Propositional Logic and the Satisfiability Problem

- Propositional Logic
- The Satisfiability Problem
- Semantic Equivalence
- Normal Forms
- Complexity
Propositional Logic

Definition An alphabet of propositional logic consists of

- a (countably) infinite set $\mathcal{R} = \{p_1, p_2, \ldots\}$ of propositional variables,
- the set $\{\neg/1, \land/2, \lor/2, \rightarrow/2, \leftrightarrow/2\}$ of connectives, and
- the special characters “(” and “)”.

Indices are sometimes omitted.

We will occasionally use other letters to denote variables.

Different alphabets of propositional logic differ in $\mathcal{R}$.

Instead of $p_i$ we could have used $i$ as propositional variable.
Propositional Formulas

Definition  An atomic formula, briefly called atom, is a propositional variable.

Definition  The set of propositional formulas is the smallest set \( \mathcal{L}(\mathcal{R}) \) of strings over an alphabet of propositional logic with the following properties:

1. If \( F \) is an atomic formula, then \( F \in \mathcal{L}(\mathcal{R}) \).
2. If \( F \in \mathcal{L}(\mathcal{R}) \), then \( \neg F \in \mathcal{L}(\mathcal{R}) \).
3. If \( \circ/2 \) is a binary connective, \( F, G \in \mathcal{L}(\mathcal{R}) \), then \( (F \circ G) \in \mathcal{L}(\mathcal{R}) \).

Definition  A literal is an atom, or a negated atom. The complement \( \overline{L} \) of a literal \( L \) is defined as follows:

- If \( L \) is an atom \( A \), then \( \overline{L} = \neg A \),
- if \( L \) is a negated atom \( \neg A \), then \( \overline{L} = A \).

A pair \( L, \overline{L} \) of literals is said to be complementary.
Some Notation

- **Notation**
  - $A$ (possibly indexed) denotes an atom,
  - $L$ (possibly indexed) denotes a literal,
  - $F, G, H$ (possibly indexed) denote propositional formulas,
  - $\mathcal{F}, \mathcal{G}, \mathcal{H}$ denote sets of propositional formulas.
3.2 Semantics

- The set of truth values $\mathcal{W}$ is the set $\{\top, \bot\}$.

- We consider the following functions on $\mathcal{W}$:
  - Negation $\neg^* / 1$.
  - Conjunction $\land^* / 2$.
  - Disjunction $\lor^* / 2$.
  - Implication $\rightarrow^* / 2$.
  - Equivalence $\leftrightarrow^* / 2$.

\[
\begin{array}{c|ccccc}
\text{Truth Values} & \neg^* & \land^* & \lor^* & \rightarrow^* & \leftrightarrow^* \\
\hline
\top & \bot & \top & \top & \top & \top \\
\top & \bot & \bot & \top & \bot & \bot \\
\bot & \top & \bot & \top & \bot & \bot \\
\bot & \bot & \bot & \bot & \bot & \bot \\
\end{array}
\]
3.2.2 Interpretations, Models and Logic Consequences

Definition An interpretation \( I = (\mathcal{W}, \cdot^I) \) consists of the set \( \mathcal{W} \) and a mapping \( \cdot^I : \mathcal{L}(\mathcal{R}) \rightarrow \mathcal{W} \) with:

\[
[F]^I = \begin{cases} 
  w \in \mathcal{W} & \text{if } F \in \mathcal{R}, \\
  \neg^* [G]^I & \text{if } F \text{ is of the form } \neg G, \\
  ([G_1]^I \circ^* [G_2]^I) & \text{if } F \text{ is of the form } (G_1 \circ G_2).
\end{cases}
\]

Definition An interpretation \( I = (\mathcal{W}, \cdot^I) \) is called a model for a propositional formula \( F \), in symbols \( I \models F \), if \( [F]^I = \top \); in this case we say that \( I \) satisfies \( F \).

Definition \( F \) is unsatisfiable if it has no models. \( F \) is valid if all interpretations are models.

Definition An interpretation \( I = (\mathcal{W}, \cdot^I) \) is called a model for a set \( \mathcal{G} \) of propositional formulas, in symbols \( I \models \mathcal{G} \), if \( [F]^I = \top \) for all \( F \in \mathcal{G} \); in this case we say that \( I \) satisfies \( \mathcal{G} \).
Representation of Interpretations

- An interpretation $I = (\forall, \cdot^I)$ is uniquely defined by specifying how $\cdot^I$ acts on atoms.
  - $I$ can be represented by $\hat{I} = \{ L \in \mathcal{L}(\mathcal{R}) \mid [L]^I = \top \}$.

