were sets.
- why can’t we use a set like {Bill, Bob} for the extension of the two-place predicate L2L^2?

- because {Bill, Bob} and {Bob, Bill} are the same set and we want to be about to say that Bill loves Bob without saying that Bob loves Bill.

- **extensionality of sets**: If everything in A is in B and vice versa then A = B.
- {1, 2, 3} = {2, 1, 3} and {1, 2} = {1, 1, 2}.

- to solve this problem we introduce a new notion which indicates a group of things *taken in order*.
- ⟨\langle 1, 2, 3⟩\rangle ≠\neq ⟨\langle 2, 1, 3⟩\rangle and ⟨\langle 1, 2⟩\rangle ≠\neq ⟨\langle 1, 1, 2⟩\rangle.

- an ordered nn-tuple is an ordered list of things with nn positions in the list 1st through nnth.
- there need not be nn distinct things involved: the very same object might occupy more than one place in the list.
- an ordered 2-tuple is called an ordered pair and an ordered 3-tuple is called an ordered triple.

- the **extension** of an nn-place predicate is a set of ordered nn-tuples of members of the domain
- our new semantic clause looks like this:

1. Pna1…an_\underline{P^{n}a_{1}} \ldots \underline{a_n} is true in 𝓜\mathcal{M} iff the ordered *n*-tuple consisting of the referents in 𝓜\mathcal{M} of a1_\underline{a_{1}} through an_\underline{a_{n}} in that order is in the extension in 𝓜\mathcal{M} of Pn_\underline{P^{n}}.

- let’s work through some examples to get a feel for the semantics of GPL

- our model:
  - domain: {Alice, Bob, Carol, Dave, Edwina}
  - referents: aa: Alice, bb: Bob, cc: Carol, dd: Dave, ee: Edwina
  - extensions: MM: {Bob, Dave}, FF: {Alice, Carol, Edwina}, TT: {⟨\langleAlice, Bob⟩\rangle, ⟨\langleAlice, Carol⟩\rangle, ⟨\langleAlice, Dave⟩\rangle, ⟨\langleAlice, Edwina⟩\rangle, ⟨\langleBob, Carol⟩\rangle, ⟨\langle Bob, Dave⟩\rangle, ⟨\langleBob, Edwina⟩\rangle, ⟨\langleCarol, Dave⟩\rangle, ⟨\langleCarol, Edwina⟩\rangle, ⟨\langleDave, Edwina⟩\rangle}
- are the following formulas true or false in this model?
- TabTab
  - true. the referent of aa, namely Alice, and the referent of bb, namely Bob, taken in that order, are in the extension of TT.
- TbaTba
  - false. the referent of aa, namely Alice, and the referent of bb, namely Bob, taken in that order, are not in the extension of TT.
- TacTac
  - true. the referent of aa, namely Alice, and the referent of cc, namely Carol, taken in that order, are in the extension of TT.

- TceTce
  - true. the referent of cc, namely Carol, and the referent of ee, namely Edwina, taken in that order, are in the extension of TT.
- TaaTaa
  - false. the referent of aa, namely Alice, and the referent of aa, namely Alice, taken in that order, are not in the extension of TT.
- TeeTee
  - false. the referent of ee, namely Edwina, and the referent of ee, namely Edwina, taken in that order, are not in the extension of TT.
- TbeTbe
  - true. the referent of bb, namely Bob, and the referent of ee, namely Edwina, taken in that order, are in the extension of TT.

trees for GPL

trees for GPL

- in moving from MPL to GPL, we did not add any logical operators—so there are no new tree rules.
- the only change was that we added nn-place predicates—and in the semantics, we added extensions for these (sets of nn-tuples).
- so now all we need to do is see how to read off the extension of a many-place predicate from an open (saturated) path.

---

1. Fa
2. $\neg Fb$
3. Gc
4. Ga

- domain: {1, 2, 3}
- referents: a: 1, b: 2, c: 3
- extensions: F: {1}, G: {1, 3}

---

1. Taa
2. $\neg Tbc$
3. Tab

- domain: {1, 2, 3}
- referents: a: 1, b: 2, c: 3
- extensions: T: {⟨\langle 1,1 ⟩\rangle, ⟨\langle 1,2 ⟩\rangle}

---

1. $Babc$
2. $\neg Bccc$
3. $Bbcd$
4. $Baaa$

- domain: {1, 2, 3, 4}
- referents: a: 1, b: 2, c: 3, d: 4
- extensions: B: {⟨\langle 1, 2, 3 ⟩\rangle, ⟨\langle 2, 3, 4 ⟩\rangle, ⟨\langle 1, 1, 1 ⟩\rangle}

example 1

To prove: whether $\exists x \forall y Lxy \therefore \forall y \exists x Lxy$ is valid.

