NAIL062 Propositional & Predicate Logic: Lecture 1

Slides by Petr Gregor with minor modifications by Jakub Bulín

October 5, 2020

Overview

What is logic? [Answer]

Logic in mathematics:

- formal methods, go beyond capabilities of intuition
- automated theorem proving (and formal verification)

Logic in computer science:

- theoretical foundations (Turing machines, limits of computation)
- complexity theory: Boolean functions and circuits, decision trees, ...
- artificial intelligence: automated inference, resolution, multiagent systems & modal logic, concurrent systems & temporal logic,...

Logic in computer engineering & business applications:

- formal speficication & verification, automated testing (hardware & software)
- SAT and SMT solving, constraint logic programming, declarative programming, functional programming
- database theory (Structures, Datalog), ...

Overview

- logic for computer science
  + resolution in predicate logic, unification, “background” of Prolog
  - less of model theory, ...
- tableau method instead of Hilbert-style calculi
  + algorithmically more intuitive, (sometimes) more elegant proofs
  - uncovered (much) in usual textbooks, restriction to countable languages
- propositional logic entirely before predicate logic
  + ideal “playground” for comprehension of foundational concepts
  - slower pace of lectures at the beginning
- undecidability and incompleteness less formally
  + emphasis on principles
  - a risk of inaccuracy

History 1

- Aristotle (384-322 B.C.E.) - theory of syllogistic, e.g.
  from *'no Q is R'* and *'every P is Q'* infer *'no P is R'*.
- Euclid: *Elements* (about 330 B.C.E.) - axiomatic approach to geometry
  "There is at most one line that can be drawn parallel to another given one through an external point." (5th postulate)
- Descartes: *Geometry* (1637) - algebraic approach to geometry
- Leibniz - dream of *"lingua characteristica, calculus ratiocinator"* (1679-90)
- De Morgan - introduction of propositional connectives (1847)

$$\neg(p \vee q) \leftrightarrow \neg p \wedge \neg q$$

$$\neg(p \wedge q) \leftrightarrow \neg p \vee \neg q$$

- Boole - propositional functions, algebra of logic (1847)
- Schröder - semantics of predicate logic, concept of a model (1890-1905)

History 2

- Cantor - *intuitive* set theory (1878), e.g. the comprehension principle

  "For every property $\varphi(x)$ there exists a set $\{x \mid \varphi(x)\}$."

- Frege - first formal system with quantifiers and relations, concept of proofs based on inference, axiomatic set theory (1879, 1884)
- Russel - Frege's set theory is contradictory (1903)

  For a set $a = \{x \mid \neg(x \in x)\}$ is $a \in a$?

- Russel, Whitehead - theory of types (1910-13)
- Zermelo (1908), Fraenkel (1922) - *standard* set theory *ZFC*, e.g.

  "For every property $\varphi(x)$ and a set y there is a set $\{x \in y \mid \varphi(x)\}$."

- Bernays (1937), Gödel (1940) - set theory based on classes, e.g.

  "For every property of sets $\varphi(x)$ there exists a class $\{x \mid \varphi(x)\}$."

History 3

- Hilbert - complete axiomatizaton of Euclidean geometry (1899), formalism - strict divorce from the intended meanings

  "It could be shown that all of mathematics follows from a correctly chosen finite system of axioms."

- Post - completeness of propositional logic (Gödel: predicate)
- Gödel - incompleteness theorems (1931)
- Kleene, Post, Church, Turing - formalizations of algorithm, an existence of algorithmically undecidable problems (1936)
- Robinson - resolution method (1965)
- Kowalski; Colmerauer, Roussel - Prolog (1972), logic programming

Levels of language

We will formalize the notion of proof and validity of mathematical statements.

We distinguish different levels of logic according to the means of language, in particular to which level of quantification is admitted.

- propositional connectives *propositional logic*

  This allows to form combined propositions from the basic ones.
- variables for objects, symbols for relations and functions, quantifiers *first-order logic*

  This allows to form statements on objects, their properties and relations.

  The (standard) set theory is also described by a first-order language.

In higher-order languages we have, in addition,

- variables for sets of objects (also relations, functions) *second-order*
- variables for sets of sets of objects, *etc.* *third-order*

Examples of statements of various orders

- *"If it will not rain, we will not get wet. And if it will rain, we will get wet,*

  but then we will get dry on the sun." *proposition*

$$(\neg r \to \neg w) \wedge (r \to (w \wedge d))$$

- *"There exists the smallest element."* *first-order*

$$\exists x \; \forall y \; (x \leq y)$$

- The axiom of induction. *second-order*

$$\forall X \; ((X(0) \wedge \forall y (X(y) \to X(y+1))) \to \forall y \; X(y))$$

- *"Every union of open sets is an open set."* *third-order*

$$\forall \mathcal{X} \forall Y ((\forall X (\mathcal{X}(X) \to \mathcal{O}(X)) \wedge \forall z (Y(z) \leftrightarrow \exists X (\mathcal{X}(X) \wedge X(z)))) \to \mathcal{O}(Y))$$

Syntax and semantics

We will consider relations between syntax and semantics:

- *syntax*: language, rules for formation of formulas, interference rules, formal proof system, proof, provability,
- *semantics*: interpreted meaning, structures, models, satisfiability, validity.

