## Syntax vs. Semantics

### Syntax

- Rules for allowable formulas in the language
- Syntax for propositional logic:

$$P ::= a \mid P \wedge P \mid P \vee P \mid P \to P \mid \neg P$$

### Semantics

- Assigning meaning/interpretations with formulas
- Semantics for propositional logic: This lecture!

### Syntax and Semantics for the English language?

- Syntax: alphabet and grammar
- Semantics: meanings for words

## Semantics for Propositional Logic

Semantics assigns **meanings/interpretrations** with **formulas**

The basic notion we use is **"truth value"**

The two standard truth values are "true" and "false"
We use the symbols **T** and **F** respectively

This is a classical notion of truth

- i.e., interpretation of each proposition is either true or false
- **Excluded Middle**: for each $A$ we have $A \vee \neg A$
- Here it means for each $A$, we have that $A$ is either true or false.

WARNING: This is just one possible way to assign meanings!

## Semantics for Propositional Logic (continued)

### Truth assignment

- Function assigning a truth value for each atomic proposition
- E.g., given 2 atomic propositions $p, q$, if the formula is $p \vee q$
- then one truth assignment $\phi$ is $\phi(p) = \mathbf{T}$ and $\phi(q) = \mathbf{F}$
- Also called an "interpretation" or a "valuation"

How many truth valuations do we need to consider for $p \vee q$?

- $2^2 = 4$
- $\phi(p) = \mathbf{T}, \phi(q) = \mathbf{T}$ and $\phi(p) = \mathbf{T}, \phi(q) = \mathbf{F}$ and $\phi(p) = \mathbf{F}, \phi(q) = \mathbf{T}$ and $\phi(p) = \mathbf{F}, \phi(q) = \mathbf{F}$

### Conventions:

- The atoms $\top, \bot$ have the interpretations **T**, **F** respectively
- $\phi(\top) = \mathbf{T}$ and $\phi(\bot) = \mathbf{F}$

How to extend the notion of semantics to **compound formulas**?
Define semantics for the four logical connectives: $\vee, \wedge, \to, \neg$

This is done **recursively bottom-up** over the structure of propositions.

For example given a conjunction $A \wedge B$, we first have to evaluate the truth-values of $A$ and $B$ to compute the truth-value of $A \wedge B$.

I.e., $\phi(A \wedge B) = \mathbf{T}$ iff both $\phi(A) = \mathbf{T}$ and $\phi(B) = \mathbf{T}$.

The **extended valuation function** is recursively defined as follows:

- $\phi(\top) = \mathbf{T}$
- $\phi(\bot) = \mathbf{F}$
- $\phi(A \vee B) = \mathbf{T}$ iff either $\phi(A) = \mathbf{T}$ or $\phi(B) = \mathbf{T}$
- $\phi(A \wedge B) = \mathbf{T}$ iff both $\phi(A) = \mathbf{T}$ and $\phi(B) = \mathbf{T}$
- $\phi(A \to B) = \mathbf{T}$ iff $\phi(B) = \mathbf{T}$ whenever $\phi(A) = \mathbf{T}$
- $\phi(\neg A) = \mathbf{T}$ iff $\phi(A) = \mathbf{F}$

## Satisfiability & validity

A formula is **satisfiable** iff there is a valuation that satisfies it
i.e., if there is a **T** in the rightmost column of its truth table
example: $p \wedge q$ because of the valuation $\phi(p) = \mathbf{T}, \phi(q) = \mathbf{T}$

A formula is **falsifiable** iff there is a valuation that makes it false
i.e., if there is a **F** in the rightmost column of its truth table
example: $p \wedge q$ because of the valuation $\phi(p) = \mathbf{F}, \phi(q) = \mathbf{T}$

A formula is **unsatisfiable** iff no valuation satisfies it
i.e., the cells of the rightmost column of its truth table all contain **F**
example: $p \wedge \neg p$ (contradiction)

