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

Motivation: Deductive Agent

```
1: function action in (∆ ∈ D) out (α ∈ Ac)
2: for all α ∈ Ac do
3: if (∆ ⊨ p) Do(α) then
4:   return α
5: end if
6: end for
7: for all α ∈ Ac do
8: if (∆ ⊭ p) ¬Do(α) then
9:   return α
10: end if
11: end for
12: return null
```

- ∆: Set of formulae written in some logic.
- ⊨: Relation that holds between ∆s and formulae that can be derived from ∆.
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

2 Syntax

- **Countable alphabet** \(\Sigma\) of propositional variables: \(a, b, c, \ldots\)
- **Propositional formulae** are built according to the following rule:

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

Parentheses can be omitted if no ambiguity arises.
**Operator precedence**: \(! > \land > \lor > \rightarrow = \leftrightarrow\).

Language and meta-language

- \((a \lor b)\) is an expression of the language of propositional logic.
- \(\varphi ::= a | \ldots | (\varphi' \leftrightarrow \varphi'')\) 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.
3 Semantics

Propositional Logic
Syntax
Semantics
Terminology

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:

\[ I : \Sigma \rightarrow \{ T, F \} \]

A formula \(\psi\) is true under \(I\) or is satisfied by \(I\) (symb. \(I \models \psi\)):

- \(I \models a\) 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 \(I \models \varphi'\)
- \(I \models \varphi \implies \varphi'\) iff if \(I \models \varphi\) then \(I \models \varphi'\)
- \(I \models \varphi \iff \varphi'\) iff \(I \models \varphi\) if and only if \(I \models \varphi'\)

Example

Given

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

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

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

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

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

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

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

**Propositional Logic**

- **Syntax**
- **Semantics**
- **Terminology**

**Terminology**

An interpretation $I$ is a model of $\varphi$ iff $I \models \varphi$.

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

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

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

**Examples**

Satisfiable, unsatisfiable, falsifiable, valid?

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

- **satisfiable**: $a \mapsto T, b \mapsto F, d \mapsto F, \ldots$
- **falsifiable**: $a \mapsto F, b \mapsto F, c \mapsto T, \ldots$

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

- **satisfiable**: $a \mapsto T, b \mapsto T$
- **valid**: Consider all interpretations or argue about falsifying ones.

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

- **Of course, equivalent (de Morgan).**

**Some obvious consequences**

**Proposition**

$\varphi$ is valid iff $\neg \varphi$ is unsatisfiable.

$\varphi$ is satisfiable iff $\neg \varphi$ is falsifiable.

**Proposition**

$\varphi \equiv \psi$ iff $\varphi \leftrightarrow \psi$ is valid.

**Theorem**

If $\varphi \equiv \psi$, and $\chi'$ results from substituting $\varphi$ by $\psi$ in $\chi$, then $\chi' \equiv \chi$.
Some equivalences

- **Simplifications**
  \[ \varphi \rightarrow \psi \equiv \neg \varphi \lor \psi \quad \varphi \leftrightarrow \psi \equiv \left( \varphi \rightarrow \psi \right) \land \left( \psi \rightarrow \varphi \right) \]

- **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**
  \[ \left( \varphi \lor \psi \right) \lor \chi \equiv \varphi \lor \left( \psi \lor \chi \right) \quad \left( \varphi \land \psi \right) \land \chi \equiv \varphi \land \left( \psi \land \chi \right) \]

- **Absorption**
  \[ \varphi \lor \left( \varphi \land \psi \right) \equiv \varphi \quad \varphi \land \left( \varphi \lor \psi \right) \equiv \varphi \]

- **Distributivity**
  \[ \varphi \lor \left( \psi \land \chi \right) \equiv \left( \varphi \lor \psi \right) \land \left( \varphi \lor \chi \right) \quad \varphi \land \left( \psi \lor \chi \right) \equiv \left( \varphi \land \psi \right) \lor \left( \varphi \land \chi \right) \]

