Why Logic?

- Logic is one of the best developed system for representing knowledge.
- Can be used for analysis, design and specification.
- Understanding formal logic is a prerequisite for understanding most research papers in KRR.
The Right Logic . . .

- Logics of different orders (1st, 2nd, ...)
  - Modal logics
    - epistemic
    - temporal
    - dynamic (program)
    - multi-
    - ...
  - Many-valued logics
  - Conditional logics
  - Nonmonotonic logics
  - Linear logics
  - ...

Classical Logic

Propositional Logic
Syntax
Semantics
Terminology
Normal forms
Decision Problems
Resolution
The Right Logic . . .

- Logics of different orders (1st, 2nd, ...)
- **Modal** logics
  - epistemic
  - temporal
  - dynamic (program)
  - multi-
  - . . .
- Many-valued logics
- Conditional logics
- Nonmonotonic logics
- Linear logics
- . . .
Logics of different orders (1st, 2nd, ...)

Modal logics
- epistemic
- temporal
- dynamic (program)
- multi-
- ...

Many-valued logics

Conditional logics

Nonmonotonic logics

Linear logics

...
The Right Logic . . .

- Logics of different orders (1st, 2nd, ...)
- **Modal** logics
  - epistemic
  - temporal
  - dynamic (program)
  - multi-
  - . . .
- **Many-valued** logics
  - Conditional logics
  - Nonmonotonic logics
  - Linear logics
  - . . .
The Right Logic . . .

- Logics of different **orders** (1st, 2nd, ...)
- **Modal** logics
  - epistemic
  - temporal
  - dynamic (program)
  - multi-
  - . . .
- **Many-valued** logics
- **Conditional** logics
  - Nonmonotonic logics
  - Linear logics
  - . . .
The Right Logic . . .

- Logics of different orders (1st, 2nd, ...)
- **Modal** logics
  - epistemic
  - temporal
  - dynamic (program)
  - multi-
  - . . .
- **Many-valued** logics
- **Conditional** logics
- **Nonmonotonic** logics
  - Linear logics
  - . . .
The Right Logic . . .

- Logics of different orders (1st, 2nd, ...)
- **Modal** logics
  - epistemic
  - temporal
  - dynamic (program)
  - multi-
  - . . .
- **Many-valued** logics
- **Conditional** logics
- **Nonmonotonic** logics
- **Linear** logics
  - . . .
The Right Logic . . .

- Logics of different orders (1st, 2nd, ...)
- **Modal** logics
  - epistemic
  - temporal
  - dynamic (program)
  - multi-
  - . . .
- **Many-valued** logics
- **Conditional** logics
- **Nonmonotonic** logics
- **Linear** logics
- . . .
The Logical Approach

- Define a **formal language**
- logical & non-logical symbols, syntax rules
- Provide language with **compositional semantics**
  - Fix *universe* of discourse
  - Specify how the non-logical symbols can be *interpreted* interpretation
  - Rules how to combine interpretation of single symbols
  - Satisfying interpretation = model
  - From that logical implication/entailment follows

- Specify a **calculus** that allows to derive new formulae from old ones – according to the entailment relation
The Logical Approach

- Define a **formal language**
- logical & non-logical symbols, syntax rules
- Provide language with **compositional semantics**
  - Fix **universe** of discourse
  - Specify how the non-logical symbols can be **interpreted**
  - Interpretation
  - Rules how to **combine** interpretation of single symbols
  - Satisfying interpretation = model
  - From that logical implication/entailment follows
- Specify a **calculus** that allows to derive new formulae from old ones – according to the entailment relation
The Logical Approach

- Define a **formal language**
- logical & non-logical symbols, syntax rules
- Provide language with **compositional semantics**
  - Fix **universe** of discourse
  - Specify how the non-logical symbols can be **interpreted**
  - interpretation
  - Rules how to **combine** interpretation of single symbols
  - **Satisfying interpretation = model**
  - From that logical implication/entailment follows

- Specify a **calculus** that allows to derive new formulae from old ones – according to the entailment relation
The Logical Approach

- Define a **formal language**
  - logical & non-logical symbols, syntax rules
- Provide language with **compositional semantics**
  - Fix **universe** of discourse
  - Specify how the non-logical symbols can be **interpreted**
  - interpretation
  - Rules how to **combine** interpretation of single symbols
  - Satisfying interpretation = model
  - From that logical implication/entailment follows

- Specify a **calculus** that allows to derive new formulae from old ones – according to the entailment relation
The Logical Approach

- Define a **formal language**
- logical & non-logical symbols, syntax rules
- Provide language with **compositional semantics**
  - Fix **universe** of discourse
  - Specify how the non-logical symbols can be **interpreted** interpretation
  - Rules how to **combine** interpretation of single symbols
  - **Satisfying interpretation** = **model**
  - From that logical implication/entailment follows

- Specify a **calculus** that allows to derive new formulae from old ones – according to the entailment relation
Propositional Logic: Main Ideas

- Non-logical symbols: propositional variables or atoms
  - representing propositions which cannot be decomposed
  - which can be true or false
  - for example:
    - “Snow is white”
    - “It rains”

- Logical Symbols: propositional connectives such as and \((\land)\), or \((\lor)\), and not \((\neg)\).

- Formulae: built out of atoms and connectives

- Universe of discourse: truth values
Propositional Logic: Main Ideas

- Non-logical symbols: propositional variables or atoms
  - representing propositions which cannot be decomposed
  - which can be true or false
  - for example:
    - “Snow is white”
    - “It rains”

- Logical Symbols: propositional connectives such as and (\(\wedge\)), or (\(\lor\)), and not (\(\neg\)).

- Formulae: built out of atoms and connectives

- Universe of discourse: truth values
Propositional Logic: Main Ideas

- Non-logical symbols: propositional **variables** or **atoms**
  - representing **propositions** which cannot be decomposed
  - which can be **true** or **false**
  - for example:
    - “Snow is white”
    - “It rains”

- Logical Symbols: propositional connectives such as **and** ($\land$), **or** ($\lor$), and **not** ($\neg$).

- Formulae: built out of atoms and connectives
- Universe of discourse: truth values
Propositional Logic: Main Ideas

- Non-logical symbols: propositional variables or atoms
  - representing propositions which cannot be decomposed
  - which can be true or false
  - for example:
    - “Snow is white”
    - “It rains”

- Logical Symbols: propositional connectives such as and ($\wedge$), or ($\vee$), and not ($\neg$).

- Formulae: built out of atoms and connectives

- Universe of discourse: truth values
Countable alphabet \( \Sigma \) of atomic propositions: \( a, b, c, \ldots \)

Propositional formulae are built according to the following rule:

- \( \varphi \rightarrow a \) \hspace{1cm} \text{atomic formula}
- \( \bot \) \hspace{1cm} \text{falsity}
- \( \top \) \hspace{1cm} \text{truth}
- \( (\neg \varphi') \) \hspace{1cm} \text{negation}
- \( (\varphi' \land \varphi'') \) \hspace{1cm} \text{conjunction}
- \( (\varphi' \lor \varphi'') \) \hspace{1cm} \text{disjunction}
- \( (\varphi' \rightarrow \varphi'') \) \hspace{1cm} \text{implication}
- \( (\varphi' \leftrightarrow \varphi'') \) \hspace{1cm} \text{equivalence}

Parenthesis can be omitted if no ambiguity arises.

Operator precedence: \( \neg > \land > \lor > \rightarrow = \leftrightarrow \).
Countable alphabet $\Sigma$ of atomic propositions: $a, b, c, \ldots$

Propositional formulae are built according to the following rule:

<table>
<thead>
<tr>
<th>$\varphi$</th>
<th>$a$</th>
<th>atomic formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\bot$</td>
<td></td>
<td>falsity</td>
</tr>
<tr>
<td>$\top$</td>
<td></td>
<td>truth</td>
</tr>
<tr>
<td>$\neg \varphi'$</td>
<td></td>
<td>negation</td>
</tr>
<tr>
<td>$\varphi' \land \varphi''$</td>
<td></td>
<td>conjunction</td>
</tr>
<tr>
<td>$\varphi' \lor \varphi''$</td>
<td></td>
<td>disjunction</td>
</tr>
<tr>
<td>$\varphi' \rightarrow \varphi''$</td>
<td></td>
<td>implication</td>
</tr>
<tr>
<td>$\varphi' \leftrightarrow \varphi''$</td>
<td></td>
<td>equivalence</td>
</tr>
</tbody>
</table>

Parenthesis can be omitted if no ambiguity arises.

Operator precedence: $\neg > \land > \lor > \rightarrow = \leftrightarrow$. 
Countable alphabet $\Sigma$ of atomic propositions: $a, b, c, \ldots$

Propositional formulae are built according to the following rule:

<table>
<thead>
<tr>
<th>$\varphi$</th>
<th>$a$</th>
<th>atomic formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\bot$</td>
<td>falsity</td>
<td></td>
</tr>
<tr>
<td>$\top$</td>
<td>truth</td>
<td></td>
</tr>
<tr>
<td>$\neg \varphi'$</td>
<td>negation</td>
<td></td>
</tr>
<tr>
<td>$\varphi' \land \varphi''$</td>
<td>conjunction</td>
<td></td>
</tr>
<tr>
<td>$\varphi' \lor \varphi''$</td>
<td>disjunction</td>
<td></td>
</tr>
<tr>
<td>$\varphi' \to \varphi''$</td>
<td>implication</td>
<td></td>
</tr>
<tr>
<td>$\varphi' \leftrightarrow \varphi''$</td>
<td>equivalence</td>
<td></td>
</tr>
</tbody>
</table>

Parenthesis can be omitted if no ambiguity arises.

Operator precedence: $\neg > \land > \lor > \to = \leftrightarrow$. 
Syntax

Countable alphabet \( \Sigma \) of atomic propositions: \( a, b, c, \ldots \)

Propositional formulae are built according to the following rule:

- \( \varphi \rightarrow a \) \( \text{atomic formula} \)
- \( \bot \) \( \text{falsity} \)
- \( \top \) \( \text{truth} \)
- \( \neg \varphi' \) \( \text{negation} \)
- \( \varphi' \land \varphi'' \) \( \text{conjunction} \)
- \( \varphi' \lor \varphi'' \) \( \text{disjunction} \)
- \( \varphi' \rightarrow \varphi'' \) \( \text{implication} \)
- \( \varphi' \leftrightarrow \varphi'' \) \( \text{equivalence} \)

Parenthesis can be omitted if no ambiguity arises.

Operator precedence: \( \neg > \land > \lor > \rightarrow = \leftrightarrow \).
Countable alphabet $\Sigma$ of atomic propositions: $a, b, c, \ldots$

Propositional formulae are built according to the following rule:

- $\varphi \rightarrow a$ \hspace{1cm} \textit{atomic formula}
- $\bot$ \hspace{1cm} \textit{falsity}
- $\top$ \hspace{1cm} \textit{truth}
- $(\neg \varphi')$ \hspace{1cm} \textit{negation}
- $(\varphi' \land \varphi'')$ \hspace{1cm} \textit{conjunction}
- $(\varphi' \lor \varphi'')$ \hspace{1cm} \textit{disjunction}
- $(\varphi' \rightarrow \varphi'')$ \hspace{1cm} \textit{implication}
- $(\varphi' \leftrightarrow \varphi'')$ \hspace{1cm} \textit{equivalence}

Parenthesis can be omitted if no ambiguity arises.

Operator precedence: $\neg > \land > \lor > \rightarrow = \leftrightarrow$. 
Syntax

Countable alphabet $\Sigma$ of atomic propositions: $a, b, c, \ldots$

Propositional formulae are built according to the following rule:

- $a$ \hspace{1cm} atomic formula
- $\bot$ \hspace{1cm} falsity
- $\top$ \hspace{1cm} truth
- $(\neg \varphi')$ \hspace{1cm} negation
- $(\varphi' \land \varphi'')$ \hspace{1cm} conjunction
- $(\varphi' \lor \varphi'')$ \hspace{1cm} disjunction
- $(\varphi' \rightarrow \varphi'')$ \hspace{1cm} implication
- $(\varphi' \leftrightarrow \varphi'')$ \hspace{1cm} equivalence

Parenthesis can be omitted if no ambiguity arises.

Operator precedence: $\neg > \land > \lor > \rightarrow = \leftrightarrow.$
Countable alphabet $\Sigma$ of atomic propositions: $a, b, c, \ldots$

Propositional formulae are built according to the following rule:

$$
\varphi \rightarrow a \quad \text{atomic formula}
$$

- $\bot$ falsity
- $\top$ truth
- $(\neg \varphi')$ negation
- $(\varphi' \land \varphi'')$ conjunction
- $(\varphi' \lor \varphi'')$ disjunction
- $(\varphi' \rightarrow \varphi'')$ implication
- $(\varphi' \leftrightarrow \varphi'')$ equivalence

Parenthesis can be omitted if no ambiguity arises.

Operator precedence: $\neg > \land > \lor > \rightarrow = \leftrightarrow$. 

Syntax

Countable alphabet $\Sigma$ of atomic propositions: $a, b, c, \ldots$

Propositional formulae are built according to the following rule:

- $\varphi \rightarrow a$ \hspace{1cm} atomic formula
- $\bot$ \hspace{1cm} falsity
- $\top$ \hspace{1cm} truth
- $(\neg \varphi')$ \hspace{1cm} negation
- $(\varphi' \land \varphi''')$ \hspace{1cm} conjunction
- $(\varphi' \lor \varphi''')$ \hspace{1cm} disjunction
- $(\varphi' \rightarrow \varphi''')$ \hspace{1cm} implication
- $(\varphi' \leftrightarrow \varphi''')$ \hspace{1cm} equivalence

Parenthesis can be omitted if no ambiguity arises.

Operator precedence: $\neg > \land > \lor > \rightarrow = \leftrightarrow$. 
Countable alphabet \( \Sigma \) of atomic propositions: \( a, b, c, \ldots \)
Propositional formulae are built according to the following rule:

\[
\varphi \quad \rightarrow \quad a \quad \text{atomic formula}
\]

\[
\rightarrow \quad \bot \quad \text{falsity}
\]

\[
\rightarrow \quad \top \quad \text{truth}
\]

\[
\rightarrow \quad (\neg \varphi') \quad \text{negation}
\]

\[
\rightarrow \quad (\varphi' \land \varphi'') \quad \text{conjunction}
\]

\[
\rightarrow \quad (\varphi' \lor \varphi'') \quad \text{disjunction}
\]

\[
\rightarrow \quad (\varphi' \rightarrow \varphi'') \quad \text{implication}
\]

\[
\rightarrow \quad (\varphi' \leftrightarrow \varphi'') \quad \text{equivalence}
\]

Parenthesis can be omitted if no ambiguity arises.
Operator precedence: \( \neg > \land > \lor > \rightarrow = \leftrightarrow \).
Countable alphabet $\Sigma$ of atomic propositions: $a, b, c, \ldots$

Propositional formulae are built according to the following rule:

- $\phi$ → $a$ \hspace{1cm} \text{atomic formula}
- $\bot$ \hspace{1cm} \text{falsity}
- $\top$ \hspace{1cm} \text{truth}
- $(\neg \phi)$ \hspace{1cm} \text{negation}
- $(\phi \land \phi'')$ \hspace{1cm} \text{conjunction}
- $(\phi' \lor \phi'')$ \hspace{1cm} \text{disjunction}
- $(\phi' \to \phi'')$ \hspace{1cm} \text{implication}
- $(\phi' \leftrightarrow \phi'')$ \hspace{1cm} \text{equivalence}

Parenthesis can be omitted if no ambiguity arises.

Operator precedence: $\neg > \land > \lor > \to = \leftrightarrow$. 
Countable alphabet \( \Sigma \) of atomic propositions: \( a, b, c, \ldots \)

Propositional formulae are built according to the following rule:

\[
\varphi \quad \rightarrow \quad \begin{align*}
\downarrow & \quad \text{atomic formula} \\
\top & \quad \text{falsity} \\

(\neg \varphi') & \quad \text{negation} \\
(\varphi' \land \varphi'') & \quad \text{conjunction} \\
(\varphi' \lor \varphi'') & \quad \text{disjunction} \\
(\varphi' \rightarrow \varphi'') & \quad \text{implication} \\
(\varphi' \leftrightarrow \varphi'') & \quad \text{equivalence}
\end{align*}
\]

Parenthesis can be omitted if no ambiguity arises.