A formula is **valid** iff every valuation satisfies it
i.e., the cells of the rightmost column of its truth table all contain **T**
example: $p \vee \neg p$ (tautology)

We can now construct a truth table for any propositional formula

- consider all possible truth assignments for the atoms
- then use truth tables for each connective recursively

What is the truth table for $(p \to q) \wedge \neg q$?

| $p$ | $q$ | $p \to q$ | $\neg q$ | $(p \to q) \wedge \neg q$ |
|---|---|---|---|---|
| **T** | **T** | **T** | **F** | **F** |
| **T** | **F** | **F** | **T** | **F** |
| **F** | **T** | **T** | **F** | **F** |
| **F** | **F** | **T** | **T** | **T** |

- 2 atoms, and hence $2^2 = 4$ rows (one per interpretation)
- Use intermediate columns to evaluate sub-formulas
- 2 atoms and 3 connectives hence $2 + 3 = 5$ columns
- Rightmost column gives values of the formula

# Logical equivalences using truth tables

Classically, two formulas are logically equivalent if they have the same semantics.

I.e., they have the same truth values for all valuations.

E.g., an implication and its contrapositive are logically equivalent:

Show that $(A \to B) \leftrightarrow (\neg B \to \neg A)$ using a truth table

| $A$ | $B$ | $A \to B$ | $\neg B$ | $\neg A$ | $\neg B \to \neg A$ |
|---|---|---|---|---|---|
| **T** | **T** | **T** | **F** | **F** | **T** |
| **T** | **F** | **F** | **T** | **F** | **F** |
| **F** | **T** | **T** | **F** | **T** | **T** |
| **F** | **F** | **T** | **T** | **T** | **T** |

**The two formulas are equivalent** because the two columns for $A \to B$ and $\neg B \to \neg A$ are identical

# Validity of arguments using semantics

## Validity of an argument

- **syntactically:** we can derive the conclusion from the premises
- **semantically:** the conclusion is true whenever the premises are

Formally, we write

$$P_1, \ldots, P_n \models C$$

if the corresponding argument is **semantically valid**

i.e., every valuation that evaluates each of the premises $P_1, \ldots, P_n$ to **T** also evaluates the conclusion $C$ to **T**

## Checking validity

- Already seen how to do this using "natural deduction"
- Truth tables is yet another way
- Bonus: yields counterexample if argument is invalid

# Checking (semantic) validity

Is $P \to Q, \neg Q \models \neg P$ (semantically) valid?

| $P$ | $Q$ | $P \to Q$ | $\neg Q$ | $\neg P$ |
|---|---|---|---|---|
| **T** | **T** | **T** | **F** | **F** |
| **T** | **F** | **F** | **T** | **F** |
| **F** | **T** | **T** | **F** | **T** |
| **F** | **F** | **T** | **T** | **T** |

Argument is valid: any row where conclusion is **F** then at least one of the premises is also **F**

Argument is valid: any row where conclusion is **F** then at least one of the premises is also **F**

Note that checking $P_1, \ldots, P_n \models C$ is equivalent to checking the validity of $P_1 \rightarrow \cdots P_n \rightarrow C$

i.e., that the cells of the rightmost column of the truth table for $P_1 \rightarrow \cdots P_n \rightarrow C$ all contain **T**

Is $\neg P \rightarrow \neg R, R \models \neg P$ (semantically) valid?

| $P$ | $R$ | $\neg P$ | $\neg R$ | $\neg P \rightarrow \neg R$ | $R$ | $\neg P$ |
|---|---|---|---|---|---|---|
| **T** | **T** | **F** | **F** | **T** | **T** | **F** |
| **T** | **F** | **F** | **T** | **T** | **F** | **F** |
| **F** | **T** | **T** | **F** | **F** | **T** | **T** |
| **F** | **F** | **T** | **T** | **T** | **F** | **T** |

Argument is invalid

- Look at the first row
- Conclusion is **F**, but both premises are **T**
- Can we add a premise to make the argument valid?
  - Yes, we can add $\neg R$, which would be **F** in the first row

