

Chapter 2

Logics

Course “Specification and Verification of Parallel Systems”

Juliane Päßler, Tobias John

Spring 2024



Chapter 2

Learning Targets of Chapter “Logics”.

The chapter gives some basic information about “standard” logics, namely propositional logics and (classical) first-order logics



Chapter 2

Outline of Chapter “Logics”.

Propositional logic

First-order logic (FO)



Section

Propositional logic

Spring 2024

Syntax (formulas, aka propositions)

$$\begin{array}{rcll} \varphi & ::= & p \mid \top \mid \bot & \text{atomic propositions} \\ & & \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \varphi \to \varphi & \text{compound propositions} \end{array}$$



Semantics: the meaning of propositions

different truth values (we only consider $\mathbb{B} = \{\top, \bot\}$)
Variable assignment:

$$\sigma : \mathrm{AP} \to \mathbb{B} \ .$$

"Semantic brackets":

$$[\![_]\!]- : (\mathrm{AP} \to \mathbb{B}) \to \Phi \to \mathbb{B}.$$

Model relation:

$$\sigma \models \varphi \text{ iff } [\![\varphi]\!]^\sigma = \top$$

How are $\sigma \models \varphi$ and $[\![\varphi]\!]^\sigma$ defined?



Definition of $\sigma \models \varphi$ and $[\![\varphi]\!]^\sigma$

remember: $\sigma \models \varphi$ iff $[\![\varphi]\!]^\sigma = \top$ and $\sigma \not\models \varphi$ iff $[\![\varphi]\!]^\sigma = \bot$

- $\sigma \models p$ iff $\sigma(p) = \top$ for $p \in P$
- $\sigma \models \top$
- $\sigma \not\models \bot$
- $\sigma \models \neg\varphi$ iff $\sigma \not\models \varphi$
- $\sigma \models \varphi \wedge \varphi'$ iff $\sigma \models \varphi$ and $\sigma \models \varphi'$
- $\sigma \models \varphi \vee \varphi'$ iff $\sigma \models \varphi$ or $\sigma \models \varphi'$
- $\sigma \models \varphi \to \varphi'$ iff $\sigma \not\models \varphi$ or $\sigma \models \varphi'$



Interesting problems

model checking

$$\sigma \overset{?}{\models} \varphi$$

(in P)

satisfiability

$$? \models \varphi$$

(NP-complete)

validity

$$\overset{?}{\models} \varphi$$

(co-NP-complete)



Proof theory

- different questions about formula:
  how to *infer*, derive, deduce formulas (from others), i.e., how to do a proof?
- mechanical process
- *proof* = deduction (sequence or tree of steps)
- theorem = formula that is always derivable
  - corresponds to a formula which is valid (semantical)
- soundness and completeness of interest

How could such a process look like?



Deductions and proof systems

A **proof system** for a given logic consists of

- **axioms** (or *axiom schemata*), which are formulae assumed to be true, and
- **inference rules**, of approx. the form

$$\frac{\varphi_1 \quad \dots \quad \varphi_n}{\psi}$$

- $\varphi_1, \dots, \varphi_n$ are **premises** and ψ **conclusion**.



A simple form of derivation

Derivation of φ

Sequence of formulas, where each formula is

- an axiom or
- can be obtained by applying an inference rule to formulae earlier in the sequence.

- Let Γ be a set of axioms (formulas):

$$\Gamma \vdash \varphi$$

- special case (theorems): $\vdash \varphi$
- proof = derivation



A proof system for propositional logic

$$\frac{}{\varphi \to (\psi \to \varphi)}\ \mathrm{Ax}_1$$

$$\frac{}{(\varphi \to (\psi \to \chi)) \to ((\varphi \to \psi) \to (\varphi \to \chi))}\ \mathrm{Ax}_2$$

$$\frac{}{((\varphi \to \bot) \to \bot) \to \varphi}\ \mathrm{DN}$$

$$\frac{\varphi \qquad \varphi \to \psi}{\psi}\ \mathrm{MP}$$

Task: prove $p \to p$



A small derivation

Example

$p \to p$ is a theorem of the proof system:

$(p \to ((p \to p) \to p)) \to ((p \to (p \to p)) \to (p \to p))$	Ax_2 $(\varphi = p, \psi = (p \to p), \chi = p)$	(1)
$p \to ((p \to p) \to p)$	Ax_1 $(\varphi = p, \psi = (p \to p))$	(2)
$(p \to (p \to p)) \to (p \to p)$	MP on (1) and (2)	(3)
$p \to (p \to p)$	Ax_1 $(\varphi = p, \psi = p)$	(4)
$p \to p$	MP on (3) and (4)	(5)



Proof systems and proofs: remarks

- "definitions" from the previous slides: not very formal
- many different approaches of how to draw conclusions exist, the one sketched on the previous slide
  - works with "sequences"
  - corresponds to the historically oldest approach to proof systems ("Hilbert-style")
  - **but** usually sound & complete



Summary: proof theory

- decidable, so a “trivial problem” in that sense
- truth tables (brute force)
- deduction systems
- one can try to do better, different derivation strategies (resolution, refutation, . . .)



Example: Therac-25 radiation therapy machine



an example of a linear accelerator

(Urology San Antonio, CC BY-NC-SA 2.0)

Express requirements using propositional logic!
What requirements can you not express with propositional logic?





Section

First-order logic (FO)

Spring 2024

Logics so far: rather sterile

- propositional logic: cannot speak about "things" just regulates "truth" in a logical vacuum, but not "truth **of** something"
- **predicates**

Example ("even" as predicate)

- $even(4)$: true,
- $even(5)$, $even(4+1)$: false,
- $even(x)$: depends on value of *variable* x



Sorts

- **Sort**
  - name of a domain (like `Nat`)
  - restricted form of type
- single-sorted vs. multi-sorted case
- single-sorted
  - one sort only
  - “degenerated”
  - we will only consider this case



Signature

- fixes the "syntactic playground"
- selection of symbols
  - *functional*
  - *relational*

  together with "arity" (or sort-information)

Definition (Signature)

A **Signature** is a 3-tuple $\Sigma = (\mathrm{Fun}, \mathrm{Rel}, \mathrm{ar})$ with

- Fun is a set of *function symbols*
- Rel is a set of *relation symbols*
- $\mathrm{Fun} \cap \mathrm{Rel} = \emptyset$
- $\mathrm{ar} : \mathrm{Fun} \cup \mathrm{Rel} \to \mathbb{N}$ is the *arity function*



Terms

given:

- signature Σ
- set of variables X (with typical elements $x, y, \ldots$)

Definition (Σ-terms)

The *set of terms* w.r.t. a signature Σ and a set of variables X is recursively defined by the following grammar:



Terms

given:

- signature Σ
- set of variables X (with typical elements $x, y, \ldots$)

Definition (Σ-terms)

The *set of terms* w.r.t. a signature Σ and a set of variables X is recursively defined by the following grammar:

$$\begin{array}{lll} t ::= x & & \text{for all } x \in Var \\ \quad \mid f(t_1, \ldots, t_k) & & f \in \mathrm{Fun} \text{ and } \mathrm{ar}(f) = k \end{array}$$

We denote this set by $T_\Sigma(X)$.



Syntax: formulas of first-order logics

- given: first-order signature Σ

Definition (FO formulas)

The *first-order formulas* w.r.t. a signature Σ and a set of variables X is recursively defined by the following grammar, with $P \in \mathrm{Rel}$, $\mathrm{ar}(P) = k$, $t_i \in T_\Sigma(X)$ for all $i \in \mathbb{N}$:

$$\begin{array}{rcll} \varphi & ::= & P(t_1, \ldots, t_k) \mid t_1 \doteq t_2 \mid \top \mid \bot & \text{atomic formulas} \\ & \mid & \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \varphi \to \varphi & \\ & \mid & \forall x.\varphi \mid \exists x.\varphi & \end{array}$$



Free variables

- quantifiers *bind* variables
- variables can *occur* free or not (= *bound*) in a formula



Free variables

- quantifiers *bind* variables
- variables can *occur* free or not (= *bound*) in a formula

How can we define "free variables"?