We will introduce the notion of proof as a well-defined syntactical object.

A formal proof system is

- *sound*, if every provable formula is valid,
- *complete*, if every valid formula is provable.

We will show that predicate logic (first-order logic) has formal proof systems

that are both sound and complete. This does not hold for higher order logics.

Paradoxes

"Paradoxes" show us the need of precise definitions of foundational concepts.

- *Cretan paradox*

  Cretan said: "All Cretans are liars."

- *Barber paradox*

  There is a barber in a town who shaves all that do not shave themselves.

  Does he shave himself?

- *Liar paradox*

  This sentence is false.

- *Berry paradox*

  The expression "The smallest positive integer not definable in under eleven words" defines it in ten words.

Overview

1. Introduction
2. Propositional Logic
   - Basic syntax
   - Basic semantics
   - Normal forms

Language

Propositional logic is a *"logic of propositional connectives"*. We start from a (nonempty) set $\mathbb{P}$ of *propositional letters* (*variables*), e.g.

$$\mathbb{P} = \{p, p_1, p_2, \dots, q, q_1, q_2, \dots\}$$

We usually assume that $\mathbb{P}$ is countable.

The *language* of propositional logic (over $\mathbb{P}$) consists of symbols

- propositional letters from $\mathbb{P}$
- propositional connectives $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$
- parentheses (,)

Thus the language is given by the set $\mathbb{P}$. We say that connectives and parentheses are *symbols of logic*.

We also use symbols for constants $\top$ (true), $\bot$ (false) which are introduced as shortcuts for $p \vee \neg p$, resp. $p \wedge \neg p$ where p is any fixed variable from $\mathbb{P}$.

Formula

Propositional formulae (*propositions*) (over $\mathbb{P}$) are given inductively by

- every propositional letter from $\mathbb{P}$ is a proposition,
- if φ, ψ are propositions, then also

$$(\neg\varphi)\ , (\varphi \wedge \psi)\ , (\varphi \vee \psi)\ , (\varphi \rightarrow \psi)\ , (\varphi \leftrightarrow \psi)$$

  are propositions,
- every proposition is formed by a finite number of steps (*i*), (*ii*).

  - Thus propositions are (well-formed) finite sequences of symbols from the given language (strings).
  - A proposition that is a part of another proposition φ as a substring is called a *subformula* (*subproposition*) of φ.
  - The set of all propositions over $\mathbb{P}$ is denoted by $\mathrm{VF}_{\mathbb{P}}$.
  - The set of all letters (variables) that occur in φ is denoted by $\mathrm{var}(\varphi)$.

Conventions

After introducing (standard) *priorities* for connectives we are allowed in a concise form to omit parentheses that are around a subformula formed by a connective of a higher priority.

- $\rightarrow$, $\leftrightarrow$
- $\wedge$, $\vee$
- $\neg$

The outer parentheses can be omitted as well, e.g.

$$(((\neg p) \wedge q) \rightarrow (\neg(p \vee (\neg q)))) \quad \text{is shortly} \quad \neg p \wedge q \rightarrow \neg(p \vee \neg q)$$

Note If we do not respect the priorities, we can obtain an ambiguous form or even a concise form of a non-equivalent proposition.

Further possibilities to omit parentheses follow from semantical properties of connectives (associativity of $\vee$, $\wedge$).

Formation trees

A *formation tree* is a finite ordered tree whose nodes are labeled with propositions according to the following rules

- leaves (and only leaves) are labeled with propositional letters,
- if a node has label $(\neg\varphi)$, then it has a single son labeled with φ,
- if a node has label $(\varphi \wedge \psi)$, $(\varphi \vee \psi)$, $(\varphi \rightarrow \psi)$, or $(\varphi \leftrightarrow \psi)$, then it has two sons, the left son labeled with φ, and the right son labeled with ψ.

A *formation tree of a proposition* φ is a formation tree with the root labeled with φ.

Proposition *Every proposition is associated with a unique formation tree.*

Proof By induction on the number of nested parentheses. $\square$

Note Such proofs are called proofs by the structure of the formula or by the depth of the formation tree.

