Lecture 1: Propositional Logic

- Syntax
- Semantics
- Truth tables
- Implications and Equivalences
- Normal forms
- Valid and Invalid Arguments
- Davis-Putnam Algorithm
- Normal forms
An atomic proposition is a statement or assertion that must be true or false.

Propositional formulas are constructed from atomic propositions by using logical connectives.

Examples of atomic propositions are: "5 is a prime" and "Program P terms ales a."
Well-formed propositional formulas

Well-formed formulas of propositional logic are obtained by using the construction rules below:

- An atomic proposition is a well-formed formula.

If \( \phi \) is a well-formed formula, then so is \( \neg \phi \).

\[ \begin{align*}
\phi & \iff \phi, \\
\phi & \iff \phi, \wedge \phi, \\
\phi & \iff \phi, \vee \phi, \\
\phi & \iff \text{atomic proposition}
\end{align*} \]

Alternatively, can use Backus-Naur Form (BNF):

\[
\text{formula} ::= \text{Atomic Proposition} \mid \text{formula} \mid \text{formula} \mid \text{formula} \mid \text{formula} \mid \text{formula} \mid \text{formula} \mid \text{formula} \mid \text{formula} \mid \text{formula}
\]

If \( \phi \) and \( \psi \) are well-formed formulas, then so are \( \psi \land \phi \) and \( \phi \lor \psi \).

If \( \phi \) is a well-formed formula, then so is \( \neg \phi \).

An atomic proposition is a well-formed formula.

Below:

The well-formed formulas of propositional logic are obtained by using the construction rules.

\[
\text{well-formed formulas}
\]
Truth functions are sometimes called Boolean functions.

The other logical connectives can be handled in a similar manner:

<table>
<thead>
<tr>
<th>$([a]B)'$</th>
<th>$[a]B'$</th>
</tr>
</thead>
<tbody>
<tr>
<td>max</td>
<td>$B \land \forall a$</td>
</tr>
<tr>
<td>min</td>
<td>$B \lor \forall a$</td>
</tr>
<tr>
<td>$I$</td>
<td>$B \land \forall a$</td>
</tr>
<tr>
<td>$0$</td>
<td>$B \lor \forall a$</td>
</tr>
<tr>
<td>$[a]S$</td>
<td>$S'$</td>
</tr>
</tbody>
</table>

If we identify 0 with false and 1 with true, we can easily determine the truth value of

\[ (u_x, \ldots, x_n, x_n) \]$ under a \( \xi \).

Let \( \xi \) be a truth assignment for \( x_1, x_2, \ldots \). Let \( \xi \) be a truth assignment that associates a truth value with each of the atomic propositions \( x_1, x_2, \ldots \). A truth assignment is a mapping that associates a truth value with each of the atomic propositions \( x_1, x_2, \ldots \).

The truth of a propositional formula \( S \) is a function of the truth values of the atomic propositions \( x_1, x_2, \ldots \).
A truth table shows whether a propositional formula is true or false for each possible truth assignment.

If we know how the five basic logical connectives work, it is easy (in principle) to construct a truth table.

<table>
<thead>
<tr>
<th>X</th>
<th>X</th>
<th>X</th>
<th>X</th>
<th>X</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</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>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>A (\iff) X</th>
<th>A (\rightarrow) X</th>
<th>A (\land) X</th>
<th>A (\lor) X</th>
<th>A (\neg) X</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

Truth tables for basic logical connectives
Mistake in table for implication?
Mistake in table for implication?

First Argument:

- If we used T3, then X \land \neg \Rightarrow \land would have the same table as X \land \neg even worse!
- If we used T2, then X \land \neg \Rightarrow \land would have the same table as X \land.
- If we used T1, then X \land \neg \Rightarrow \land would have the same table as X \land \forall.

Clearly, each of these three alternatives is unreasonable. Table T4 is the only remaining possibility.

---

Mistake in table for implication?
Mistake in table for implication?

<table>
<thead>
<tr>
<th></th>
<th>T1</th>
<th>T2</th>
<th>T3</th>
<th>T4</th>
<th>(A \land A) \leftarrow X</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>I</td>
</tr>
<tr>
<td>I</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>I</td>
</tr>
<tr>
<td>I</td>
<td>I</td>
<td>0</td>
<td>0</td>
<td>I</td>
<td>I</td>
</tr>
<tr>
<td>I</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Only T4 makes the implication a tautology.

Second Argument: We would certainly want (A \land X) \leftarrow X to be a tautology. Let's test each of the four possible choices for X.

\[
\begin{align*}
\lambda \leftarrow X, & \\
\end{align*}
\]
In general, if the truth of a formula depends on \( n \) propositions, its truth table will have \( 2^n \) rows.