**Operator precedence:** \( \neg > \land > \lor > \rightarrow = \leftrightarrow \).
Atomic propositions can be true (1, T) or false (0, F).

Provided the truth values of the atoms have been fixed (truth assignment or interpretation), the truth value of a formula can be computed from the truth values of the atoms and the connectives.

Example:

\[(a \lor b) \land c\]

is true iff \(c\) is true and additionally \(a\) or \(b\) is true.

Logical implication can then be defined as follows:

\(\varphi\) is implied by the formulae \(\Theta\) iff \(\varphi\) is true for all truth assignments (world states) that make all formulae in \(\Theta\) true.
Atomic propositions can be true (1, $T$) or false (0, $F$).

Provided the truth values of the atoms have been fixed (truth assignment or interpretation), the truth value of a formula can be computed from the truth values of the atoms and the connectives.

Example:

$$(a \lor b) \land c$$

is true iff $c$ is true and additionally $a$ or $b$ is true.

Logical implication can then be defined as follows:

$\varphi$ is implied by the formulae $\Theta$ iff $\varphi$ is true for all truth assignments (world states) that make all formulae in $\Theta$ true.
Semantics: Idea

- Atomic propositions can be true (1, T) or false (0, F).
- Provided the truth values of the atoms have been fixed (truth assignment or interpretation), the truth value of a formula can be computed from the truth values of the atoms and the connectives.

Example:

\[(a \lor b) \land c\]

is true iff \(c\) is true and additionally \(a\) or \(b\) is true.

- Logical implication can then be defined as follows:
  - \(\varphi\) is implied by the formulae \(\Theta\) iff \(\varphi\) is true for all truth assignments (world states) that make all formulae in \(\Theta\) true.
Semantics: Idea

- Atomic propositions can be true (1, T) or false (0, F).
- Provided the truth values of the atoms have been fixed (truth assignment or interpretation), the truth value of a formula can be computed from the truth values of the atoms and the connectives.
- Example:

\[(a \lor b) \land c\]

is true iff \(c\) is true and additionally \(a\) or \(b\) is true.

- Logical implication can then be defined as follows:
- \(\varphi\) is implied by the formulae \(\Theta\) iff \(\varphi\) is true for all truth assignments (world states) that make all formulae in \(\Theta\) true.
An interpretation or truth assignment over $\Sigma$ is a function: $I : \Sigma \rightarrow \{T, F\}$.

A formula $\psi$ is true under $I$ or is satisfied by $I$ (symbolically $I \models \psi$):

- $I \models a$ iff $I(a) = T$
- $I \models T$
- $I \not\models \bot$
- $I \models \neg \varphi$ iff $I \not\models \varphi$
- $I \models \varphi \land \varphi'$ iff $I \models \varphi$ and $I \models \varphi'$
- $I \models \varphi \lor \varphi'$ iff $I \models \varphi$ or $I \models \varphi'$
- $I \models \varphi \rightarrow \varphi'$ iff if $I \models \varphi$, then $I \models \varphi'$
- $I \models \varphi \leftrightarrow \varphi'$ iff $I \models \varphi$ if and only if $I \models \varphi'$
Formal Semantics

An interpretation or truth assignment over $\Sigma$ is a function: $\mathcal{I}: \Sigma \rightarrow \{T, F\}$. A formula $\psi$ is true under $\mathcal{I}$ or is satisfied by $\mathcal{I}$ (symbolically $\mathcal{I} \models \psi$):

$\mathcal{I} \models a \quad \text{iff} \quad \mathcal{I}(a) = T$

$\mathcal{I} \models T$

$\mathcal{I} \not\models \bot$

$\mathcal{I} \models \neg \varphi \quad \text{iff} \quad \mathcal{I} \not\models \varphi$

$\mathcal{I} \models \varphi \land \varphi' \quad \text{iff} \quad \mathcal{I} \models \varphi \text{ and } \mathcal{I} \models \varphi'$

$\mathcal{I} \models \varphi \lor \varphi' \quad \text{iff} \quad \mathcal{I} \models \varphi \text{ or } \mathcal{I} \models \varphi'$

$\mathcal{I} \models \varphi \rightarrow \varphi' \quad \text{iff} \quad \text{if } \mathcal{I} \models \varphi, \text{ then } \mathcal{I} \models \varphi'$

$\mathcal{I} \models \varphi \leftrightarrow \varphi' \quad \text{iff} \quad \mathcal{I} \models \varphi \text{ if and only if } \mathcal{I} \models \varphi'$
Formal Semantics

An interpretation or truth assignment over $\Sigma$ is a function: $I : \Sigma \rightarrow \{T, F\}$.

A formula $\psi$ is true under $I$ or is satisfied by $I$ (symbolically $I \models \psi$):

\[
\begin{align*}
I \models a & \quad \text{iff} \quad I(a) = T \\
I \models T & \\
I \not\models \bot & \\
I \models \neg \varphi & \quad \text{iff} \quad I \not\models \varphi \\
I \models \varphi \land \varphi' & \quad \text{iff} \quad I \models \varphi \text{ and } I \models \varphi' \\
I \models \varphi \lor \varphi' & \quad \text{iff} \quad I \models \varphi \text{ or } I \models \varphi' \\
I \models \varphi \rightarrow \varphi' & \quad \text{iff} \quad \text{if } I \models \varphi, \text{ then } I \models \varphi' \\
I \models \varphi \leftrightarrow \varphi' & \quad \text{iff} \quad I \models \varphi \text{ if and only if } I \models \varphi'
\end{align*}
\]
Formal Semantics

An interpretation or truth assignment over $\Sigma$ is a function: $I : \Sigma \to \{T, F\}$.
A formula $\psi$ is true under $I$ or is satisfied by $I$ (symbolically $I \models \psi$):

- $I \models a$ iff $I(a) = T$
- $I \models \top$
- $I \not\models \bot$
- $I \models \neg \varphi$ iff $I \not\models \varphi$
- $I \models \varphi \land \varphi'$ iff $I \models \varphi$ and $I \models \varphi'$
- $I \models \varphi \lor \varphi'$ iff $I \models \varphi$ or $I \models \varphi'$
- $I \models \varphi \rightarrow \varphi'$ iff if $I \models \varphi$, then $I \models \varphi'$
- $I \models \varphi \leftrightarrow \varphi'$ iff $I \models \varphi$ if and only if $I \models \varphi'$
Formal Semantics

An interpretation or truth assignment over $\Sigma$ is a function: $I : \Sigma \rightarrow \{T, F\}$.
A formula $\psi$ is true under $I$ or is satisfied by $I$ (symbolically $I \models \psi$):

- $I \models a$ iff $I(a) = T$
- $I \models \top$
- $I \not\models \bot$
- $I \models \neg \varphi$ iff $I \not\models \varphi$
- $I \models \varphi \land \varphi'$ iff $I \models \varphi$ and $I \models \varphi'$
- $I \models \varphi \lor \varphi'$ iff $I \models \varphi$ or $I \models \varphi'$
- $I \models \varphi \rightarrow \varphi'$ iff if $I \models \varphi$, then $I \models \varphi'$
- $I \models \varphi \leftrightarrow \varphi'$ iff $I \models \varphi$ if and only if $I \models \varphi'$
Formal Semantics

An interpretation or truth assignment over $\Sigma$ is a function: $\mathcal{I} : \Sigma \rightarrow \{T, F\}$.

A formula $\psi$ is true under $\mathcal{I}$ or is satisfied by $\mathcal{I}$ (symbolically $\mathcal{I} \models \psi$):

- $\mathcal{I} \models a$ iff $\mathcal{I}(a) = T$
- $\mathcal{I} \models T$
- $\mathcal{I} \not\models \bot$
- $\mathcal{I} \models \neg \varphi$ iff $\mathcal{I} \not\models \varphi$
- $\mathcal{I} \models \varphi \wedge \varphi'$ iff $\mathcal{I} \models \varphi$ and $\mathcal{I} \models \varphi'$
- $\mathcal{I} \models \varphi \vee \varphi'$ iff $\mathcal{I} \models \varphi$ or $\mathcal{I} \models \varphi'$
- $\mathcal{I} \models \varphi \rightarrow \varphi'$ iff if $\mathcal{I} \models \varphi$, then $\mathcal{I} \models \varphi'$
- $\mathcal{I} \models \varphi \leftrightarrow \varphi'$ iff $\mathcal{I} \models \varphi$ if and only if $\mathcal{I} \models \varphi'$
Formal Semantics

An interpretation or truth assignment over $\Sigma$ is a function: $I : \Sigma \rightarrow \{T, F\}$.

A formula $\psi$ is true under $I$ or is satisfied by $I$ (symbolically $I \models \psi$):

$I \models a$ iff $I(a) = T$

$I \models T$

$I \not\models \bot$

$I \models \neg \varphi$ iff $I \not\models \varphi$

$I \models \varphi \land \varphi'$ iff $I \models \varphi$ and $I \models \varphi'$

$I \models \varphi \lor \varphi'$ iff $I \models \varphi$ or $I \models \varphi'$

$I \models \varphi \rightarrow \varphi'$ iff if $I \models \varphi$, then $I \models \varphi'$

$I \models \varphi \leftrightarrow \varphi'$ iff $I \models \varphi$ if and only if $I \models \varphi'$
An interpretation or truth assignment over $\Sigma$ is a function: $\mathcal{I} : \Sigma \rightarrow \{T, F\}$.

A formula $\psi$ is true under $\mathcal{I}$ or is satisfied by $\mathcal{I}$ (symbolically $\mathcal{I} \models \psi$):

$\mathcal{I} \models a$ iff $\mathcal{I}(a) = T$

$\mathcal{I} \models \top$

$\mathcal{I} \not\models \bot$

$\mathcal{I} \models \neg \varphi$ iff $\mathcal{I} \not\models \varphi$

$\mathcal{I} \models \varphi \land \varphi'$ iff $\mathcal{I} \models \varphi$ and $\mathcal{I} \models \varphi'$

$\mathcal{I} \models \varphi \lor \varphi'$ iff $\mathcal{I} \models \varphi$ or $\mathcal{I} \models \varphi'$

$\mathcal{I} \models \varphi \rightarrow \varphi'$ iff if $\mathcal{I} \models \varphi$, then $\mathcal{I} \models \varphi'$

$\mathcal{I} \models \varphi \leftrightarrow \varphi'$ iff $\mathcal{I} \models \varphi$ if and only if $\mathcal{I} \models \varphi'$
Example

**Given**

\[ I : a \mapsto T, \ b \mapsto F, \ c \mapsto F, \ d \mapsto T, \]

Is \(((a \lor b) \leftrightarrow (c \lor d)) \land (\neg(a \land c) \lor (c \land \neg d))\) true or false?
Example

Given

\[ \mathcal{I} : a \mapsto T, \ b \mapsto F, \ c \mapsto F, \ d \mapsto T, \]

Is \((a \lor b) \leftrightarrow (c \lor d)) \land \neg(a \land c) \lor (c \land \neg d))\) true or false?

\( ((a \lor b) \leftrightarrow (c \lor d)) \land \neg(a \land c) \lor (c \land \neg d)) \)
Example

Given

\[ \mathcal{I} : a \mapsto T, \ b \mapsto F, \ c \mapsto F, \ d \mapsto T, \]

Is \((a \lor b) \leftrightarrow (c \lor d)) \land (\neg(a \land c) \lor (c \land \neg d))\) true or false?

\((a \lor b) \leftrightarrow (c \lor d)) \land (\neg(a \land c) \lor (c \land \neg d))\)

\((a \lor b) \leftrightarrow (c \lor d)) \land (\neg(a \land c) \lor (c \land \neg d))\)

\((a \lor b) \leftrightarrow (c \lor d)) \land (\neg(a \land c) \lor (c \land \neg d))\)

\((a \lor b) \leftrightarrow (c \lor d)) \land (\neg(a \land c) \lor (c \land \neg d))\)
Example

Given

\[ I: a \mapsto T, \quad b \mapsto F, \quad c \mapsto F, \quad d \mapsto T, \]

Is \(((a \lor b) \leftrightarrow (c \lor d)) \land \neg(a \land c) \lor (c \land \neg d))\) true or false?

\[((a \lor b) \leftrightarrow (c \lor d)) \land (\neg(a \land c) \lor (c \land \neg d))\]

\[((a \lor b) \leftrightarrow (c \lor d)) \land (\neg(a \land c) \lor (c \land \neg d))\]

\[((a \lor b) \leftrightarrow (c \lor d)) \land (\neg(a \land c) \lor (c \land \neg d))\]

\[((a \lor b) \leftrightarrow (c \lor d)) \land (\neg(a \land c) \lor (c \land \neg d))\]
Example

Given

\[ I : a \mapsto T, \ b \mapsto F, \ c \mapsto F, \ d \mapsto T, \]

Is \((a \lor b) \leftrightarrow (c \lor d)) \land (\neg(a \land c) \lor (c \land \neg d))\) true or false?

\[ ((a \lor b) \leftrightarrow (c \lor d)) \land (\neg(a \land c) \lor (c \land \neg d)) \]

\[ ((a \lor b) \leftrightarrow (c \lor d)) \land (\neg(a \land c) \lor (c \land \neg d)) \]

\[ ((a \lor b) \leftrightarrow (c \lor d)) \land (\neg(a \land c) \lor (c \land \neg d)) \]

\[ ((a \lor b) \leftrightarrow (c \lor d)) \land (\neg(a \land c) \lor (c \land \neg d)) \]
Example

Given

\[ \mathcal{I} : a \mapsto T, \ b \mapsto F, \ c \mapsto F, \ d \mapsto T, \]

Is \((a \lor b) \leftrightarrow (c \lor d)) \land (\neg(a \land c) \lor (c \land \neg d))\) true or false?

\[
((a \lor b) \leftrightarrow (c \lor d)) \land (\neg(a \land c) \lor (c \land \neg d))
\]

\[
((a \lor b) \leftrightarrow (c \lor d)) \land (\neg(a \land c) \lor (c \land \neg d))
\]

\[
((a \lor b) \leftrightarrow (c \lor d)) \land (\neg(a \land c) \lor (c \land \neg d))
\]

\[
((a \lor b) \leftrightarrow (c \lor d)) \land (\neg(a \land c) \lor (c \land \neg d))
\]

\[
((a \lor b) \leftrightarrow (c \lor d)) \land (\neg(a \land c) \lor (c \land \neg d))
\]
An interpretation $\mathcal{I}$ is a model of $\varphi$ iff

$$\mathcal{I} \models \varphi$$

A formula $\varphi$ is

- **satisfiable** iff there is $\mathcal{I}$ such that $\mathcal{I} \models \varphi$,
- **unsatisfiable** otherwise, and
- **valid** iff $\mathcal{I} \models \varphi$ for all $\mathcal{I}$,
- **falsifiable** otherwise.

Two formulae $\varphi$ and $\psi$ are **logically equivalent** (symbolically $\varphi \equiv \psi$) iff for all interpretations $\mathcal{I}$

$$\mathcal{I} \models \varphi \text{ iff } \mathcal{I} \models \psi.$$
Terminology

An interpretation $\mathcal{I}$ is a model of $\varphi$ iff

$$\mathcal{I} \models \varphi$$

A formula $\varphi$ is

- **satisfiable** iff there is $\mathcal{I}$ such that $\mathcal{I} \models \varphi$,
- **unsatisfiable** otherwise, and
- **valid** iff $\mathcal{I} \models \varphi$ for all $\mathcal{I}$,
- **falsifiable** otherwise.

