Philosophy 220

Trees for PL

Consistency on truth trees:

- A set of sentences is consistent if and only if there is at least one truth value assignment [of the constituents of the set of sentences] on which all the members of the set are true.
- We now know how to check for consistency using a tree, and can recover specific truth-value assignments on which all members of a given set come out true (if the set is consistent).

FLASHBACK !!!

Contradiction

Definition

- A sentence is a contradiction if and only if it is false on every possible truth-value assignment of its constituents.

Explained via consistency

- A sentence P, is a contradiction if and only if {P} is inconsistent.
- Since inconsistent sets are sets whose members can never all be true at the same time, and since {P} has only one member, that member must never be true for the set to be inconsistent.

Setting up a tree to check for contradiction

- A sentence **P** of PL is a contradiction if and only if {**P**} has a closed truth tree (meaning {**P**} is inconsistent).
- If the tree for {**P**} closes, it means that it is impossible for **P** to be true.

Tautology

Definition

- A sentence of PL is a tautology if and only if it is true on every possible truth-value assignment of its constituents.

Explained via Consistency

- A sentence P is a tautology if and only if {~P} is truth-functionally inconsistent.
- Since inconsistent sets are sets whose members can never all be true, and since {~P} has only one member and is a negation, P must be always true for ~P to be never true.

Setting up a tree to check for tautology

- A sentence **P** is a tautology if and only if {~**P**} has a closed truth tree (meaning {~**P**} is inconsistent).
- If the tree for {~**P**} closes, it means that it is impossible for ~**P** to be true, which in turn means it is impossible for **P** to be false.

Contingency

Definition

- A sentence is contingent if and only if it is neither a tautology nor a contradiction.

Explained via consistency

- A sentence P is contingent if and only if both {~P} and {P} are truth-functionally consistent.
- Since P is either a tautology or a contradiction if one of the above sets is inconsistent, P is contingent if it is neither a tautology nor a contradiction.

Setting up a tree to check for contingency

- A sentence **P** is contingent if and only if neither {~**P**} nor {**P**} has a closed truth tree.
- If the tree for {~**P**} is open and the tree for {**P**} is open, then it means that **P** can be either true or false.

Equivalence

Definition

- Sentences P and Q are equivalent if and only if there is no truth value assignment on which P and Q have different truth-values.

Explained via consistency

- Sentences P and Q are equivalent if and only if $\{\sim(P \equiv Q)\}$ is inconsistent
- Since only contradictions make sets of which they are sole members inconsistent, the negation of a sentence asserting that P and Q have different truth-values being contradictory means that P and Q <u>must</u> have the same truth-value.

Setting up a tree to check for equivalence

- Sentences P and Q are equivalent if and only if {~(P ≡ Q)} has a closed truth tree.
- If P and Q are equivalent, then (P ≡ Q) is a tautology because the two sentences always have the same truth-value. That would make ~(P ≡ Q) a contradiction.
- So to check for equivalence of any two sentences on a tree, join them with a biconditional, negate the biconditional, and check for consistency of the set with that negated biconditional as its only member.

Entailment

Definition

- A set Γ of sentences entails a sentence **P** if and only if there is no truth-value assignment on which every member of Γ is true and **P** is false.

Explained via Consistency

- $\Gamma \models$ **P** if and only if $\Gamma \cup \{\sim P\}$ is inconsistent.

Entailment on a tree

- A finite set Γ entails a sentence **P** if and only if the set $\Gamma \cup \{\sim$**P**$\}$ has a closed tree (is inconsistent).
- So to check if some finite set entails some sentence, represent each member of the set along with the negation of what you're checking to see whether the set entails.
- If the table closes, then it is impossible for all members of the set Γ to be true while **P** is false.

Validity

- Since validity is simply a special case of entailment, the same procedure can demonstrate that validity can be described in terms of consistency.
- If an argument is valid, then the union of the set of its premises and the negation of its conclusion will form an inconsistent set.

Validity on a tree

- An argument with a finite number of premises is valid if and only if the set consisting of all and only its premises and the negation of the conclusion has a closed tree.
- A tree that closes when you include every premise and the negation of the conclusion means that the conclusion cannot be false while the premises are true. (or else that the premises are inconsistent, in which case they entail anything)

New stuff:

- All of the previous things remain true in PL.
- We do, however, need a rule for decomposing Universal and Existential quantifiers.
- Those rules are a direct result of the semantics of quantified sentences.

∀D rule:

- ∀D is a non-branching rule:

M. $(\forall x)P$	--
N. $P(a/x)$	M, ∀D

Important: Universally quantified sentences are never checked off. Since every substitution instance of a true universally quantified statement is true, an infinite number of substitution instances can be decomposed into the tree.

In general, we will extract one substitution instance of a universal for every individual constant in the branch.

∃D rule:

- ∃D is a non-branching rule

M. (∃x)P	√	--
N. P(a/x)		M, ∃D

Important: a must be a constant that is foreign to the branch in which the existentially quantified sentence is decomposed. When an existentially quantified sentence is true, we are entitled only to the truth of a single substitution instance of it, and we are not entitled to infer any other properties of a while we are at it.