- **Note**
  - $\hat{I}$ does not contain a complementary pair of literals.
  - $\cdot^I$ is a total mapping from $\mathcal{L}(\mathcal{R})$ to $\forall$.
    - Hence, for each $A \in \mathcal{L}(\mathcal{R})$ either $A \in \hat{I}$ or $\overline{A} \in \hat{I}$ but not both.

- In the sequel, we will identify $I$ with $\hat{I}$.

- **Definition** Let $J$ be a set of literals not containing a complementary pair. $J$ is a partial interpretation if there is an $A$ such that neither $A \in J$ nor $\overline{A} \in J$. 
Some Additional Notation

- $I, J$ (possibly indexed) denote interpretations.
- We often write $F^I$ instead of $[F]^I$.
- We define the following precedence hierarchy among connectives:
  
  $\neg \succ \{\lor, \land\} \succ \rightarrow \succ \leftrightarrow$ .

- We sometimes omit parentheses taking into account that conjunction and disjunction are associative and commutative.
Propositional Satisfiability Problems

Definition A propositional satisfiability problem, briefly called SAT, consists of a formula $F \in \mathcal{L}(\mathcal{R})$, and is the problem to decide whether $F$ is satisfiable.

SAT is a combinatorial decision problem.

- Decision variant yes/no answer.
- Search variant find a model if $F$ is satisfiable.
A Simple SAT Instance

- Let \( F = p_1 \)
  \[ \land (p_1 \lor p_2) \]
  \[ \land (p_1 \rightarrow p_3) \]
  \[ \land (p_1 \land p_3 \rightarrow p_4) \]
  \[ \land (p_5 \lor p_6) \]
  \[ \land (p_5 \rightarrow p_7) \]
  \[ \land (\neg p_5 \lor p_8) \]
  \[ \land (\neg p_7 \lor \neg p_8). \]

- \( \{p_1, p_2, p_3, p_4, \neg p_5, p_6, \neg p_7, \neg p_8\} \cup \{p_i \mid 8 < i\} \) is a model for \( F \).

- Hence, \( F \) is satisfiable.

- How can we find such a model?
Subformulas

► Definition Let $F$ be a propositional formula. The set of subformulas of $F$ is the smallest set of formulas $S(F)$ satisfying the following conditions:

1. $F \in S(F)$.
2. If $\neg G \in S(F)$, then $G \in S(F)$.
3. If $(G_1 \circ G_2) \in S(F)$, then $G_1, G_2 \in S(F)$.

► Example

\[ S(\neg((p_1 \rightarrow p_2) \lor p_1)) \]
\[ = \{ \neg((p_1 \rightarrow p_2) \lor p_1), ((p_1 \rightarrow p_2) \lor p_1), (p_1 \rightarrow p_2), p_1, p_2 \}. \]
Semantic Equivalence

- **Definition** Two propositional formulas $F$ and $G$ are **semantically equivalent**, in symbols $F \equiv G$, if for all interpretations $I$ we have: $I \models F$ iff $I \models G$.

- **Some equivalence laws**:

  - $\neg\neg F \equiv F$  \hspace{1cm} \text{double negation}
  - $\neg(F \land G) \equiv (\neg F \lor \neg G)$  \hspace{1cm} \text{de Morgan}
  - $\neg(F \lor G) \equiv (\neg F \land \neg G)$
  - $(F \land (G \lor H)) \equiv ((F \land G) \lor (F \land H))$  \hspace{1cm} \text{distributivity}
  - $(F \lor (G \land H)) \equiv ((F \lor G) \land (F \lor H))$
  - $(F \leftrightarrow G) \equiv ((F \land G) \lor (\neg G \land \neg F))$  \hspace{1cm} \text{equivalence}
  - $(F \rightarrow G) \equiv (\neg F \lor G)$  \hspace{1cm} \text{implication}
  - $(F \lor G) \equiv F$, if $F$ is valid
  - $(F \land G) \equiv G$, if $F$ is valid  \hspace{1cm} \text{tautology}
  - $(F \lor G) \equiv G$, if $F$ is unsatisfiable
  - $(F \land G) \equiv F$, if $F$ is unsatisfiable  \hspace{1cm} \text{unsatisfiability}
Replacement

- $F[G \leftrightarrow H]$ denotes the formula obtained from $F$ by replacing an occurrence of $G \in S(F)$ by $H$.