To prove: whether $\exists x \forall y Lxy \therefore \forall y \exists x Lxy$ is valid.

1. $\exists x \forall y Lxy$
2. $\neg \forall y \exists x Lxy$

To prove: whether $\exists x \forall y Lxy \therefore \forall y \exists x Lxy$ is valid.

1. $\exists x \forall y Lxy$
2. $\neg \forall y \exists x Lxy \checkmark$
3. $\exists y \neg \exists x Lxy$

To prove: whether $\exists x \forall y Lxy \therefore \forall y \exists x Lxy$ is valid.

1. $\exists x \forall y Lxy \checkmark a$
2. $\neg \forall y \exists x Lxy \checkmark$
3. $\exists y \neg \exists x Lxy$
4. $\forall y Lay$

To prove: whether $\exists x \forall y Lxy \therefore \forall y \exists x Lxy$ is valid.

1. $\exists x \forall y Lxy \checkmark a$
2. $\neg \forall y \exists x Lxy$
3. $\exists y \neg \exists x Lxy \checkmark b$
4. $\forall y Lay$
5. $\neg \exists y Lxb$

To prove: whether $\exists x \forall y Lxy \therefore \forall y \exists x Lxy$ is valid.

1. $\exists x \forall y Lxy \checkmark a$
2. $\neg \forall y \exists x Lxy \checkmark$
3. $\exists y \neg \exists x Lxy \checkmark b$
4. $\forall y Lay$
5. $\neg \exists y Lxb \checkmark$
6. $\forall x \neg Lxb$

To prove: whether $\exists x \forall y Lxy \therefore \forall y \exists x Lxy$ is valid.

1. $\exists x \forall y Lxy \checkmark a$
2. $\neg \forall y \exists x Lxy \checkmark$
3. $\exists y \neg \exists x Lxy \checkmark b$
4. $\forall y Lay \backslash b$
5. $\neg \exists y Lxb \checkmark$
6. $\forall x \neg Lxb$
7. Lab

To prove: whether $\exists x \forall y Lxy \therefore \forall y \exists x Lxy$ is valid.

1. $\exists x \forall y Lxy \checkmark a$
2. $\neg \forall y \exists x Lxy \checkmark$
3. $\exists y \neg \exists x Lxy \checkmark b$
4. $\forall x Lay \backslash b$
5. $\neg \exists y Lxb \checkmark$
6. $\forall x \neg Lxb \backslash a$
7. Lab
8. $\neg Lab$

To prove: whether $\exists x \forall y Lxy \therefore \forall y \exists x Lxy$ is valid.

1. $\exists x \forall y Lxy \checkmark a$
2. $\neg \forall y \exists x Lxy \checkmark$
3. $\exists y \neg \exists x Lxy \checkmark b$
4. $\forall x Lay \backslash b$
5. $\neg \exists y Lxb \checkmark$
6. $\forall x \neg Lxb \backslash a$
7. Lab
8. $\neg Lab$

$\otimes$

example 2

To prove: whether $\exists y \forall x Lxy \therefore \forall x \exists y Lxy$ is valid.

To prove: whether $\exists y \forall x Lxy \therefore \forall x \exists y Lxy$ is valid.

1. $\exists y \forall x Lxy$
2. $\neg \forall x \exists y Lxy$

To prove: whether $\exists y \forall x Lxy \therefore \forall x \exists y Lxy$ is valid.

1. $\exists y \forall x Lxy$
2. $\neg \forall x \exists y Lxy \checkmark$
3. $\exists x \neg \exists y Lxy$

To prove: whether $\exists y \forall x Lxy \therefore \forall x \exists y Lxy$ is valid.

1. $\exists y \forall x Lxy \checkmark a$
2. $\neg \forall x \exists y Lxy \checkmark$
3. $\exists x \neg \exists y Lxy$
4. $\forall x Lxa$

To prove: whether $\exists y \forall x Lxy \therefore \forall x \exists y Lxy$ is valid.

1. $\exists y \forall x Lxy \checkmark a$
2. $\neg \forall x \exists y Lxy \checkmark$
3. $\exists x \neg \exists y Lxy \checkmark b$
4. $\forall x Lxa$
5. $\neg \exists y Lby$

To prove: whether $\exists y \forall x Lxy \therefore \forall x \exists y Lxy$ is valid.

1. $\exists y \forall x Lxy \checkmark a$
2. $\neg \forall x \exists y Lxy \checkmark$
3. $\exists x \neg \exists y Lxy \checkmark b$
4. $\forall x Lxa$
5. $\neg \exists y Lby \checkmark$
6. $\forall y \neg Lby$

To prove: whether $\exists y \forall x Lxy \therefore \forall x \exists y Lxy$ is valid.