Semantics

- We consider only two-valued logic.
- Propositional letters represent (atomic) statements whose 'meaning' is given by an assignment of *truth values* 0 *(false)* or 1 *(true)*.
- Semantics of propositional connectives is given by their *truth tables*.

p	q	$\neg p$	$p \wedge q$	$p \vee q$	$p \rightarrow q$	$p \leftrightarrow q$
0	0	1	0	0	1	1
0	1	1	0	1	1	0
1	0	0	0	1	0	0
1	1	0	1	1	1	1

  This determines the truth value of every proposition based on the values assigned to its propositional letters.
- Thus we may assign *"truth tables"* also to all propositions. We say that propositions represent Boolean functions
- A *Boolean function* is an n-ary operation on $2 = \{0, 1\}$, i.e., $f : \{0, 1\}^n \rightarrow \{0, 1\}$.

Truth valuations

- A *truth assignment* is a function $v\colon \mathbb{P} \to \{0,1\}$, i.e. $v \in {}^{\mathbb{P}}2$.
- A *truth value* $\overline{v}(\varphi)$ of a proposition φ for a truth assignment v is given by

$$\begin{array}{rclrcl} \overline{v}(p) &=& v(p) \ \text{ if } \ p \in \mathbb{P} & \overline{v}(\neg\varphi) &=& -_1(\overline{v}(\varphi)) \\ \overline{v}(\varphi \wedge \psi) &=& \wedge_1(\overline{v}(\varphi), \overline{v}(\psi)) & \overline{v}(\varphi \vee \psi) &=& \vee_1(\overline{v}(\varphi), \overline{v}(\psi)) \\ \overline{v}(\varphi \to \psi) &=& \to_1(\overline{v}(\varphi), \overline{v}(\psi)) & \overline{v}(\varphi \leftrightarrow \psi) &=& \leftrightarrow_1(\overline{v}(\varphi), \overline{v}(\psi)) \end{array}$$

  where $-_1$, $\wedge_1$, $\vee_1$, $\to_1$, $\leftrightarrow_1$ are the Boolean functions given by the tables.

Proposition *The truth value of a proposition φ depends only on the truth assignment of $\mathrm{var}(\varphi)$.*

Proof Easily by induction on the structure of the formula. $\square$

Note Since the function $\overline{v}\colon \mathrm{VF}_{\mathbb{P}} \to 2$ is a unique extension of the function v, we can (unambiguously) write v instead of $\overline{v}$.

Semantic notions

A proposition φ over $\mathbb{P}$ is

- *is true in* (*satisfied by*) *an assignment* $v \in {}^{\mathbb{P}}2$, if $\overline{v}(\varphi) = 1$. Then v is a *satisfying assignment* for φ, denoted by $v \models \varphi$.
- *valid* (*a tautology*), if $\overline{v}(\varphi) = 1$ for every $v \in {}^{\mathbb{P}}2$, i.e. φ is satisfied by every assignment, denoted by $\models \varphi$.
- *unsatisfiable* (*a contradiction*), if $\overline{v}(\varphi) = 0$ for every $v \in {}^{\mathbb{P}}2$, i.e. $\neg\varphi$ is valid.
- *independent* (*a contingency*), if $\overline{v_1}(\varphi) = 0$ and $\overline{v_2}(\varphi) = 1$ for some $v_1, v_2 \in {}^{\mathbb{P}}2$, i.e. φ is neither a tautology nor a contradiction.
- *satisfiable*, if $\overline{v}(\varphi) = 1$ for some $v \in {}^{\mathbb{P}}2$, i.e. φ is not a contradiction.

Propositions φ and ψ are (logically) *equivalent*, denoted by $\varphi \sim \psi$, if $\overline{v}(\varphi) = \overline{v}(\psi)$ for every $v \in {}^{\mathbb{P}}2$, i.e. the proposition $\varphi \leftrightarrow \psi$ is valid.

Models

We reformulate these semantic notions in the terminology of models.

A *model of a language* $\mathbb{P}$ is a truth assignment of $\mathbb{P}$. The class of all models of $\mathbb{P}$ is denoted by $M(\mathbb{P})$, so $M(\mathbb{P}) = {}^{\mathbb{P}}2$. A proposition φ over $\mathbb{P}$ is

- *true in a model* $v \in M(\mathbb{P})$, if $\overline{v}(\varphi) = 1$. Then v is a *model of* φ, denoted by $v \models \varphi$, and the *class of all models* of φ is

$$M^{\mathbb{P}}(\varphi) = \{v \in M(\mathbb{P}) \mid v \models \varphi\}$$

- *valid* (*a tautology*) if it is true in every model of the language, denoted by $\models \varphi$.
- *unsatisfiable* (*a contradiction*) if it does not have a model.
- *independent* if it is true in some model and false in other.
- *satisfiable* if it has a model.

Propositions φ and ψ are (logically) *equivalent*, denoted by $\varphi \sim \psi$, if they have same models.

Adequacy

Language of propositional logic has *basic* connectives $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$.
In general, we can introduce *n*-ary connective for any Boolean function, e.g.

