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 $\circ \in \{\land, \lor, \rightarrow, \leftrightarrow\}$

<table>
<thead>
<tr>
<th>Symbol</th>
<th>is called</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\neg$</td>
<td>negation</td>
</tr>
<tr>
<td>$\land$</td>
<td>conjunction</td>
</tr>
<tr>
<td>$\lor$</td>
<td>disjunction</td>
</tr>
<tr>
<td>$\rightarrow$</td>
<td>implication</td>
</tr>
<tr>
<td>$\leftrightarrow$</td>
<td>bi-implication</td>
</tr>
</tbody>
</table>
Parentheses

**Precedence** of logical operators in decreasing order:

\[ \neg \land \lor \rightarrow \leftrightarrow \]

Operators with higher precedence bind more strongly.

**Example**

Instead of \((A \rightarrow ((B \land \neg(C \lor D)) \lor E))\),
we can write \(A \rightarrow B \land \neg(C \lor D) \lor E\).

Outermost parentheses can be dropped.
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.
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 \{\land, \lor, \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!
The elements of the set \{0, 1\} are called \textbf{truth values}. (You may call 0 “false” and 1 “true”)

An \textbf{assignment} is a function \( \mathcal{A} : Atoms \rightarrow \{0, 1\} \) where \( Atoms \) is the set of all atoms.

We extend \( \mathcal{A} \) to a function \( \hat{\mathcal{A}}: Formulas \rightarrow \{0, 1\} \)
Semantics of propositional logic (II)

\[
\hat{A}(A_i) = A(A_i)
\]

\[
\hat{A}(\neg F) = \begin{cases} 
1 & \text{if } \hat{A}(F) = 0 \\
0 & \text{otherwise}
\end{cases}
\]

\[
\hat{A}(F \land G) = \begin{cases} 
1 & \text{if } \hat{A}(F) = 1 \text{ and } \hat{A}(G) = 1 \\
0 & \text{otherwise}
\end{cases}
\]

\[
\hat{A}(F \lor G) = \begin{cases} 
1 & \text{if } \hat{A}(F) = 1 \text{ or } \hat{A}(G) = 1 \\
0 & \text{otherwise}
\end{cases}
\]

\[
\hat{A}(F \rightarrow G) = \begin{cases} 
1 & \text{if } \hat{A}(F) = 0 \text{ or } \hat{A}(G) = 1 \\
0 & \text{otherwise}
\end{cases}
\]

Instead of \(\hat{A}\) we simply write \(A\)
Truth tables (I)

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

<table>
<thead>
<tr>
<th>$\neg$</th>
<th>$A$</th>
<th>$A \lor B$</th>
<th>$A \land B$</th>
<th>$A \rightarrow B$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Using arithmetic:

$A(F \land G) = \min(A(F), A(G))$

$A(F \lor G) = \max(A(F), A(G))$
Abbreviations

A, B, C,

P, Q, R, or . . . instead of A₁, A₂, A₃ . . .

\( F_1 \leftrightarrow F_2 \) abbreviates \((F_1 \land F_2) \lor (\neg F_1 \land \neg F_2)\)

\( \bigvee_{i=1}^{n} F_i \) abbreviates \((\ldots ((F_1 \lor F_2) \lor F_3) \lor \ldots \lor F_n)\)

\( \bigwedge_{i=1}^{n} F_i \) abbreviates \((\ldots ((F_1 \land F_2) \land F_3) \land \ldots \land F_n)\)

Special cases:

\( \bigvee_{i=1}^{0} F_i = \bigvee \emptyset = \bot \)

\( \bigwedge_{i=1}^{0} F_i = \bigwedge \emptyset = \top \)
Truth tables (II)

<table>
<thead>
<tr>
<th>$A$</th>
<th>$B$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Lemma

Let $A_1$ and $A_2$ be two assignments. If $A_1(A_i) = A_2(A_i)$ for all atoms $A_i$ in some formula $F$, then $A_1(F) = A_2(F)$.

Proof.

Exercise.
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 $|= F$ if $F$ is valid, and $\not|= F$ 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 assignment that is a model of every formula in $S$. 
## Exercise

<table>
<thead>
<tr>
<th></th>
<th>Valid</th>
<th>Satisfiable</th>
<th>Unsatisfiable</th>
</tr>
</thead>
<tbody>
<tr>
<td>( A )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( A \lor B )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( A \lor \neg A )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( A \land \neg A )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( A \rightarrow \neg A )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( A \rightarrow (B \rightarrow A) )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( A \rightarrow (A \rightarrow B) )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( A \leftrightarrow \neg A )</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Exercise

Which of the following statements are true?

