

# Theory of Computer Science

## B1. Propositional Logic I

Malte Helmert

University of Basel

February 27, 2017

# Theory of Computer Science

February 27, 2017 — B1. Propositional Logic I

B1.1 Motivation

B1.2 Syntax

B1.3 Semantics

B1.4 Properties of Propositional Formulas

B1.5 Summary

## B1.1 Motivation

## Why Logic?

- ▶ formalizing mathematics
  - ▶ What is a true statement?
  - ▶ What is a valid proof?
- ▶ basis of many tools in computer science
  - ▶ design of digital circuits
  - ▶ meaning of programming languages
  - ▶ semantics of databases; query optimization
  - ▶ verification of safety-critical hardware/software
  - ▶ knowledge representation in artificial intelligence
  - ▶ ...

## Example: Group Theory

Example of a **group** (in mathematics):  $\langle \mathbb{Z}, + \rangle$

- ▶ the set of integers with the addition operation

A **group** in general:  $\langle G, \circ \rangle$

- ▶  $G$  is a set and  $\circ : G \times G \rightarrow G$  is called the **group operation**; we write “ $x \circ y$ ” instead of “ $\circ(x, y)$ ” (**infix notation**)

For  $\langle G, \circ \rangle$  to be a group, it must satisfy the **group axioms**:

- ▶ (G1) For all  $x, y, z \in G$ ,  $(x \circ y) \circ z = x \circ (y \circ z)$ .
- ▶ There exists  $e \in G$  (called the **neutral element**) such that:
  - ▶ (G2) for all  $x \in G$ ,  $x \circ e = x$ , and
  - ▶ (G3) for all  $x \in G$ , there is a  $y \in G$  with  $x \circ y = e$ .

**German:** Gruppe, Verknüpfung, Infix, Gruppenaxiome, neutrales Element

## Example: Group Theory

**Theorem (Existence of a left inverse)**

Let  $\langle G, \circ \rangle$  be a group with neutral element  $e$ .

For all  $x \in G$  there is a  $y \in G$  with  $y \circ x = e$ .

**Proof.**

Consider an arbitrary  $x \in G$ .

Because of G3, there is a  $y$  with  $x \circ y = e$  (\*).

Also because of G3, for this  $y$  there is a  $z$  with  $y \circ z = e$  (\*\*).

It follows that:

$$\begin{aligned} y \circ x &\stackrel{(G2)}{=} (y \circ x) \circ e \stackrel{(**)}{=} (y \circ x) \circ (y \circ z) \\ &\stackrel{(G1)}{=} y \circ (x \circ (y \circ z)) \stackrel{(G1)}{=} y \circ ((x \circ y) \circ z) \\ &\stackrel{(*)}{=} y \circ (e \circ z) \stackrel{(G1)}{=} (y \circ e) \circ z \\ &\stackrel{(G2)}{=} y \circ z \stackrel{(**)}{=} e \end{aligned}$$

□

## What Logic is About

**General Question:**

- ▶ Given a set of axioms (e. g., group axioms)
- ▶ what can we **derive** from them?  
(e. g., theorem about the existence of a left inverse)
- ▶ And on what basis may we argue?  
(e. g., why does  $y \circ x = (y \circ x) \circ e$  follow from axiom G2?)

↪ **logic**

**Goal:** “mechanical” proofs

- ▶ formal “game with letters”
- ▶ detached from a concrete meaning

## Propositional Logic

**Propositional logic** is a simple logic without numbers or objects.

Building blocks of propositional logic:

- ▶ **propositions** are statements that can be either true or false
- ▶ **atomic propositions** cannot be split into sub-propositions
- ▶ **logical connectives** connect propositions to form new ones

**German:** Aussagenlogik, Aussage, atomare Aussage, Junktoren

## Examples for Building Blocks



If I don't **drink beer** to a meal, then I always **eat fish**. Whenever I **have fish and beer** with the same meal, I abstain from **ice cream**. When I **eat ice cream** or don't **drink beer**, then I never touch **fish**.