Negated Quantifiers

- We can’t really do anything with negated quantifiers, so we just convert them to equivalent forms:
  - Since ~(∀x)~ is equivalent to (∃x) and ~(∃x)~ is equivalent to (∀x):
  - We change ‘~(∀x)P’ to ‘(∃x)~P’
  - We change ‘~(∃x)P’ to ‘(∀x)~P’
  - And then we check off the negated quantified sentence.
  - These rules are known as ~∀D and ~∃D, respectively

9.1E d.

1. (∃x)(Fx & ~Gx) SM
2. (∀x)Fx ⊃ (∀x)Gx SM

9.1E d.

1. $(\exists x)(Fx \& {\sim}Gx)$ SM
2. $(\forall x)Fx \supset (\forall x)Gx$ SM

Here we have the set members. First, we will decompose 1. because it doesn't branch.

9.1E d.

1. (∃x)(Fx & ~Gx) √ SM
2. (∀x)Fx ⊃ (∀x)Gx SM
3. Fa & ~Ga √ 1, ∃D
4. Fa 3, &D
5. ~Ga 3,&D

Now we do 3. because it doesn't branch.

9.1E d.

1. (∃x)(Fx & ~Gx) √ SM
2. (∀x)Fx ⊃ (∀x)Gx √ SM
3. Fa & ~Ga √ 1, ∃D
4. Fa 3, &D
5. ~Ga 3,&D
6. ~(∀x)Fx (∀x)Gx 2, ⊃D

Now we do 2. Next, we convert the negated universal.

9.1E d.

1.	(∃x)(Fx & ~Gx) √		SM
2.	(∀x)Fx ⊃ (∀x)Gx √		SM
3.	Fa & ~Ga √		1, ∃D
4.	Fa		3, &D
5.	~Ga		3,&D
6.	~(∀x)Fx√	(∀x)Gx	2, ⊃D
7.	(∃x)~Fx		6, ~∀D

9.1E d.

1.	$(\exists x)(Fx \,\&\, {\sim}Gx)$ √		SM
2.	$(\forall x)Fx \supset (\forall x)Gx$ √		SM
3.	$Fa \,\&\, {\sim}Ga$ √		1, ∃D
4.	Fa		3, &D
5.	${\sim}Ga$		3,&D
6.	${\sim}(\forall x)Fx$√	$(\forall x)Gx$	2, ⊃D
7.	$(\exists x){\sim}Fx$		6, ~∀D

Now we must eliminate the existential statement, so that we may check it off.

9.1E d.

1. $(\exists x)(Fx \ \& \ {\sim}Gx)$ √ SM
2. $(\forall x)Fx \supset (\forall x)Gx$ √ SM
3. $Fa \ \& \ {\sim}Ga$ √ 1, $\exists$D
4. Fa 3, &D
5. ${\sim}Ga$ 3,&D
6. ${\sim}(\forall x)Fx$ √ $(\forall x)Gx$ 2, $\supset$D
7. $(\exists x){\sim}Fx$ √ 6, ${\sim}\forall$D
8. ${\sim}Fb$ 7, $\exists$D

Remember that the constant for an existential must be foreign to the branch!

9.1E d.

1.	(∃x)(Fx & ~Gx) √		SM
2.	(∀x)Fx ⊃ (∀x)Gx √		SM
3.	Fa & ~Ga √		1, ∃D
4.	Fa		3, &D
5.	~Ga		3,&D
6.	~(∀x)Fx √	(∀x)Gx	2, ⊃D
7.	(∃x)~Fx √		6, ~∀D
8.	~Fb		7, ∃D

Now we can see that we have an open branch, and so don't really have to bother with the one on the right, but lets see what we can do with that branch:

9.1E d.

1.	(∃x)(Fx & ~Gx) √		SM
2.	(∀x)Fx ⊃ (∀x)Gx √		SM
3.	Fa & ~Ga √		1, ∃D
4.	Fa		3, &D
5.	~Ga		3,&D
6.	~(∀x)Fx √	(∀x)Gx	2, ⊃D
7.	(∃x)~Fx √		6, ~∀D
8.	~Fb		7, ∃D
9.		Ga	6, ∀D
10.			

Using ‘a’ as a substitution constant (remembering not to check off the universal), we can close the right branch, though the left branch remains open.

9.1E d.

1.	$(\exists x)(Fx \;\&\; \sim Gx)$ √		SM
2.	$(\forall x)Fx \supset (\forall x)Gx$ √		SM
3.	$Fa \;\&\; \sim Ga$ √		1, $\exists$D
4.	Fa		3, &D
5.	$\sim Ga$		3,&D
6.	$\sim(\forall x)Fx$ √	$(\forall x)Gx$	2, $\supset$D
7.	$(\exists x)\sim Fx$ √		6, $\sim\forall$D
8.	$\sim Fb$		7, $\exists$D
9.		Ga	6, $\forall$D
10.	O	X	

Using ‘a’ as a substitution constant (remembering not to check off the universal), we can close the right branch, though the left branch remains open.