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

Propositional Logic Syntax Semantics Terminology



Motivation: Deductive Agent



- 1: function action in $(\Delta \in D)$ out $(\alpha \in Ac)$ 2: for all $\alpha \in Ac$ do
- 3: **if** $\Delta \vdash_{\rho} Do(\alpha)$ **then**
- 4: return α
- 5: end if
- 6: end for
- 7: for all $\alpha \in Ac$ do
- 8: **if** $\Delta \not\vdash_{\rho} \neg Do(\alpha)$ **then**
- 9: return α
- 10: end if
- 11: end for
- 12: return null
 - Δ : Set of formulae written in some logic.
 - \vdash : Relation that holds between Δ s and formulae that can be derived from Δ .

Semantics

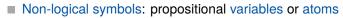
1 Propositional Logic



Propositional Logic

Syntax

Semantics



- 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 (∧), or (∨), and not (¬)
- Formulae: built out of atoms and connectives
- Universe of discourse: truth values

Propositional Logic Syntax Semantics





Propositional Logic

Syntax

Semantics

Syntax



Countable alphabet Σ of propositional variables: a, b, c, ...Propositional formulae are built according to the following rule:

φ	::=	а	atomic formula
		\perp	falsity
		Т	truth
		eg arphi'	negation
		$(arphi \wedge arphi'')$	conjunction
		$(arphi^\prime ee arphi^{\prime\prime})$	disjunction
		(arphi' ightarrow arphi'')	implication
		$(arphi'\leftrightarrowarphi'')$	equivalence

Propositional Logic

Syntax

Semantics

Terminology

Parentheses can be omitted if no ambiguity arises.

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

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



Syntax

Semantics

3 Semantics



Propositional Logic

Syntax

Semantics

- 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:

• φ is implied by a set of formulae Θ iff φ is true for all truth assignments (world states) that make all formulae in Θ true.



Syntax

Semantics

An interpretation (or truth assignment) over $\boldsymbol{\Sigma}$ is a function:

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

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

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

22

Syntax

Semantics

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





Propositional Logic Syntax

Semantics

4 Terminology



Propositional Logic

Syntax

Semantics

18 / 28

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

Terminology

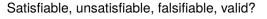
Formulae φ and ψ are logically equivalent (symb. $\varphi \equiv \psi$) if for all interpretations \mathscr{I} ,

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

UNI FREIBURG

> Propositional Logic Syntax Semantics

Examples



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

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

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

- \rightsquigarrow satisfiable: $a \mapsto T, b \mapsto T$
- 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).





Proposition

 φ is valid iff $\neg \varphi$ is unsatisfiable. φ is satisfiable iff $\neg \varphi$ is falsifiable.

Proposition

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

Theorem

If $\varphi \equiv \psi$, and χ' results from substituting φ by ψ in χ , then $\chi' \equiv \chi$.

Propositional Logic Syntax

Semantics

Some equivalences



simplifications	$oldsymbol{arphi} ightarrow oldsymbol{\psi}$	≡	$ eg \phi \lor \psi$	$oldsymbol{arphi} \leftrightarrow oldsymbol{\psi}$	≡	$egin{array}{lll} (arphi ightarrow \psi) \wedge \ (\psi ightarrow arphi) \end{array}$	Proposi- tional Logic
idempotency	$\boldsymbol{\varphi} \lor \boldsymbol{\varphi}$	\equiv	φ	$oldsymbol{arphi}\wedgeoldsymbol{arphi}$	\equiv	φ	Syntax
commutativity	$\pmb{\varphi} \lor \pmb{\psi}$	\equiv	$\psi \lor \varphi$	$oldsymbol{arphi}\wedgeoldsymbol{\psi}$	\equiv	$oldsymbol{\psi} \wedge oldsymbol{arphi}$	Semantics
associativity	$(\varphi \lor \psi) \lor \chi$	\equiv	$\varphi \lor (\psi \lor \chi)$	$(\phi \wedge \psi) \wedge \chi$	\equiv	$arphi \wedge (\psi \wedge \chi)$	Terminology
absorption	$arphi ee (arphi \wedge \psi)$	\equiv	φ	$oldsymbol{arphi} \wedge (oldsymbol{arphi} ee oldsymbol{\psi})$	\equiv	ϕ	
distributivity	$\varphi \wedge (\psi \lor \chi)$	\equiv	$(\phi \wedge \psi) \lor$	$\varphi \lor (\psi \land \chi)$	\equiv	$(arphi ee \psi) \land$	
			$(\phi \wedge \chi)$			$(\varphi \lor \chi)$	
double negation	$ eg \neg \phi$	\equiv	φ				
constants	$\neg \top$	\equiv	\perp	$\neg \bot$	\equiv	Т	
De Morgan	$ eg(\varphi \lor \psi)$	\equiv	$\neg \phi \land \neg \psi$	$ eg(\varphi \wedge \psi)$	\equiv	$ eg \phi \lor eg \psi$	
truth	arphi ee o o	\equiv	Т	$oldsymbol{arphi}\wedge op$	\equiv	ϕ	
falsity	$arphi ee \bot$	\equiv	φ	$arphi \wedge ot$	\equiv	\perp	
taut./contrad.	$arphi ee \neg arphi$	\equiv	Т	$oldsymbol{arphi}\wedge eg oldsymbol{arphi}$	\equiv	\perp	

... for a given finite alphabet Σ ?

- 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 Σ with $n = |\Sigma|$, there are 2^n different interpretations.
 - There are 2^(2ⁿ) different sets of interpretations.
 - There are 2^(2ⁿ) (logical) equivalence classes of formulae.

Propositional Logic Syntax

Semantics





tional Logic Syntax Semantics

Proposi-

Terminology

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

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

■ Deduction theorem: $\Theta \cup \{\phi\} \models \psi$ iff $\Theta \models \phi \rightarrow \psi$

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

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

- Now negate and test for unsatisfiability using tableaux.
- Different approach: Try to derive φ from Θ find a proof of φ from Θ .
- Use inference rules to derive new formulae from Θ.
 Continue to deduce new formulae until φ can be deduced.

Nebel, Engesser, Bergdoll - MAS

Propositional Logic Syntax Semantics

Terminology

UNI FREIBU

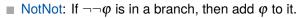


- Propositional Logic Syntax Semantics
 - Terminology

- 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.

- 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: $\land \Theta \land \neg \phi$
- The tableaux can be incrementally extended by applying rules:
 - And-Rule: If $\phi \land \psi$ is in a branch, then add ϕ and ψ to it.
 - Or-Rule: If $\varphi \lor \psi$ is in a branch, then add φ to it, add a new branch, and add ψ to it.
 - Implication: If $\varphi \rightarrow \psi$ is in a branch, then add $\neg \varphi$ to it, add a new branch, and add ψ to it.





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



Semantics



Propositional Logic Syntax

Semantics

- A branch is saturated if no more rule can be applied.
- A branch is closed if it contains formulae φ 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.