Propositional Logic

Propositional Formulas are recursively defined by the following operators:

\[ \top \] always true
\[ \bot \] always false
\[ \land \] ... and ...
\[ \lor \] ... or ...
\[ \rightarrow \] if ... then ...
\[ \leftrightarrow \] ... if-and-only-if ...

In addition to this, there are propositional variables \( A, B, C, P, Q, R, \ldots \), whose truth value depends on the interpretation.
Semantics for Propositional Logic

An interpretation (also truth-assignment, valuation) of a set of propositional formulas $S$ is a function that assigns elements of $\{f, t\}$ to the propositional variables in $S$. 
Truth of a Formula in Propositional Logic

The following table defines how the propositional logical operators propagate their truth values:

<table>
<thead>
<tr>
<th>$A$</th>
<th>$B$</th>
<th>$\perp$</th>
<th>$\top$</th>
<th>$\neg A$</th>
<th>$A \lor B$</th>
<th>$A \land B$</th>
<th>$A \rightarrow B$</th>
<th>$A \leftrightarrow B$</th>
</tr>
</thead>
<tbody>
<tr>
<td>f</td>
<td>f</td>
<td>f</td>
<td>t</td>
<td>t</td>
<td>f</td>
<td>f</td>
<td>t</td>
<td>t</td>
</tr>
<tr>
<td>f</td>
<td>t</td>
<td>f</td>
<td>t</td>
<td>t</td>
<td>t</td>
<td>f</td>
<td>t</td>
<td>f</td>
</tr>
<tr>
<td>t</td>
<td>f</td>
<td>f</td>
<td>t</td>
<td>f</td>
<td>t</td>
<td>f</td>
<td>f</td>
<td>f</td>
</tr>
<tr>
<td>t</td>
<td>t</td>
<td>f</td>
<td>t</td>
<td>f</td>
<td>t</td>
<td>f</td>
<td>f</td>
<td>f</td>
</tr>
<tr>
<td>t</td>
<td>t</td>
<td>f</td>
<td>t</td>
<td>f</td>
<td>t</td>
<td>t</td>
<td>t</td>
<td>t</td>
</tr>
</tbody>
</table>
Truth of a Formula in an Interpretation

Consider the truth assignment $I_1$, defined by

$$I_1(P) = f, \quad I_1(Q) = t, \quad I_1(R) = t,$$

and $I_2$, which is defined by

$$I_2(P) = t, \quad I_2(Q) = f, \quad I_2(R) = t.$$

Then:

$$I_1(P \land (Q \rightarrow R)) = f, \quad I_2(P \land (Q \rightarrow R)) = t,$$

$$I_1(Q \leftrightarrow R) = t, \quad I_2(Q \leftrightarrow R) = f,$$

$$I_1(\neg P \rightarrow \neg Q) = f, \quad I_2(\neg P \rightarrow \neg Q) = t.$$
Some Basic Definitions:

Definition: Let $F$ be a formula, let $I$ be an interpretation. If $I(F) = t$, then $I$ is called a model of $F$. One usually writes $I \models F$.

Now let $F_1, \ldots, F_n$ be a sequence of formulas. Let $I$ be an interpretation. We say that $I$ is a model of $F_1, \ldots, F_n$, if for every $F_i$, $I \models F_i$. 
Basic Definitions (2)

Definition: Let $F$ be a formula. We call $F$ satisfiable if there exists an interpretation $I$, such that $I \models F$.

Similarly, we call a sequence $F_1, \ldots, F_n$ of formulas satisfiable if there exists an interpretation $I$, s.t. $I \models F_1, \ldots, F_n$.

Definition: A formula is universally valid if for every interpretation $I$,

$$I \models F.$$  

A formula $G$ is a logical consequence of $F_1, \ldots, F_n$, if for every interpretation $I$, for which

$$I \models F_1, \ldots, F_n,$$ also $I \models G$.