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>I</td>
</tr>
<tr>
<td>I</td>
<td>0</td>
<td>I</td>
<td>0</td>
<td>0</td>
<td>I</td>
<td>I</td>
</tr>
<tr>
<td>I</td>
<td>I</td>
<td>0</td>
<td>I</td>
<td>I</td>
<td>0</td>
<td>I</td>
</tr>
<tr>
<td>I</td>
<td>0</td>
<td>0</td>
<td>I</td>
<td>0</td>
<td>0</td>
<td>I</td>
</tr>
<tr>
<td>I</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>0</td>
</tr>
<tr>
<td>I</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>I</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

In this case there are \( 2^3 = 8 \) such truth assignments. Hence, the table for \( S \) will have 8 rows.

To construct the truth table for \( S \), we must consider all possible truth assignments for \( A \), \( B \), and \( C \).

\[
S \leftarrow A \quad B \leftarrow A \quad C \leftarrow (A \lor B) \lor (A \land C)
\]

Let \( S \) be the formula

A more complex truth table
A propositional formula is a tautology if
\[ \forall x. \] is a tautology.

A formula is a contradiction if
\[ \forall x. \neg \] is a contradiction.

A formula is satisfiable if
\[ \exists x. \] is satisfiable.

\textbf{Major open problem:} Is there a more efficient way to determine if a formula is a tautology (is satisfiable) than by constructing its truth table?

\textbf{Note that:}

The truth table on the previous page shows that the formula \( S \) is a tautology.

It is easy to see that \( \forall x. \neg \lor \) is a tautology and that \( \forall x. \lor \neg \) is a contradiction.

\textbf{Special formulas:}

\begin{itemize}
  \item satisfiable if \( S\mid a = 1 \) for some a.
  \item a contradiction if \( S\mid a = 0 \) for all a.
  \item a tautology if \( S\mid a = 1 \) for all a.
\end{itemize}
Implications

In the formula $A \implies B$:

- $A$ is the antecedent, hypothesis or premise.
- $B$ is the consequent, hypothesis or conclusion.

There are 3 variants:

- **Converse**: $B \implies A$
- **Inverse**: $\neg A \implies \neg B$
- **Contrapositive**: $\neg B \implies \neg A$

Can be associated with 3 variants:

- $A$ is the antecedent, hypothesis or premise.
- $B$ is the consequent, hypothesis or conclusion.
- In the formula $A \implies B$

*Modus Tollens*: Given $A$ and $\neg B$, conclude $\neg A$.

*Modus Ponens*: Given $A$ and $B$, conclude $B$. An implication and its contrapositive are equivalent.
Equivalences

Two formulae \( \phi \) and \( \psi \) are equivalent iff for any truth assignment \( \alpha \), we have \( \phi[\alpha] = \psi[\alpha] \).

Claim: \( S \) and \( T \) are equivalent iff \( S \) is a tautology.

Some Useful Equivalences that can be used to simplify complex formulas:

1. \( B \land A \leftrightarrow B \leftrightarrow A \)
2. \((B \rightarrow A) \leftrightarrow B \lor A \)
3. \((C \lor A) \land (B \lor A) \leftrightarrow (C \land B) \lor A \)
4. \(C \lor (B \lor A) \leftrightarrow (C \lor B) \lor A \)
5. \(A \lor B \leftrightarrow B \lor A \)
6. \(A \iff A \lor A \)
7. \(A \iff A \lor A \)
8. \(0 \iff 0 \lor A \)

Equivalences
When is an argument valid?

An argument is an assertion that a set of statements, called the premises, yields another statement, called the conclusion. An argument is valid if and only if the conjunction of the premises implies the conclusion. In other words, if we grant that the premises are all true, then the conclusion must be true also.

An invalid argument is called a fallacy. Unfortunately, fallacies are probably more common than valid arguments. In many cases, the validity of an argument can be checked by constructing a truth table. All we have to do is show that the conjunction of the premises implies the conclusion.
Which of the following arguments are valid?

1. If I am wealthy, then I am happy. I am happy. Therefore, I am wealthy.
2. If John drinks beer, he is at least 18 years old. John does not drink beer. Therefore, John is not yet 18 years old.
3. If girls are blonde, they are popular with boys. Ugly girls are unpopular with boys. Intellectual girls are ugly. Therefore, blonde girls are not intellectual.
4. If I study, then I will not fail basket weaving 101. If I do not play cards too often, then I will study. Therefore, I played cards too often.

Valid and Invalid Arguments
The following example is due to Lewis Carroll. Prove that it is a valid argument.