- ▶ Every sentence is a proposition that consists of sub-propositions (e. g., "eat ice cream or don't drink beer").
- ▶ atomic propositions "**drink beer**", "**eat fish**", "**eat ice cream**"
- ▶ logical connectives "and", "or", negation, "if, then"

Exercise from U. Schöning: Logik für Informatiker  
Picture courtesy of graur razvan ionut / FreeDigitalPhotos.net

## Examples for Building Blocks



If I **don't** drink beer to a meal, **then** I always eat fish. **Whenever** I have fish **and** beer with the same meal, I **abstain** from ice cream. **When** I eat ice cream **or** **don't** drink beer, **then** I **never** touch fish.

- ▶ Every sentence is a proposition that consists of sub-propositions (e. g., "eat ice cream or don't drink beer").
- ▶ atomic propositions "drink beer", "eat fish", "eat ice cream"
- ▶ logical connectives "**and**", "**or**", **negation**, "**if, then**"

Exercise from U. Schöning: Logik für Informatiker  
Picture courtesy of graur razvan ionut / FreeDigitalPhotos.net

## Problems with Natural Language



If I don't drink beer **to a meal**, then I **always** eat fish.  
Whenever I have fish and beer **with the same meal**, I abstain from ice cream.  
When I eat ice cream or don't drink beer, then I never touch fish.

- ▶ "**irrelevant**" information

Exercise from U. Schöning: Logik für Informatiker  
Picture courtesy of graur razvan ionut / FreeDigitalPhotos.net

## Problems with Natural Language



If I **don't** drink beer, then I eat fish.  
Whenever I have fish and beer, I **abstain** from ice cream.  
When I eat ice cream or **don't** drink beer, then I **never** touch fish.

- ▶ "**irrelevant**" information
- ▶ **different formulations for the same connective/proposition**

Exercise from U. Schöning: Logik für Informatiker  
Picture courtesy of graur razvan ionut / FreeDigitalPhotos.net

## Problems with Natural Language



If not DrinkBeer, then EatFish.  
 If EatFish and DrinkBeer,  
 then not EatIceCream.  
 If EatIceCream or not DrinkBeer,  
 then not EatFish.

- ▶ “irrelevant” information
- ▶ different formulations for the same connective/proposition

Exercise from U. Schöning: Logik für Informatiker  
 Picture courtesy of graur razvan ionut / FreeDigitalPhotos.net

## What is Next?

- ▶ What are meaningful (well-defined) sequences of atomic propositions and connectives?  
 “if then EatIceCream not or DrinkBeer and” not meaningful  
 → **syntax**
- ▶ What does it mean if we say that a statement is true?  
 Is “DrinkBeer and EatFish” true?  
 → **semantics**
- ▶ When does a statement logically follow from another?  
 Does “EatFish” follow from “if DrinkBeer, then EatFish”?  
 → **logical entailment**
- ▶ **German:** Syntax, Semantik, logische Folgerung

## B1.2 Syntax

### Syntax of Propositional Logic

#### Definition (Syntax of Propositional Logic)

Let  $A$  be a set of **atomic propositions**. The set of **propositional formulas** (over  $A$ ) is inductively defined as follows:

- ▶ Every **atom**  $a \in A$  is a propositional formula over  $A$ .
- ▶ If  $\varphi$  is a propositional formula over  $A$ , then so is its **negation**  $\neg\varphi$ .
- ▶ If  $\varphi$  and  $\psi$  are propositional formulas over  $A$ , then so is the **conjunction**  $(\varphi \wedge \psi)$ .
- ▶ If  $\varphi$  and  $\psi$  are propositional formulas over  $A$ , then so is the **disjunction**  $(\varphi \vee \psi)$ .

The **implication**  $(\varphi \rightarrow \psi)$  is an abbreviation for  $(\neg\varphi \vee \psi)$ .