Two formulae $\varphi$ and $\psi$ are **logically equivalent** (symbolically $\varphi \equiv \psi$) iff for all interpretations $\mathcal{I}$

$$\mathcal{I} \models \varphi \text{ iff } \mathcal{I} \models \psi.$$
An interpretation $I$ is a model of $\varphi$ iff

$$I \models \varphi$$

A formula $\varphi$ is

- **satisfiable** iff there is $I$ such that $I \models \varphi$,
- **unsatisfiable** otherwise, and
- **valid** iff $I \models \varphi$ for all $I$,
- **falsifiable** otherwise.

Two formulae $\varphi$ and $\psi$ are logically equivalent (symbolically $\varphi \equiv \psi$) iff for all interpretations $I$

$$I \models \varphi \text{ iff } I \models \psi.$$
Terminology

An interpretation $\mathcal{I}$ is a **model** of $\varphi$ iff

$$\mathcal{I} \models \varphi$$

A formula $\varphi$ is

- **satisfiable** iff there is $\mathcal{I}$ such that $\mathcal{I} \models \varphi$,
- **unsatisfiable** otherwise, and
- **valid** iff $\mathcal{I} \models \varphi$ for all $\mathcal{I}$,
- **falsifiable** otherwise.

Two formulae $\varphi$ and $\psi$ are **logically equivalent** (symbolically $\varphi \equiv \psi$) iff for all interpretations $\mathcal{I}$

$$\mathcal{I} \models \varphi \text{ iff } \mathcal{I} \models \psi.$$
Examples

Satisfiable, unsatisfiable, falsifiable, valid?

\[(a \lor b \lor \neg c) \land (\neg a \lor \neg b \lor d) \land (\neg a \lor b \lor \neg d)\]

\(
\rightarrow\) satisfiable: \(a \mapsto T, b \mapsto F, d \mapsto F, \ldots\)
\(
\rightarrow\) falsifiable: \(a \mapsto F, b \mapsto F, c \mapsto T, \ldots\)

\[((\neg a \rightarrow \neg b) \rightarrow (b \rightarrow a))\]

\(
\rightarrow\) satisfiable: \(a \mapsto T, b \mapsto T\)

\(
\rightarrow\) valid: Consider all interpretations or argue about falsifying ones.

Equivalence?

\(\neg (a \lor b) \equiv \neg a \land \neg b\)

\(
\rightarrow\) Of course, equivalent (de Morgan).
Examples

Satisfiable, unsatisfiable, falsifiable, valid?

\[ (a \lor b \lor \neg c) \land (\neg a \lor \neg b \lor d) \land (\neg a \lor b \lor \neg d) \]

\( \Rightarrow \text{satisfiable: } a \mapsto T, b \mapsto F, d \mapsto F, \ldots \)

\( \Rightarrow \text{falsifiable: } a \mapsto F, b \mapsto F, c \mapsto T, \ldots \)

\((\neg a \rightarrow \neg b) \rightarrow (b \rightarrow a))

\( \Rightarrow \text{satisfiable: } a \mapsto T, b \mapsto T \)

\( \Rightarrow \text{valid: Consider all interpretations or argue about falsifying ones.} \)

Equivalence?

\[ \neg(a \lor b) \equiv \neg a \land \neg b \]

\( \Rightarrow \text{Of course, equivalent (de Morgan).} \)
Examples

Satisfiable, unsatisfiable, falsifiable, valid?

\[(a \lor b \lor \neg c) \land (\neg a \lor \neg b \lor d) \land (\neg a \lor b \lor \neg d)\]

\[\rightsquigarrow \text{satisfiable: } a \mapsto T, b \mapsto F, d \mapsto F, \ldots\]

\[\rightsquigarrow \text{falsifiable: } a \mapsto F, b \mapsto F, c \mapsto T, \ldots\]

\[
\left(\neg a \rightarrow \neg b\right) \rightarrow \left(b \rightarrow a\right)
\]

\[\rightsquigarrow \text{satisfiable: } a \mapsto T, b \mapsto T\]

\[\rightsquigarrow \text{valid: Consider all interpretations or argue about falsifying ones.}\]

Equivalence?

\[\neg (a \lor b) \equiv \neg a \land \neg b\]

\[\rightsquigarrow \text{Of course, equivalent (de Morgan).}\]
Examples

Satisfiable, unsatisfiable, falsifiable, valid?

\((a \lor b \lor \neg c) \land (\neg a \lor \neg b \lor d) \land (\neg a \lor b \lor \neg d)\)

\(\leadsto\) satisfiable: \(a \mapsto T, b \mapsto F, d \mapsto F, \ldots\)

\(\leadsto\) falsifiable: \(a \mapsto F, b \mapsto F, c \mapsto T, \ldots\)

\(((\neg a \rightarrow \neg b) \rightarrow (b \rightarrow a))\)

\(\leadsto\) satisfiable: \(a \mapsto T, b \mapsto T\)

\(\leadsto\) valid: Consider all interpretations or argue about falsifying ones.

Equivalence?

\(\neg (a \lor b) \equiv \neg a \land \neg b\)

\(\leadsto\) Of course, equivalent (de Morgan).
Examples

Satisfiable, unsatisfiable, falsifiable, valid?

\((a \lor b \lor \neg c) \land (\neg a \lor b \lor d) \land (\neg a \lor b \lor \neg d)\)

\[\implies\text{satisfiable: } a \leftrightarrow T, b \leftrightarrow F, d \leftrightarrow F, \ldots\]

\[\implies\text{falsifiable: } a \leftrightarrow F, b \leftrightarrow F, c \leftrightarrow T, \ldots\]

\[((\neg a \rightarrow \neg b) \rightarrow (b \rightarrow a))\]

\[\implies\text{satisfiable: } a \leftrightarrow T, b \leftrightarrow T\]

\[\implies\text{valid: Consider all interpretations or argue about falsifying ones.}\]

Equivalence?

\(\neg(a \lor b) \equiv \neg a \land \neg b\)

\[\implies\text{Of course, equivalent (de Morgan).}\]
Examples

Satisfiable, unsatisfiable, falsifiable, valid?
\((a \lor b \lor \neg c) \land (\neg a \lor \neg b \lor d) \land (\neg a \lor b \lor \neg d)\)

\(\therefore\) satisfiable: \(a \mapsto T, b \mapsto F, d \mapsto F, \ldots\)

\(\therefore\) falsifiable: \(a \mapsto F, b \mapsto F, c \mapsto T, \ldots\)

\(((\neg a \rightarrow \neg b) \rightarrow (b \rightarrow a))\)

\(\therefore\) satisfiable: \(a \mapsto T, b \mapsto T\)

\(\therefore\) valid: Consider all interpretations or argue about falsifying ones.

Equivalence?
\(\neg(a \lor b) \equiv \neg a \land \neg b\)

\(\therefore\) Of course, equivalent (de Morgan).
Examples

Satisfiable, unsatisfiable, falsifiable, valid?
\((a \lor b \lor \neg c) \land (\neg a \lor \neg b \lor d) \land (\neg a \lor b \lor \neg d)\)

\(\leadsto\) satisfiable: \(a \mapsto T, b \mapsto F, d \mapsto F, \ldots\)

\(\leadsto\) falsifiable: \(a \mapsto F, b \mapsto F, c \mapsto T, \ldots\)

\(((\neg a \rightarrow \neg b) \rightarrow (b \rightarrow a))\)

\(\leadsto\) satisfiable: \(a \mapsto T, b \mapsto T\)

\(\leadsto\) valid: Consider all interpretations or argue about falsifying ones.

Equivalence?

\(\neg(a \lor b) \equiv \neg a \land \neg b\)

\(\leadsto\) Of course, equivalent (de Morgan).
Examples

Satisfiable, unsatisfiable, falsifiable, valid?

\[(a \lor b \lor \neg c) \land (\neg a \lor \neg b \lor d) \land (\neg a \lor b \lor \neg d)\]

\(\leadsto\) satisfiable: \(a \mapsto T, b \mapsto F, d \mapsto F, \ldots\)

\(\leadsto\) falsifiable: \(a \mapsto F, b \mapsto F, c \mapsto T, \ldots\)

\[((\neg a \rightarrow \neg b) \rightarrow (b \rightarrow a))\]

\(\leadsto\) satisfiable: \(a \mapsto T, b \mapsto T\)

\(\leadsto\) valid: Consider all interpretations or argue about falsifying ones.

Equivalence?

\(\neg(a \lor b) \equiv \neg a \land \neg b\)

\(\leadsto\) Of course, equivalent (de Morgan).
Some Obvious Consequences

**Proposition**

\[ \varphi \text{ is valid iff } \neg \varphi \text{ is unsatisfiable and } \varphi \text{ is satisfiable iff } \neg \varphi \text{ is falsifiable.} \]

**Proposition**

\[ \varphi \equiv \psi \text{ iff } \varphi \leftrightarrow \psi \text{ is valid.} \]

**Theorem**

*If* \( \varphi \equiv \psi \) *and* \( \chi' \) *results from substituting* \( \varphi \) *by* \( \psi \) *in* \( \chi \), *then* \( \chi' \equiv \chi \).
Some Obvious Consequences

**Proposition**

ϕ is valid iff ¬ϕ is unsatisfiable and ϕ is satisfiable iff ¬ϕ is falsifiable.

**Proposition**

ϕ ≡ ψ iff ϕ ↔ ψ is valid.

**Theorem**

If ϕ ≡ ψ and χ′ results from substituting ϕ by ψ in χ, then χ′ ≡ χ.
Some Obvious Consequences

Proposition

\( \varphi \text{ is valid iff } \neg \varphi \text{ is unsatisfiable and } \varphi \text{ is satisfiable iff } \neg \varphi \text{ is falsifiable.} \)

Proposition

\( \varphi \equiv \psi \text{ iff } \varphi \leftrightarrow \psi \text{ is valid.} \)

Theorem

If \( \varphi \equiv \psi \) and \( \chi' \) results from substituting \( \varphi \) by \( \psi \) in \( \chi \), then \( \chi' \equiv \chi \).
Some Equivalences

<table>
<thead>
<tr>
<th>Simplifications</th>
<th>( \varphi \rightarrow \psi \equiv \neg \varphi \lor \psi )</th>
<th>( \varphi \leftrightarrow \psi \equiv (\varphi \rightarrow \psi) \land (\psi \rightarrow \varphi) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Idempotency</td>
<td>( \varphi \lor \varphi \equiv \varphi )</td>
<td>( \varphi \land \varphi \equiv \varphi )</td>
</tr>
<tr>
<td>Commutativity</td>
<td>( \varphi \lor \psi \equiv \psi \lor \varphi )</td>
<td>( \varphi \land \psi \equiv \psi \land \varphi )</td>
</tr>
<tr>
<td>Associativity</td>
<td>( (\varphi \lor \psi) \lor \chi \equiv \varphi \lor (\psi \lor \chi) )</td>
<td>( (\varphi \land \psi) \land \chi \equiv \varphi \land (\psi \land \chi) )</td>
</tr>
<tr>
<td>Absorption</td>
<td>( \varphi \lor (\varphi \land \psi) \equiv \varphi \lor (\varphi \lor \psi) )</td>
<td>( \varphi \lor (\varphi \land \psi) \equiv \varphi \lor (\varphi \lor \psi) )</td>
</tr>
<tr>
<td>Distributivity</td>
<td>( \varphi \land (\psi \lor \chi) \equiv (\varphi \land \psi) \lor (\varphi \land \chi) )</td>
<td>( \varphi \lor (\psi \land \chi) \equiv (\varphi \lor \psi) \lor (\varphi \lor \chi) )</td>
</tr>
<tr>
<td>Double Negation</td>
<td>( \neg \neg \varphi \equiv \varphi )</td>
<td>( \neg \top \equiv \bot )</td>
</tr>
<tr>
<td>Constants</td>
<td>( \neg \top \equiv \bot )</td>
<td>( \neg \bot \equiv \top )</td>
</tr>
<tr>
<td>De Morgan</td>
<td>( \neg (\varphi \lor \psi) \equiv \neg \varphi \land \neg \psi )</td>
<td>( \neg (\varphi \lor \psi) \equiv \neg \varphi \lor \neg \psi )</td>
</tr>
<tr>
<td>Truth</td>
<td>( \varphi \lor \top \equiv \top )</td>
<td>( \varphi \land \top \equiv \varphi )</td>
</tr>
<tr>
<td>Falsity</td>
<td>( \varphi \lor \bot \equiv \varphi )</td>
<td>( \varphi \land \bot \equiv \bot )</td>
</tr>
</tbody>
</table>
Some Equivalences

<table>
<thead>
<tr>
<th>Simplifications</th>
<th>Simplifications: $\varphi \to \psi \equiv \neg \varphi \lor \psi$</th>
<th>Simplifications: $\varphi \leftrightarrow \psi \equiv (\varphi \to \psi) \land (\psi \to \varphi)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Idempotency</td>
<td>$\varphi \lor \varphi \equiv \varphi$</td>
<td>$\varphi \land \varphi \equiv \varphi$</td>
</tr>
<tr>
<td>Commutativity</td>
<td>$\varphi \lor \psi \equiv \psi \lor \varphi$</td>
<td>$\varphi \land \psi \equiv \psi \land \varphi$</td>
</tr>
<tr>
<td>Associativity</td>
<td>$(\varphi \lor \psi) \lor \chi \equiv \varphi \lor (\psi \lor \chi)$</td>
<td>$(\varphi \land \psi) \land \chi \equiv \varphi \land (\psi \land \chi)$</td>
</tr>
<tr>
<td>Absorption</td>
<td>$\varphi \lor (\varphi \land \psi) \equiv (\varphi \land \psi) \lor \varphi$</td>
<td>$\varphi \land (\varphi \lor \psi) \equiv (\varphi \lor \psi) \land \varphi$</td>
</tr>
<tr>
<td>Distributivity</td>
<td>$\varphi \land (\psi \lor \chi) \equiv (\varphi \land \psi) \lor (\varphi \land \chi)$</td>
<td>$\varphi \lor (\psi \land \chi) \equiv (\varphi \lor \psi) \land (\varphi \lor \chi)$</td>
</tr>
<tr>
<td>Double Negation</td>
<td>$\neg \neg \varphi \equiv \varphi$</td>
<td>$\neg \neg \bot \equiv \top$</td>
</tr>
<tr>
<td>Constants</td>
<td>$\neg \top \equiv \bot$</td>
<td>$\neg \bot \equiv \top$</td>
</tr>
<tr>
<td>De Morgan</td>
<td>$\neg (\varphi \lor \psi) \equiv \neg \varphi \land \neg \psi$</td>
<td>$\neg (\varphi \land \psi) \equiv \neg \varphi \lor \neg \psi$</td>
</tr>
<tr>
<td>Truth</td>
<td>$\varphi \lor \top \equiv \top$</td>
<td>$\varphi \land \top \equiv \varphi$</td>
</tr>
<tr>
<td>Falsity</td>
<td>$\varphi \lor \bot \equiv \varphi$</td>
<td>$\varphi \land \bot \equiv \bot$</td>
</tr>
</tbody>
</table>
### Some Equivalences

<table>
<thead>
<tr>
<th>Feature</th>
<th>Expression</th>
</tr>
</thead>
</table>
| **simplifications** | $\varphi \rightarrow \psi \equiv \neg \varphi \lor \psi$  
|                  | $\varphi \leftrightarrow \psi \equiv (\varphi \rightarrow \psi) \land (\psi \rightarrow \varphi)$  
| **idempotency**   | $\varphi \lor \varphi \equiv \varphi$  
|                  | $\varphi \land \varphi \equiv \varphi$  
| **commutativity** | $\varphi \lor \psi \equiv \psi \lor \varphi$  
|                  | $\varphi \land \psi \equiv \psi \land \varphi$  
| **associativity** | $(\varphi \lor \psi) \lor \chi \equiv \varphi \lor (\psi \lor \chi)$  
|                  | $(\varphi \land \psi) \land \chi \equiv \varphi \land (\psi \land \chi)$  
| **absorption**    | $\varphi \lor (\varphi \land \psi) \equiv \varphi$  
|                  | $\varphi \land (\varphi \lor \psi) \equiv \varphi$  
| **distributivity** | $\varphi \land (\psi \lor \chi) \equiv (\varphi \land \psi) \lor (\varphi \land \chi)$  
|                  | $\varphi \lor (\psi \land \chi) \equiv (\varphi \lor \psi) \land (\varphi \lor \chi)$  
| **double negation** | $\neg \neg \varphi \equiv \varphi$  
| **constants**    | $\neg \top \equiv \bot$  
|                  | $\neg \bot \equiv \top$  
| **De Morgan**    | $\neg (\varphi \lor \psi) \equiv \neg \varphi \land \neg \psi$  
|                  | $\neg (\varphi \land \psi) \equiv \neg \varphi \lor \neg \psi$  
| **truth**        | $\varphi \lor \top \equiv \top$  
|                  | $\varphi \land \top \equiv \varphi$  
| **falsity**      | $\varphi \lor \bot \equiv \varphi$  
|                  | $\varphi \land \bot \equiv \bot$  