  - Usually, the context determines which occurrence is meant.
  - Sometimes the condition $G \in S(F)$ is omitted. In this case, if $G \notin S(F)$, then $F[G \leftrightarrow H] = F$.

- Replacement Theorem If $G \equiv H$, then $F[G \leftrightarrow H] \equiv F$.

- Exercise Proof the replacement theorem by structural induction.
Generalized Disjunctions and Conjunctions

- **Generalized disjunction**

\[
[F_1, \ldots, F_n] = (\ldots ((F_1 \lor F_2) \lor F_3) \lor \ldots \lor F_n)
\]

- **Generalized conjunction**

\[
\langle F_1, \ldots, F_n \rangle = (\ldots ((F_1 \land F_2) \land F_3) \land \ldots \land F_n)
\]

- **Empty generalized disjunction:** \([\ ]\) with \([\ ]^I = \bot\) for all \(I\).

- **Empty generalized conjunction:** \(\langle \rangle\) with \(\langle \rangle^I = \top\) for all \(I\).

- We may extend our language by adding \(\langle \rangle\) and \([\ ]\) to the alphabet and treating them as atoms.
Conjunctive Normal Form

- **Definition**
  - A clause is a generalized disjunction \([L_1, \ldots, L_n]\), \(n \geq 0\), where every \(L_i, 1 \leq i \leq n\), is a literal.
  - A formula is in conjunctive normal form (clause form, CNF) iff it is of the form \(\langle C_1, \ldots, C_m\rangle\), \(m \geq 0\), and if every \(C_j, 1 \leq j \leq m\), is a clause.

- A formula \(F\) in CNF is said to be in \(n\)-CNF if each clause occurring in \(F\) has at most \(n\) literals.
More on Clauses

- A clause is a Horn clause if at most one disjunct is an atom.
- A formula $F$ in CNF is a Horn formula if it contains only Horn clauses.
- A clause is a unit clause if it contains precisely one literal.
- A clause is a binary clause if it contains precisely two literals.
- Observation A clause containing a complementary pair of literals is valid.
More Notation

- \( C \) (possibly indexed) denotes a clause.

- Clauses and formulas in CNF are sometimes considered as sets of literals and clauses, respectively, in which case
  - \( L_i, 1 \leq i \leq n \), are said to be elements of \([L_1, \ldots, L_n]\) and
  - \( C_j, 1 \leq j \leq m \), are said to be elements of \(\langle C_1, \ldots, C_m \rangle\).

Note that in sets duplicates are removed!
Transformation into Clause Form

- **Theorem** There is an algorithm which transforms any propositional formula into a semantically equivalent formula in clause form.

- **Observation**
  - All equivalences can be eliminated using the law
    \[ F \leftrightarrow G \equiv (F \land G) \lor (\neg F \land \neg G). \]
  - All implications can be eliminated using the law
    \[ F \rightarrow G \equiv \neg F \lor G. \]
  - Hence, we can assume that only the connectives \( \neg, \land \) and \( \lor \) occur in formulas.
An Algorithm for the Transformation into Clause Form

- **Input:** A propositional formula $F$.
- **Output:** A formula, which is in conjunctive normal form and is equivalent to $F$.

$G := \langle[F]\rangle$. ($G$ is a conjunction of disjunctions.)

While $G$ is not in conjunctive normal form do:
- Select a non-clausal element $H$ from $G$.
- Select a non-literal element $K$ from $H$.
- Apply the rule among the following ones which is applicable.

\[
\begin{align*}
\neg\neg D & \quad \neg(D_1 \lor D_2) \\
D & \quad D_1 \lor D_2 \\
\neg(D_1 \land D_2) & \quad D_1, D_2 \\
\neg D_1, \neg D_2 & \quad \neg(D_1 \lor D_2) \\
\end{align*}
\]

- A rule $\frac{D}{D'}$ is applicable to $K$ if $K$ is of the form $D$.
- If applied, then $K$ is replaced by $D'$.