The **biconditional**  $(\varphi \leftrightarrow \psi)$  is an abbrev. for  $((\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi))$ .

**German:** atomare Aussage, aussagenlogische Formel, Atom, Negation, Konjunktion, Disjunktion, Implikation, Bikonditional

## Syntax: Examples

Which of the following sequences of symbols are propositional formulas over the set of all possible letter sequences?

- ▶  $(A \wedge (B \vee C))$
- ▶  $((\text{EatFish} \wedge \text{DrinkBeer}) \rightarrow \neg \text{EatIceCream})$
- ▶  $\neg(\wedge \text{Rain} \vee \text{StreetWet})$
- ▶  $\neg(\text{Rain} \vee \text{StreetWet})$
- ▶  $\neg(A = B)$
- ▶  $(A \wedge \neg(B \leftrightarrow C))$
- ▶  $(A \vee \neg(B \leftrightarrow C))$
- ▶  $((A \leq B) \wedge C)$
- ▶  $((A_1 \wedge A_2) \vee \neg(A_3 \leftrightarrow A_2))$

Which kinds of formula are they (atom, conjunction, ...)?

## B1.3 Semantics

## Meaning of Propositional Formulas?

So far propositional formulas are only symbol sequences without any meaning.

For example, what does this mean:

$((\text{EatFish} \wedge \text{DrinkBeer}) \rightarrow \neg \text{EatIceCream})?$

▶ **We need semantics!**

## Semantics of Propositional Logic

### Definition (Semantics of Propositional Logic)

A **truth assignment** (or **interpretation**) for a set of atomic propositions  $A$  is a function  $\mathcal{I} : A \rightarrow \{0, 1\}$ .

A propositional **formula**  $\varphi$  (over  $A$ ) **holds under**  $\mathcal{I}$  (written as  $\mathcal{I} \models \varphi$ ) according to the following definition:

$$\begin{aligned} \mathcal{I} \models a & \quad \text{iff} \quad \mathcal{I}(a) = 1 & \quad (\text{for } a \in A) \\ \mathcal{I} \models \neg\varphi & \quad \text{iff} \quad \text{not } \mathcal{I} \models \varphi \\ \mathcal{I} \models (\varphi \wedge \psi) & \quad \text{iff} \quad \mathcal{I} \models \varphi \text{ and } \mathcal{I} \models \psi \\ \mathcal{I} \models (\varphi \vee \psi) & \quad \text{iff} \quad \mathcal{I} \models \varphi \text{ or } \mathcal{I} \models \psi \end{aligned}$$

**Question:** should we define semantics of  $(\varphi \rightarrow \psi)$  and  $(\varphi \leftrightarrow \psi)$ ?

**German:** Wahrheitsbelegung/Interpretation,  $\varphi$  gilt unter  $\mathcal{I}$

## Semantics of Propositional Logic: Terminology

- ▶ For  $\mathcal{I} \models \varphi$  we also say  $\mathcal{I}$  is a **model of  $\varphi$**  and that  $\varphi$  is **true under  $\mathcal{I}$** .
- ▶ If  $\varphi$  does not hold under  $\mathcal{I}$ , we write this as  $\mathcal{I} \not\models \varphi$  and say that  $\mathcal{I}$  is **no model of  $\varphi$**  and that  $\varphi$  is **false under  $\mathcal{I}$** .
- ▶ **Note:**  $\models$  is not part of the formula but part of the **meta language** (speaking **about** a formula).

**German:**  $\mathcal{I}$  ist ein/kein Modell von  $\varphi$ ;  $\varphi$  ist wahr/falsch unter  $\mathcal{I}$ ;  
**Metasprache**

## Semantics: Example (1)

$$A = \{\text{DrinkBeer}, \text{EatFish}, \text{EatIceCream}\}$$

$$\mathcal{I} = \{\text{DrinkBeer} \mapsto 1, \text{EatFish} \mapsto 0, \text{EatIceCream} \mapsto 1\}$$

$$\varphi = (\neg \text{DrinkBeer} \rightarrow \text{EatFish})$$

Do we have  $\mathcal{I} \models \varphi$ ?