**Syntactic Terminology:**
- **Normal forms**
- **Decision Problems**
- **Resolution**
### Some Equivalences

<table>
<thead>
<tr>
<th>Simplifications</th>
<th>$\varphi \rightarrow \psi \equiv \neg \varphi \lor \psi$</th>
<th>$\varphi \leftrightarrow \psi \equiv (\varphi \rightarrow \psi) \land (\psi \rightarrow \varphi)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Idempotency</td>
<td>$\varphi \lor \varphi \equiv \varphi$</td>
<td>$\varphi \land \varphi \equiv \varphi$</td>
</tr>
<tr>
<td>Commutativity</td>
<td>$\varphi \lor \psi \equiv \psi \lor \varphi$</td>
<td>$\varphi \land \psi \equiv \psi \land \varphi$</td>
</tr>
<tr>
<td>Associativity</td>
<td>$(\varphi \lor \psi) \lor \chi \equiv \varphi \lor (\psi \lor \chi)$</td>
<td>$(\varphi \land \psi) \land \chi \equiv \varphi \land (\psi \land \chi)$</td>
</tr>
<tr>
<td>Absorption</td>
<td>$\varphi \lor (\varphi \land \psi) \equiv \varphi$</td>
<td>$\varphi \land (\varphi \lor \psi) \equiv \varphi$</td>
</tr>
<tr>
<td>Distributivity</td>
<td>$\varphi \land (\psi \lor \chi) \equiv (\varphi \land \psi) \lor (\varphi \land \chi)$</td>
<td>$\varphi \lor (\psi \land \chi) \equiv (\varphi \lor \psi) \land (\varphi \lor \chi)$</td>
</tr>
<tr>
<td>Double Negation</td>
<td>$\neg \neg \varphi \equiv \varphi$</td>
<td>$\neg \bot \equiv \top$</td>
</tr>
<tr>
<td>Constants</td>
<td>$\neg \top \equiv \bot$</td>
<td>$\neg \bot \equiv \top$</td>
</tr>
<tr>
<td>De Morgan</td>
<td>$\neg (\varphi \lor \psi) \equiv \neg \varphi \land \neg \psi$</td>
<td>$\neg (\varphi \land \psi) \equiv \neg \varphi \lor \neg \psi$</td>
</tr>
<tr>
<td>Truth</td>
<td>$\varphi \lor \top \equiv \top$</td>
<td>$\varphi \land \top \equiv \varphi$</td>
</tr>
<tr>
<td>Falsity</td>
<td>$\varphi \lor \bot \equiv \varphi$</td>
<td>$\varphi \land \bot \equiv \bot$</td>
</tr>
</tbody>
</table>
## Some Equivalences

<table>
<thead>
<tr>
<th>Simplifications</th>
<th>$\varphi \to \psi \equiv \neg \varphi \vee \psi$</th>
<th>$\varphi \leftrightarrow \psi \equiv (\varphi \to \psi) \land (\psi \to \varphi)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Idempotency</td>
<td>$\varphi \vee \varphi \equiv \varphi$</td>
<td>$\varphi \land \varphi \equiv \varphi$</td>
</tr>
<tr>
<td>Commutativity</td>
<td>$\varphi \vee \psi \equiv \psi \vee \varphi$</td>
<td>$\varphi \land \psi \equiv \psi \land \varphi$</td>
</tr>
<tr>
<td>Associativity</td>
<td>$(\varphi \vee \psi) \vee \chi \equiv \varphi \vee (\psi \vee \chi)$</td>
<td>$(\varphi \land \psi) \land \chi \equiv \varphi \land (\psi \land \chi)$</td>
</tr>
<tr>
<td>Absorption</td>
<td>$\varphi \vee (\varphi \land \psi) \equiv \varphi$</td>
<td>$\varphi \land (\varphi \vee \psi) \equiv \varphi$</td>
</tr>
<tr>
<td>Distributivity</td>
<td>$\varphi \land (\psi \vee \chi) \equiv (\varphi \land \psi) \lor (\varphi \land \chi)$</td>
<td>$\varphi \lor (\psi \land \chi) \equiv (\varphi \lor \psi) \land (\varphi \lor \chi)$</td>
</tr>
<tr>
<td>Double Negation</td>
<td>$\neg \neg \varphi \equiv \varphi$</td>
<td></td>
</tr>
<tr>
<td>Constants</td>
<td>$\neg \top \equiv \bot$</td>
<td>$\neg \bot \equiv \top$</td>
</tr>
<tr>
<td>De Morgan</td>
<td>$\neg (\varphi \lor \psi) \equiv \neg \varphi \land \neg \psi$</td>
<td>$\neg (\varphi \land \psi) \equiv \neg \varphi \lor \neg \psi$</td>
</tr>
<tr>
<td>Truth</td>
<td>$\varphi \lor \top \equiv \top$</td>
<td>$\varphi \land \top \equiv \varphi$</td>
</tr>
<tr>
<td>Falsity</td>
<td>$\varphi \lor \bot \equiv \varphi$</td>
<td>$\varphi \land \bot \equiv \bot$</td>
</tr>
</tbody>
</table>
Some Equivalences

<table>
<thead>
<tr>
<th>Concept</th>
<th>Expression 1</th>
<th>Expression 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>simplifications</td>
<td>$\varphi \rightarrow \psi \equiv \neg \varphi \lor \psi$</td>
<td>$\varphi \leftrightarrow \psi \equiv (\varphi \rightarrow \psi) \land (\psi \rightarrow \varphi)$</td>
</tr>
<tr>
<td>idempotency</td>
<td>$\varphi \lor \varphi \equiv \varphi$</td>
<td>$\varphi \land \varphi \equiv \varphi$</td>
</tr>
<tr>
<td>commutativity</td>
<td>$\varphi \lor \psi \equiv \psi \lor \varphi$</td>
<td>$\varphi \land \psi \equiv \psi \land \varphi$</td>
</tr>
<tr>
<td>associativity</td>
<td>$(\varphi \lor \psi) \lor \chi \equiv \varphi \lor (\psi \lor \chi)$</td>
<td>$(\varphi \land \psi) \land \chi \equiv \varphi \land (\psi \land \chi)$</td>
</tr>
<tr>
<td>absorption</td>
<td>$\varphi \lor (\varphi \land \psi) \equiv \varphi$</td>
<td>$\varphi \land (\varphi \lor \psi) \equiv \varphi$</td>
</tr>
<tr>
<td>distributivity</td>
<td>$\varphi \land (\psi \lor \chi) \equiv (\varphi \land \psi) \lor (\varphi \land \chi)$</td>
<td>$(\varphi \lor \psi) \land (\varphi \lor \chi) \equiv (\varphi \lor \psi) \land (\varphi \lor \chi)$</td>
</tr>
<tr>
<td>double negation</td>
<td>$\neg \neg \varphi \equiv \varphi$</td>
<td></td>
</tr>
<tr>
<td>constants</td>
<td>$\neg \top \equiv \bot$</td>
<td>$\neg \bot \equiv \top$</td>
</tr>
<tr>
<td>De Morgan</td>
<td>$\neg (\varphi \lor \psi) \equiv \neg \varphi \land \neg \psi$</td>
<td>$\neg (\varphi \land \psi) \equiv \neg \varphi \lor \neg \psi$</td>
</tr>
<tr>
<td>truth</td>
<td>$\varphi \lor \top \equiv \top$</td>
<td>$\varphi \land \top \equiv \varphi$</td>
</tr>
<tr>
<td>falsity</td>
<td>$\varphi \lor \bot \equiv \varphi$</td>
<td>$\varphi \land \bot \equiv \bot$</td>
</tr>
</tbody>
</table>
**Some Equivalences**

<table>
<thead>
<tr>
<th>Category</th>
<th>Equivalence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simplifications</td>
<td>$\phi \to \psi \equiv \neg \phi \lor \psi$</td>
</tr>
<tr>
<td></td>
<td>$\phi \leftrightarrow \psi \equiv (\phi \to \psi) \land (\psi \to \phi)$</td>
</tr>
<tr>
<td>Idempotency</td>
<td>$\phi \lor \phi \equiv \phi$</td>
</tr>
<tr>
<td></td>
<td>$\phi \land \phi \equiv \phi$</td>
</tr>
<tr>
<td>Commutativity</td>
<td>$\phi \lor \psi \equiv \psi \lor \phi$</td>
</tr>
<tr>
<td></td>
<td>$\phi \land \psi \equiv \psi \land \phi$</td>
</tr>
<tr>
<td>Associativity</td>
<td>$(\phi \lor \psi) \lor \chi \equiv \phi \lor (\psi \lor \chi)$</td>
</tr>
<tr>
<td></td>
<td>$(\phi \land \psi) \land \chi \equiv \phi \land (\psi \land \chi)$</td>
</tr>
<tr>
<td>Absorption</td>
<td>$\phi \lor (\phi \land \psi) \equiv \phi$</td>
</tr>
<tr>
<td></td>
<td>$\phi \land (\phi \lor \psi) \equiv \phi$</td>
</tr>
<tr>
<td>Distributivity</td>
<td>$\phi \land (\psi \lor \chi) \equiv (\phi \land \psi) \lor (\phi \land \chi)$</td>
</tr>
<tr>
<td></td>
<td>$\phi \lor (\psi \land \chi) \equiv (\phi \lor \psi) \land (\phi \lor \chi)$</td>
</tr>
<tr>
<td>Double Negation</td>
<td>$\neg \neg \phi \equiv \phi$</td>
</tr>
<tr>
<td>Constants</td>
<td>$\neg \top \equiv \bot$</td>
</tr>
<tr>
<td>De Morgan</td>
<td>$\neg (\phi \lor \psi) \equiv \neg \phi \land \neg \psi$</td>
</tr>
<tr>
<td></td>
<td>$\neg (\phi \land \psi) \equiv \neg \phi \lor \neg \psi$</td>
</tr>
<tr>
<td>Truth</td>
<td>$\phi \lor \top \equiv \top$</td>
</tr>
<tr>
<td></td>
<td>$\phi \land \top \equiv \phi$</td>
</tr>
<tr>
<td>Falsity</td>
<td>$\phi \lor \bot \equiv \phi$</td>
</tr>
<tr>
<td></td>
<td>$\phi \land \bot \equiv \bot$</td>
</tr>
</tbody>
</table>
Some Equivalences

<table>
<thead>
<tr>
<th>Simplifications</th>
<th>( \varphi \rightarrow \psi \equiv \neg \varphi \lor \psi )</th>
<th>( \varphi \leftrightarrow \psi \equiv (\varphi \rightarrow \psi) \land (\psi \rightarrow \varphi) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Idempotency</td>
<td>( \varphi \lor \varphi \equiv \varphi )</td>
<td>( \varphi \land \varphi \equiv \varphi )</td>
</tr>
<tr>
<td>Commutativity</td>
<td>( \varphi \lor \psi \equiv \psi \lor \varphi )</td>
<td>( \varphi \land \psi \equiv \psi \land \varphi )</td>
</tr>
<tr>
<td>Associativity</td>
<td>( (\varphi \lor \psi) \lor \chi \equiv \varphi \lor (\psi \lor \chi) )</td>
<td>( (\varphi \land \psi) \land \chi \equiv \varphi \land (\psi \land \chi) )</td>
</tr>
<tr>
<td>Absorption</td>
<td>( \varphi \lor (\varphi \land \psi) \equiv \varphi )</td>
<td>( \varphi \land (\varphi \lor \psi) \equiv \varphi )</td>
</tr>
<tr>
<td>Distributivity</td>
<td>( \varphi \land (\psi \lor \chi) \equiv (\varphi \land \psi) \lor (\varphi \land \chi) )</td>
<td>( \varphi \lor (\varphi \land \psi) \equiv (\varphi \lor \psi) \land (\varphi \lor \chi) )</td>
</tr>
<tr>
<td>Double Negation</td>
<td>( \neg \neg \varphi \equiv \varphi )</td>
<td></td>
</tr>
<tr>
<td>Constants</td>
<td>( \neg \top \equiv \bot )</td>
<td>( \neg \bot \equiv \top )</td>
</tr>
<tr>
<td>De Morgan</td>
<td>( \neg (\varphi \lor \psi) \equiv \neg \varphi \land \neg \psi )</td>
<td>( \neg (\varphi \land \psi) \equiv \neg \varphi \lor \neg \psi )</td>
</tr>
<tr>
<td>Truth</td>
<td>( \varphi \lor \top \equiv \top )</td>
<td>( \varphi \land \top \equiv \varphi )</td>
</tr>
<tr>
<td>Falsity</td>
<td>( \varphi \lor \bot \equiv \varphi \bot )</td>
<td>( \varphi \land \bot \equiv \bot )</td>
</tr>
</tbody>
</table>
Some Equivalences

simplifications $\varphi \rightarrow \psi \equiv \neg \varphi \lor \psi \quad \varphi \leftrightarrow \psi \equiv (\varphi \rightarrow \psi) \land (\psi \rightarrow \varphi)$

idempotency $\varphi \lor \varphi \equiv \varphi \quad \varphi \land \varphi \equiv \varphi$

commutativity $\varphi \lor \psi \equiv \psi \lor \varphi \quad \varphi \land \psi \equiv \psi \land \varphi$

associativity $(\varphi \lor \psi) \lor \chi \equiv \varphi \lor (\psi \lor \chi) \quad (\varphi \land \psi) \land \chi \equiv \varphi \land (\psi \land \chi)$

absorption $\varphi \lor (\varphi \land \psi) \equiv \varphi \quad \varphi \land (\varphi \lor \psi) \equiv \varphi$

distributivity $\varphi \land (\psi \lor \chi) \equiv (\varphi \land \psi) \lor (\varphi \land \chi) \quad \varphi \lor (\psi \land \chi) \equiv (\varphi \lor \psi) \land (\varphi \lor \chi)$

double negation $\neg \neg \varphi \equiv \varphi$

constants $\neg \top \equiv \bot \quad \neg \bot \equiv \top$

De Morgan $\neg (\varphi \lor \psi) \equiv \neg \varphi \land \neg \psi \quad \neg (\varphi \land \psi) \equiv \neg \varphi \lor \neg \psi$

truth $\varphi \lor \top \equiv \top \quad \varphi \land \top \equiv \varphi$

falsity $\varphi \lor \bot \equiv \varphi \quad \varphi \land \bot \equiv \bot$
Some Equivalences

simplifications \( \varphi \rightarrow \psi \equiv \neg \varphi \lor \psi \) \( \varphi \leftrightarrow \psi \equiv (\varphi \rightarrow \psi) \land (\psi \rightarrow \varphi) \)

idempotency \( \varphi \lor \varphi \equiv \varphi \) \( \varphi \land \varphi \equiv \varphi \)

commutativity \( \varphi \lor \psi \equiv \psi \lor \varphi \) \( \varphi \land \psi \equiv \psi \land \varphi \)

associativity \( (\varphi \lor \psi) \lor \chi \equiv \varphi \lor (\psi \lor \chi) \) \( (\varphi \land \psi) \land \chi \equiv \varphi \land (\psi \land \chi) \)

absorption \( \varphi \lor (\varphi \land \psi) \equiv \varphi \) \( \varphi \land (\varphi \lor \psi) \equiv \varphi \)

distributivity \( \varphi \land (\psi \lor \chi) \equiv (\varphi \land \psi) \lor (\varphi \land \chi) \) \( \varphi \lor (\psi \land \chi) \equiv (\varphi \lor \psi) \land (\varphi \lor \chi) \)

double negation \( \neg \neg \varphi \equiv \varphi \)

constants \( \neg \top \equiv \bot \) \( \neg \bot \equiv \top \)