1. $\exists y \forall x Lxy$ ✓ a
2. $\neg \forall x \exists y Lxy$ ✓
3. $\exists x \neg \exists y Lxy$ ✓ b
4. $\forall x Lxa \backslash b$
5. $\neg \exists y Lby$ ✓
6. $\forall y \neg Lby$
7. Lba

To prove: whether $\exists y \forall x Lxy \therefore \forall x \exists y Lxy$ is valid.

1. $\exists y \forall x Lxy$ ✓ a
2. $\neg \forall x \exists y Lxy$ ✓
3. $\exists x \neg \exists y Lxy$ ✓ b
4. $\forall x Lxa \backslash b$
5. $\neg \exists y Lby$ ✓
6. $\forall y \neg Lby \backslash a$
7. Lba
8. $\neg Lba$

To prove: whether $\exists y \forall x Lxy \therefore \forall x \exists y Lxy$ is valid.

1. $\exists y \forall x Lxy$ ✓ a
2. $\neg \forall x \exists y Lxy$ ✓
3. $\exists x \neg \exists y Lxy$ ✓ b
4. $\forall x Lxa \backslash b$
5. $\neg \exists y Lby$ ✓
6. $\forall y \neg Lby \backslash a$
7. Lba
8. $\neg Lba$

$\otimes$

example 3

To prove: whether $\exists x(Hx \wedge Tax) \rightarrow \exists x(Ax \wedge Tax)$ a tautology.

To prove: whether $\exists x(Hx \wedge Tax) \rightarrow \exists x(Ax \wedge Tax)$ a tautology.

1. $\neg(\exists x(Hx \wedge Tax) \rightarrow \exists x(Ax \wedge Tax))$

To prove: whether $\exists x(Hx \wedge Tax) \rightarrow \exists x(Ax \wedge Tax)$ a tautology.

1. $\neg(\exists x(Hx \wedge Tax) \rightarrow \exists x(Ax \wedge Tax))$ ✓
2. $\exists x(Hx \wedge Tax)$
3. $\neg \exists x(Ax \wedge Tax)$

To prove: whether $\exists x(Hx \wedge Tax) \rightarrow \exists x(Ax \wedge Tax)$ a tautology.

1. $\neg(\exists x(Hx \wedge Tax) \rightarrow \exists x(Ax \wedge Tax))$ ✓
2. $\exists x(Hx \wedge Tax)$
3. $\neg \exists x(Ax \wedge Tax)$ ✓
4. $\forall x \neg(Ax \wedge Tax)$

To prove: whether $\exists x(Hx \wedge Tax) \rightarrow \exists x(Ax \wedge Tax)$ a tautology.

1. $\neg(\exists x(Hx \wedge Tax) \rightarrow \exists x(Ax \wedge Tax))$ ✓
2. $\exists x(Hx \wedge Tax)$ ✓b
3. $\neg\exists x(Ax \wedge Tax)$ ✓
4. $\forall x \neg(Ax \wedge Tax)$
5. $Hb \wedge Tab$

To prove: whether $\exists x(Hx \wedge Tax) \rightarrow \exists x(Ax \wedge Tax)$ a tautology.

1. $\neg(\exists x(Hx \wedge Tax) \rightarrow \exists x(Ax \wedge Tax))$ ✓
2. $\exists x(Hx \wedge Tax)$ ✓b
3. $\neg\exists x(Ax \wedge Tax)$ ✓
4. $\forall x \neg(Ax \wedge Tax)$
5. $Hb \wedge Tab$ ✓
6. Hb
7. Tab

To prove: whether $\exists x(Hx \wedge Tax) \rightarrow \exists x(Ax \wedge Tax)$ a tautology.

1. $\neg(\exists x(Hx \wedge Tax) \rightarrow \exists x(Ax \wedge Tax))$ ✓
2. $\exists x(Hx \wedge Tax)$ ✓b
3. $\neg\exists x(Ax \wedge Tax)$ ✓
4. $\forall x \neg(Ax \wedge Tax) \backslash b$
5. $Hb \wedge Tab$ ✓
6. Hb
7. Tab
8. $\neg(Ab \wedge Tab)$

To prove: whether $\exists x(Hx \wedge Tax) \rightarrow \exists x(Ax \wedge Tax)$ a tautology.

