Principles of Knowledge Representation and Reasoning

Propositional Logic
1 Why Logic?
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 KR&R.
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
- ...

April 17, 2018 Nebel, Lindner, Engesser – KR&R
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
2 Propositional Logic
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
3 Syntax
Syntax

Countable alphabet $\Sigma$ of propositional variables: $a, b, c, \ldots$

Propositional formulae are built according to the following rule:

$$\varphi ::= a \quad \text{atomic formula}$$

$$\begin{align*}
\perp & \quad \text{falsity} \\
\top & \quad \text{truth} \\
\neg \varphi' & \quad \text{negation} \\
(\varphi' \land \varphi'') & \quad \text{conjunction} \\
(\varphi' \lor \varphi'') & \quad \text{disjunction} \\
(\varphi' \rightarrow \varphi'') & \quad \text{implication} \\
(\varphi' \leftrightarrow \varphi'') & \quad \text{equivalence}
\end{align*}$$

Parentheses can be omitted if no ambiguity arises.

Operator precedence: $\neg > \land > \lor > \rightarrow = \leftrightarrow$. 
Language and meta-language

- $(a \lor b)$ is an expression of the language of propositional logic.
- $\phi ::= a | \ldots | (\phi' \leftrightarrow \phi'')$ is a statement about how expressions in the language of propositional logic can be formed. It is stated using meta-language.
- In order to describe how expressions (in this case formulae) can be formed, we use meta-language.
- When we describe how to interpret formulae, we use meta-language expressions.
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 iff \(c\) 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 \(\Theta\) iff \(\varphi\) is true for all truth assignments (world states) that make all formulae in \(\Theta\) true.
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 $\mathcal{I}$ (symb. $\mathcal{I} \models \psi$):

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

$$\mathcal{I} \models \top$$

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

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

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

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

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

$$\mathcal{I} \models \phi \leftrightarrow \phi' \quad \text{iff} \quad \mathcal{I} \models \phi \text{ if and only if } \mathcal{I} \models \phi'$$
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?

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

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

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

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

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

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

A formula $\varphi$ is

- **satisfiable** if there is an $\mathcal{I}$ such that $\mathcal{I} \models \varphi$;
- **unsatisfiable**, otherwise; and
- **valid** if $\mathcal{I} \models \varphi$ for each $\mathcal{I}$ (or *tautology*);
- **falsifiable**, otherwise.

Formulae $\varphi$ and $\psi$ are **logically equivalent** (symb. $\varphi \equiv \psi$) if for all interpretations $\mathcal{I}$,

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

Satisfiable, unsatisfiable, falsifiable, valid?

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

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

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

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

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

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

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

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

**Proposition**

ϕ is valid iff ¬ϕ is unsatisfiable.
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 equivalences