De Morgan \( \neg (\varphi \lor \psi) \equiv \neg \varphi \land \neg \psi \) \( \neg (\varphi \land \psi) \equiv \neg \varphi \lor \neg \psi \)

truth \( \varphi \lor \top \equiv \top \) \( \varphi \land \top \equiv \varphi \)

falsity \( \varphi \lor \bot \equiv \varphi \) \( \varphi \land \bot \equiv \bot \)
### Some Equivalences

<table>
<thead>
<tr>
<th>Simplifications</th>
<th>( \phi \rightarrow \psi \equiv \neg \phi \lor \psi )</th>
<th>( \phi \leftrightarrow \psi \equiv (\phi \rightarrow \psi) \land (\psi \rightarrow \phi) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Idempotency</td>
<td>( \phi \lor \phi \equiv \phi )</td>
<td>( \phi \land \phi \equiv \phi )</td>
</tr>
<tr>
<td>Commutativity</td>
<td>( \phi \lor \psi \equiv \psi \lor \phi )</td>
<td>( \phi \land \psi \equiv \psi \land \phi )</td>
</tr>
<tr>
<td>Associativity</td>
<td>( (\phi \lor \psi) \lor \chi \equiv \phi \lor (\psi \lor \chi) )</td>
<td>( (\phi \land \psi) \land \chi \equiv \phi \land (\psi \land \chi) )</td>
</tr>
<tr>
<td>Absorption</td>
<td>( \phi \lor (\phi \land \psi) \equiv \phi )</td>
<td>( \phi \land (\phi \lor \psi) \equiv \phi )</td>
</tr>
<tr>
<td>Distributivity</td>
<td>( \phi \land (\psi \lor \chi) \equiv (\phi \land \psi) \lor (\phi \land \chi) )</td>
<td>( \phi \lor (\psi \land \chi) \equiv (\phi \lor \psi) \land (\phi \lor \chi) )</td>
</tr>
<tr>
<td>Double Negation</td>
<td>( \neg \neg \phi \equiv \phi )</td>
<td>( \neg \top \equiv \bot )</td>
</tr>
<tr>
<td>Constants</td>
<td>( \neg \top \equiv \bot )</td>
<td>( \neg \bot \equiv \top )</td>
</tr>
<tr>
<td>De Morgan</td>
<td>( \neg (\phi \lor \psi) \equiv \neg \phi \land \neg \psi )</td>
<td>( \neg (\phi \land \psi) \equiv \neg \phi \lor \neg \psi )</td>
</tr>
<tr>
<td>Truth</td>
<td>( \phi \lor \top \equiv \top )</td>
<td>( \phi \land \top \equiv \phi )</td>
</tr>
<tr>
<td>Falsity</td>
<td>( \phi \lor \bot \equiv \phi )</td>
<td>( \phi \land \bot \equiv \bot )</td>
</tr>
</tbody>
</table>
How Many Different Formulae Are There . . .

. . . for a given *finite* alphabet $\Sigma$?

- Infinitely many: $a, a \lor a, a \land a, a \lor a \lor a, \ldots$
- How many different logically distinguishable (non-equivalent) formulae?
  - For $\Sigma$ with $n = |\Sigma|$, there are $2^n$ different interpretations.
  - A formula can be characterized by its set of models (if two formulae are logically non-equivalent then their sets of models differ).
  - There are $2^{(2^n)}$ different sets of interpretations.
  - There are $2^{(2^n)}$ logical equivalence classes of formulae.
How Many Different Formulae Are There . . .

. . . for a given *finite* alphabet $\Sigma$?

- **Infinitely many**: $a, a \lor a, a \land a, a \lor a \lor a, \ldots$

- How many different logically distinguishable (non-equivalent) formulae?
  - For $\Sigma$ with $n = |\Sigma|$, there are $2^n$ different interpretations.
  - A formula can be characterized by its set of models (if two formulae are logically non-equivalent then their sets of models differ).
  - There are $2^{(2^n)}$ different sets of interpretations.
  - There are $2^{(2^n)}$ logical equivalence classes of formulae.
How Many Different Formulae Are There . . .

. . . for a given $finite$ alphabet $\Sigma$?

- Infinitely many: $a, a \lor a, a \land a, a \lor a \lor a, . . .$
- How many different logically distinguishable (non-equivalent) formulae?
  - For $\Sigma$ with $n = |\Sigma|$, there are $2^n$ different interpretations.
  - A formula can be characterized by its set of models (if two formulae are logically non-equivalent then their sets of models differ).
  - There are $2^{(2^n)}$ different sets of interpretations.
  - There are $2^{(2^n)}$ logical equivalence classes of formulae.
How Many Different Formulae Are There . . .

. . . for a given \textit{finite} alphabet $\Sigma$?

- Infinitely many: $a, a \lor a, a \land a, a \lor a \lor a, \ldots$
- How many different logically distinguishable (non-equivalent) formulae?
  - For $\Sigma$ with $n = |\Sigma|$, there are $2^n$ different interpretations.
  - A formula can be characterized by its set of models (if two formulae are logically non-equivalent then their sets of models differ).
  - There are $2^{2^n}$ different sets of interpretations.
  - There are $2^{2^n}$ logical equivalence classes of formulae.
How Many Different Formulae Are There . . .

. . . for a given *finite* alphabet $\Sigma$?

- Infinitely many: $a, a \lor a, a \land a, a \lor a \lor a, \ldots$
- How many different logically distinguishable (non-equivalent) formulae?
  - For $\Sigma$ with $n = |\Sigma|$, there are $2^n$ different interpretations.
  - A formula can be characterized by its set of models (if two formulae are logically non-equivalent then their sets of models differ).
  - There are $2^{(2^n)}$ different sets of interpretations.
  - There are $2^{(2^n)}$ logical equivalence classes of formulae.
Logical Implication

- Extension of the relation $\models$ to sets $\Theta$ of formulae:

$$\mathcal{I} \models \Theta \text{ iff } \mathcal{I} \models \varphi \text{ for all } \varphi \in \Theta.$$ 

- $\varphi$ is logically implied by $\Theta$ (symbolically $\Theta \models \varphi$) iff $\varphi$ is true in all models of $\Theta$:

$$\Theta \models \varphi \text{ iff } \mathcal{I} \models \varphi \text{ for all } \mathcal{I} \text{ such that } \mathcal{I} \models \Theta.$$ 

- Some consequences:
  - Deduction theorem: $\Theta \cup \{\varphi\} \models \psi$ iff $\Theta \models \varphi \rightarrow \psi$
  - Contraposition: $\Theta \cup \{\varphi\} \models \neg \psi$ iff $\Theta \cup \{\psi\} \models \neg \varphi$
  - Contradiction: $\Theta \cup \{\varphi\}$ is unsatisfiable iff $\Theta \models \neg \varphi$
Logical Implication

- Extension of the relation $\models$ to sets $\Theta$ of formulae:

  $$\mathcal{I} \models \Theta \text{ iff } \mathcal{I} \models \varphi \text{ for all } \varphi \in \Theta.$$  

- $\varphi$ is logically implied by $\Theta$ (symbolically $\Theta \models \varphi$) iff $\varphi$ is true in all models of $\Theta$:

  $$\Theta \models \varphi \text{ iff } \mathcal{I} \models \varphi \text{ for all } \mathcal{I} \text{ such that } \mathcal{I} \models \Theta.$$  

Some consequences:

- Deduction theorem: $\Theta \cup \{\varphi\} \models \psi$ iff $\Theta \models \varphi \rightarrow \psi$
- Contraposition: $\Theta \cup \{\varphi\} \models \neg \psi$ iff $\Theta \cup \{\psi\} \models \neg \varphi$
- Contradiction: $\Theta \cup \{\varphi\}$ is unsatisfiable iff $\Theta \models \neg \varphi$
Logical Implication

- Extension of the relation |= to sets \( \Theta \) of formulae:

\[
I \models \Theta \iff I \models \varphi \text{ for all } \varphi \in \Theta.
\]

- \( \varphi \) is logically implied by \( \Theta \) (symbolically \( \Theta \models \varphi \)) iff \( \varphi \) is true in all models of \( \Theta \):

\[
\Theta \models \varphi \iff I \models \varphi \text{ for all } I \text{ such that } I \models \Theta
\]

- Some consequences:
  - Deduction theorem: \( \Theta \cup \{ \varphi \} \models \psi \iff \Theta \models \varphi \rightarrow \psi \)
  - Contraposition: \( \Theta \cup \{ \varphi \} \models \neg \psi \iff \Theta \cup \{ \psi \} \models \neg \varphi \)
  - Contradiction: \( \Theta \cup \{ \varphi \} \) is unsatisfiable iff \( \Theta \models \neg \varphi \)
Logical Implication

- Extension of the relation $|$ to sets $\Theta$ of formulae:

$$\mathcal{I} \models \Theta \text{ iff } \mathcal{I} \models \varphi \text{ for all } \varphi \in \Theta.$$  

- $\varphi$ is logically implied by $\Theta$ (symbolically $\Theta \models \varphi$) iff $\varphi$ is true in all models of $\Theta$:

$$\Theta \models \varphi \text{ iff } \mathcal{I} \models \varphi \text{ for all } \mathcal{I} \text{ such that } \mathcal{I} \models \Theta.$$  

- Some consequences:
  - Deduction theorem: $\Theta \cup \{\varphi\} \models \psi$ iff $\Theta \models \varphi \rightarrow \psi$
  - Contraposition: $\Theta \cup \{\varphi\} \models \neg \psi$ iff $\Theta \cup \{\psi\} \models \neg \varphi$
  - Contradiction: $\Theta \cup \{\varphi\}$ is unsatisfiable iff $\Theta \models \neg \varphi$
Logical Implication

- Extension of the relation $\models$ to sets $\Theta$ of formulae:

$$\mathcal{I} \models \Theta \iff \mathcal{I} \models \varphi \text{ for all } \varphi \in \Theta.$$ 

- $\varphi$ is logically implied by $\Theta$ (symbolically $\Theta \models \varphi$) iff $\varphi$ is true in all models of $\Theta$:

$$\Theta \models \varphi \iff \mathcal{I} \models \varphi \text{ for all } \mathcal{I} \text{ such that } \mathcal{I} \models \Theta$$

- Some consequences:
  - Deduction theorem: $\Theta \cup \{\varphi\} \models \psi$ iff $\Theta \models \varphi \to \psi$
  - Contraposition: $\Theta \cup \{\varphi\} \models \neg \psi$ iff $\Theta \cup \{\psi\} \models \neg \varphi$
  - Contradiction: $\Theta \cup \{\varphi\}$ is unsatisfiable iff $\Theta \models \neg \varphi$
Normal Forms

Terminology:

- Atomic formulae $a$, negated atomic formulae $\neg a$, truth $\top$ and falsity $\bot$ are literals.
- A disjunction of literals is a clause.

  - If $\neg$ only occurs in front of an atom and there are no occurrences of $\rightarrow$ and $\leftrightarrow$, the formula is in negation normal form (NNF).
  - Example: $(\neg a \lor \neg b) \land c$, but not: $\neg(a \land b) \land c$

- A conjunction of clauses is in conjunctive normal form (CNF).
  - Example: $(a \lor b) \land (\neg a \lor c)$

- The dual form (disjunction of conjunctions of literals) is in disjunctive normal form (DNF).
  - Example: $(a \land b) \lor (\neg a \land c)$
Normal Forms

Terminology:

- Atomic formulae $a$, negated atomic formulae $\neg a$, truth $\top$ and falsity $\bot$ are literals.
- A disjunction of literals is a clause.
- If $\neg$ only occurs in front of an atom and there are no occurrences of $\rightarrow$ and $\leftrightarrow$, the formula is in negation normal form (NNF).
  Example: $(\neg a \lor \neg b) \land c$, but not: $\neg(a \land b) \land c$
- A conjunction of clauses is in conjunctive normal form (CNF).
  Example: $(a \lor b) \land (\neg a \lor c)$
- The dual form (disjunction of conjunctions of literals) is in disjunctive normal form (DNF).
  Example: $(a \land b) \lor (\neg a \land c)$
Normal Forms

Terminology:
- Atomic formulae $a$, negated atomic formulae $\neg a$, truth $\top$ and falsity $\bot$ are literals.
- A disjunction of literals is a clause.
- If $\neg$ only occurs in front of an atom and there are no occurrences of $\rightarrow$ and $\leftrightarrow$, the formula is in negation normal form (NNF).
  Example: $(\neg a \lor \neg b) \land c$, but not: $\neg(a \land b) \land c$
- A conjunction of clauses is in conjunctive normal form (CNF).
  Example: $(a \lor b) \land (\neg a \lor c)$
- The dual form (disjunction of conjunctions of literals) is in disjunctive normal form (DNF).
  Example: $(a \land b) \lor (\neg a \land c)$
Terminology:

- Atomic formulae $a$, negated atomic formulae $\neg a$, truth $\top$ and falsity $\bot$ are literals.
- A disjunction of literals is a clause.
- If $\neg$ only occurs in front of an atom and there are no occurrences of $\to$ and $\leftrightarrow$, the formula is in negation normal form (NNF).
  Example: $(\neg a \lor \neg b) \land c$, but not: $\neg(a \land b) \land c$
- A conjunction of clauses is in conjunctive normal form (CNF).
  Example: $(a \lor b) \land (\neg a \lor c)$
- The dual form (disjunction of conjunctions of literals) is in disjunctive normal form (DNF).
  Example: $(a \land b) \lor (\neg a \land c)$
Negation Normal Form

**Theorem**

*For each propositional formula there is a logically equivalent formula in NNF.*

**Proof.**

First eliminate $\rightarrow$ and $\leftrightarrow$ by the appropriate equivalences. The rest of the proof is by structural induction.

Base case: Claim is true for $a$, $\neg a$, $\top$, $\bot$.

Inductive case: Assume claim is true for all formulae $\varphi$ (up to a certain number of connectives) and call its NNF $\text{nnf}(\varphi)$.

- $\text{nnf}(\varphi \land \psi) = \text{nnf}(\varphi) \land \text{nnf}(\psi)$
- $\text{nnf}(\varphi \lor \psi) = \text{nnf}(\varphi) \lor \text{nnf}(\psi)$
- $\text{nnf}(\neg (\varphi \land \psi)) = \text{nnf}(\neg \varphi) \lor \text{nnf}(\neg \psi)$
- $\text{nnf}(\neg (\varphi \lor \psi)) = \text{nnf}(\neg \varphi) \land \text{nnf}(\neg \psi)$
- $\text{nnf}(\neg (\neg \varphi)) = \text{nnf}(\varphi)$
Negation Normal Form

Theorem

For each propositional formula there is a logically equivalent formula in NNF.

Proof.

First eliminate $\rightarrow$ and $\leftrightarrow$ by the appropriate equivalences. The rest of the proof is by structural induction.

Base case: Claim is true for $a$, $\neg a$, $\top$, $\bot$.

Inductive case: Assume claim is true for all formulae $\varphi$ (up to a certain number of connectives) and call its NNF $\text{nnf}(\varphi)$.

