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** $(\Delta \in D)$ **out** $(\alpha \in Ac)$
for all $\alpha \in Ac$ **do**
  if $\Delta \vdash_\rho Do(\alpha)$ **then**
    return α
  end if
end for
for all $\alpha \in Ac$ **do**
  if $\Delta \not\vdash_\rho \neg Do(\alpha)$ **then**
    return α
  end if
end for
return *null*

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

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 Σ 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:

$$(a \vee b) \wedge c$$

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.

Formal semantics

An interpretation (or truth assignment) over Σ is a function:

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

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

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

Example

Given

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

Is $((a \vee b) \leftrightarrow (c \vee d)) \wedge (\neg(a \wedge c) \vee (c \wedge \neg d))$ true or false?

$$((\mathbf{a} \vee \mathbf{b}) \leftrightarrow (\mathbf{c} \vee \mathbf{d})) \wedge (\neg(\mathbf{a} \wedge \mathbf{c}) \vee (\mathbf{c} \wedge \neg \mathbf{d}))$$

$$((\mathbf{a} \vee \mathbf{b}) \leftrightarrow (\mathbf{c} \vee \mathbf{d})) \wedge (\neg(\mathbf{a} \wedge \mathbf{c}) \vee (\mathbf{c} \wedge \neg \mathbf{d}))$$

$$((\mathbf{a} \vee \mathbf{b}) \leftrightarrow (\mathbf{c} \vee \mathbf{d})) \wedge (\neg(\mathbf{a} \wedge \mathbf{c}) \vee (\mathbf{c} \wedge \neg \mathbf{d}))$$

$$((\mathbf{a} \vee \mathbf{b}) \leftrightarrow (\mathbf{c} \vee \mathbf{d})) \wedge (\neg(\mathbf{a} \wedge \mathbf{c}) \vee (\mathbf{c} \wedge \neg \mathbf{d}))$$

$$((\mathbf{a} \vee \mathbf{b}) \leftrightarrow (\mathbf{c} \vee \mathbf{d})) \wedge (\neg(\mathbf{a} \wedge \mathbf{c}) \vee (\mathbf{c} \wedge \neg \mathbf{d}))$$

4 Terminology

Terminology

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.

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

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

Examples

Satisfiable, unsatisfiable, falsifiable, valid?

$(a \vee b \vee \neg c) \wedge (\neg a \vee \neg b \vee d) \wedge (\neg a \vee b \vee \neg d)$

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

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

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

Equivalence? $\neg(a \vee b) \equiv \neg a \wedge \neg b$

- $\leadsto$ Of course, equivalent (de Morgan).

Some obvious consequences

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 $\chi' \equiv \chi$.

Some equivalences

simplifications	$\varphi \rightarrow \psi$	$\equiv$	$\neg\varphi \lor \psi$	$\varphi \leftrightarrow \psi$	$\equiv$	$(\varphi \rightarrow \psi) \land (\psi \rightarrow \varphi)$
idempotency	$\varphi \lor \varphi$	$\equiv$	φ	$\varphi \land \varphi$	$\equiv$	φ
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	$\varphi \lor (\varphi \land \psi)$	$\equiv$	φ	$\varphi \land (\varphi \lor \psi)$	$\equiv$	φ
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	$\neg\neg\varphi$	$\equiv$	φ			
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	$\varphi \lor \top$	$\equiv$	$\top$	$\varphi \land \top$	$\equiv$	φ
falsity	$\varphi \lor \bot$	$\equiv$	φ	$\varphi \land \bot$	$\equiv$	$\bot$
taut./contrad.	$\varphi \lor \neg\varphi$	$\equiv$	$\top$	$\varphi \land \neg\varphi$	$\equiv$	$\bot$

How many different formulae are there …

…for a given finite alphabet Σ?

- Infinitely many: $a, a \vee a, a \wedge a, a \vee a \vee a, \ldots$
- 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 $2^{(2^n)}$ different sets of interpretations.
  - There are $2^{(2^n)}$ (logical) equivalence classes of formulae.

Logical implication

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

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

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

Deciding entailment

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

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

- 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.

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 φ and ψ to it.
  - Or-Rule: If $\varphi \vee \psi$ 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.

Propositional Tableaux

- NotNot: If $\neg\neg\varphi$ is in a branch, then add φ 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 φ 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 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.