One usually writes $F_1, \ldots, F_n \models G$. 
Examples

The following formulas are universally valid:

\[ \neg A \lor A, \quad A \rightarrow A, \quad (\neg \neg A) \leftrightarrow A, \quad (A \lor B) \land (\neg A \lor C) \rightarrow B \lor C. \]

The right hand sides are consequences of the left hand sides:

\[ \neg \neg A \models A, \]

\[ A \rightarrow B, \quad B \rightarrow C \models \neg C \rightarrow \neg A, \]

\[ \neg (A \lor B) \models \neg A \land \neg B, \]

\[ \neg (A \land B) \models \neg A \lor \neg B, \]

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

\[ A, \neg A \models B. \]
Reducing Logical Consequence to Satisfiability

Theorem Let $F_1, \ldots, F_n$ be a sequence of formulas. Let $G$ be a formula.

Then $F_1, \ldots, F_n \models G$ iff $F_1, \ldots, F_n, \neg G$ is unsatisfiable.

proof On black board.

Nearly all automatic procedures are based on satisfiability testing.
Algorithms for Satisfiability Testing

- Simplest way: Write a table of all possible interpretations, and check if the formula is true in one of them. ⇒ Exponential.
  
- Write a backtracking tree. ⇒ Still exponential, but better.
  
- Use resolution or DPLL algorithm.

Satisfiability for propositional logic is NP-complete, so a better-than-exponential algorithm is unlikely.
Choosing your Algorithm

• Is efficiency important?
• If formula is unsatisfiable, do you want a proof?
• If formula is satisfiable, do you want a model? Do you want all models?
• Should the algorithm also work for non-propositional formulas?
Types of Deduction Systems

The most important types of deduction systems are:

- **Natural Deduction:** Natural Deduction follows the natural style of reasoning, as it can be found in mathematical textbooks or in spoken arguments. Most of the proof consists of forward reasoning, that is deriving conclusions, deriving new conclusions from these conclusions, etc. Occasionally additional assumptions are introduced or dropped.

- **Sequent Calculus:** In sequent calculus, conclusions and premises are treated in the same way. The reasoning proceeds by deriving relations between formulas, instead of deriving only conclusions. This is different from the style found in textbooks, but the resulting calculus is easier to use.
• **Axiomatic Method:** Axiomatic Methods are historically the oldest proof systems, but they are not important anymore. Their distinguishing feature is that logical operators are defined by axioms. There are usually three deduction rules, modus ponens:

If $A$ and $A \rightarrow B$ are provable, then so is $B$, generalization

If $A$ is provable, then so is $\forall x \ A$

and formula instantiation:

If $A$ is provable, then so is $A[X := F]$. 
Sequent Calculus for Propositional Logic

A multiset is a set that can distinguish how often an element occurs in it, (or alternatively it is a list, that cannot see the order of its elements).

Examples:

\[ A \lor B, \ A \land B, \ A \land B, \]
\[ A \lor B, \ A \land B, \ C \rightarrow D, \]
\[ A \land B, \ A \lor B, \ A \land B. \]

The first and the last multiset are equal.

A sequent is an object of form \( \Gamma \vdash \Delta \), in which \( \Gamma \) and \( \Delta \) are multisets of formulas.

The meaning is: Whenever all of the \( \Gamma \) are true, then at least one of the \( \Delta \) is true.
Structural Rules:

(axiom) \[ \Gamma, A \vdash \Delta, A \]

(cut) \[ \frac{\Gamma, A \vdash \Delta \quad \Gamma \vdash \Delta, A}{\Gamma \vdash \Delta} \]

Structural Rules:

(weakening left) \[ \frac{\Gamma \vdash \Delta}{\Gamma, A \vdash \Delta} \]

(weakening right) \[ \frac{\Gamma \vdash \Delta}{\Gamma \vdash \Delta, A} \]

(contraction left) \[ \frac{\Gamma, A, A \vdash \Delta}{\Gamma, A \vdash \Delta} \]

(contraction right) \[ \frac{\Gamma \vdash \Delta, A, A}{\Gamma \vdash \Delta, A} \]
Rules for the truth constants:

\[
\begin{align*}
\text{(⊤-left)} & \quad \frac{\Gamma \vdash \Delta}{\Gamma, \top \vdash \Delta} \\
\text{(⊥-left)} & \quad \frac{\Gamma \vdash \Delta}{\Gamma, \bot \vdash \Delta} \\
\text{(⊤-right)} & \quad \frac{\Delta, \top}{\Gamma, \Delta} \\
\text{(⊥-right)} & \quad \frac{\Delta, \bot}{\Gamma, \Delta}
\end{align*}
\]