Free variables

- quantifiers *bind* variables
- variables can *occur* free or not (= *bound*) in a formula

How can we define "free variables"?

Definition (Free variables)

Let Σ be a signature and X a set of variables. The set of *free variables* $\mathrm{FV}(\varphi) \subseteq X$ of a FO formula φ is defined as:

$$\mathrm{FV}(\varphi) = \begin{cases} \text{variables in } \varphi & \varphi \text{ is atomic formula} \\ \mathrm{FV}(\psi) & \varphi = \neg\psi \\ \mathrm{FV}(\psi) \cup \mathrm{FV}(\chi) & \varphi = \psi \text{ with } \{\wedge, \vee, \rightarrow\} \\ \mathrm{FV}(\psi) \setminus \{x\} & \varphi = \forall x.\psi \text{ or } \varphi = \exists x.\psi \end{cases}$$



Exercise: free variables

What are the free variables in the following formulas?

- $\forall x.P(a)$
- $\forall x.Q(x,x) \rightarrow \exists x.Q(x,y)$
- $\forall x.(P(y) \wedge \forall y.P(x))$
- $\exists x.(\neg P(x) \vee P(f(a)))$
- $\exists x.\forall x.Q(x,f(x))$



First-order structures and models

- given: Σ

first-order structure (or model) $M = (A, I)$

- A a nonempty set (domain)
- **interpretation** I (respecting arity)

$$[\![f]\!]^I : A^{\mathrm{ar}(f)} \to A \qquad \text{for } f \in \mathrm{Fun}$$
$$[\![P]\!]^I \subseteq A^{\mathrm{ar}(P)} \qquad \text{for } P \in \mathrm{Rel}$$



Variable assignments

Definition (Variable assignment)

Given a set of variables X and nonempty set A (domain), a *variable assignment* is a function

$$\sigma : X \to A$$

- alternative names: “valuation” or “state”

Extension

For $x \in X$ and $a \in A$ we define $\sigma[x \mapsto a] : X \to A$ as

$$(\sigma[x \mapsto a])(y) = \begin{cases} a & y = x \\ \sigma(y) & \text{else} \end{cases}$$



Variable assignments

Definition (Variable assignment)

Given a set of variables X and nonempty set A (domain), a *variable assignment* is a function

$$\sigma : X \to A$$

- alternative names: "valuation" or "state"

Extension

For $x \in X$ and $a \in A$ we define $\sigma[x \mapsto a] : X \to A$ as

$$(\sigma[x \mapsto a])(y) = \begin{cases} a & y = x \\ \sigma(y) & \text{else} \end{cases}$$

How to define σ for terms?



Assignment for terms

Definition (Assignment for Terms)

Let Σ be a signature, $M = (A, I)$ a corresponding FO-structure and $\sigma : X \to A$ a variable assignment. We define $\sigma' : T_\Sigma(X) \to A$ as

$$\sigma'(t) = \begin{cases} \sigma(t) & t \in X \\ [\![f]\!]^I(\sigma'(t_1), \ldots, \sigma'(t_k)) & t = f(t_1, \ldots, t_k) \text{ with } \\ & f \in \mathrm{Fun}, \mathrm{ar}(f) = k, t_1, \ldots, t_k \in T_\Sigma(\mathit{Var}) \end{cases}$$

We will use σ for σ'.



Example: Variable assignment

- Let's consider the domain of natural numbers with the usual interpretation of the function symbols. Let id be the identity function ($\mathrm{id} : x \mapsto x$ for all $x \in \mathbb{N}$), inc be the increment function and 1 a constant.

How are the terms evaluated?

