Lecture 4

Hans Halvorson

Princeton University

September 29, 2025

Midterm Exam

- Monday, October 6 at 1:20pm
- 80 minutes to complete exam
- Cheat sheet: You may bring one sheet of paper with whatever information you can fit on it (front and back)
- No precepts next week (after exam)
- No pset this week
- To do: Work on practice midterm
- To do: Practice problems

Plan for today

- Not much new content — mostly stuff that will help you become more confident with proofs.
- Semantics (truth-tables) again
  - New: Biconditional
  - New: Classification of sentences
- Meta-rules for proofs
- Inferring the semantic type of compound sentences

Semantics

Truth table: Biconditional

P	Q	$P \leftrightarrow Q$
1	1	1
1	0	0
0	1	0
0	0	1

The biconditional $P \leftrightarrow Q$ is true (1) exactly when P and Q have the same truth value.

Semantic classification of sentences

Tautology: The column under the main connective is always True (1)

Inconsistency: The column under the main connective is always False (0)

Contingency: The column under the main connective is a mix of True (1) and False (0)

Semantic classification of sentences

$$(P \leftrightarrow Q) \vee ((Q \leftrightarrow R) \vee (P \leftrightarrow R))$$

This sentence is a tautology: for any three sentences P, Q, R, at least two must have the same truth-value.

Equivalent sentences

Two sentences are said to be **logically equivalent** just in case they have the same truth-value in all rows of their joint truth table.

P	Q	P	$\to$	Q	$\neg$	P	$\vee$	Q
1	1	1	1	1	0	1	1	1
1	0	1	0	0	0	1	0	0
0	1	0	1	1	1	0	1	1
0	0	0	1	0	1	0	1	0

Equivalent sentences

P	Q	$\neg$	$(P$	$\to$	$Q)$	P	$\wedge$	$\neg$	Q
1	1	0	1	1	1	1	0	0	1
1	0	1	1	0	0	1	1	1	0
0	1	0	0	1	1	0	0	0	1
0	0	0	0	1	0	0	0	1	0

Equivalent sentences

$$
\begin{aligned}
P \rightarrow Q &\equiv \neg P \vee Q \\
\neg(P \rightarrow Q) &\equiv P \wedge \neg Q \\
\neg(P \vee Q) &\equiv \neg P \wedge \neg Q \\
\neg(P \wedge Q) &\equiv \neg P \vee \neg Q
\end{aligned}
$$

Equivalent sentences

$$
\begin{aligned}
P \wedge Q &\equiv Q \wedge P \\
P \wedge P &\equiv P \\
P \vee P &\equiv P \\
P \rightarrow \neg P &\equiv \neg P
\end{aligned}
$$

Meta-theorems

Summary

- Soundness: If an argument form has a counterexample, then it cannot be proven.
- Completeness: If an argument form has no counterexample, then it can be proven.
- Cut: Proven sequents can act as **derived rules**.
- Replacement: Replacing a subformula of φ with an equivalent subformula results in an equivalent formula φ'.

Soundness

If the argument from $A_1, \ldots, A_j$ to B is **not** truth-functionally valid (if it has a counterexample), then $A_1, \ldots, A_j \vdash B$ can **not** be proven.

Completeness

If the argument from $A_1, \ldots, A_j$ to B is truth-functionally valid, then there is a proof of $A_1, \ldots, A_j \vdash B$.

- If $A_1, \ldots, A_j \nvDash B$, then no correct proof can end with $A_1, \ldots, A_j$ (n) B.
- If $A_1, \ldots, A_j \vDash B$, then there is a correct proof that ends with that line.

Consequences of soundness and completeness

Two sentences are **logically equivalent** if and only if they are **inter-derivable**.

$$\begin{aligned} P \to Q &\equiv \neg P \lor Q \\ \neg(P \to Q) &\equiv P \land \neg Q \\ \neg(P \lor Q) &\equiv \neg P \land \neg Q \\ \neg(P \land Q) &\equiv \neg P \lor \neg Q \end{aligned}$$

Fragment check I

Can there be a correct proof with these line fragments?