- **De Morgan**
  \[ \neg \left( \varphi \lor \psi \right) \equiv \neg \varphi \land \neg \psi \quad \neg \left( \varphi \land \psi \right) \equiv \neg \varphi \lor \neg \psi \]

- **Double Negation**
  \[ \neg \neg \varphi \equiv \varphi \]

- **Constants**
  \[ \neg \top \equiv \bot \quad \neg \bot \equiv \top \]

- **Truth**
  \[ \varphi \lor \top \equiv \top \]

- **Falsity**
  \[ \varphi \lor \bot \equiv \varphi \quad \varphi \land \bot \equiv \bot \]

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**:
  \[ \Theta \cup \{ \varphi \} \text{ is unsatisfiable} \iff \Theta \models \neg \varphi \]

...for a given finite alphabet \( \Sigma \)?

- **Infinitely many**:
  \[ a, a \lor a, a \land a, a \lor \top, \ldots \]

- **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.

### 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 DPLL.**

- **Different approach**: Try to derive \( \varphi \) from \( \Theta \) – find a proof of \( \varphi \) from \( \Theta \).

- **Use inference rules** to derive new formulæ from \( \Theta \).

  Continue to deduce new formulæ until \( \varphi \) can be deduced.

- **One particular calculus**: **tableaux**.
Propositional Tableaux

- **Goal**: Prove the unsatisfiability of a formula.
- Tableaux algorithm for propositional logic is sound and complete.
- **General principle**: Break each formula into its components up to the simplest one, where contradiction is easy to spot.

**April 17, 2018 Nebel, Lindner, Engesser – MAS 25 / 28**

Propositional Tableaux

- **NotNot**: If \( \neg \neg \varphi \) is in a branch, then add \( \varphi \) to it.
- **NotAnd**: If \( \neg (\varphi \land \psi) \) is in a branch, then add \( \neg \varphi \) to it, add a new branch, and add \( \neg \psi \) to it.
- **NotOr**: If \( \neg (\varphi \lor \psi) \) is in a branch, then add \( \neg \varphi \) and \( \neg \psi \) to it.
- **NotImplication**: If \( \neg (\varphi \rightarrow \psi) \) is in a branch, then add \( \varphi \) and \( \neg \psi \) to that branch.

**April 17, 2018 Nebel, Lindner, Engesser – MAS 26 / 28**

Propositional Tableaux

- A tableaux is a tree. Each branch of that tree corresponds to one attempt to find a **model** for the input formula.
- Initial Tableaux consists of the node: \( \land \Theta \land \neg \varphi \)
  - \( \Theta \models \varphi \) iff \( \land \Theta \rightarrow \varphi \) is valid iff \( \neg (\land \Theta \rightarrow \varphi) \) is unsatisfiable iff \( \land \Theta \land \neg \varphi \) is unsatisfiable
- The tableaux can be incrementally extended by applying rules:
  - **And-Rule**: If \( \varphi \land \psi \) is in a branch, then add \( \varphi \) and \( \psi \) to it.
  - **Or-Rule**: If \( \varphi \lor \psi \) is in a branch, then add \( \varphi \) to it, add a new branch, and add \( \psi \) to it.
  - **Implication**: If \( \varphi \rightarrow \psi \) is in a branch, then add \( \neg \varphi \) to it, add a new branch, and add \( \psi \) to it.

**April 17, 2018 Nebel, Lindner, Engesser – MAS 27 / 28**

Propositional Tableaux: Closed Tableaux

- A **branch is saturated** if no more rule can be applied.
- **A branch is closed** if it contains formulae \( \varphi \) and \( \neg \varphi \).
- **A tableaux is closed** if all branches are closed.
- If the tableaux is closed, this means no model for the input formula could be found, hence, its negation is valid.

**April 17, 2018 Nebel, Lindner, Engesser – MAS 28 / 28**