Rules for ¬:

\[
\begin{align*}
\text{(¬-left)} & \quad \frac{\Gamma \vdash \Delta, A}{\Gamma, \neg A \vdash \Delta} \\
\text{(¬-right)} & \quad \frac{\Gamma, A \vdash \Delta}{\Gamma, \Delta, \neg A}
\end{align*}
\]
Rules for $\land$ and $\lor$:

($\land$-left) \[ \frac{\Gamma, A, B \vdash \Delta}{\Gamma, A \land B \vdash \Delta} \]  

($\land$-right) \[ \frac{\Gamma \vdash \Delta, A \quad \Gamma \vdash \Delta, B}{\Gamma \vdash \Delta, A \land B} \]

($\lor$-left) \[ \frac{\Gamma, A \vdash \Delta \quad \Gamma, B \vdash \Delta}{\Gamma, A \lor B \vdash \Delta} \]  

($\lor$-right) \[ \frac{\Gamma \vdash \Delta, A, B}{\Gamma \vdash \Delta, A \lor B} \]

(one can see from this, that premisses and conclusions are treated in completely the same way)
Rules for $\rightarrow$ and $\leftrightarrow$: 

$(\rightarrow$-left $) \quad \frac{\Gamma \vdash \Delta, A \quad \Gamma, B \vdash \Delta}{\Gamma, A \rightarrow B \vdash \Delta}$  

$(\rightarrow$-right $) \quad \frac{\Gamma, A \vdash \Delta, B}{\Gamma \vdash \Delta, A \rightarrow B}$

$(\leftrightarrow$-left $) \quad \frac{\Gamma, A \rightarrow B, B \rightarrow A \vdash \Delta}{\Gamma, A \leftrightarrow B \vdash \Delta}$

$(\leftrightarrow$-right $) \quad \frac{\Gamma \vdash \Delta, A \rightarrow B \quad \Gamma \vdash \Delta, B \rightarrow A}{\Gamma \vdash \Delta, A \leftrightarrow B}$
Correctness of Sequent Calculus

**Definition:** A sequent $\Gamma \vdash \Delta$ is valid if

- for every interpretation $I$, s.t. for all $A \in \Gamma$, $I(A) = t$,
- there is a $B \in \Delta$, s.t. $I(B) = t$.

**Theorem:** Every provable sequent is valid.

**proof:** It is enough to show preservation of validity for each of the rules. We do a few on the next slides:
Correctness of \(-\)-left

Assume that $\Gamma \vdash \Delta$, $A$ is a valid sequent. We show that $\Gamma, \neg A \vdash \Delta$ is a valid sequent as well.

In order to do this, we assume an arbitrary truth-assignment $I$, which makes all formulas $F \in \Gamma \cup \{\neg A\}$ true, and show that it makes one of the formulas $G \in \Delta$ true.

Assume that $I$ is a truth-assignment, s.t. for all $F \in \Gamma$, $I(F) = t$, and $I(\neg A) = t$. From the truth-table of $\neg$, it follows that $I(A) = f$.

Because $\Gamma \vdash \Delta$, $A$ is a valid sequent, there must be a formula $G \in \Delta \cup \{A\}$, s.t. $I(G) = t$. Since $I(A) = f$, it cannot be $A$. Therefore it must be a $G \in \Delta$. 
Correctness of $\lor$-left

Assume that $\Gamma, A \vdash \Delta$ and $\Gamma, B \vdash \Delta$ are valid sequents. We show that $\Gamma, A \lor B \vdash \Delta$ is a valid sequent as well.

Let $I$ be an interpretation, s.t. for all $F \in \Gamma \cup [A \lor B]$, we have $I(F) = t$. In particular, $I(A \lor B) = t$. From the truth-table of $\lor$, we see that either $I(A) = t$, or $I(B) = t$.

If $I(A) = t$, then all $F \in \Gamma \cup [A]$ are true. Therefore, we can use the validity of the first sequent to obtain that one $G \in \Delta$ is true.