1	(1)	$P \vee Q$	A
2	(2)	$P \vee \neg Q$	A
	⋮		
1,2	(n)	P	

Yes, $P \vee Q,\ P \vee \neg Q \vDash P$ (easy truth-table reasoning). By completeness, some proof exists.

Fragment check II: Explosion from inconsistency

1	(1)	$\neg(P \leftrightarrow Q) \wedge (\neg(Q \leftrightarrow R) \wedge \neg(P \leftrightarrow R))$	A
	$\vdots$		
1	(n)	$P \wedge \neg P$	

Line 1 is inconsistent. From an inconsistency one can derive any formula. By completeness, there is a correct proof to $P \wedge \neg P$ depending only on 1.

Fragment check III: Tautology does not entail contingency

1	(1)	$P \vee \neg P$	A
	⋮		
1	(n)	Q	

$P \vee \neg P$ is a tautology; Q is a contingency. Since $P \vee \neg P \not\models Q$, soundness forbids such a proof.

Derived rules

Derived rules

- The relationship between the basic rules and derived rules is like the relationship between machine language and a high-level programming language (such as Python).
- Your thinking can operate at two levels: you can use derived rules to find a path to a proof, and then fill out the details with basic rules.
- Two kinds of derived rules:
    - **Cut:** Inference rules that operate on entire lines
    - **Replacement:** Inference rules that operate on subformulas

Ex Falso Quodlibet is a derived inference rule

1	(1)	$\neg P$	A
2	(2)	P	A
3	(3)	$\neg Q$	A
1,2	(4)	$P \wedge \neg P$	2,1 $\wedge$I
1,2	(5)	$\neg\neg Q$	3,4 RA
1,2	(6)	Q	5 DN

Negative paradox is a derived inference rule

1	(1)	$\neg P$	A
2	(2)	P	A
1,2	(3)	Q	1,2 EFQ
1	(4)	$P \rightarrow Q$	2,3 CP

Chain order from derived rules

$\vdash (P \to Q) \lor (Q \to P)$

$\varnothing$	(1)	$Q \lor \neg Q$	Excluded middle
2	(2)	Q	A
2	(3)	$P \to Q$	Positive paradox
2	(4)	$(P \to Q) \lor (Q \to P)$	3 $\lor$I
5	(5)	$\neg Q$	A
5	(6)	$Q \to P$	Negative paradox
5	(7)	$(P \to Q) \lor (Q \to P)$	6 $\lor$I
$\varnothing$	(8)	$(P \to Q) \lor (Q \to P)$	1,2,4,5,7 $\lor$E

Using derived rules

$P \to (Q \vee R) \vdash (P \to Q) \vee R$

1	(1)	$P \to (Q \vee R)$	A
2	(2)	$\neg(P \to Q)$	A
2	(3)	P	2 Material conditional
1,2	(4)	$Q \vee R$	1,3 MP
2	(5)	$\neg Q$	2 Material conditional
1,2	(6)	R	4,5 Disjunctive syllogism
1	(7)	$\neg(P \to Q) \to R$	2,6 CP
1	(8)	$(P \to Q) \vee R$	7 Material conditional

Using derived rules

$$(P \wedge Q) \rightarrow R \vdash (P \rightarrow R) \vee (Q \rightarrow R)$$

1	(1)	$(P \wedge Q) \rightarrow R$	A
2	(2)	$\neg(P \rightarrow R)$	A
2	(3)	$\neg R$	2 Material conditional
1,2	(4)	$\neg(P \wedge Q)$	1,3 MT
1,2	(5)	$\neg P \vee \neg Q$	4 DeMorgans
2	(6)	P	2 Material conditional
1,2	(7)	$\neg Q$	5,6 Disjunctive syllogism
1,2	(8)	$Q \rightarrow R$	7 Negative paradox
1	(9)	$\neg(P \rightarrow R) \rightarrow (Q \rightarrow R)$	2,8 CP
1	(10)	$(P \rightarrow R) \vee (Q \rightarrow R)$	9 Material conditional

Substitution instances

Substitution instances

We implicitly assumed that proof rules should be read **schematically**: while written as $P \to Q, P \vdash P$ with specific propositional constants P and Q, it applies to any sentences of these forms.