1. $\neg(\exists x(Hx \wedge Tax) \rightarrow \exists x(Ax \wedge Tax))$ ✓
2. $\exists x(Hx \wedge Tax)$ ✓b
3. $\neg\exists x(Ax \wedge Tax)$ ✓
4. $\forall x \neg(Ax \wedge Tax) \backslash b$
5. $Hb \wedge Tab$ ✓
6. Hb
7. Tab
8. $\neg(Ab \wedge Tab)$ ✓
9. $\neg Ab \qquad \neg Tab$

To prove: whether $\exists x(Hx \wedge Tax) \rightarrow \exists x(Ax \wedge Tax)$ a tautology.

1. $\neg(\exists x(Hx \wedge Tax) \rightarrow \exists x(Ax \wedge Tax))$ ✓
2. $\exists x(Hx \wedge Tax)$ ✓b
3. $\neg\exists x(Ax \wedge Tax)$ ✓
4. $\forall x \neg(Ax \wedge Tax) \backslash b$
5. $Hb \wedge Tab$ ✓
6. Hb
7. Tab
8. $\neg(Ab \wedge Tab)$ ✓
9. $\neg Ab \qquad \neg Tab$

$\qquad\qquad \otimes$

To prove: whether $\exists x(Hx \wedge Tax) \rightarrow \exists x(Ax \wedge Tax)$ a tautology.

1. $\neg(\exists x(Hx \wedge Tax) \rightarrow \exists x(Ax \wedge Tax))$ ✓
2. $\exists x(Hx \wedge Tax)$ ✓b
3. $\neg\exists x(Ax \wedge Tax)$ ✓
4. $\forall x\neg(Ax \wedge Tax) \backslash b, a$
5. $Hb \wedge Tab$ ✓
6. Hb
7. Tab
8. $\neg(Ab \wedge Tab)$ ✓
9. $\neg Ab$ $\qquad \neg Tab$
10. $\neg(Aa \wedge Taa)$ $\qquad \otimes$

To prove: whether $\exists x(Hx \wedge Tax) \rightarrow \exists x(Ax \wedge Tax)$ a tautology.

1. $\neg(\exists x(Hx \wedge Tax) \rightarrow \exists x(Ax \wedge Tax))$ ✓
2. $\exists x(Hx \wedge Tax)$ ✓b
3. $\neg\exists x(Ax \wedge Tax)$ ✓
4. $\forall x\neg(Ax \wedge Tax) \backslash b, a$
5. $Hb \wedge Tab$ ✓
6. Hb
7. Tab
8. $\neg(Ab \wedge Tab)$ ✓
9. $\neg Ab$ $\qquad \neg Tab$
10. $\neg(Aa \wedge Taa)$ ✓ $\qquad \otimes$
11. $\neg Aa$ $\qquad \neg Taa$

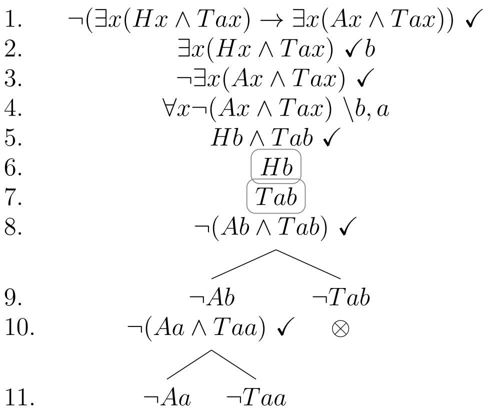
- domain:
- referents:
- extensions:

To prove: whether $\exists x(Hx \wedge Tax) \rightarrow \exists x(Ax \wedge Tax)$ a tautology.

1. $\neg(\exists x(Hx \wedge Tax) \rightarrow \exists x(Ax \wedge Tax))$ ✓
2. $\exists x(Hx \wedge Tax)$ ✓b
3. $\neg\exists x(Ax \wedge Tax)$ ✓
4. $\forall x\neg(Ax \wedge Tax) \backslash b, a$
5. $Hb \wedge Tab$ ✓
6. Hb
7. Tab
8. $\neg(Ab \wedge Tab)$ ✓
9. $\neg Ab$ $\qquad \neg Tab$
10. $\neg(Aa \wedge Taa)$ ✓ $\qquad \otimes$
11. $\neg Aa$ $\qquad \neg Taa$

- domain: {1, 2}
- referents: aa: 1, bb:2
- extensions:

To prove: whether $\exists x(Hx \wedge Tax) \rightarrow \exists x(Ax \wedge Tax)$ a tautology.



- domain: {1, 2}
- referents: aa: 1, bb:2
- extensions: HH: {2}, TT: { ⟨\langle 1, 2 ⟩\rangle }, AA: {}
- note: this handout contains a correction from the lecture regarding the extension of AA

wrapping up

this lecture

- trees for GPL
- reading models off open paths

next lecture

- lecture 21, the formal language GPLI