<table>
<thead>
<tr>
<th></th>
<th>Y</th>
<th>C.ex.</th>
</tr>
</thead>
<tbody>
<tr>
<td>If $F$ is valid, then $F$ is satisfiable</td>
<td></td>
<td></td>
</tr>
<tr>
<td>If $F$ is satisfiable, then $\neg F$ is satisfiable</td>
<td></td>
<td></td>
</tr>
<tr>
<td>If $F$ is valid, then $\neg F$ is unsatisfiable</td>
<td></td>
<td></td>
</tr>
<tr>
<td>If $F$ is unsatisfiable, then $\neg F$ is unsatisfiable</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Mirroring principle

<table>
<thead>
<tr>
<th>valid formulas</th>
<th>satisfiable but not valid formulas</th>
<th>unsatisfiable formulas</th>
</tr>
</thead>
<tbody>
<tr>
<td>$G$</td>
<td>$F$</td>
<td>$\neg G$</td>
</tr>
<tr>
<td></td>
<td>$\neg F$</td>
<td></td>
</tr>
</tbody>
</table>
Definition
A formula $G$ is a \textit{(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
$A \lor B, \ A \rightarrow B, \ B \land R \rightarrow \neg A, \ R \models (R \land \neg A) \land B$
Example

\[ A \vee B, \ A \rightarrow B, \ B \land R \rightarrow \neg A, \ R \models (R \land \neg A) \land B \]

Proof:
Assume \( \mathcal{A} \models F \) for all \( F \in M \).
We need to prove \( \mathcal{A} \models (R \land \neg A) \land B \).
From \( \mathcal{A} \models A \lor B \) and \( \mathcal{A} \models A \rightarrow B \) follows \( \mathcal{A} \models B \):
Proof by cases:
If \( \mathcal{A}(A) = 0 \) then \( \mathcal{A}(B) = 1 \) because \( \mathcal{A} \models A \lor B \)
If \( \mathcal{A}(A) = 1 \) then \( \mathcal{A}(B) = 1 \) because \( \mathcal{A} \models A \rightarrow B \)
From \( \mathcal{A} \models B \) and \( \mathcal{A} \models R \) follows \( \mathcal{A} \models \neg A \) because . . .
From \( \mathcal{A} \models B, \ \mathcal{A} \models R, \) and \( \mathcal{A} \models \neg A \) follows \( \mathcal{A} \models (R \land \neg A) \land B \)
Exercise

<table>
<thead>
<tr>
<th>$M$</th>
<th>$F$</th>
<th>$M \models F$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A$</td>
<td>$A \lor B$</td>
<td></td>
</tr>
<tr>
<td>$A$</td>
<td>$A \land B$</td>
<td></td>
</tr>
<tr>
<td>$A, B$</td>
<td>$A \lor B$</td>
<td></td>
</tr>
<tr>
<td>$A, B$</td>
<td>$A \land B$</td>
<td></td>
</tr>
<tr>
<td>$A \land B$</td>
<td>$A$</td>
<td></td>
</tr>
<tr>
<td>$A \lor B$</td>
<td>$A$</td>
<td></td>
</tr>
<tr>
<td>$A, A \rightarrow B$</td>
<td>$B$</td>
<td></td>
</tr>
</tbody>
</table>
Consequence

Exercise

The following statements are equivalent:

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

Proof of “if $F_1, \ldots, F_k \models G$ then $\models (\bigwedge_{i=1}^{k} F_i) \rightarrow G$”.

Assume $F_1, \ldots, F_k \models G$.
We need to prove $\models H$, i.e. $A(H) = 1$ for all $A$.
We pick an arbitrary $A$ and show $A(H) = 1$.
Proof by cases.

If $A(\bigwedge F_i) = 0$ then $A(H) = 1$ because $H = \bigwedge F_i \rightarrow G$
If $A(\bigwedge F_i) = 1$ then $A(F_i) = 1$ for all $i$.
Therefore $A$ is a model of $F_1, \ldots, F_k$.
Therefore $A \models G$ because $F_1, \ldots, F_k \models G$.
Therefore $A(H) = 1$
Validity and satisfiability

Exercise

The following statements are equivalent:

1. $F \implies G$ is valid.
2. $F \land \lnot G$ is unsatisfiable.
Exercise

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

<table>
<thead>
<tr>
<th>Y/N</th>
<th>C.ex.</th>
</tr>
</thead>
<tbody>
<tr>
<td>If $F$ satisfiable then $M \models F$.</td>
<td></td>
</tr>
<tr>
<td>If $F$ valid then $M \models F$.</td>
<td></td>
</tr>
<tr>
<td>If $F \in M$ then $M \models F$.</td>
<td></td>
</tr>
<tr>
<td>If $F \models G$ then $\neg F \models \neg G$.</td>
<td></td>
</tr>
</tbody>
</table>
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 $F \in S$, $M \models F$