1	(1)	$(P \wedge Q) \to (Q \to R)$	A
2	(2)	$P \wedge Q$	A
1,2	(3)	$Q \to R$	1,2 MP

More precisely: the rule applies to **substitution instance** of $P \to Q$ and P.

Substitution Instances

Definition

A **substitution instance** of a formula schema is obtained by uniformly replacing its propositional variables with arbitrary sentences of propositional logic.

Schema: $P \to Q$

- Substitution $P := R \wedge S$, $Q := T$

$$(R \wedge S) \to T$$

- Substitution $P := \neg R$, $Q := (S \vee T)$

$$\neg R \to (S \vee T)$$

Each of these is a substitution instance of the schema $P \to Q$.

What is *not* a substitution instance?

Reminder

A substitution instance of a formula results from *uniformly replacing* its propositional variables with formulas. It does *not* allow adding, deleting, or re-arranging structure.

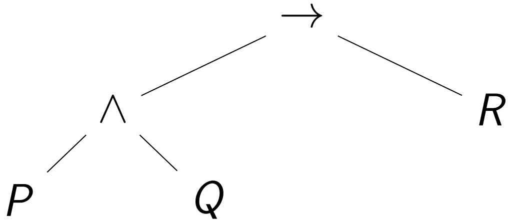
Not substitution instances:

- Q is not a substitution instance of $\neg P$. (We cannot "drop" the negation sign by substitution.)
- $S \to T$ is not a substitution instance of $P \to (Q \to P)$. (No substitution for P, Q will collapse the schema into $S \to T$.)

Moral: Substitution preserves the *tail form* of the formula.

Parse trees

A substitution instance of a formula results from extending the leaves in that formula's parse tree.



How to generate a substitution instance

Idea

A substitution maps each propositional variable to a formula. To generate a substitution instance, recursively replace variables.

Pseudo-Python:

```
def substitute(formula, mapping):
    if is_var(formula):
        return mapping[formula]
    elif is_neg(formula):              # ¬φ
        return Neg(substitute(formula.arg, mapping))
    elif is_and(formula):              # φ ∧ ψ
        return And(substitute(formula.left, mapping),
                   substitute(formula.right, mapping))
    elif is_or(formula):               # φ ∨ ψ
        return Or(substitute(formula.left, mapping),
                  substitute(formula.right, mapping))
```

A substitution consequence

Substitution of $R \mapsto P \wedge Q$ in the provable sequent

$$(P \wedge Q) \rightarrow R \;\vdash\; (P \rightarrow R) \vee (Q \rightarrow R),$$

yields

$$(P \wedge Q) \rightarrow (P \wedge Q) \;\vdash\; (P \rightarrow (P \wedge Q)) \vee (Q \rightarrow (P \wedge Q)).$$

Since the premise of the latter sequent is a tautology, its conclusion is a tautology.

Using already proven results

$\vdash\ (P \to (P \wedge Q)) \vee (Q \to (P \wedge Q))$

$\varnothing$	(1)	$Q \vee \neg Q$	Excluded middle
2	(2)	Q	A
3	(3)	P	A
2,3	(4)	$P \wedge Q$	3,2 $\wedge$I
2	(5)	$P \to (P \wedge Q)$	2,4 CP
6	(6)	$\neg Q$	A
6	(7)	$Q \to (P \wedge Q)$	6 Negative paradox
$\varnothing$	(8)	$(P \to (P \wedge Q)) \vee (Q \to (P \wedge Q))$	1,2,5,6,7 $\vee$E*

Replacement rules

An unsound rule

$\wedge$E$^+$: Any subformula $P \wedge Q$ may be replaced by P.

1	(1)	$(P \wedge Q) \rightarrow R$	A
1	(2)	$P \rightarrow R$	1 $\wedge$E$^+$

Line (2) is not semantically valid: if P is true and Q and R are false, then the dependency is true but $P \rightarrow R$ is false.

A sound rule

Material conditional: Any occurence of $P \to Q$ as a subformula may be replaced by $\neg P \vee Q$.