$p \downarrow q$	*"neither p nor q"*	(NOR, Peirce arrow)
$p \uparrow q$	*"not both p and q"*	(NAND, Sheffer stroke)

A set of connectives is *adequate* if they can express any Boolean function by some (well) formed proposition from them.

Proposition $\{\neg , \wedge , \vee\}$ *is adequate.*

Proof Any $f \colon {}^n2 \rightarrow 2$ is expressed by the proposition $\bigvee_{v \in f^{-1}[1]} \bigwedge_{i=0}^{n-1} p_i^{v(i)}$ where $p_i^{v(i)}$ stands for the proposition p_i if $v(i) = 1$; and for $\neg p_i$ if $v(i) = 0$. For $f^{-1}[1] = \emptyset$ we take the proposition $\bot$. $\square$

Proposition $\{\neg , \rightarrow\}$ *is adequate.*

Proof $(p \wedge q) \sim \neg(p \rightarrow \neg q)$, $(p \vee q) \sim (\neg p \rightarrow q)$. $\square$

CNF and DNF

- A *literal* is a propositional letter or its negation. For a propositional letter p let p^0 denote the literal $\neg p$ and let p^1 denote the literal p. For a literal l let $\bar{l}$ denote the *complementary* literal of l.
- A *clause* is a disjunction of literals, by the empty clause we mean $\perp$.
- A proposition is in *conjunctive normal form* (*CNF*) if it is a conjunction of clauses. By the empty proposition in CNF we mean $\top$.
- An *elementary conjunction* is a conjunction of literals, by the empty conjunction we mean $\top$.
- A proposition is in *disjunctive normal form* (*DNF*) if it is a disjunction of elementary conjunctions. By the empty proposition in DNF we mean $\perp$.

Note A clause or an elementary conjunction is both in CNF and DNF.

Observation *A proposition in CNF is valid if and only if each of its clauses contains a pair of complementary literals. A proposition in DNF is satisfiable if and only if at least one of its elementary conjunctions does not contain a pair of complementary literals.*

Transformations by tables

Proposition *Let $K \subseteq {}^{\mathbb{P}}2$ where $\mathbb{P}$ is finite. Denote $\overline{K} = {}^{\mathbb{P}}2 \setminus K$. Then*

$$M^{\mathbb{P}}\Big(\bigvee_{v \in K} \bigwedge_{p \in \mathbb{P}} p^{v(p)}\Big) = K = M^{\mathbb{P}}\Big(\bigwedge_{v \in \overline{K}} \bigvee_{p \in \mathbb{P}} \overline{p^{v(p)}}\Big)$$

Proof The first equality follows from $\overline{w}(\bigwedge_{p \in \mathbb{P}} p^{v(p)}) = 1$ whenever $w = v$, for every $w \in {}^{\mathbb{P}}2$. Similarly, the second one follows from $\overline{w}(\bigvee_{p \in \mathbb{P}} \overline{p^{v(p)}}) = 1$ whenever $w \neq v$. □

For example, $K = \{(1,0,0),(1,1,0),(0,1,0),(1,1,1)\}$ can be modeled by

$$(p \wedge \neg q \wedge \neg r) \vee (p \wedge q \wedge \neg r) \vee (\neg p \wedge q \wedge \neg r) \vee (p \wedge q \wedge r) \sim$$
$$(p \vee q \vee r) \wedge (p \vee q \vee \neg r) \wedge (p \vee \neg q \vee \neg r) \wedge (\neg p \vee q \vee \neg r)$$

Corollary *Every proposition has CNF and DNF equivalents.*

Proof The value of a proposition φ depends only on the assignment of $\mathrm{var}(\varphi)$ which is finite. Hence we can apply the above proposition for $K = M^{\mathbb{P}}(\varphi)$ and $\mathbb{P} = \mathrm{var}(\varphi)$. □

Transformations by rules

Proposition *Let φ' be the proposition obtained from φ by replacing some occurrences of a subformula ψ with ψ'. If $\psi \sim \psi'$, then $\varphi \sim \varphi'$.*

Proof Easily by induction on the structure of the formula. $\square$

(1) $(\varphi \rightarrow \psi) \sim (\neg\varphi \vee \psi), \quad (\varphi \leftrightarrow \psi) \sim ((\neg\varphi \vee \psi) \wedge (\neg\psi \vee \varphi))$

(2) $\neg\neg\varphi \sim \varphi, \quad \neg(\varphi \wedge \psi) \sim (\neg\varphi \vee \neg\psi), \quad \neg(\varphi \vee \psi) \sim (\neg\varphi \wedge \neg\psi)$

(3) $(\varphi \vee (\psi \wedge \chi)) \sim ((\psi \wedge \chi) \vee \varphi) \sim ((\varphi \vee \psi) \wedge (\varphi \vee \chi))$

(3)' $(\varphi \wedge (\psi \vee \chi)) \sim ((\psi \vee \chi) \wedge \varphi) \sim ((\varphi \wedge \psi) \vee (\varphi \wedge \chi))$

Proposition *Every proposition can be transformed into CNF / DNF applying the transformation rules $(1), (2), (3)/(3)'$.*

Proof Easily by induction on the structure of the formula. $\square$

Proposition *Assume that φ contains only $\neg$, $\wedge$, $\vee$ and φ^* is obtained from φ by interchanging $\wedge$ and $\vee$, and by complementing all literals. Then* $\neg\varphi \sim \varphi^*$.