<table>
<thead>
<tr>
<th>Simplifications</th>
<th>$\varphi \rightarrow \psi$ ≡ $\neg \varphi \lor \psi$</th>
<th>$\varphi \leftrightarrow \psi$ ≡ $$(\varphi \rightarrow \psi) \land (\psi \rightarrow \varphi)$$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Idempotency</td>
<td>$\varphi \lor \varphi$ ≡ $\varphi$</td>
<td>$\varphi \land \varphi$ ≡ $\varphi$</td>
</tr>
<tr>
<td>Commutativity</td>
<td>$\varphi \lor \psi$ ≡ $\psi \lor \varphi$</td>
<td>$\varphi \land \psi$ ≡ $\psi \land \varphi$</td>
</tr>
<tr>
<td>Associativity</td>
<td>$(\varphi \lor \psi) \lor \chi$ ≡ $\varphi \lor (\psi \lor \chi)$</td>
<td>$(\varphi \land \psi) \land \chi$ ≡ $\varphi \land (\psi \land \chi)$</td>
</tr>
<tr>
<td>Absorption</td>
<td>$\varphi \lor (\varphi \land \psi)$ ≡ $\varphi$</td>
<td>$\varphi \land (\varphi \lor \psi)$ ≡ $\varphi$</td>
</tr>
<tr>
<td>Distributivity</td>
<td>$\varphi \land (\psi \lor \chi)$ ≡ $(\varphi \land \psi) \lor (\varphi \land \chi)$</td>
<td>$\varphi \lor (\psi \land \chi)$ ≡ $(\varphi \lor \psi) \land (\varphi \lor \chi)$</td>
</tr>
<tr>
<td>Double Negation</td>
<td>$\neg \neg \varphi$ ≡ $\varphi$</td>
<td>$\neg \neg \perp$ ≡ $\perp$</td>
</tr>
<tr>
<td>Constants</td>
<td>$\neg \top$ ≡ $\bot$</td>
<td>$\neg \bot$ ≡ $\top$</td>
</tr>
<tr>
<td>De Morgan</td>
<td>$\neg (\varphi \lor \psi)$ ≡ $\neg \varphi \land \neg \psi$</td>
<td>$\neg (\varphi \land \psi)$ ≡ $\neg \varphi \lor \neg \psi$</td>
</tr>
<tr>
<td>Truth</td>
<td>$\varphi \lor \top$ ≡ $\top$</td>
<td>$\varphi \land \top$ ≡ $\varphi$</td>
</tr>
<tr>
<td>Falsity</td>
<td>$\varphi \lor \bot$ ≡ $\varphi$</td>
<td>$\varphi \land \bot$ ≡ $\bot$</td>
</tr>
<tr>
<td>Taut./Contrad.</td>
<td>$\varphi \lor \neg \varphi$ ≡ $\top$</td>
<td>$\varphi \land \neg \varphi$ ≡ $\bot$</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 (not equivalent) formulae?
  - A formula can be characterized by its set of models (if two formulae are not logically equivalent, then their sets of models differ).
  - For $\Sigma$ with $n = |\Sigma|$, there are $2^n$ different interpretations.
  - 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:

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

- $\varphi$ 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 $\Theta \models \varphi \rightarrow \psi$

- **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 $a, \neg a, \top, \bot$.
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 exist logically equivalent formulae in CNF and DNF, respectively.

Proof.

The claim is true for $a$, $\neg a$, $\top$, $\bot$.

Let us assume it is true for all formulae $\phi$ (up to a certain number of connectives) and call its CNF $\text{cnf}(\phi)$ (and its DNF $\text{dnf}(\phi)$).

- $\text{cnf}(\neg \phi) = \text{nff}(\neg \text{dnf}(\phi))$ and $\text{cnf}(\phi \land \psi) = \text{cnf}(\phi) \land \text{cnf}(\psi)$.

- Assume $\text{cnf}(\phi) = \land_i \chi_i$ and $\text{cnf}(\psi) = \land_j \rho_j$ with $\chi_i, \rho_j$ being clauses.
  
  Then $\text{cnf}(\phi \lor \psi) = \text{cnf}((\land_i \chi_i) \lor (\land_j \rho_j)) = \land_i \land_j (\chi_i \lor \rho_j)$ (by distributivity)

Similar for $\text{dnf}(\phi)$.
6 Decision Problems and Resolution

- Completeness
- Resolution Strategies
- Horn Clauses
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) $\rightsquigarrow$ Davis-Putnam-Logemann-Loveland.
Deciding entailment

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

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

- 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.
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 \(\square\)) and empty set of clauses (symbolically \(\emptyset\)) 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:

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

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

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

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

**Theorem (Soundness of resolution)**

*Let* *D* *be a clause. If* *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 ∪ \{\l\}* be the parent clauses of *D = C_1 ∪ C_2*.

Assume *I ⊨ Δ*, we have to show *I ⊨ D*.

Case 1: *I ⊨ l* then \(∃ m ∈ C_2 \text{ s.t. } I ⊨ m\). This implies *I ⊨ D*.

Case 2: *I ⊨ \l* similarly, \(∃ m ∈ C_1 \text{ s.t. } I ⊨ m\).

This means that each model *I* of *Δ* also satisfies *D*, i.e., *Δ ⊨ D*. \(\square\)
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**:

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

**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.
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 \(\Delta \subseteq R^*_U(\Delta)\)).
Literature

Harry R. Lewis and Christos H. Papadimitriou.  
*Elements of the Theory of Computation.*  

Volker Sperschneider and Grigorios Antoniou.  
*Logic – A Foundation for Computer Science.*  
Addison-Wesley, Reading, MA, 1991 (Chapters 1–3).

*Einführung in die mathematische Logik.*  

U. Schöning.  
*Logik für Informatiker.*  