- $\text{nnf}(\varphi \land \psi) = \text{nnf}(\varphi) \land \text{nnf}(\psi)$
- $\text{nnf}(\varphi \lor \psi) = \text{nnf}(\varphi) \lor \text{nnf}(\psi)$
- $\text{nnf}(\neg(\varphi \land \psi)) = \text{nnf}(\neg \varphi) \lor \text{nnf}(\neg \psi)$
- $\text{nnf}(\neg(\varphi \lor \psi)) = \text{nnf}(\neg \varphi) \land \text{nnf}(\neg \psi)$
- $\text{nnf}(\neg(\neg \varphi)) = \text{nnf}(\varphi)$
Negation Normal Form

**Theorem**

*For each propositional formula there is a logically equivalent formula in NNF.*

**Proof.**

First eliminate \(\rightarrow\) and \(\leftrightarrow\) by the appropriate equivalences. The rest of the proof is by structural induction.

**Base case:** Claim is true for \(a, \neg a, \top, \bot\).

**Inductive case:** Assume claim is true for all formulae \(\varphi\) (up to a certain number of connectives) and call its NNF \(\text{nnf}(\varphi)\).

- \(\text{nnf}(\varphi \land \psi) = \text{nnf}(\varphi) \land \text{nnf}(\psi)\)
- \(\text{nnf}(\varphi \lor \psi) = \text{nnf}(\varphi) \lor \text{nnf}(\psi)\)
- \(\text{nnf}(\neg(\varphi \land \psi)) = \text{nnf}(\neg \varphi) \lor \text{nnf}(\neg \psi)\)
- \(\text{nnf}(\neg(\varphi \lor \psi)) = \text{nnf}(\neg \varphi) \land \text{nnf}(\neg \psi)\)
- \(\text{nnf}(\neg(\neg \varphi)) = \text{nnf}(\varphi)\)
Negation Normal Form

**Theorem**

*For each propositional formula there is a logically equivalent formula in NNF.*

**Proof.**

First eliminate $\rightarrow$ and $\leftrightarrow$ by the appropriate equivalences. The rest of the proof is by structural induction.

**Base case:** Claim is true for $a$, $\neg a$, $\top$, $\bot$.

**Inductive case:** Assume claim is true for all formulae $\varphi$ (up to a certain number of connectives) and call its NNF $\text{nnf}(\varphi)$.

- $\text{nnf}(\varphi \land \psi) = \text{nnf}(\varphi) \land \text{nnf}(\psi)$
- $\text{nnf}(\varphi \lor \psi) = \text{nnf}(\varphi) \lor \text{nnf}(\psi)$
- $\text{nnf}(\neg(\varphi \land \psi)) = \text{nnf}(\neg \varphi) \lor \text{nnf}(\neg \psi)$
- $\text{nnf}(\neg(\varphi \lor \psi)) = \text{nnf}(\neg \varphi) \land \text{nnf}(\neg \psi)$
- $\text{nnf}(\neg(\neg \varphi)) = \text{nnf}(\varphi)$
Negation Normal Form

**Theorem**

*For each propositional formula there is a logically equivalent formula in NNF.*

**Proof.**

First eliminate $\rightarrow$ and $\leftrightarrow$ by the appropriate equivalences. The rest of the proof is by structural induction.

**Base case:** Claim is true for $a$, $\neg a$, $\top$, $\bot$.

**Inductive case:** Assume claim is true for all formulae $\varphi$ (up to a certain number of connectives) and call its NNF $nnf(\varphi)$.

- $nnf(\varphi \land \psi) = nnf(\varphi) \land nnf(\psi)$
- $nnf(\varphi \lor \psi) = nnf(\varphi) \lor nnf(\psi)$
- $nnf(\neg(\varphi \land \psi)) = nnf(\neg \varphi) \lor nnf(\neg \psi)$
- $nnf(\neg(\varphi \lor \psi)) = nnf(\neg \varphi) \land nnf(\neg \psi)$
- $nnf(\neg(\neg \varphi)) = nnf(\varphi)$
Theorem

For each propositional formula there is a logically equivalent formula in NNF.

Proof.

First eliminate $\to$ and $\leftrightarrow$ by the appropriate equivalences. The rest of the proof is by structural induction.  

Base case: Claim is true for $a, \neg a, \top, \bot$.  

Inductive case: Assume claim is true for all formulae $\varphi$ (up to a certain number of connectives) and call its NNF $nnf(\varphi)$.

- $nnf(\varphi \land \psi) = nnf(\varphi) \land nnf(\psi)$
- $nnf(\varphi \lor \psi) = nnf(\varphi) \lor nnf(\psi)$
- $nnf(\neg(\varphi \land \psi)) = nnf(\neg \varphi) \lor nnf(\neg \psi)$
- $nnf(\neg(\varphi \lor \psi)) = nnf(\neg \varphi) \land nnf(\neg \psi)$
- $nnf(\neg(\neg \varphi)) = nnf(\varphi)$
**Theorem**

*For each propositional formula there is a logically equivalent formula in NNF.*

**Proof.**

First eliminate $→$ and $↔$ by the appropriate equivalences. The rest of the proof is by structural induction.

**Base case:** Claim is true for $a$, $¬a$, $⊤$, $⊥$.

**Inductive case:** Assume claim is true for all formulae $φ$ (up to a certain number of connectives) and call its NNF $nnf(φ)$.

- $nnf(φ \land ψ) = nnf(φ) \land nnf(ψ)$
- $nnf(φ \lor ψ) = nnf(φ) \lor nnf(ψ)$
- $nnf(¬(φ \land ψ)) = nnf(¬φ) \lor nnf(¬ψ)$
- $nnf(¬(φ \lor ψ)) = nnf(¬φ) \land nnf(¬ψ)$
- $nnf(¬(¬φ)) = nnf(φ)$
Theorem

For each propositional formula there is a logically equivalent formula in NNF.

Proof.

First eliminate $\rightarrow$ and $\leftrightarrow$ by the appropriate equivalences. The rest of the proof is by structural induction.

Base case: Claim is true for $a, \neg a, \top, \bot$.

Inductive case: Assume claim is true for all formulae $\varphi$ (up to a certain number of connectives) and call its NNF $\text{nnf}(\varphi)$.

- $\text{nnf}(\varphi \land \psi) = \text{nnf}(\varphi) \land \text{nnf}(\psi)$
- $\text{nnf}(\varphi \lor \psi) = \text{nnf}(\varphi) \lor \text{nnf}(\psi)$
- $\text{nnf}(\neg(\varphi \land \psi)) = \text{nnf}(\neg \varphi) \lor \text{nnf}(\neg \psi)$
- $\text{nnf}(\neg(\varphi \lor \psi)) = \text{nnf}(\neg \varphi) \land \text{nnf}(\neg \psi)$
- $\text{nnf}(\neg(\neg \varphi)) = \text{nnf}(\varphi)$
Conjunctive Normal Form

Theorem

For each propositional formula there is a logically equivalent formula in CNF. A similar argument works for DNF!

- True for \( a, \neg a, \top, \bot \).
- Let us assume it is true for all formulae \( \varphi \) (up to a certain number of connectives) and call its CNF \( \text{cnf}(\varphi) \).
  - \( \text{cnf}(\neg \varphi) = \text{cnf}(\text{nnf}(\neg \varphi)) \)
  - \( \text{cnf}(\varphi \land \psi) = \text{cnf}(\varphi) \land \text{cnf}(\psi) \)
  - Assume \( \text{cnf}(\varphi) = \bigwedge_i \chi_i \) and \( \text{cnf}(\psi) = \bigwedge_j \rho_j \) with \( \chi_i, \rho_j \) being clauses. Then

\[
\text{cnf}(\varphi \lor \psi) = \text{cnf}(\bigwedge_i \chi_i \lor (\bigwedge_j \rho_j)) \\
= \bigwedge_i \bigwedge_j (\chi_i \lor \rho_j) \quad \text{(by distributivity)}
\]
Conjunctive Normal Form

**Theorem**

*For each propositional formula there is a logically equivalent formula in CNF. A similar argument works for DNF!*

- **True for** $a, \neg a, \top, \bot$.
- Let us assume it is true for all formulae $\varphi$ (up to a certain number of connectives) and call its CNF $\text{cnf}(\varphi)$.
  - $\text{cnf}(\neg \varphi) = \text{cnf}(\text{nnf}(\neg \varphi))$
  - $\text{cnf}(\varphi \land \psi) = \text{cnf}(\varphi) \land \text{cnf}(\psi)$
  - Assume $\text{cnf}(\varphi) = \bigwedge_i \chi_i$ and $\text{cnf}(\psi) = \bigwedge_j \rho_j$ with $\chi_i, \rho_j$ being clauses. Then
    $$\text{cnf}(\varphi \lor \psi) = \text{cnf}(\bigwedge_i \chi_i \lor \bigwedge_j \rho_j)$$
    $$= \bigwedge_i \bigwedge_j (\chi_i \lor \rho_j) \quad \text{(by distributivity)}$$
For each propositional formula there is a logically equivalent formula in CNF. A similar argument works for DNF!

- True for $a$, $\neg a$, $\top$, $\bot$.
- Let us assume it is true for all formulae $\varphi$ (up to a certain number of connectives) and call its CNF $\text{cnf}(\varphi)$.
  - $\text{cnf}(\neg \varphi) = \text{cnf}(\text{nnf}(\neg \varphi))$
  - $\text{cnf}(\varphi \land \psi) = \text{cnf}(\varphi) \land \text{cnf}(\psi)$
  - Assume $\text{cnf}(\varphi) = \bigwedge_i \chi_i$ and $\text{cnf}(\psi) = \bigwedge_j \rho_j$ with $\chi_i$, $\rho_j$ being clauses. Then

\[
\text{cnf}(\varphi \lor \psi) = \text{cnf}((\bigwedge_i \chi_i) \lor (\bigwedge_j \rho_j)) = \bigwedge_i \bigwedge_j (\chi_i \lor \rho_j) \quad \text{(by distributivity)}
\]
Theorem

For each propositional formula there is a logically equivalent formula in CNF. A similar argument works for DNF!

- True for $a, \neg a, \top, \bot$.
- Let us assume it is true for all formulae $\varphi$ (up to a certain number of connectives) and call its CNF $\text{cnf}(\varphi)$.
  - $\text{cnf}(\neg \varphi) = \text{cnf}(\text{nnf}(\neg \varphi))$
  - $\text{cnf}(\varphi \land \psi) = \text{cnf}(\varphi) \land \text{cnf}(\psi)$
  - Assume $\text{cnf}(\varphi) = \bigwedge_i \chi_i$ and $\text{cnf}(\psi) = \bigwedge_j \rho_j$ with $\chi_i, \rho_j$ being clauses. Then

$$\text{cnf}(\varphi \lor \psi) = \text{cnf}(\bigwedge_i \chi_i \lor \bigwedge_j \rho_j))$$

$$= \bigwedge_i \bigwedge_j (\chi_i \lor \rho_j) \quad \text{(by distributivity)}$$
Conjunctive Normal Form

**Theorem**

*For each propositional formula there is a logically equivalent formula in CNF. A similar argument works for DNF!*

- True for \( a, \neg a, \top, \bot \).
- Let us assume it is true for all formulae \( \varphi \) (up to a certain number of connectives) and call its CNF \( \text{cnf}(\varphi) \).
  - \( \text{cnf}(\neg \varphi) = \text{cnf}(\text{nnf}(\neg \varphi)) \)
  - \( \text{cnf}(\varphi \land \psi) = \text{cnf}(\varphi) \land \text{cnf}(\psi) \)
  - Assume \( \text{cnf}(\varphi) = \bigwedge_i \chi_i \) and \( \text{cnf}(\psi) = \bigwedge_j \rho_j \) with \( \chi_i, \rho_j \) being clauses. Then

\[
\text{cnf}(\varphi \lor \psi) = \text{cnf}((\bigwedge_i \chi_i) \lor (\bigwedge_j \rho_j))
\]

\[
= \bigwedge_i \bigwedge_j (\chi_i \lor \rho_j) \quad \text{(by distributivity)}
\]
Theorem

For each propositional formula there is a logically equivalent formula in CNF. A similar argument works for DNF!

- True for $a, \neg a, \top, \bot$.
- Let us assume it is true for all formulae $\varphi$ (up to a certain number of connectives) and call its CNF $\text{cnf}(\varphi)$.
  - $\text{cnf}(\neg \varphi) = \text{cnf}(\text{nnf}(\neg \varphi))$
  - $\text{cnf}(\varphi \land \psi) = \text{cnf}(\varphi) \land \text{cnf}(\psi)$
  - Assume $\text{cnf}(\varphi) = \bigwedge_i \chi_i$ and $\text{cnf}(\psi) = \bigwedge_j \rho_j$ with $\chi_i, \rho_j$ being clauses. Then

$$
\text{cnf}(\varphi \lor \psi) = \text{cnf}(\bigwedge_i \chi_i \lor (\bigwedge_j \rho_j))
$$

$$
= \bigwedge_i \bigwedge_j (\chi_i \lor \rho_j) \quad \text{(by distributivity)}
$$
Theorem

*For each propositional formula there is a logically equivalent formula in CNF. A similar argument works for DNF!*

- True for $a, \neg a, \top, \bot$.
- Let us assume it is true for all formulae $\varphi$ (up to a certain number of connectives) and call its CNF $\text{cnf}(\varphi)$.
  - $\text{cnf}(\neg \varphi) = \text{cnf}(\text{nff}(\neg \varphi))$
  - $\text{cnf}(\varphi \land \psi) = \text{cnf}(\varphi) \land \text{cnf}(\psi)$
  - Assume $\text{cnf}(\varphi) = \bigwedge_i \chi_i$ and $\text{cnf}(\psi) = \bigwedge_j \rho_j$ with $\chi_i, \rho_j$ being clauses. Then

\[
\text{cnf}(\varphi \lor \psi) = \text{cnf}((\bigwedge_i \chi_i) \lor (\bigwedge_j \rho_j))
\]

\[
= \bigwedge_i \bigwedge_j (\chi_i \lor \rho_j) \quad \text{(by distributivity)}
\]
Conjunctive Normal Form

Theorem

For each propositional formula there is a logically equivalent formula in CNF. A similar argument works for DNF!

- True for $a, \neg a, \top, \bot$.
- Let us assume it is true for all formulae $\varphi$ (up to a certain number of connectives) and call its CNF $\text{cnf}(\varphi)$.
  - $\text{cnf}(\neg \varphi) = \text{cnf}(\text{nnf}(\neg \varphi))$
  - $\text{cnf}(\varphi \land \psi) = \text{cnf}(\varphi) \land \text{cnf}(\psi)$
  - Assume $\text{cnf}(\varphi) = \bigwedge_i \chi_i$ and $\text{cnf}(\psi) = \bigwedge_j \rho_j$ with $\chi_i, \rho_j$ being clauses. Then

\[
\text{cnf}(\varphi \lor \psi) = \text{cnf}((\bigwedge_i \chi_i) \lor (\bigwedge_j \rho_j))
\]
\[
= \bigwedge_i \bigwedge_j (\chi_i \lor \rho_j) \quad \text{(by distributivity)}
\]
How do we decide whether a formula is satisfiable, unsatisfiable, valid, or falsifiable?

Note: Satisfiability and falsifiability are NP-complete. Validity and unsatisfiability are co-NP-complete.

- A CNF formula is valid iff all clauses contain two complementary literals or $\top$.
- A DNF formula is satisfiable iff one disjunct does not contain $\bot$ or two complementary literals.
- However, transformation to CNF or DNF may take exponential time (and space!).
- One can try out all truth assignments.
- One can test systematically for satisfying truth assignments (backtracking search) $\leadsto$ Davis-Putnam procedure (DP)
How to Decide Properties of Formulae

How do we decide whether a formula is satisfiable, unsatisfiable, valid, or falsifiable?
Note: Satisfiability and falsifiability are NP-complete. Validity and unsatisfiability are co-NP-complete.

- A CNF formula is valid iff all clauses contain two complementary literals or $\top$.
- A DNF formula is satisfiable iff one disjunct does not contain $\bot$ or two complementary literals.
- However, transformation to CNF or DNF may take exponential time (and space!).
- One can try out all truth assignments.
- One can test systematically for satisfying truth assignments (backtracking search) $\rightsquigarrow$ Davis-Putnam procedure (DP)
How do we decide whether a formula is satisfiable, unsatisfiable, valid, or falsifiable?