Proof Easily by induction on the structure of the formula. $\square$

NAIL062 Propositional & Predicate Logic: Lecture 2

Slides by Petr Gregor with minor modifications by Jakub Bulín

October 12, 2020

- Please enroll in our Moodle course (if you haven't done so yet): https://dl1.cuni.cz/course/view.php?id=10128
- Please use the discussion forum on Moodle whenever possible, and Moodle messages.
- The Limnu whiteboards are only available for 14 days (save them manually if you want).
- Let me know if you want to schedule office hours!
- The issue with my microphone should be fixed now. (Let me know in case it reappears!)
- In the slides, $\mathrm{VF}_{\mathbb{P}}$ stands for "very many, actually, all propositions over the language $\mathbb{P}$", the notation $\mathrm{PF}_{\mathbb{P}}$ is reserved for "predicate (first-order) formulas".

Table of Contents

Semantics

- We consider only two-valued logic.
- Propositional letters represent (atomic) statements whose 'meaning' is given by an assignment of *truth values* 0 *(false)* or 1 *(true)*.
- Semantics of propositional connectives is given by their *truth tables*.

p	q	$\neg p$	$p \wedge q$	$p \vee q$	$p \rightarrow q$	$p \leftrightarrow q$
0	0	1	0	0	1	1
0	1	1	0	1	1	0
1	0	0	0	1	0	0
1	1	0	1	1	1	1

  This determines the truth value of every proposition based on the values assigned to its propositional letters.

- Thus we may assign *"truth tables"* also to all propositions. We say that propositions represent Boolean functions
- A *Boolean function* is an n-ary operation on $2 = \{0, 1\}$, i.e., $f : \{0, 1\}^n \rightarrow \{0, 1\}$.

Truth valuations

- A *truth assignment* is a function $v\colon \mathbb{P} \to \{0,1\}$, i.e. $v \in {}^{\mathbb{P}}2$.
- A *truth value* $\overline{v}(\varphi)$ of a proposition φ for a truth assignment v is given by

$$\begin{array}{rr} \overline{v}(p) = v(p) \text{ if } p \in \mathbb{P} & \overline{v}(\neg\varphi) = -_1(\overline{v}(\varphi)) \\ \overline{v}(\varphi \wedge \psi) = \wedge_1(\overline{v}(\varphi), \overline{v}(\psi)) & \overline{v}(\varphi \vee \psi) = \vee_1(\overline{v}(\varphi), \overline{v}(\psi)) \\ \overline{v}(\varphi \to \psi) = \to_1(\overline{v}(\varphi), \overline{v}(\psi)) & \overline{v}(\varphi \leftrightarrow \psi) = \leftrightarrow_1(\overline{v}(\varphi), \overline{v}(\psi)) \end{array}$$

  where $-_1$, $\wedge_1$, $\vee_1$, $\to_1$, $\leftrightarrow_1$ are the Boolean functions given by the tables.

Proposition *The truth value of a proposition φ depends only on the truth assignment of $\mathrm{var}(\varphi)$.*

Proof Easily by induction on the structure of the formula. $\square$

Note Since the function $\overline{v}\colon \mathrm{VF}_{\mathbb{P}} \to 2$ is a unique extension of the function v, we can (unambiguously) write v instead of $\overline{v}$.

Semantic notions

A proposition φ over $\mathbb{P}$ is

- *is true in* (*satisfied by*) *an assignment* $v \in {}^{\mathbb{P}}2$, if $\overline{v}(\varphi) = 1$. Then v is a *satisfying assignment* for φ, denoted by $v \models \varphi$.
- *valid* (*a tautology*), if $\overline{v}(\varphi) = 1$ for every $v \in {}^{\mathbb{P}}2$, i.e. φ is satisfied by every assignment, denoted by $\models \varphi$.
- *unsatisfiable* (*a contradiction*), if $\overline{v}(\varphi) = 0$ for every $v \in {}^{\mathbb{P}}2$, i.e. $\neg\varphi$ is valid.
- *independent* (*a contingency*), if $\overline{v_1}(\varphi) = 0$ and $\overline{v_2}(\varphi) = 1$ for some $v_1, v_2 \in {}^{\mathbb{P}}2$, i.e. φ is neither a tautology nor a contradiction.
- *satisfiable*, if $\overline{v}(\varphi) = 1$ for some $v \in {}^{\mathbb{P}}2$, i.e. φ is not a contradiction.