## Semantics: Example (2)

**Goal:** prove  $\mathcal{I} \models \varphi$ .

Let us use the definitions we have seen:

$$\begin{aligned} \mathcal{I} \models \varphi &\text{ iff } \mathcal{I} \models (\neg \text{DrinkBeer} \rightarrow \text{EatFish}) \\ &\text{ iff } \mathcal{I} \models (\neg \neg \text{DrinkBeer} \vee \text{EatFish}) \\ &\text{ iff } \mathcal{I} \models \neg \neg \text{DrinkBeer} \text{ or } \mathcal{I} \models \text{EatFish} \end{aligned}$$

This means that if we want to prove  $\mathcal{I} \models \varphi$ , it is sufficient to prove

$$\mathcal{I} \models \neg \neg \text{DrinkBeer}$$

or to prove

$$\mathcal{I} \models \text{EatFish}.$$

We attempt to prove the first of these statements.

## Semantics: Example (3)

**New goal:** prove  $\mathcal{I} \models \neg \neg \text{DrinkBeer}$ .

We again use the definitions:

$$\begin{aligned} \mathcal{I} \models \neg \neg \text{DrinkBeer} &\text{ iff not } \mathcal{I} \models \neg \text{DrinkBeer} \\ &\text{ iff not not } \mathcal{I} \models \text{DrinkBeer} \\ &\text{ iff } \mathcal{I} \models \text{DrinkBeer} \\ &\text{ iff } \mathcal{I}(\text{DrinkBeer}) = 1 \end{aligned}$$

The last statement is true for our interpretation  $\mathcal{I}$ .

To write this up as a **proof** of  $\mathcal{I} \models \varphi$ , we can go through this line of reasoning back-to-front, starting from our assumptions and ending with the conclusion we want to show.

## Semantics: Example (4)

Let  $\mathcal{I} = \{\text{DrinkBeer} \mapsto 1, \text{EatFish} \mapsto 0, \text{EatIceCream} \mapsto 1\}$ .

Proof that  $\mathcal{I} \models (\neg\text{DrinkBeer} \rightarrow \text{EatFish})$ :

- (1) We have  $\mathcal{I} \models \text{DrinkBeer}$   
(uses defn. of  $\models$  for atomic props. and fact  $\mathcal{I}(\text{DrinkBeer}) = 1$ ).
- (2) From (1), we get  $\mathcal{I} \not\models \neg\text{DrinkBeer}$   
(uses defn. of  $\models$  for negations).
- (3) From (2), we get  $\mathcal{I} \models \neg\neg\text{DrinkBeer}$   
(uses defn. of  $\models$  for negations).
- (4) From (3), we get  $\mathcal{I} \models (\neg\neg\text{DrinkBeer} \vee \psi)$  for all formulas  $\psi$ ,  
in particular  $\mathcal{I} \models (\neg\neg\text{DrinkBeer} \vee \text{EatFish})$   
(uses defn. of  $\models$  for disjunctions).
- (5) From (4), we get  $\mathcal{I} \models (\neg\text{DrinkBeer} \rightarrow \text{EatFish})$   
(uses defn. of " $\rightarrow$ "). □

## B1.4 Properties of Propositional Formulas

## Properties of Propositional Formulas

A propositional formula  $\varphi$  is

- ▶ **satisfiable** if  $\varphi$  has at least one model
- ▶ **unsatisfiable** if  $\varphi$  is not satisfiable
- ▶ **valid** (or a **tautology**) if  $\varphi$  is true under every interpretation
- ▶ **falsifiable** if  $\varphi$  is no tautology

German: erfüllbar, unerfüllbar, gültig/eine Tautologie, falsifizierbar

How can we show that a formula has one of these properties?