1. All the dated letters in this room are written on blue paper.
2. None of them are in black ink, except those that are written on more than one sheet.
3. I have not read any of those that I can read.
4. None of those that are written on one sheet are undated.
5. All of those that are not crossed out are in black ink.
6. All of those that are written by Brown begin with "Dear Sir."
7. All of those that are written on blue paper are filed.
8. None of those that are written on more than one sheet are crossed out.
9. None of those that begin with "Dear Sir." are written in the third person.

Therefore, I cannot read any of Brown’s letters.

A More Complicated Example:
Let 

```
fi
x
m
n
n
p
s
b
d
```

Let

Lewis Carroll example (cont.)
Now, we can write the argument in propositional logic:

Therefore $n \leftarrow x$

6

$s \leftarrow h$

7

$m \leftarrow n$

8

$t \leftarrow b$

5

$h \leftarrow x$

6

$t \leftarrow m$

4

$d \leftarrow a$

3

$n \leftarrow t$

2

$s \leftarrow t$

1

$b \leftarrow d$
Some more useful equivalences:

\begin{align*}
(\emptyset \vdash \lor d) & \iff (\emptyset \vdash d) \downarrow \bullet \\
(\emptyset \vdash \land d) & \iff (\emptyset \vdash d) \downarrow \bullet \\
(\emptyset \vdash \lor d) & \iff (\emptyset \vdash d) \downarrow \bullet \\
\lor d & \iff d \downarrow \bullet \\
\end{align*}
The resulting formula is said to be in negation normal form.

\[
(S \lor (\neg H \land d \neg)) \lor \emptyset
\]

\[
(S \lor ([\neg H \neg] \land d \neg)) \lor \emptyset
\]

\[
(S \lor ([\neg H \lor d] \neg)) \lor \emptyset
\]

\[
([\neg H \neg] \lor ([\neg H \lor d] \neg)) \lor \emptyset
\]

\[
([S \neg] \land ([\neg H \lor d] \neg)) \lor \emptyset
\]

\[
([S \neg] \land ([\neg H \lor d] \neg)) \leftarrow \emptyset
\]

This may not be very useful. Often desirable to simplify formula as much as possible using tautologies above.

\[
([S \neg] \land ([\neg H \lor d] \neg)) \leftarrow \emptyset
\]

is simply

\[
(S \neg \land (H \lor d)) \leftarrow \emptyset
\]

The negation of

**Negation Normal Form**
Every propositional formula is equivalent to a formula in disjunctive normal form (DNF):

\[
\bigwedge (\bigvee (\bigwedge (\bigvee (\bigwedge \cdots \cdots \bigvee \bigwedge \cdots \cdots \cdots)) \cdots)
\]

where each \( \varphi \) is a literal.

In short:

\[
\forall \lnot \forall
\]

Every propositional formula is equivalent to a formula in conjunctive normal form (CNF):

\[
\bigvee (\bigwedge (\bigvee (\bigwedge \cdots \cdots \bigwedge \cdots \cdots \cdots)) \cdots)
\]

where each \( \varphi \) is a literal.

In short:

\[
\forall \forall \lnot
\]

Disjunctive Normal Form

Satisfiability instead?

How hard is it to check if CNF formula is a tautology? How about DNF? How about checking for satisfiability instead?
Connectives

\[ (\bar{x} \lor x) \leftrightarrow (\bar{x} \land x) \]

Consider the binary connective \( \times \).

\[ (\bar{x} \land x) \leftrightarrow (\bar{x} \lor x) \]

\[ \text{Claim: } \{ \times \} \text{ alone is sufficient.} \]

\[ \text{But is not sufficient.} \]

\[ \text{Likewise, } \{ \land \} \text{ is not sufficient.} \]

\[ \text{Likewise, } \{ \lor \} \text{ is not sufficient.} \]

\[ \text{Likewise, } \{ \neg \} \text{ is not sufficient.} \]

\[ \{ \land \} \text{ alone is sufficient.} \]

Since \( A \lor B \) is equivalent to \( \neg (\neg A \land \neg B) \), we only need \( \{ \land \} \) and \( \{ \neg \} \), which are really needed.

From CNF (or DNF) it follows that no connectives other than \( \{ \land \}, \{ \lor \}, \{ \neg \} \) are really needed.
Deciding satisfiability

The fastest known algorithms for deciding propositional satisfiability are based on the

*Davis-Putnam Algorithm.*

A *unit clause* is a clause that consists of a single literal.

**end function**

```
end function
else return FALSE end if
else if Satisfiable (T \ \ S) then return TRUE
else if Satisfiable (S \ \ T) then return TRUE
else if S is empty then return TRUE

end for

while no further changes result end repeat
else if null clause is in S then return FALSE end if

choose a literal occurring in S

/* splitting */

for each clause of T in which it occurs delete from T
for each clause of S \ \ T do delete from S
```

```