Note: Satisfiability and falsifiability are NP-complete. Validity and unsatisfiability are co-NP-complete.

- A CNF formula is valid iff all clauses contain two complementary literals or $\top$.
- A DNF formula is satisfiable iff one disjunct does not contain $\bot$ or two complementary literals.

However, transformation to CNF or DNF may take exponential time (and space!).

- One can try out all truth assignments.
- One can test systematically for satisfying truth assignments (backtracking search) \(\leadsto\) Davis-Putnam procedure (DP)
How do we decide whether a formula is satisfiable, unsatisfiable, valid, or falsifiable? 
Note: Satisfiability and falsifiability are NP-complete. Validity and unsatisfiability are co-NP-complete.

- A CNF formula is valid iff all clauses contain two complementary literals or $\top$.
- A DNF formula is satisfiable iff one disjunct does not contain $\bot$ or two complementary literals.
- However, transformation to CNF or DNF may take exponential time (and space!).

- One can try out all truth assignments.
- One can test systematically for satisfying truth assignments (backtracking search) $\leadsto$ Davis-Putnam procedure (DP)
How to Decide Properties of Formulae

How do we decide whether a formula is satisfiable, unsatisfiable, valid, or falsifiable?

Note: Satisfiability and falsifiability are NP-complete. Validity and unsatisfiability are co-NP-complete.

- A CNF formula is valid iff all clauses contain two complementary literals or $\top$.
- A DNF formula is satisfiable iff one disjunct does not contain $\bot$ or two complementary literals.
- However, transformation to CNF or DNF may take exponential time (and space!).
- One can try out all truth assignments.
- One can test systematically for satisfying truth assignments (backtracking search) $\leadsto$ Davis-Putnam procedure (DP)
How do we decide whether a formula is satisfiable, unsatisfiable, valid, or falsifiable?

Note: Satisfiability and falsifiability are NP-complete. Validity and unsatisfiability are co-NP-complete.

- A CNF formula is valid iff all clauses contain two complementary literals or $\top$.

- A DNF formula is satisfiable iff one disjunct does not contain $\bot$ or two complementary literals.

- However, transformation to CNF or DNF may take exponential time (and space!).

- One can try out all truth assignments.

- One can test systematically for satisfying truth assignments (backtracking search) $\rightsquigarrow$ Davis-Putnam procedure (DP)
Deciding Entailment

- We want to decide $\Theta \models \varphi$.
- Use deduction theorem and reduce to validity:
  
  $$\Theta \models \varphi \text{ iff } \bigwedge \Theta \rightarrow \varphi \text{ is valid.}$$

- Now negate and test for unsatisfiability using DP.
- Different approach: Try to derive $\varphi$ from $\Theta$ – find a proof of $\varphi$ from $\Theta$.
- Use inference rules to derive new formulae from $\Theta$. Continue to deduce new formulae until $\varphi$ can be deduced.
- One particular calculus: resolution
Deciding Entailment

- We want to decide $\Theta \models \varphi$.
- Use deduction theorem and reduce to validity:

$$\Theta \models \varphi \iff \bigwedge \Theta \rightarrow \varphi \text{ is valid.}$$

- Now negate and test for unsatisfiability using DP.
- Different approach: Try to derive $\varphi$ from $\Theta$ – find a proof of $\varphi$ from $\Theta$.
- Use inference rules to derive new formulae from $\Theta$. Continue to deduce new formulae until $\varphi$ can be deduced.
- One particular calculus: resolution
Deciding Entailment

- We want to decide $\Theta \models \varphi$.
- Use deduction theorem and reduce to validity:

$$\Theta \models \varphi \text{ iff } \bigwedge \Theta \rightarrow \varphi \text{ is valid.}$$

- Now negate and test for unsatisfiability using DP.
- Different approach: Try to derive $\varphi$ from $\Theta$ – find a proof of $\varphi$ from $\Theta$

  - Use inference rules to derive new formulae from $\Theta$.
  - Continue to deduce new formulae until $\varphi$ can be deduced.

- One particular calculus: resolution
Deciding Entailment

- We want to decide $\Theta \models \varphi$.
- Use deduction theorem and reduce to validity:
  $$\Theta \models \varphi \text{ iff } \bigwedge \Theta \rightarrow \varphi \text{ is valid.}$$
- Now negate and test for unsatisfiability using DP.
- Different approach: Try to derive $\varphi$ from $\Theta$ — find a proof of $\varphi$ from $\Theta$.
- Use inference rules to derive new formulae from $\Theta$. Continue to deduce new formulae until $\varphi$ can be deduced.
- One particular calculus: resolution
Deciding Entailment

- We want to decide $\Theta \models \varphi$.
- Use deduction theorem and reduce to validity:

  $$\Theta \models \varphi \iff \bigwedge \Theta \to \varphi \text{ is valid.}$$

- Now negate and test for unsatisfiability using DP.
- Different approach: Try to derive $\varphi$ from $\Theta$ — find a proof of $\varphi$ from $\Theta$.
- Use inference rules to derive new formulae from $\Theta$. Continue to deduce new formulae until $\varphi$ can be deduced.
- One particular calculus: resolution
Resolution: Representation

- We assume that all formulae are in CNF.
  - Can be generated using the described method.
  - Often formulae are already close to CNF.
  - There is a “cheap” conversion from arbitrary formulae to CNF that preserves satisfiability – which is enough as we will see.

- More convenient representation
  - CNF formula is represented as set.
  - Each clause is a set of literals.
  - \((a \lor \neg b) \land (\neg a \lor c) \leadsto \{\{a, \neg b\}, \{\neg a, c\}\}

- Empty clause (symbolically \(\Box\)) and empty set of clauses (symbolically \(\emptyset\)) are different!
Resolution: Representation

- We assume that all formulae are in CNF.
  - Can be generated using the described method.
  - Often formulae are already close to CNF.
  - There is a “cheap” conversion from arbitrary formulae to CNF that preserves satisfiability – which is enough as we will see.

- More convenient representation
  - CNF formula is represented as set.
  - Each clause is a set of literals.
  - \((a \lor \neg b) \land (\neg a \lor c) \leadsto \{\{a, \neg b\}, \{\neg a, c\}\}

- Empty clause (symbolically \(\Box\)) and empty set of clauses (symbolically \(\emptyset\)) are different!
Resolution: Representation

- We assume that all formulae are in CNF.
  - Can be generated using the described method.
  - Often formulae are already close to CNF.
  - There is a “cheap” conversion from arbitrary formulae to CNF that preserves satisfiability – which is enough as we will see.

- More convenient representation
  - CNF formula is represented as set.
  - Each clause is a set of literals.
  - \((a \lor \neg b) \land (\neg a \lor c) \leadsto \{\{a, \neg b\}, \{\neg a, c\}\}\)

- Empty clause (symbolically □) and empty set of clauses (symbolically ∅) are different!
Resolution: The Inference Rule

Let $l$ be a literal and $\bar{l}$ its complement.

The resolution rule

\[
\frac{C_1 \cup \{l\}, C_2 \cup \{\bar{l}\}}{C_1 \cup C_2}
\]

$C_1 \cup C_2$ is the resolvent of the parent clauses $C_1 \cup \{l\}$ and $C_2 \cup \{\bar{l}\}$. $l$ and $\bar{l}$ are the resolution literals.

Example: $\{a, b, \neg c\}$ resolves with $\{a, d, c\}$ to $\{a, b, d\}$.

Note: The resolvent is not logically equivalent to the set of parent clauses!

Notation:

\[
R(\Delta) = \Delta \cup \{C | C \text{ is resolvent of two clauses in } \Delta\}
\]
Resolution: The Inference Rule

Let \( l \) be a literal and \( \overline{l} \) its complement.

The resolution rule

\[
\frac{C_1 \cup \{l\}, \ C_2 \cup \{\overline{l}\}}{C_1 \cup \overline{C_2}}
\]

\( C_1 \cup C_2 \) is the resolvent of the parent clauses \( C_1 \cup \{l\} \) and \( C_2 \cup \{\overline{l}\} \). \( l \) and \( \overline{l} \) are the resolution literals.

Example: \( \{a, b, \neg c\} \) resolves with \( \{a, d, c\} \) to \( \{a, b, d\} \).

Note: The resolvent is not logically equivalent to the set of parent clauses!

Notation:

\[
R(\Delta) = \Delta \cup \{C | C \text{ is resolvent of two clauses in } \Delta\}
\]
Resolution: The Inference Rule

Let \( l \) be a literal and \( \bar{l} \) its complement.

**The resolution rule**

\[
\frac{C_1 \cup \{l\}, C_2 \cup \{\bar{l}\}}{C_1 \cup C_2}
\]

\( C_1 \cup C_2 \) is the **resolvent** of the parent clauses \( C_1 \cup \{l\} \) and \( C_2 \cup \{\bar{l}\} \). \( l \) and \( \bar{l} \) are the **resolution literals**.

**Example:** \( \{a, b, \neg c\} \) resolves with \( \{a, d, c\} \) to \( \{a, b, d\} \).

Note: The resolvent is **not** logically equivalent to the set of parent clauses!

**Notation:**

\[
R(\Delta) = \Delta \cup \{C|C \text{ is resolvent of two clauses in } \Delta\}
\]
Resolution: The Inference Rule

Let $l$ be a literal and $\overline{l}$ its complement.

The resolution rule

\[
\frac{C_1 \cup \{l\}, \, C_2 \cup \{\overline{l}\}}{C_1 \cup C_2}
\]

$C_1 \cup C_2$ is the resolvent of the parent clauses $C_1 \cup \{l\}$ and $C_2 \cup \{\overline{l}\}$. $l$ and $\overline{l}$ are the resolution literals.

Example: $\{a, b, \neg c\}$ resolves with $\{a, d, c\}$ to $\{a, b, d\}$.

Note: The resolvent is not logically equivalent to the set of parent clauses!

Notation:

\[
R(\Delta) = \Delta \cup \{C|C \text{ is resolvent of two clauses in } \Delta\}
\]
Resolution: The Inference Rule

Let \( l \) be a literal and \( \bar{l} \) its complement.

**The resolution rule**

\[
\frac{C_1 \cup \{ l \}, C_2 \cup \{ \bar{l} \}}{C_1 \cup C_2}
\]

\( C_1 \cup C_2 \) is the **resolvent** of the parent clauses \( C_1 \cup \{ l \} \) and \( C_2 \cup \{ \bar{l} \} \). \( l \) and \( \bar{l} \) are the **resolution literals**.

**Example:** \( \{ a, b, \neg c \} \) resolves with \( \{ a, d, c \} \) to \( \{ a, b, d \} \).

**Note:** The resolvent is **not** logically equivalent to the set of parent clauses!

**Notation:**

\[
R(\Delta) = \Delta \cup \{ C | C \text{ is resolvent of two clauses in } \Delta \}
\]
Resolution: Derivations

\( D \) can be derived from \( \Delta \) by resolution (symbolically \( \Delta \vdash D \)) if there is a sequence \( C_1, \ldots, C_n \) of clauses such that

1. \( C_n = D \) and
2. \( C_i \in R(\Delta \cup \{C_1, \ldots, C_{i-1}\}) \), for all \( i \in \{1, \ldots, n\} \).

Define \( R^*(\Delta) = \{D | \Delta \vdash D \} \).

Theorem (Soundness of resolution)

Let \( D \) be a clause. If \( \Delta \vdash D \) then \( \Delta \models D \).

Proof idea.

Show \( \Delta \models D \) if \( D \in R(\Delta) \) and use induction on proof length.
Let \( C_1 \cup \{l\} \) and \( C_2 \cup \{\bar{l}\} \) be the parent clauses of \( D = C_1 \cup C_2 \).
Assume \( I \models \Delta \), we have to show \( I \models D \).
Case 1: \( I \models l \) then there must be a literal \( m \in C_2 \) s.t. \( I \models m \). This implies \( I \models D \).
Case 2: \( I \models \bar{l} \) similarly, there is \( m \in C_1 \) s.t. \( I \models m \).
This means that each model \( I \) of \( \Delta \) also satisfies \( D \), i.e., \( \Delta \models D \).
Resolution: Derivations

$D$ can be derived from $\Delta$ by resolution (symbolically $\Delta \vdash D$) if there is a sequence $C_1, \ldots, C_n$ of clauses such that

1. $C_n = D$ and
2. $C_i \in R(\Delta \cup \{C_1, \ldots, C_{i-1}\})$, for all $i \in \{1, \ldots, n\}$.

Define $R^*(\Delta) = \{D | \Delta \vdash D\}$.

Theorem (Soundness of resolution)

Let $D$ be a clause. If $\Delta \vdash D$ then $\Delta \models D$.

Proof idea.

Show $\Delta \models D$ if $D \in R(\Delta)$ and use induction on proof length.

Let $C_1 \cup \{l\}$ and $C_2 \cup \{\bar{l}\}$ be the parent clauses of $D = C_1 \cup C_2$.

Assume $I \models \Delta$, we have to show $I \models D$.

Case 1: $I \models l$ then there must be a literal $m \in C_2$ s.t. $I \models m$. This implies $I \models D$.

Case 2: $I \models \bar{l}$ similarly, there is $m \in C_1$ s.t. $I \models m$.

This means that each model $I$ of $\Delta$ also satisfies $D$, i.e., $\Delta \models D$. 

Resolution: Derivations

$D$ can be derived from $\Delta$ by resolution (symbolically $\Delta \vdash D$) if there is a sequence $C_1, \ldots, C_n$ of clauses such that

1. $C_n = D$ and
2. $C_i \in R(\Delta \cup \{C_1, \ldots, C_{i-1}\})$, for all $i \in \{1, \ldots, n\}$.

Define $R^*(\Delta) = \{D | \Delta \vdash D\}$.

**Theorem (Soundness of resolution)**

*Let $D$ be a clause. If $\Delta \vdash D$ then $\Delta \models D$.*

**Proof idea.**

Show $\Delta \models D$ if $D \in R(\Delta)$ and use induction on proof length.

Let $C_1 \cup \{l\}$ and $C_2 \cup \{\bar{l}\}$ be the parent clauses of $D = C_1 \cup C_2$.

Assume $\mathcal{I} \models \Delta$, we have to show $\mathcal{I} \models D$.

Case 1: $\mathcal{I} \models l$ then there must be a literal $m \in C_2$ s.t. $\mathcal{I} \models m$. This implies $\mathcal{I} \models D$.

Case 2: $\mathcal{I} \models \bar{l}$ similarly, there is $m \in C_1$ s.t. $\mathcal{I} \models m$.

This means that each model $\mathcal{I}$ of $\Delta$ also satisfies $D$, i.e., $\Delta \models D$. 


Resolution: Derivations

A clause $D$ can be derived from $\Delta$ by resolution (symbolically $\Delta \vdash D$) if there is a sequence $C_1, \ldots, C_n$ of clauses such that

1. $C_n = D$ and
2. $C_i \in R(\Delta \cup \{C_1, \ldots, C_{i-1}\})$, for all $i \in \{1, \ldots, n\}$.

Define $R^*(\Delta) = \{D | \Delta \vdash D\}$.

Theorem (Soundness of resolution)

Let $D$ be a clause. If $\Delta \vdash D$ then $\Delta \models D$.

Proof idea.

Show $\Delta \models D$ if $D \in R(\Delta)$ and use induction on proof length.
Let $C_1 \cup \{l\}$ and $C_2 \cup \{\bar{l}\}$ be the parent clauses of $D = C_1 \cup C_2$.
Assume $\mathcal{I} \models \Delta$, we have to show $\mathcal{I} \models D$.

Case 1: $\mathcal{I} \models l$ then there must be a literal $m \in C_2$ s.t. $\mathcal{I} \models m$. This implies $\mathcal{I} \models D$.

Case 2: $\mathcal{I} \models \bar{l}$ similarly, there is $m \in C_1$ s.t. $\mathcal{I} \models m$.

This means that each model $\mathcal{I}$ of $\Delta$ also satisfies $D$, i.e., $\Delta \models D$. 