Similarly, if $I(B) = t$, all $F \in \Gamma \cup [B]$ are true, and we can use the validity of the second sequent.

In both cases, there is a $G \in \Delta$, s.t. $I(G) = t$. 

Completeness of Sequent Calculus

For every deduction system/algorithm, one can ask the following questions:

- **correctness/soundness** Is every provable object valid?
- **completeness** Is every valid object provable?

In practice, correctness is more important than completeness. Many logics/deduction systems/algorithms, that are used in practice, are not complete, and nobody cares.
Completeness of Sequent Calculus

We define a algorithm Chck(Γ ⊢ Δ) that either returns a proof of Γ ⊢ Δ, or a counter interpretation.

- If Γ ⊢ Δ contains only propositional variables, then there are two possibilities:
  1. Γ ∩ Δ ≠ ∅. Γ ⊢ Δ can be written in form Γ′, A ⊢ Δ′, A.
     Return the proof that consists of the single axiom
     \[ Γ′, A ⊢ Δ′, A. \]
  2. Γ ∩ Δ = ∅. Construct a truth-assignment I as follows:
     For all \( F \in Γ \), \( I(F) = t \), for all \( G \in Δ \), \( I(G) = f \).
     Return I.
• If Γ ⊨ Δ has form Γ′, ⊥ ⊨ Δ, then return the proof consisting of the single axiom Γ′, ⊥ ⊨ Δ.

• If Γ ⊨ Δ has form Γ ⊨ Δ′, ⊥, then let

\[ \pi = \text{Chck}(\Gamma ⊨ \Delta′). \]

If \( \pi \) is an interpretation, then return \( \pi \). Otherwise, return

\[ \frac{\pi}{\Gamma ⊨ \Delta′, ⊥} \] (weakening right)
• If $\Gamma \vdash \Delta$ has form $\Gamma \vdash \Delta', \neg A$, then let

$$\pi = \text{Chck}(\Gamma, A \vdash \Delta').$$

If $\pi$ is an interpretation, then return $\pi$. Otherwise, return

$$\frac{\pi}{\Gamma \vdash \Delta', \neg A} (\neg\text{-right})$$

• If $\Gamma \vdash \Delta$ has form $\Gamma', \neg A \vdash \Delta$, then let

$$\pi = \text{Chck}(\Gamma' \vdash \Delta, A).$$

If $\pi$ is an interpretation, then return $\pi$. Otherwise, return

$$\frac{\pi}{\Gamma', \neg A \vdash \Delta} (\neg\text{-left})$$
• If $\Gamma \vdash \Delta$ has form $\Gamma \vdash \Delta', \ A \lor \ B$, then let

$$\pi = \text{Chck}(\Gamma \vdash \Delta', A, B).$$

If $\pi$ is an interpretation, then return $\pi$. Otherwise, return

$$\Gamma \vdash \Delta', A \lor B \quad (\lor\text{-right})$$

• If $\Gamma \vdash \Delta$ has form $\Gamma', A \lor B \vdash \Delta$, then let

$$\pi_1 = \text{Chck}(\Gamma', A \vdash \Delta), \ \pi_2 = \text{Chck}(\Gamma', B \vdash \Delta).$$

If $\pi_1$ is an interpretation, then return $\pi_1$. If $\pi_2$ is an interpretation, then return $\pi_2$.

(Both of them are proofs) Return

$$\Gamma', A \lor B \vdash \Delta \quad (\lor\text{-left})$$
Exercise

Fill in the missing cases.
Properties of Algorithm Chck

• Algorithm Chck terminates.

• If Chck(Γ ⊢ Δ) returns an interpretation $I$, then $I$ is a counter interpretation of $Γ ⊢ Δ$.

• If Chck(Γ ⊢ Δ) returns a proof $π$, then $π$ is a proof of $Γ ⊢ Δ$.

Completeness of sequent calculus follows from the correctness of Chck.
Some More Observations/Questions:

• In case, more than one rule can be applied, non-branching rules should be preferred over branching rules.

• Not all rules of propositional sequent calculus are used by algorithm Chck. Which rules? What can be concluded?

• What is the time complexity of Chck?

• What is its space complexity?