Why logic?

- Logic is one of the best developed systems 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-modal logics
  - ...
- Many-valued logics
- Nonmonotonic logics
- Intuitionistic 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**
  - Semantics often entails concept of **logical implication/entailment**
- 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
### Syntax

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

Propositional formulae are built according to the following rule:

<table>
<thead>
<tr>
<th>$\varphi$ $\longrightarrow$</th>
<th>a</th>
<th>atomic formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>⊥</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>$(\varphi' \rightarrow \varphi'')$</td>
<td>implication</td>
<td></td>
</tr>
<tr>
<td>$(\varphi' \leftrightarrow \varphi'')$</td>
<td>equivalence</td>
<td></td>
</tr>
</tbody>
</table>

Parentheses can be omitted if no ambiguity arises.

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

### 4 Semantics

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 if $c$ is true and, additionally, $a$ or $b$ is true.

Logical implication can then be defined as follows:

- $\varphi$ is implied by a set of formulae $\Theta$ if $\varphi$ 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 $I$ (symb. $I \models \psi$):
\[
\begin{align*}
I \models a & \iff I(a) = T \\
I \models \top & \\
I \models \bot & \\
I \models \neg \phi & \iff I \not\models \phi \\
I \models \phi \land \phi' & \iff I \models \phi \text{ and } I \models \phi' \\
I \models \phi \lor \phi' & \iff I \models \phi \text{ or } I \models \phi' \\
I \models \phi \rightarrow \phi' & \iff \text{if } I \models \phi \text{ then } I \models \phi' \\
I \models \phi \leftrightarrow \phi' & \iff I \models \phi \text{ if and only if } I \models \phi'.
\end{align*}
\]

An interpretation $I$ is a model of $\phi$ iff
\[
I \models \phi.
\]
A formula $\phi$ is
- satisfiable if there is an $I$ such that $I \models \phi$;  
- unsatisfiable, otherwise; and  
- valid if $I \models \phi$ for each $I$ (or tautology);  
- falsifiable, otherwise.

Two formulae $\phi$ and $\psi$ are logically equivalent (symb. $\phi \equiv \psi$) if for all interpretations $I$,
\[
I \models \phi \iff I \models \psi.
\]
Why Logic?

Propositional Logic
Syntax
Semantics
Terminology
Decision Problems and Resolution

Examples

Satisfiable, unsatisfiable, falsifiable, valid?

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

\[\rightsquigarrow\]

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

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

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

\[\rightsquigarrow\]

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

\[\rightsquigarrow\]

valid: Consider all interpretations or argue about falsifying ones.

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

\[\rightsquigarrow\]

Of course, equivalent (de Morgan).

Some obvious consequences

Proposition

\(\phi\) is valid iff \(\neg\phi\) is unsatisfiable and \(\phi\) is satisfiable iff \(\neg\phi\) is falsifiable.

Proposition

\(\phi \equiv \psi\) iff \(\phi \leftrightarrow \psi\) is valid.

Theorem

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

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>
<th>Why Logic?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Idempotency</td>
<td>(\phi \lor \phi) (\equiv) (\phi)</td>
<td>(\phi \land \phi) (\equiv) (\phi)</td>
<td>Propositio-</td>
</tr>
<tr>
<td>Commutativity</td>
<td>(\phi \lor \psi) (\equiv) (\psi \lor \phi)</td>
<td>(\phi \land \psi) (\equiv) (\psi \land \phi)</td>
<td>nal Logic</td>
</tr>
<tr>
<td>Associativity</td>
<td>(\phi \lor (\psi \land \chi)) (\equiv) ((\phi \lor \psi) \land \chi)</td>
<td>(\phi \land (\psi \lor \chi)) (\equiv) ((\phi \land \psi) \lor \chi)</td>
<td>Syntax</td>
</tr>
<tr>
<td>Absorption</td>
<td>(\phi \lor (\phi \land \psi)) (\equiv) (\phi)</td>
<td>(\phi \land (\phi \lor \psi)) (\equiv) (\phi)</td>
<td>Semantics</td>
</tr>
<tr>
<td>Distributivity</td>
<td>(\phi \land (\psi \lor \chi)) (\equiv) ((\phi \land \psi) \lor \chi)</td>
<td>(\phi \lor (\phi \land \psi)) (\equiv) (\phi)</td>
<td>Terminology</td>
</tr>
<tr>
<td>Double negation</td>
<td>(\neg\neg\phi) (\equiv) (\phi)</td>
<td>(\neg\neg\phi) (\equiv) (\phi)</td>
<td>Decision</td>
</tr>
<tr>
<td>Truth</td>
<td>(\phi \lor \bot) (\equiv) (T)</td>
<td>(\phi \land \bot) (\equiv) (\bot)</td>
<td>Problems</td>
</tr>
<tr>
<td>Falsity</td>
<td>(\phi \lor \bot) (\equiv) (\bot)</td>
<td>(\phi \land \bot) (\equiv) (\bot)</td>
<td>and</td>
</tr>
<tr>
<td>Taut./contrad.</td>
<td>(\phi \lor \neg\phi) (\equiv) (T)</td>
<td>(\phi \land \neg\phi) (\equiv) (\bot)</td>
<td>Resolution</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 (not 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 not logically 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:
  \[ I \models \Theta \text{ iff } I \models \varphi \text{ for all } \varphi \in \Theta. \]