Classical Logic

Propositional Logic

Syntax

Semantics

Terminology

Normal forms

Decision Problems

Resolution

Derivations

Completeness

Resolution strategies

Horn clauses
Resolution: Derivations

\( D \) can be derived from \( \Delta \) by resolution (symbolically \( \Delta \vdash D \)) if there is a sequence \( C_1, \ldots, C_n \) of clauses such that

1. \( C_n = D \) and
2. \( C_i \in R(\Delta \cup \{C_1, \ldots, C_{i-1}\}) \), for all \( i \in \{1, \ldots, n\} \).

Define \( R^*(\Delta) = \{ D | \Delta \vdash D \} \).

Theorem (Soundness of resolution)

Let \( D \) be a clause. If \( \Delta \vdash D \) then \( \Delta \models D \).

Proof idea.

Show \( \Delta \models D \) if \( D \in R(\Delta) \) and use induction on proof length.
Let \( C_1 \cup \{l\} \) and \( C_2 \cup \{\overline{l}\} \) be the parent clauses of \( D = C_1 \cup C_2 \).
Assume \( I \models \Delta \), we have to show \( I \models D \).

Case 1: \( I \models l \) then there must be a literal \( m \in C_2 \) s.t. \( I \models m \). This implies \( I \models D \).

Case 2: \( I \models \overline{l} \) similarly, there is \( m \in C_1 \) s.t. \( I \models m \).

This means that each model \( I \) of \( \Delta \) also satisfies \( D \), i.e., \( \Delta \models D \).
Resolution: Derivations

*D* can be derived from *Δ* by resolution (symbolically *Δ ⊢ D*) if there is a sequence *C_1, . . . , C_n* of clauses such that

1. *C_n = D* and
2. *C_i ∈ R(Δ ∪ {C_1, . . . , C_{i−1}})*, for all *i ∈ {1, . . . , n}.*

Define *R*(Δ) = {*D|Δ ⊢ D*}.

**Theorem (Soundness of resolution)**

*Let D be a clause. If Δ ⊢ D then Δ |= D.*

**Proof idea.**

Show Δ |= D if *D ∈ R*(Δ) and use induction on proof length.

Let *C_1 ∪ {l}* and *C_2 ∪ {l̅}* be the parent clauses of *D = C_1 ∪ C_2.*

Assume *I |= Δ*, we have to show *I |= D*.

Case 1: *I |= l* then there must be a literal *m ∈ C_2* s.t. *I |= m*. This implies *I |= D*.

Case 2: *I |= l̅* similarly, there is *m ∈ C_1* s.t. *I |= m*.

This means that each model *I* of *Δ* also satisfies *D*, i.e., *Δ |= D*. 
Resolution: Derivations

A clause $D$ can be derived from a set of clauses $\Delta$ (symbolically $\Delta \vdash D$) if there is a sequence $C_1, \ldots, C_n$ of clauses such that:

1. $C_n = D$ and
2. $C_i \in R(\Delta \cup \{C_1, \ldots, C_{i-1}\})$, for all $i \in \{1, \ldots, n\}$.

Define $R^*(\Delta) = \{ D \mid \Delta \vdash D \}$.

Theorem (Soundness of resolution)

Let $D$ be a clause. If $\Delta \vdash D$ then $\Delta \models D$.

Proof idea.

Show $\Delta \models D$ if $D \in R(\Delta)$ and use induction on proof length. Let $C_1 \cup \{l\}$ and $C_2 \cup \{\bar{l}\}$ be the parent clauses of $D = C_1 \cup C_2$. Assume $I \models \Delta$, we have to show $I \models D$.

Case 1: $I \models l$ then there must be a literal $m \in C_2$ s.t. $I \models m$. This implies $I \models D$.

Case 2: $I \models \bar{l}$ similarly, there is $m \in C_1$ s.t. $I \models m$.

This means that each model $I$ of $\Delta$ also satisfies $D$, i.e., $\Delta \models D$. 

Resolution: Derivations

\( D \) can be derived from \( \Delta \) by resolution (symbolically \( \Delta \vdash D \)) if there is a sequence \( C_1, \ldots, C_n \) of clauses such that

1. \( C_n = D \) and
2. \( C_i \in R(\Delta \cup \{C_1, \ldots, C_{i-1}\}) \), for all \( i \in \{1, \ldots, n\} \).

Define \( R^*(\Delta) = \{ D | \Delta \vdash D \} \).

Theorem (Soundness of resolution)

\textit{Let} \( D \) \textit{be a clause. If} \( \Delta \vdash D \) \textit{then} \( \Delta \models D \).

Proof idea.

Show \( \Delta \models D \) if \( D \in R(\Delta) \) and use induction on proof length.

Let \( C_1 \cup \{l\} \) and \( C_2 \cup \{\bar{l}\} \) be the parent clauses of \( D = C_1 \cup C_2 \).

Assume \( I \models \Delta \), we have to show \( I \models D \).

Case 1: \( I \models l \) then there must be a literal \( m \in C_2 \) s.t. \( I \models m \). This implies \( I \models D \).

Case 2: \( I \models \bar{l} \) similarly, there is \( m \in C_1 \) s.t. \( I \models m \).

This means that each model \( I \) of \( \Delta \) also satisfies \( D \), i.e., \( \Delta \models D \).
Resolution: Derivations

\(D\) can be derived from \(\Delta\) by resolution (symbolically \(\Delta \vdash D\)) if there is a sequence \(C_1, \ldots, C_n\) of clauses such that

1. \(C_n = D\) and
2. \(C_i \in R(\Delta \cup \{C_1, \ldots, C_{i-1}\})\), for all \(i \in \{1, \ldots, n\}\).

Define \(R^*(\Delta) = \{D | \Delta \vdash D\}\).

**Theorem (Soundness of resolution)**

*Let \(D\) be a clause. If \(\Delta \vdash D\) then \(\Delta \models D\).*

**Proof idea.**

Show \(\Delta \models D\) if \(D \in R(\Delta)\) and use induction on proof length.

Let \(C_1 \cup \{l\}\) and \(C_2 \cup \{\bar{l}\}\) be the parent clauses of \(D = C_1 \cup C_2\).

Assume \(I \models \Delta\), we have to show \(I \models D\).

Case 1: \(I \models l\) then there must be a literal \(m \in C_2\) s.t. \(I \models m\). This implies \(I \models D\).

Case 2: \(I \models \bar{l}\) similarly, there is \(m \in C_1\) s.t. \(I \models m\).

This means that each model \(I\) of \(\Delta\) also satisfies \(D\), i.e., \(\Delta \models D\).
Resolution: Completeness?

Do we have

$$\Delta \models \varphi \text{ implies } \Delta \vdash \varphi?$$

Of course, could only hold for CNF. However:

$$\left\{\{a, b\}, \{-b, c\}\right\} = \{a, b, c\} \nvdash \{a, b, c\}$$

However, one can show that resolution is refutation complete:

$$\Delta \text{ is unsatisfiable } \iff \Delta \vdash \Box.$$  

Entailment: Reduce to unsatisfiability testing and decide by resolution.
Resolution: Completeness?

Do we have
\[ \Delta \models \varphi \text{ implies } \Delta \vdash \varphi? \]

Of course, could only hold for CNF. However:
\[
\{\{a, b\}, \{-b, c\}\} \models \{a, b, c\} \\
\not\vdash \{a, b, c\}
\]

However, one can show that resolution is refutation complete:
\[ \Delta \text{ is unsatisfiable iff } \Delta \vdash \square. \]

Entailment: Reduce to unsatisfiability testing and decide by resolution.
Resolution: Completeness?

Do we have

\[ \Delta \models \varphi \text{ implies } \Delta \vdash \varphi? \]

Of course, could only hold for CNF. However:

\[ \left\{ \{a, b\}, \{-b, c\} \right\} \models \{a, b, c\} \quad \not\vdash \{a, b, c\} \]

However, one can show that resolution is \textit{refutation complete}:

\[ \Delta \text{ is unsatisfiable iff } \Delta \vdash \square. \]

Entailment: Reduce to unsatisfiability testing and decide by resolution.
Resolution: Completeness?

Do we have

\[ \Delta \models \varphi \implies \Delta \vdash \varphi? \]

Of course, could only hold for CNF. However:

\[ \left\{ \{a, b\}, \{\neg b, c\} \right\} \models \{a, b, c\} \]

\[ \not\vdash \{a, b, c\} \]

However, one can show that resolution is **refutation complete**:

\[ \Delta \text{ is unsatisfiable } \iff \Delta \vdash \Box. \]

Entailment: Reduce to unsatisfiability testing and decide by resolution.
Resolution: Completeness?

Do we have

$$\Delta \models \varphi \text{ implies } \Delta \vdash \varphi?$$

Of course, could only hold for CNF. However:

$$\left\{ \{a, b\}, \{-b, c\} \right\} \models \{a, b, c\}
\not\vdash \{a, b, c\}$$

However, one can show that resolution is **refutation complete**:

$$\Delta \text{ is unsatisfiable } \iff \Delta \vdash \square.$$  

Entailment: Reduce to unsatisfiability testing and decide by resolution.
Resolution Strategies

- Trying out all different resolutions can be very costly, and might not be necessary.
- There are different resolution strategies.
- Examples:
  - Input resolution \( R_I(\cdot) \): In each resolution step, one of the parent clauses must be a clause of the input set.
  - Unit resolution \( R_U(\cdot) \): In each resolution step, one of the parent clauses must be a unit clause.
  - Not all strategies are (refutation) completeness preserving. Neither input nor unit resolution is. However, there are others.
Resolution Strategies

- Trying out all different resolutions can be very costly, and might not be necessary.
- There are different resolution strategies.
- Examples:
  - Input resolution ($R_I(\cdot)$): In each resolution step, one of the parent clauses must be a clause of the input set.
  - Unit resolution ($R_U(\cdot)$): In each resolution step, one of the parent clauses must be a unit clause.
  - Not all strategies are (refutation) completeness preserving. Neither input nor unit resolution is. However, there are others.
Resolution Strategies

- Trying out all different resolutions can be very costly, and might not be necessary.
- There are different resolution strategies.
- Examples:
  - **Input resolution** ($R_I(\cdot)$): In each resolution step, one of the parent clauses must be a clause of the input set.
  - **Unit resolution** ($R_U(\cdot)$): In each resolution step, one of the parent clauses must be a unit clause.
  - Not all strategies are (refutation) completeness preserving. Neither input nor unit resolution is. However, there are others.
Resolution Strategies

- Trying out all different resolutions can be very costly, and might not be necessary.
- There are different resolution strategies.
- Examples:
  - **Input resolution** ($R_I(\cdot)$): In each resolution step, one of the parent clauses must be a clause of the input set.
  - **Unit resolution** ($R_U(\cdot)$): In each resolution step, one of the parent clauses must be a unit clause.
  - Not all strategies are (refutation) completeness preserving. Neither input nor unit resolution is. However, there are others.
Horn Clauses & Resolution

**Horn clauses**: Clauses with at most one positive literal

Example: \((a \lor \neg b \lor \neg c), (\neg b \lor \neg c)\)

**Proposition**

*Unit resolution is refutation complete for Horn clauses.*

**Proof idea.**

Consider \(R^*_U(\Delta)\) of Horn clause set \(\Delta\). We have to show that if \(\Box \notin R^*_U(\Delta)\), then \(\Delta (\equiv R^*_U(\Delta))\) is satisfiable.

- Assign *true* to all unit clauses in \(R^*_U(\Delta)\).
- Those clauses that do not contain a literal \(l\) such that \(\{l\}\) is one of the unit clauses have at least one negative literal.
- Assign true to these literals.
- Results in satisfying truth-assignment for \(R^*_U(\Delta)\) (and \(\Delta \subseteq R^*_U(\Delta)\)).
Horn Clauses & Resolution

Horn clauses: Clauses with at most one positive literal
Example: \((a \lor \neg b \lor \neg c), (\neg b \lor \neg c)\)

Proposition

Unit resolution is refutation complete for Horn clauses.

Proof idea.

Consider \(R^*_U(\Delta)\) of Horn clause set \(\Delta\). We have to show that if \(\square \notin R^*_U(\Delta)\), then \(\Delta(\equiv R^*_U(\Delta))\) is satisfiable.

- Assign \texttt{true} to all unit clauses in \(R^*_U(\Delta)\).
- Those clauses that do not contain a literal \(l\) such that \(\{l\}\) is one of the unit clauses have at least one negative literal.
- Assign \texttt{true} to these literals.
- Results in satisfying truth-assignment for \(R^*_U(\Delta)\) (and \(\Delta \subseteq R^*_U(\Delta)\)).
Horn Clauses & Resolution

Horn clauses: Clauses with at most one positive literal
Example: \((a \lor \neg b \lor \neg c), (\neg b \lor \neg c)\)

Proposition

Unit resolution is refutation complete for Horn clauses.

Proof idea.

Consider \(R_U^*(\Delta)\) of Horn clause set \(\Delta\). We have to show that if \(\square \not\in R_U^*(\Delta)\), then \(\Delta(\equiv R_U^*(\Delta))\) is satisfiable.

- Assign true to all unit clauses in \(R_U^*(\Delta)\).
- Those clauses that do not contain a literal \(l\) such that \(\{l\}\) is one of the unit clauses have at least one negative literal.
- Assign true to these literals.
- Results in satisfying truth-assignment for \(R_U^*(\Delta)\) (and \(\Delta \subseteq R_U^*(\Delta)\)).
Horn Clauses & Resolution

**Horn clauses**: Clauses with at most one positive literal

**Example**: \((a \lor \neg b \lor \neg c), (\neg b \lor \neg c)\)

**Proposition**

*Unit resolution is refutation complete for Horn clauses.*

**Proof idea.**

Consider \(R_U^*(\Delta)\) of Horn clause set \(\Delta\). We have to show that if \(\square \notin R_U^*(\Delta)\), then \(\Delta(\equiv R_U^*(\Delta))\) is satisfiable.

- Assign *true* to all unit clauses in \(R_U^*(\Delta)\).
- Those clauses that do not contain a literal \(l\) such that \(\{l\}\) is one of the unit clauses have at least one negative literal.
  - Assign true to these literals.
  - Results in satisfying truth-assignment for \(R_U^*(\Delta)\) (and \(\Delta \subseteq R_U^*(\Delta)\)).
Horn Clauses & Resolution

**Horn clauses**: Clauses with at most one positive literal

Example: \((a \lor \neg b \lor \neg c), (\neg b \lor \neg c)\)

---

**Proposition**

*Unit resolution is refutation complete for Horn clauses.*

---

**Proof idea.**

Consider \(R^*_U(\Delta)\) of Horn clause set \(\Delta\). We have to show that if \(\square \notin R^*_U(\Delta)\), then \(\Delta(= R^*_U(\Delta))\) is satisfiable.

- Assign *true* to all unit clauses in \(R^*_U(\Delta)\).
- Those clauses that do not contain a literal \(l\) such that \(\{l\}\) is one of the unit clauses have at least one negative literal.
- Assign true to these literals.
- Results in satisfying truth-assignment for \(R^*_U(\Delta)\) (and \(\Delta \subseteq R^*_U(\Delta)\)).
Horn Clauses & Resolution

Horn clauses: Clauses with at most one positive literal
Example: \((a \lor \neg b \lor \neg c), (\neg b \lor \neg c)\)

Proposition

Unit resolution is refutation complete for Horn clauses.

Proof idea.

Consider \(R_U^*(\Delta)\) of Horn clause set \(\Delta\). We have to show that if \(\Box \notin R_U^*(\Delta)\), then \(\Delta(\equiv R_U^*(\Delta))\) is satisfiable.

- Assign *true* to all unit clauses in \(R_U^*(\Delta)\).
- Those clauses that do not contain a literal \(l\) such that \(\{l\}\) is one of the unit clauses have at least one negative literal.
- Assign true to these literals.
- Results in satisfying truth-assignment for \(R_U^*(\Delta)\) (and \(\Delta \subseteq R_U^*(\Delta)\)).