Propositional Logic
Basics

Syntax of propositional logic

Definition

An atomic formula (or atom) has the form A_i where $i = 1, 2, 3, \ldots$.
Formulas are defined inductively:

- $\bot$ ("False") and $\top$ ("True") are formulas
- All atomic formulas are formulas
- For all formulas F, $\neg F$ is a formula.
- For all formulas F und G, $(F \circ G)$ is a formula, where $\circ \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$

$\neg$	is called	negation
$\wedge$	is called	conjunction
$\vee$	is called	disjunction
$\rightarrow$	is called	implication
$\leftrightarrow$	is called	bi-implication

Parentheses

Precedence of logical operators in decreasing order:

$$\neg \quad \wedge \quad \vee \quad \rightarrow \quad \leftrightarrow$$

Operators with higher precedence bind more strongly.

Example

Instead of $(A \rightarrow ((B \wedge \neg(C \vee D)) \vee E))$
we can write $A \rightarrow B \wedge \neg(C \vee D) \vee E$.

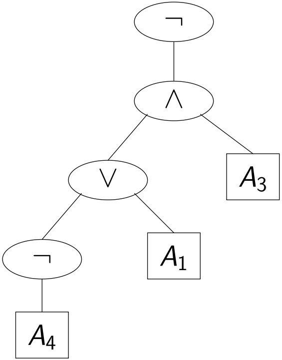
Outermost parentheses can be dropped.

Syntax tree of a formula

Every formula can be represented by a syntax tree.

Example

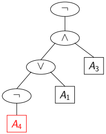
$F = \neg((\neg A_4 \lor A_1) \land A_3)$



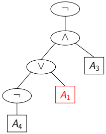
Subformulas

The subformulas of a formula are the formulas corresponding to the subtrees of its syntax tree.

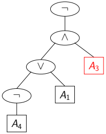
A_4



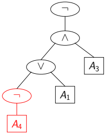
A_1



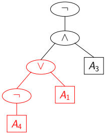
A_3



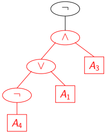
$\neg A_4$



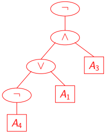
$(\neg A_4 \vee A_1)$



$((\neg A_4 \vee A_1) \wedge A_3)$



$\neg((\neg A_4 \vee A_1) \wedge A_3)$



Induction on formulas

Proof by induction on the structure of a formula:
In order to prove some property $\mathcal{P}(F)$ for all formulas F
it suffices to prove the following:

- Base cases:
  prove $\mathcal{P}(\bot)$, prove $\mathcal{P}(\top)$, and prove $\mathcal{P}(A_i)$ for all atoms A_i
- Induction step for $\neg$:
  prove $\mathcal{P}(\neg F)$ under the induction hypothesis $\mathcal{P}(F)$
- Induction step for all $\circ \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$:
  prove $\mathcal{P}(F \circ G)$ under the induction hypotheses $\mathcal{P}(F)$ and $\mathcal{P}(G)$

Operators that are merely abbreviations need not be considered!

Semantics of propositional logic (I)

The elements of the set $\{0, 1\}$ are called truth values.
(You may call 0 “false” and 1 “true”)

An assignment is a function $\mathcal{A} : \mathit{Atoms} \to \{0, 1\}$
where *Atoms* is the set of all atoms.

We extend $\mathcal{A}$ to a function $\hat{\mathcal{A}}\colon \mathit{Formulas} \to \{0, 1\}$

Semantics of propositional logic (II)

$$
\begin{array}{rcl}
\hat{\mathcal{A}}(A_i) & = & \mathcal{A}(A_i) \\
\hat{\mathcal{A}}(\neg F) & = & \begin{cases} 1 & \text{if } \hat{\mathcal{A}}(F) = 0 \\ 0 & \text{otherwise} \end{cases} \\
\hat{\mathcal{A}}(F \wedge G) & = & \begin{cases} 1 & \text{if } \hat{\mathcal{A}}(F) = 1 \text{ and } \hat{\mathcal{A}}(G) = 1 \\ 0 & \text{otherwise} \end{cases} \\
\hat{\mathcal{A}}(F \vee G) & = & \begin{cases} 1 & \text{if } \hat{\mathcal{A}}(F) = 1 \text{ or } \hat{\mathcal{A}}(G) = 1 \\ 0 & \text{otherwise} \end{cases} \\
\hat{\mathcal{A}}(F \to G) & = & \begin{cases} 1 & \text{if } \hat{\mathcal{A}}(F) = 0 \text{ or } \hat{\mathcal{A}}(G) = 1 \\ 0 & \text{otherwise} \end{cases}
\end{array}
$$

Instead of $\hat{\mathcal{A}}$ we simply write $\mathcal{A}$

Truth tables (I)

We can compute $\hat{\mathcal{A}}$ with the help of truth tables.