Why is this sound?

$$
\begin{array}{llll}
m_1, \ldots, m_j & (m) & \varphi & \\
 & \vdots & & \\
m_1, \ldots, m_j & (n) & \varphi[\neg P \vee Q / P \to Q] & \text{Material conditional}
\end{array}
$$

Replacement meta-rule

Statement

$\Gamma \vdash \varphi$ is provable if and only if $\Gamma \vdash \varphi'$ is provable, where φ' is the result of replacing some **subformula** of φ with a logically equivalent subformula.

Example:

$$\neg(P \to Q) \quad \equiv \quad P \wedge \neg Q$$

So $\Gamma \vdash \neg(P \to Q) \to R$ if and only if $\Gamma \vdash (P \wedge \neg Q) \to R$.

Useful equivalences

$$
\begin{aligned}
P \to Q &\equiv \neg P \lor Q \\
\neg(P \to Q) &\equiv P \land \neg Q \\
P \to Q &\equiv \neg Q \to \neg P \\
\neg(P \lor Q) &\equiv \neg P \land \neg Q \\
\neg(P \land Q) &\equiv \neg P \lor \neg Q \\
P \leftrightarrow Q &\equiv (P \land Q) \lor (\neg P \land \neg Q)
\end{aligned}
$$

Useful equivalences

$$
\begin{aligned}
P \vee Q &\equiv Q \vee P \\
P \vee (Q \vee R) &\equiv (P \vee Q) \vee R \\
P \vee P &\equiv P
\end{aligned}
$$

Useful equivalences

$$
\begin{aligned}
P \to (Q \to R) &\equiv\ (P \land Q) \to R \\
P \land (Q \lor R) &\equiv\ (P \land Q) \lor (P \land R) \\
P \lor (Q \land R) &\equiv\ (P \lor Q) \land (P \lor R)
\end{aligned}
$$

Chain of equivalences

$$
\begin{aligned}
(P \wedge Q) \rightarrow R &\equiv P \rightarrow (Q \rightarrow R) \\
&\equiv \neg P \vee (\neg Q \vee R) \\
&\equiv \neg P \vee (\neg Q \vee (R \vee R)) \\
&\equiv (\neg P \vee R) \vee (\neg Q \vee R) \\
&\equiv (P \rightarrow R) \vee (Q \rightarrow R)
\end{aligned}
$$

Proofs with replacement rules

$\varnothing$	(1)	$P \vee \neg P$	Excluded middle
$\varnothing$	(2)	$(\neg P \vee Q) \vee (\neg Q \vee P)$	1 $\vee$I
$\varnothing$	(3)	$(P \rightarrow Q) \vee (Q \rightarrow P)$	2 Material conditional

Translation aided by semantics

I will leave Princeton unless they give me a substantial raise.

Option 1: $R \vee \neg P$
Option 2: $\neg R \rightarrow \neg P$
Option 3: $R \rightarrow P$
Option 4: $\neg R \leftrightarrow \neg P$
Option 5: $R \leftrightarrow P$

I will stay at Princeton only if they give me a substantial raise.

Option 1: $P \rightarrow R$
Option 2: $R \rightarrow P$
Option 3: $P \leftrightarrow R$

Desmond is either in Princeton or in Queens.

Option 1: $P \vee Q$
Option 2: $P \leftrightarrow \neg Q$
Option 3: $(P \vee Q) \wedge \neg(P \wedge Q)$

Inferring types of sentences

Type of $\Phi \vee \Psi$ when both contingencies

- Cannot be an inconsistency (since Φ is true on some row, making $\Phi \vee \Psi$ true there).
- Could be a contingency (e.g. $P \vee Q$).
- Could be a tautology (e.g. $P \vee \neg P$).

Type of $\Phi \to \Psi$ when Φ is a tautology

If Φ is a tautology, then $\Phi \to \Psi \equiv \Psi$. Therefore $\Phi \to \Psi$ has the same type as Ψ (contingency if Ψ is).

Exercise. Build a 3×3 table for $\Phi \to \Psi$ over the cases where each of Φ, Ψ is a tautology, inconsistency, or contingency.

Wrap-up

- Soundness/Completeness connect proofs to truth-tables, giving another way to discern logical relations.
- Using standard moves (e.g. material conditional) plus cut/replacement can transform difficult proofs into routine exercises.
- When translating, consider whether the target sentence has the intended logical relations.