# Multi-Agent Systems

## Propositional Logic

Albert-Ludwigs-Universität Freiburg

Bernhard Nebel, Rolf Bergdoll, and Thorsten Engesser

Winter Term 2019/20

# 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



```
function action in (Δ ∈ D) out (α ∈ Ac)
for all α ∈ Ac do
  if Δ ⊢_ρ Do(α) then
    return α
  end if
end for
for all α ∈ Ac do
  if Δ ⊬_ρ ¬Do(α) then
    return α
  end if
end for
return null
```

- $\Delta$: Set of formulae written in some logic.
- $\vdash$: Relation that holds between $\Delta$s and formulae that can be derived from $\Delta$.



# 1 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 ($\wedge$), or ($\vee$), and not ($\neg$)
- Formulae: built out of atoms and connectives
- Universe of discourse: truth values





## 2 Syntax





## Syntax

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

$$
\begin{array}{rcll}
\varphi & ::= & a & \text{atomic formula} \\
 & | & \bot & \text{falsity} \\
 & | & \top & \text{truth} \\
 & | & \neg\varphi' & \text{negation} \\
 & | & (\varphi' \wedge \varphi'') & \text{conjunction} \\
 & | & (\varphi' \vee \varphi'') & \text{disjunction} \\
 & | & (\varphi' \rightarrow \varphi'') & \text{implication} \\
 & | & (\varphi' \leftrightarrow \varphi'') & \text{equivalence}
\end{array}
$$

Parentheses can be omitted if no ambiguity arises.
Operator precedence: $\neg > \wedge > \vee > \rightarrow = \leftrightarrow$.





## Language and meta-language

- $(a \vee b)$ 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





# 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:

$$\mathscr{I} : \Sigma \to \{T, F\}.$$

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

$$\begin{array}{rcl}
\mathscr{I} \models a & \text{iff} & \mathscr{I}(a) = T \\
 & \mathscr{I} \models \top & \\
 & \mathscr{I} \not\models \bot & \\
\mathscr{I} \models \neg\varphi & \text{iff} & \mathscr{I} \not\models \varphi \\
\mathscr{I} \models \varphi \land \varphi' & \text{iff} & \mathscr{I} \models \varphi \text{ and } \mathscr{I} \models \varphi' \\
\mathscr{I} \models \varphi \lor \varphi' & \text{iff} & \mathscr{I} \models \varphi \text{ or } \mathscr{I} \models \varphi' \\
\mathscr{I} \models \varphi \to \varphi' & \text{iff} & \text{if } \mathscr{I} \models \varphi \text{ then } \mathscr{I} \models \varphi' \\
\mathscr{I} \models \varphi \leftrightarrow \varphi' & \text{iff} & \mathscr{I} \models \varphi \text{ if and only if } \mathscr{I} \models \varphi'
\end{array}$$



# Example



Given

$$\mathscr{I} : a \mapsto T,\ b \mapsto F,\ c \mapsto F,\ d \mapsto T,$$

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

$$((\mathbf{a} \lor \mathbf{b}) \leftrightarrow (\mathbf{c} \lor \mathbf{d})) \land (\neg(\mathbf{a} \land \mathbf{c}) \lor (\mathbf{c} \land \neg\mathbf{d}))$$

$$((\mathbf{a} \lor \mathbf{b}) \leftrightarrow (\mathbf{c} \lor \mathbf{d})) \land (\neg(\mathbf{a} \land \mathbf{c}) \lor (\mathbf{c} \land \neg\mathbf{d}))$$

$$((\mathbf{a} \lor \mathbf{b}) \leftrightarrow (\mathbf{c} \lor \mathbf{d})) \land (\neg(\mathbf{a} \land \mathbf{c}) \lor (\mathbf{c} \land \neg\mathbf{d}))$$

$$((\mathbf{a} \lor \mathbf{b}) \leftrightarrow (\mathbf{c} \lor \mathbf{d})) \land (\neg(\mathbf{a} \land \mathbf{c}) \lor (\mathbf{c} \land \neg\mathbf{d}))$$

$$((\mathbf{a} \lor \mathbf{b}) \leftrightarrow (\mathbf{c} \lor \mathbf{d})) \land (\neg(\mathbf{a} \land \mathbf{c}) \lor (\mathbf{c} \land \neg\mathbf{d}))$$

# 4 Terminology





# Terminology



An interpretation $\mathscr{I}$ is a model of $\varphi$ iff $\mathscr{I} \models \varphi$.
A formula $\varphi$ is

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

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

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



# Examples



Satisfiable, unsatisfiable, falsifiable, valid?

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

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

$((\neg a \to \neg b) \to (b \to a))$

- $\rightsquigarrow$ 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

*$\varphi$ 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 | $\varphi \to \psi$ | $\equiv$ | $\neg\varphi \lor \psi$ | $\varphi \leftrightarrow \psi$ | $\equiv$ | $(\varphi \to \psi) \land (\psi \to \varphi)$ |
| 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 | $\varphi \lor \bot$ | $\equiv$ | $\varphi$ | $\varphi \land \bot$ | $\equiv$ | $\bot$ |
| taut./contrad. | $\varphi \lor \neg\varphi$ | $\equiv$ | $\top$ | $\varphi \land \neg\varphi$ | $\equiv$ | $\bot$ |

UNI FREIBURG





## 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.

UNI FREIBURG





## Logical implication

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

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

- Deduction theorem: $\Theta \cup \{\varphi\} \models \psi$ iff $\Theta \models \varphi \to \psi$

UNI FREIBURG





## Deciding entailment

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

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

- Now negate and test for unsatisfiability using tableaux.
- 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.

UNI FREIBURG

## Propositional Tableaux

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





## 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: $\bigwedge \Theta \wedge \neg\varphi$
  - $\Theta \models \varphi$ iff $\bigwedge \Theta \rightarrow \varphi$ is valid iff $\neg(\bigwedge \Theta \rightarrow \varphi)$ is unsatisfiable iff $\bigwedge \Theta \wedge \neg\varphi$ is unsatisfiable
- The tableaux can be incrementally extended by applying rules:
  - And-Rule: If $\varphi \wedge \psi$ is in a branch, then add $\varphi$ and $\psi$ to it.
  - Or-Rule: If $\varphi \vee \psi$ 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.





## Propositional Tableaux

- NotNot: If $\neg\neg\varphi$ is in a branch, then add $\varphi$ to it.
- NotAnd: If $\neg(\varphi \wedge \psi)$ is in a branch, then add $\neg\varphi$ to it, add a new branch, and add $\neg\psi$ to it.
- NotOr: If $\neg(\varphi \vee \psi)$ 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.





## 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.