$\neg$	A
1	0
0	1

A	$\vee$	B
0	0	0
0	1	1
1	1	0
1	1	1

A	$\wedge$	B
0	0	0
0	0	1
1	0	0
1	1	1

A	$\rightarrow$	B
0	1	0
0	1	1
1	0	0
1	1	1

Abbreviations

$A, B, C,$
$P, Q, R,$ or $\ldots$ instead of $A_1, A_2, A_3 \ldots$

$$F_1 \leftrightarrow F_2 \quad \text{abbreviates} \quad (F_1 \wedge F_2) \vee (\neg F_1 \wedge \neg F_2)$$

$$\bigvee_{i=1}^{n} F_i \quad \text{abbreviates} \quad (\ldots((F_1 \vee F_2) \vee F_3) \vee \ldots \vee F_n)$$

$$\bigwedge_{i=1}^{n} F_i \quad \text{abbreviates} \quad (\ldots((F_1 \wedge F_2) \wedge F_3) \wedge \ldots \wedge F_n)$$

Special cases:

$$\bigvee_{i=1}^{0} F_i = \bigvee \emptyset = \bot \qquad \bigwedge_{i=1}^{0} F_i = \bigwedge \emptyset = \top$$

Truth tables (II)

A	$\leftrightarrow$	B
0	1	0
0	0	1
1	0	0
1	1	1

Coincidence Lemma

Lemma

Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be two assignments.
If $\mathcal{A}_1(A_i) = \mathcal{A}_2(A_i)$ for all atoms A_i in some formula F,
then $\mathcal{A}_1(F) = \mathcal{A}_2(F)$.

Proof.

Exercise. □

Models

If $\mathcal{A}(F) = 1$	then we write	$\mathcal{A} \models F$
	and say	F is true under $\mathcal{A}$
	or	$\mathcal{A}$ is a model of F
If $\mathcal{A}(F) = 0$	then we write	$\mathcal{A} \not\models F$
	and say	F is false under $\mathcal{A}$
	or	$\mathcal{A}$ is not a model of F

Validity and satisfiability

Definition (Validity)

A formula F is valid (or a tautology)
if *every* assignment is a model of F.
We write $\models F$ if F is valid, and $\not\models F$ otherwise.

Definition (Satisfiability)

A formula F is satisfiable if it has at least one model;
otherwise F is unsatisfiable.
A (finite or infinite!) set of formulas S is satisfiable if there is an assigment that is a model of every formula in S.

Exercise

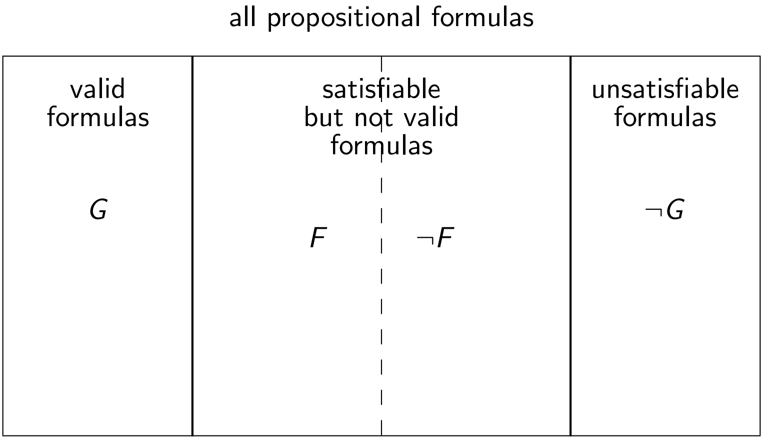
	Valid	Satisfiable	Unsatisfiable
A			
$A \vee B$			
$A \vee \neg A$			
$A \wedge \neg A$			
$A \rightarrow \neg A$			
$A \rightarrow (B \rightarrow A)$			
$A \rightarrow (A \rightarrow B)$			
$A \leftrightarrow \neg A$			

Exercise

Which of the following statements are true?

		Y	C.ex.
If F is valid,	then F is satisfiable		
If F is satisfiable,	then $\neg F$ is satisfiable		
If F is valid,	then $\neg F$ is unsatisfiable		
If F is unsatisfiable,	then $\neg F$ is unsatisfiable		

Mirroring principle



Consequence

Definition

A formula G is a (semantic) consequence of a set of formulas M if every model $\mathcal{A}$ of all $F \in M$ is also a model of G.
Then we write $M \models G$.

In a nutshell:

"Every model of M is a model of G."

Example

$A \vee B,\ A \rightarrow B,\ B \wedge R \rightarrow \neg A,\ R \ \models\ (R \wedge \neg A) \wedge B$

Exercise

M	F	$M \models F$?
A	$A \vee B$	
A	$A \wedge B$	
A, B	$A \vee B$	
A, B	$A \wedge B$	
$A \wedge B$	A	
$A \vee B$	A	
$A, A \rightarrow B$	B	

Consequence, validity, satisfiability

Exercise

The following statements are equivalent:

1. $F_1, \ldots, F_k \models G$
2. $\models (\bigwedge_{i=1}^{k} F_i) \to G$

Exercise

The following statements are equivalent:

1. $F \to G$ *is valid.*
2. $F \wedge \neg G$ *is unsatisfiable.*

Exercise

Let M be a set of formulas, and let F and G be formulas.
Which of the following statements hold?

	Y/N	C.ex.
If F satisfiable then $M \models F$.		
If F valid then $M \models F$.		
If $F \in M$ then $M \models F$.		
If $F \models G$ then $\neg F \models \neg G$.		

Notation

Warning: The symbol $\models$ is overloaded:

$\mathcal{A} \models F$

$\models F$

$M \models F$

Convenient variations for set of formulas S:

$\mathcal{A} \models S$ means that for all $F \in S$, $\mathcal{A} \models F$

$\models S$ means that for all $F \in S$, $\models F$

$M \models S$ means that for all $F \in S$, $M \models F$