Propositions φ and ψ are (logically) *equivalent*, denoted by $\varphi \sim \psi$, if $\overline{v}(\varphi) = \overline{v}(\psi)$ for every $v \in {}^{\mathbb{P}}2$, i.e. the proposition $\varphi \leftrightarrow \psi$ is valid.

Models

We reformulate these semantic notions in the terminology of models.

A *model of a language* $\mathbb{P}$ is a truth assignment of $\mathbb{P}$. The class of all models of $\mathbb{P}$ is denoted by $M(\mathbb{P})$, so $M(\mathbb{P}) = {}^{\mathbb{P}}2$. A proposition φ over $\mathbb{P}$ is

- *true in a model* $v \in M(\mathbb{P})$, if $\overline{v}(\varphi) = 1$. Then v is a *model of* φ, denoted by $v \models \varphi$, and the *class of all models* of φ is

$$M^{\mathbb{P}}(\varphi) = \{v \in M(\mathbb{P}) \mid v \models \varphi\}$$

- *valid* (*a tautology*) if it is true in every model of the language, denoted by $\models \varphi$.
- *unsatisfiable* (*a contradiction*) if it does not have a model.
- *independent* if it is true in some model and false in other.
- *satisfiable* if it has a model.

Propositions φ and ψ are (logically) *equivalent*, denoted by $\varphi \sim \psi$, if they have same models.

Adequacy

Language of propositional logic has *basic* connectives $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$.
In general, we can introduce *n*-ary connective for any Boolean function, e.g.

$p \downarrow q$	*"neither p nor q"*	(NOR, Peirce arrow)
$p \uparrow q$	*"not both p and q"*	(NAND, Sheffer stroke)

A set of connectives is *adequate* if they can express any Boolean function by some (well) formed proposition from them.

Proposition $\{\neg\ , \wedge\ , \vee\}$ *is adequate.*

Proof Any $f\colon {}^n2 \rightarrow 2$ is expressed by the proposition $\bigvee_{v \in f^{-1}[1]} \bigwedge_{i=0}^{n-1} p_i^{v(i)}$
where $p_i^{v(i)}$ stands for the proposition p_i if $v(i) = 1$; and for $\neg p_i$ if $v(i) = 0$. For $f^{-1}[1] = \emptyset$ we take the proposition $\perp$. $\square$

Proposition $\{\neg\ , \rightarrow\}$ *is adequate.*

Proof $(p \wedge q) \sim \neg(p \rightarrow \neg q)$, $(p \vee q) \sim (\neg p \rightarrow q)$. $\square$

CNF and DNF

- A *literal* is a propositional letter or its negation. For a propositional letter p let p^0 denote the literal $\neg p$ and let p^1 denote the literal p. For a literal l let $\bar{l}$ denote the *complementary* literal of l.
- A *clause* is a disjunction of literals, by the empty clause we mean $\perp$.
- A proposition is in *conjunctive normal form* (*CNF*) if it is a conjunction of clauses. By the empty proposition in CNF we mean $\top$.
- An *elementary conjunction* is a conjunction of literals, by the empty conjunction we mean $\top$.
- A proposition is in *disjunctive normal form* (*DNF*) if it is a disjunction of elementary conjunctions. By the empty proposition in DNF we mean $\perp$.

Note A clause or an elementary conjunction is both in CNF and DNF.

Observation *A proposition in CNF is valid if and only if each of its clauses contains a pair of complementary literals. A proposition in DNF is satisfiable if and only if at least one of its elementary conjunctions does not contain a pair of complementary literals.*

Transformations by tables

Proposition *Let $K \subseteq {}^{\mathbb{P}}2$ where $\mathbb{P}$ is finite. Denote $\overline{K} = {}^{\mathbb{P}}2 \setminus K$. Then*

$$M^{\mathbb{P}}\Big(\bigvee_{v \in K} \bigwedge_{p \in \mathbb{P}} p^{v(p)}\Big) = K = M^{\mathbb{P}}\Big(\bigwedge_{v \in \overline{K}} \bigvee_{p \in \mathbb{P}} \overline{p^{v(p)}}\Big)$$

Proof The first equality follows from $\overline{w}(\bigwedge_{p \in \mathbb{P}} p^{v(p)}) = 1$ whenever $w = v$, for every $w \in {}^{\mathbb{P}}2$. Similarly, the second one follows from $\overline{w}(\bigvee_{p \in \mathbb{P}} \overline{p^{v(p)}}) = 1$ whenever $w \neq v$. □

For example, $K = \{(1,0,0),(1,1,0),(0,1,0),(1,1,1)\}$ can be modeled by

$$(p \wedge \neg q \wedge \neg r) \vee (p \wedge q \wedge \neg r) \vee (\neg p \wedge q \wedge \neg r) \vee (p \wedge q \wedge r) \sim$$
$$(p \vee q \vee r) \wedge (p \vee q \vee \neg r) \wedge (p \vee \neg q \vee \neg r) \wedge (\neg p \vee q \vee \neg r)$$