- $\mathrm{id}[x \mapsto 7][y \mapsto 2](+(x, \mathrm{inc}(y)))$
- $\mathrm{id}[x \mapsto 7][y \mapsto 2][x \mapsto 2](+(x, \mathrm{inc}(y)))$
- $\mathrm{id}[y \mapsto 2](+(x, \mathrm{inc}(1)))$



Satisfaction

Definition ($\models$)

Let Σ be a signature, $M = (A, I)$ a corresponding FO-structure and $\sigma : X \to A$ a variable assignment. We define that M and σ satisfy φ ($M, \sigma \models \varphi$) as:

$M, \sigma \models$	$\top$	$M, \sigma \not\models \bot$
$M, \sigma \models$	$P(t_1, \ldots, t_k)$	$(\sigma(t_1), \ldots, \sigma(t_k)) \in [\![P]\!]^I$, $P \in \mathrm{Rel}, \mathrm{ar}(P) = k$ and $t_1, \ldots, t_k \in T_\Sigma(Var)$
$M, \sigma \models$	$t_1 \doteq t_2$	$\sigma(t_1) = \sigma(t_2)$ and $t_1, t_2 \in T_\Sigma(Var)$
$M, \sigma \models$	$\neg\varphi$	$M, \sigma \not\models \varphi$
$M, \sigma \models$	$\varphi \wedge \psi$	$M, \sigma \models \varphi$ and $M, \sigma \models \psi$
$M, \sigma \models$	$\varphi \vee \psi$	$M, \sigma \models \varphi$ or $M, \sigma \models \psi$
$M, \sigma \models$	$\varphi \to \psi$	$M, \sigma \not\models \varphi$ or $M, \sigma \models \psi$
$M, \sigma \models$	$\exists x.\varphi$	there exists $a \in A$ such that $M, \sigma[x \mapsto a] \models \varphi$
$M, \sigma \models$	$\forall x.\varphi$	$M, \sigma[x \mapsto a] \models \varphi$ holds for all $a \in A$



Interesting problems

Let's consider only formulas φ with $\mathrm{FV}(\varphi) = \emptyset$ for simplicity.

model checking

$$M \overset{?}{\models} \varphi$$

(decidable)

satisfiability

$$? \models \varphi$$

(undecidable)

validity

$$\overset{?}{\models} \varphi$$

(undecidable)



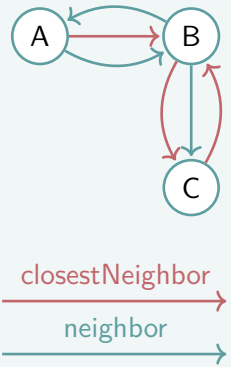
Example: Neighbors

$$\Sigma = (\{\text{closestNeighbor}\}, \{\text{neighbor}\}, \text{ar}) \text{ with}$$

- $\text{ar}(\text{closestNeighbor}) = 1$
- $\text{ar}(\text{neighbor}) = 2$

Denote the example model mathematically.

Example model



Example: Neighbors

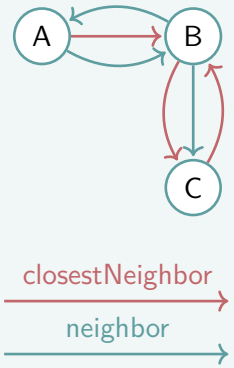
$$\Sigma = (\{\text{closestNeighbor}\}, \{\text{neighbor}\}, \text{ar}) \text{ with}$$

- $\text{ar}(\text{closestNeighbor}) = 1$
- $\text{ar}(\text{neighbor}) = 2$

Denote the example model mathematically.

1. No one is its own neighbor.
2. A closest neighbor is always a neighbor.
3. Everyone has at least one neighbor.
4. There is a person that has no other person as a neighbor.
5. Does 2. imply 3.?
6. How to model multiple “closest neighbors”?
7. How to model a situation where 2. and 4. are satisfied?

Example model



Conclusion

- Propositional logic
  - very simple
  - foundation of all logics that we are going to consider
- First-order logic
  - way more powerful
  - properties of formula often undecidable
- We are going to consider **temporal logic** next



References I