## Examples

- ▶ Show that  $(A \wedge B)$  is **satisfiable**.  
 $\mathcal{I} = \{A \mapsto 1, B \mapsto 1\}$  (+ simple proof that  $\mathcal{I} \models (A \wedge B)$ )
- ▶ Show that  $(A \wedge B)$  is **falsifiable**.  
 $\mathcal{I} = \{A \mapsto 0, B \mapsto 1\}$  (+ simple proof that  $\mathcal{I} \not\models (A \wedge B)$ )
- ▶ Show that  $(A \wedge B)$  is **not valid**.  
Follows directly from falsifiability.
- ▶ Show that  $(A \wedge B)$  is **not unsatisfiable**.  
Follows directly from satisfiability.

So far all proofs by specifying **one** interpretation.

How to prove that a given formula is valid/unsatisfiable/  
not satisfiable/not falsifiable?

$\rightsquigarrow$  must consider **all possible** interpretations

## Truth Tables

Evaluate for all possible interpretations if they are models of the considered formula.

$I(A)$	$I \models \neg A$
0	Yes
1	No

$I(A)$	$I(B)$	$I \models (A \wedge B)$	$I(A)$	$I(B)$	$I \models (A \vee B)$
0	0	No	0	0	No
0	1	No	0	1	Yes
1	0	No	1	0	Yes
1	1	Yes	1	1	Yes

## Truth Tables in General

Similarly in the case where we consider a formula whose building blocks are themselves arbitrary unspecified formulas:

$I \models \varphi$	$I \models \psi$	$I \models (\varphi \wedge \psi)$
No	No	No
No	Yes	No
Yes	No	No
Yes	Yes	Yes

**Exercises:** truth table for  $(\varphi \rightarrow \psi)$

## Truth Tables for Properties of Formulas

Is  $\varphi = ((A \rightarrow B) \vee (\neg B \rightarrow A))$  valid, unsatisfiable, ... ?

$I(A)$	$I(B)$	$I \models \neg B$	$I \models (A \rightarrow B)$	$I \models (\neg B \rightarrow A)$	$I \models \varphi$
0	0	Yes	Yes	No	Yes
0	1	No	Yes	Yes	Yes
1	0	Yes	No	Yes	Yes
1	1	No	Yes	Yes	Yes

## Connection Between Formula Properties and Truth Tables

A propositional formula  $\varphi$  is

- ▶ **satisfiable** if  $\varphi$  has at least one model  
 $\rightsquigarrow$  result in at least one row is "Yes"
- ▶ **unsatisfiable** if  $\varphi$  is not satisfiable  
 $\rightsquigarrow$  result in all rows is "No"
- ▶ **valid** (or a **tautology**) if  $\varphi$  is true under every interpretation  
 $\rightsquigarrow$  result in all rows is "Yes"
- ▶ **falsifiable** if  $\varphi$  is no tautology  
 $\rightsquigarrow$  result in at least one row is "No"

## Main Disadvantage of Truth Tables

How big is a truth table with  $n$  atomic propositions?

1	2 interpretations (rows)
2	4 interpretations (rows)
3	8 interpretations (rows)
$n$	??? interpretations

Some examples:  $2^{10} = 1024$ ,  $2^{20} = 1048576$ ,  $2^{30} = 1073741824$

↪ not viable for larger formulas; we need a different solution

- ▶ more on difficulty of satisfiability etc.: Part E of this course
- ▶ practical algorithms: Foundations of AI course

## B1.5 Summary

## Summary

- ▶ **propositional logic** based on atomic propositions
- ▶ **syntax** defines what well-formed formulas are
- ▶ **semantics** defines when a formula is true
- ▶ **interpretations** are the basis of semantics
- ▶ **satisfiability** and **validity** are important properties of formulas
- ▶ **truth tables** systematically consider all possible interpretations
- ▶ truth tables are only useful for small formulas