Corollary *Every proposition has CNF and DNF equivalents.*

Proof The value of a proposition φ depends only on the assignment of $\mathrm{var}(\varphi)$ which is finite. Hence we can apply the above proposition for $K = M^{\mathbb{P}}(\varphi)$ and $\mathbb{P} = \mathrm{var}(\varphi)$. □

Transformations by rules

Proposition *Let φ' be the proposition obtained from φ by replacing some occurrences of a subformula ψ with ψ'. If $\psi \sim \psi'$, then $\varphi \sim \varphi'$.*

Proof Easily by induction on the structure of the formula. $\square$

(1) $(\varphi \to \psi) \sim (\neg\varphi \vee \psi), \quad (\varphi \leftrightarrow \psi) \sim ((\neg\varphi \vee \psi) \wedge (\neg\psi \vee \varphi))$

(2) $\neg\neg\varphi \sim \varphi, \quad \neg(\varphi \wedge \psi) \sim (\neg\varphi \vee \neg\psi), \quad \neg(\varphi \vee \psi) \sim (\neg\varphi \wedge \neg\psi)$

(3) $(\varphi \vee (\psi \wedge \chi)) \sim ((\psi \wedge \chi) \vee \varphi) \sim ((\varphi \vee \psi) \wedge (\varphi \vee \chi))$

(3)' $(\varphi \wedge (\psi \vee \chi)) \sim ((\psi \vee \chi) \wedge \varphi) \sim ((\varphi \wedge \psi) \vee (\varphi \wedge \chi))$

Proposition *Every proposition can be transformed into CNF / DNF applying the transformation rules $(1), (2), (3)/(3)'$.*

Proof Easily by induction on the structure of the formula. $\square$

Proposition *Assume that φ contains only $\neg$, $\wedge$, $\vee$ and φ^* is obtained from φ by interchanging $\wedge$ and $\vee$, and by complementing all literals. Then $\neg\varphi \sim \varphi^*$.*

Proof Easily by induction on the structure of the formula. $\square$

Boolean Satisfiability and SAT solvers

- The *SAT* problem: Is a given propositional formula satisfiable?
- *Example (boardomino)* *Is it possible to perfectly cover a chessboard with two diagonally opposite corners removed using domino tiles?* We can easily form a propositional formula that is satisfiable, if and only if the answer is yes. Then we can test its satisfiability using a SAT solver.
- Best SAT solvers: http://www.satcompetition.org/
- We will use Glucose, and the DIMACS file format for CNF input.
- In general, can we convert all of mathematics to logical formulas? AI, theorem proving, Peano: *Formulario* (1895-1908), Mizar system,
- *Why people (usually) do not do it?* How can we solve the boardomino problem more *elegantly*? What is our approach based on?[1]

[1] Each domino tile covers one white and one black field, but there are more fields of one color since both the removed corners have the same color.

2-SAT

- A proposition in CNF is in *k-CNF* if every clause has at most k literals.
- *k-SAT* is the following problem (for fixed $k > 0$)

  INSTANCE: *A proposition φ in k-CNF.*

  QUESTION: *Is φ satisfiable?*

The problem k-SAT for $k \geq 3$ is an NP-complete problem. We will show that 2-SAT can be solved in *linear* time (with respect to the length of φ).

We will neglect implementation details (computational model, representation in memory) and use the following fact (see `[ADS I]`):

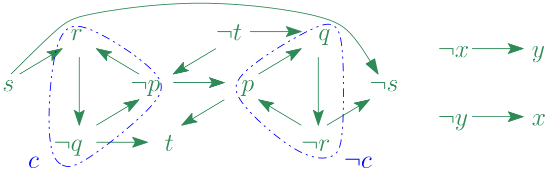
Proposition *A partition of a directed graph (V, E) to strongly connected components can be found in time $\mathcal{O}(|V| + |E|)$.*

- A directed graph G is *strongly connected* if for every two vertices u and v there are directed paths both from u to v and from v to u.
- A strongly connected *component* of a graph G is a maximal strongly connected subgraph of G.

Implication graphs

The *implication graph* G_φ of a 2-CNF proposition φ is the following directed graph:

- vertices are all the propositional letters in φ and their negations,
- a clause $l_1 \vee l_2$ in φ is represented by a pair of edges $\overline{l_1} \to l_2$, $\overline{l_2} \to l_1$,
- a clause l_1 in φ is represented by an edge $\overline{l_1} \to l_1$.