- $\varphi$ is logically implied by $\Theta$ (symbolically $\Theta \models \varphi$) iff $\varphi$ is true in all models of $\Theta$:
  \[ \Theta \models \varphi \text{ iff } I \models \varphi \text{ for all } I \text{ such that } I \models \Theta. \]

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 $\Theta \models \neg \varphi$


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 $\rightarrow$ 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: $(a \land b) \lor (\neg a \land c)$


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.
Base case: Claim is true for a, $\top, \bot$.
Inductive case: Assume claim is true for all formulae $\varphi$ (up to a certain number of connectives) and call its NNF $\text{nnf}(\varphi)$.
- $\text{nnf}(\varphi \land \psi) = (\text{nnf}(\varphi) \land \text{nnf}(\psi))$
- $\text{nnf}(\varphi \lor \psi) = (\text{nnf}(\varphi) \lor \text{nnf}(\psi))$
- $\text{nnf}(\neg(\varphi \land \psi)) = (\text{nnf}(\neg \varphi) \lor \text{nnf}(\neg \psi))$
- $\text{nnf}(\neg(\varphi \lor \psi)) = (\text{nnf}(\neg \varphi) \land \text{nnf}(\neg \psi))$
- $\text{nnf}(\neg \varphi) = \text{nnf}(\neg \varphi)$


Conjunctive normal form

Theorem
For each propositional formula there are logically equivalent formulae in CNF and DNF, respectively.

Beweis.
The claim is 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 $\text{cnf}(\varphi)$ (and its DNF $\text{dnf}(\varphi)$).
- $\text{cnf}(\neg \varphi) = \text{nnf}(\neg \text{dnf}(\varphi))$ and $\text{cnf}(\varphi \land \psi) = \text{cnf}(\varphi) \land \text{cnf}(\psi)$.
- Assume $\text{cnf}(\varphi) = \bigwedge \chi_i$ and $\text{cnf}(\psi) = \bigwedge \rho_j$ with $\chi_i, \rho_j$ being clauses. Then
  $\text{cnf}(\varphi \lor \psi) = \text{cnf}((\bigwedge \chi_i) \lor (\bigwedge \rho_j)) = \bigwedge \bigwedge (\chi_i \lor \rho_j)$ (by distributivity)
6 Decision Problems and Resolution

- Completeness
- Resolution Strategies
- Horn Clauses

October 24, 2012 Nebel, Wölfl, Hué – KRR 31 / 40

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) $\leadsto$ Davis-Putnam-Logemann-Loveland.

October 24, 2012 Nebel, Wölfl, Hué – KRR 32 / 40

Deciding entailment

- We want to decide $\Theta \models \varphi$.
- Use deduction theorem and reduce to validity:
  $\Theta \models \varphi$ iff $\bigwedge \Theta \rightarrow \varphi$ is valid.
- Now negate and test for unsatisfiability using DPLL.
- 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.

October 24, 2012 Nebel, Wölfl, Hué – KRR 33 / 40

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 a 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!

October 24, 2012 Nebel, Wölfl, Hué – KRR 34 / 40
Resolution: the inference rule

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

The resolution rule

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

\( C_1 \cup C_2 \) 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:

\[
\mathcal{R}(\Delta) = \{ C \mid C \text{ is resolvent of two clauses in } \Delta \}
\]

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

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

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

Theorem (Soundness of resolution)

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

Proof idea.

Show \( \Delta \models D \) if \( D \in \mathcal{R}(\Delta) \) 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 \( \exists m \in C_2 \text{ s.t. } I \models m \). This implies \( I \models D \).

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

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

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.

Entailment: Reduce to unsatisfiability testing and decide by 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 \(\Box \not\in R^*_U(\Delta)\), 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 \(\Delta \subseteq R^*_U(\Delta)\)).