- A rule $\frac{D}{D_1|D_2}$ is applicable to $K$ if $K$ is of the form $D$.
- If applied, $H$ is replaced by two disjunctions:
  - The first one is obtained from $H$ by replacing the occurrence of $D$ by $D_1$.
  - The second one is obtained from $H$ by replacing the occurrence of $D$ by $D_2$.
An Example

- Let $F = (p \land (p \rightarrow q)) \rightarrow q$.
- $F$ is valid.
- Eliminating implications yields:
  $$\neg(p \land (\neg p \lor q)) \lor q.$$
- Applying the algorithm yields:
  $$\langle [\neg(p \land (\neg p \lor q)) \lor q] \rangle$$
  $$\langle [\neg(p \land (\neg p \lor q), q] \rangle$$
  $$\langle [\neg p, \neg (\neg p \lor q), q] \rangle$$
  $$\langle [\neg p, \neg p \land \neg q, q] \rangle$$
  $$\langle [\neg p, \neg p, q], [\neg p, \neg q, q] \rangle$$
  $$\langle [\neg p, p, q], [\neg p, \neg q, q] \rangle$$
- Both clauses in the final formula contain a complementary pair of literals.
Definitional Transformation

- The size of a formula may grow exponentially during normalization.
- Can we do better?
  - Unfortunately, the shortest CNF of some $F$ is exponential in the size of $F$.
  - Luckily, we may use a weaker concept.

Definitional transformation (Eder, 1985)

- Let $F$ be a formula, $G \in S(F)$ and $p \notin S(F)$ a propositional variable.
- Replace $F$ by $F[G \leftrightarrow p] \land (p \leftrightarrow G)$.

Some observations

- $F \neq F[G \leftrightarrow p] \land (p \leftrightarrow G)$.
- $F$ is satisfiable iff $F[G \leftrightarrow p] \land (p \leftrightarrow G)$ is satisfiable.
- The above mentioned exponential growth can be avoided.
Reduct of a CNF-Formula

Definition Let $F$ be a CNF-formula and $J$ a partial interpretation. The reduct of $F$ wrt $J$, in symbols $F_J$, is obtained by applying the following transformations to $F$:

1. For all $L \in J$ do:
   - if $L = A$, then replace each occurrence of $A$ in $F$ by $\langle \rangle$.
   - if $L = \overline{A}$, then replace each occurrence of $A$ in $F$ by $[\ ]$.

2. Eliminate all occurrences of $[\ ]$ and $\langle \rangle$ by applying the replacement theorem using the laws of validity and unsatisfiability.

Let $F$ be the following formula:

$\langle [p_1], [p_1, p_2], [\neg p_1, p_3], [\neg p_1, \neg p_3, p_4], [p_5, p_6], [\neg p_5, p_7], [\neg p_5, p_8], [\neg p_7, \neg p_8]\rangle$.

Then,

$F_{\{p_1\}} = \langle [p_3], [\neg p_3, p_4], [p_5, p_6], [\neg p_5, p_7], [\neg p_5, p_8], [\neg p_7, \neg p_8]\rangle$.

$F_{\{p_1,p_3\}} = \langle [p_4], [p_5, p_6], [\neg p_5, p_7], [\neg p_5, p_8], [\neg p_7, \neg p_8]\rangle$.

$F_{\{p_1,p_3,p_4\}} = \langle [p_5, p_6], [\neg p_5, p_7], [\neg p_5, p_8], [\neg p_7, \neg p_8]\rangle$. 
Computational Complexity

- Let \( \mathcal{P} \) be the class of problems that can be solved by a deterministic Turing machine in polynomial time.

- Let \( \mathcal{NP} \) be the class of problems that can be solved by a nondeterministic Turing machine in polynomial time.

- Obviously, \( \mathcal{P} \subseteq \mathcal{NP} \).

- The question whether \( \mathcal{NP} \subseteq \mathcal{P} \) holds is open.

- A problem that is at least as hard as any other problem in \( \mathcal{NP} \) is called \( \mathcal{NP} \)-hard (in the sense that each problem in \( \mathcal{NP} \) can be polynomially reduced to it).

- An \( \mathcal{NP} \)-hard problem which is in \( \mathcal{NP} \) is called \( \mathcal{NP} \)-complete.
Computational Complexity and SAT

- Let $F$ be a formula containing 1000 different propositional variables.
- There are $2^{1000}$ different interpretations for $F$.
- A truth table for $F$ has $2^{1000}$ rows.
- SAT is $\mathcal{NP}$-complete.
- This holds even if SAT is restricted to CNF or even 3-CNF formulae (Garey, Johnson 1979).
- SAT is decidable in linear time for DNF, for 2-CNF (Cook 1971) or for Horn formulae (Dowling, Gallier 1984; Scutella 1990).