Is $P, \neg P \models C$ is (semantically) valid?

| $P$ | $C$ | $\neg P$ | $C$ |
|---|---|---|---|
| **T** | **T** | **F** | **T** |
| **T** | **F** | **F** | **F** |
| **F** | **T** | **T** | **T** |
| **F** | **F** | **T** | **F** |

Argument is (trivially) valid:

- Look at any row (we only have to look at rows where the conclusion is **F**)
- One of $P$ and $\neg P$ is **F**

# Soundness & Completeness

Given a deduction system such as Natural deduction, a formula is said to be **provable** if there is a proof of it in that deduction system

- This is a **syntactic** notion
- it asserts the existence of a syntactic object: a proof
- typically written $\vdash A$

A formula $A$ is **valid** if $\phi(A) = \mathbf{T}$ for all possible valuations $\phi$

- it is a **semantic** notion
- it is checked w.r.t. valuations that give meaning to formulas

**Soundness**: a deduction system is sound w.r.t. a semantics if every provable formula is valid

- i.e., if $\vdash A$ then $\models A$

**Completeness**: a deduction system is complete w.r.t. a semantics if every valid formula is provable

- i.e., if $\models A$ then $\vdash A$

# Soundness & Completeness

**Classical Natural Deduction** is

- **sound** and
- **complete**

w.r.t. the **truth table semantics**

Proving those properties is done within the **metatheory**

- Soundness is easy. It requires proving that each rule is valid.

  For example:

  $$\frac{A \quad B}{A \wedge B}\ [\wedge I]$$

  is valid because $A, B \models A \wedge B$
- Completeness is harder

We will not prove them here

Pros and cons of two ways of checking validity

| Truth tables | Natural deduction |
| --- | --- |
| shows validity in a restricted setting (Boolean truth values) | checks validity in general setting (by an actual proof!) |
| simple, easy to automate | more difficult to automate |
| size of truth table is huge: exponential in number of atoms | typically scales better than brute force search |
| generates counterexamples if invalid | no easy way to check validity (other than actually proving) |

## Semantics for "or"

$$\phi(A \vee B) = \mathbf{T} \text{ iff either } \phi(A) = \mathbf{T} \text{ or } \phi(B) = \mathbf{T}$$

## Truth table for "or"

| $A$ | $B$ | $A \vee B$ |
|---|---|---|
| **T** | **T** | **T** |
| **T** | **F** | **T** |
| **F** | **T** | **T** |
| **F** | **F** | **F** |

- One row for each valuation
- Last column has the truth value for the corresponding valuation

## Semantics for "and"

$$\phi(A \wedge B) = \mathbf{T} \text{ iff both } \phi(A) = \mathbf{T} \text{ and } \phi(B) = \mathbf{T}$$

## Truth table for "and"

| $A$ | $B$ | $A \wedge B$ |
|---|---|---|
| **T** | **T** | **T** |
| **T** | **F** | **F** |
| **F** | **T** | **F** |
| **F** | **F** | **F** |

## Semantics for "implies"

$$\phi(A \rightarrow B) = \mathbf{T} \text{ iff } \phi(B) = \mathbf{T} \text{ whenever } \phi(A) = \mathbf{T}$$

$$\phi(A \to B) = \mathbf{T} \text{ iff } \phi(B) = \mathbf{T} \text{ whenever } \phi(A) = \mathbf{T}$$

## Truth table for "implies"

| $A$ | $B$ | $A \to B$ |
|---|---|---|
| **T** | **T** | **T** |
| **T** | **F** | **F** |
| **F** | **T** | **T** |
| **F** | **F** | **T** |

## Semantics for "not"

$$\phi(\neg A) = \mathbf{T} \text{ iff } \phi(A) = \mathbf{F}$$

## Truth table for "not"

| $A$ | $\neg A$ |
|---|---|
| **T** | **F** |
| **F** | **T** |