$p \wedge (\neg p \vee q) \wedge (\neg q \vee \neg r) \wedge (p \vee r) \wedge (r \vee \neg s) \wedge (\neg p \vee t) \wedge (q \vee t) \wedge \neg s \wedge (x \vee y)$

Proposition *φ is satisfiable if and only if no strongly connected component of G_φ contains a pair of complementary literals.*

Proof Every satisfying assignment has to assign the same value to all literals in one component; the left-to-right implication (necessity) holds.

Satisfying assignment

For the right-to-left implication (sufficiency), let G^*_φ be the graph obtained from G_φ by contracting strongly connected components to single vertices.

Observation G^*_φ *is acyclic, and therefore has a topological ordering* $<$.

- A directed graph is *acyclic* if it is has no directed *cycles*.
- A linear ordering $<$ of vertices of a directed graph is *topological* if $p < q$ for every edge from p to q.

Now for every unassigned component in an increasing order by $<$, assign 0 to all its literals and 1 to all literals in the complementary component.

It remains to show that such assignment v satisfies φ. If not, then G^*_φ contains edges $p \to q$ and $\overline{q} \to \overline{p}$ with $v(p) = 1$ and $v(q) = 0$. But this contradicts the order of assigning values to components since $p < q$ and $\overline{q} < \overline{p}$. $\square$

Corollary *2-SAT can be solved in linear time.*

Horn-SAT

- A *unit clause* is a clause containing a single literal,
- a *Horn clause* is a clause containing at most one positive literal,

$$\neg p_1 \vee \cdots \vee \neg p_n \vee q \quad \sim \quad (p_1 \wedge \cdots \wedge p_n) \rightarrow q$$

- a *Horn formula* is a conjunction of Horn clauses,
- *Horn-SAT* is the problem of satisfiability of a given Horn formula.

Algorithm

(1) *if φ contains a pair of unit clauses l and $\bar{l}$, then it is not satisfiable,*

(2) *if φ contains a unit clause l, then assign 1 to l, remove all clauses containing l, remove $\bar{l}$ from all clauses, and repeat from the start,*

(3) *if φ does not contain a unit clause, then it is satisfied by assigning 0 to all remaining propositional variables.*

Step (2) is called *unit propagation*.

Unit propagation

$$(\neg p \vee q) \wedge (\neg p \vee \neg q \vee r) \wedge (\neg r \vee \neg s) \wedge (\neg t \vee s) \wedge s \qquad v(s) = 1$$
$$(\neg p \vee q) \wedge (\neg p \vee \neg q \vee r) \wedge \neg r \qquad v(\neg r) = 1$$
$$(\neg p \vee q) \wedge (\neg p \vee \neg q) \qquad v(p) = v(q) = v(t) = 0$$

Observation *Let φ^l be the proposition obtained from φ by unit propagation. Then φ^l is satisfiable if and only if φ is satisfiable.*

Corollary *The algorithm is correct (it solves Horn-SAT).*

Proof The correctness in Step (1) is obvious, in Step (2) it follows from the observation, in Step (3) it follows from the *Horn form* since every remaining clause contains at least one negative literal.

Note A direct implementation requires quadratic time, but with an appropriate representation in memory, one can achieve linear time (w.r.t. the length of φ).

Theory

Informally, a description of the "world" to which we restrict ourselves, i.e., which we want to model.

- A propositional *theory* over the language $\mathbb{P}$ is any set $T \subseteq \mathrm{VF}_{\mathbb{P}}$ if propositions. The propositions in T are *axioms* of the theory T.
- A *model of the theory* T over $\mathbb{P}$ is an assignment $v \in M(\mathbb{P})$ (i.e., a model of the language) in which all axioms of T are true. We write $v \models T$ ("v models T").
- The *class of (all) models* of T is

$$M^{\mathbb{P}}(T) = \{v \in M(\mathbb{P}) \mid v \models \varphi \ \text{ for all } \ \varphi \in T\}.$$

  For example, for $T = \{p,\ \neg p \vee \neg q,\ q \to r\}$ over $\mathbb{P} = \{p, q, r\}$:

$$M^{\mathbb{P}}(T) = \{(1,0,0),(1,0,1)\}$$

- If a theory is finite, it can be replaced by a *conjunction* of its axioms.
- We write $M(T, \varphi)$ as a shortcut for $M(T \cup \{\varphi\})$.

Semantics with respect to a theory

Semantic notions can be defined relative to a theory (more precisely, its models). Let T be a theory over $\mathbb{P}$. A proposition φ over $\mathbb{P}$ is

- *valid in T* (*true in T*) if it is true in every model of T, denoted by $T \models \varphi$, we also say that φ is a (semantic) *consequence* of T,
- *unsatisfiable* (*contradictory*) *in T* (*inconsistent with T*) if it is false in every model of T,
- *independent (or contingency) in T* if it is true in some model of T and false in some other,
- *satisfiable in T* (*consistent with T*) if it is true in some model of T.

Propositions φ and ψ are *equivalent in T* (*T-equivalent*), denoted by $\varphi \sim_T \psi$, if for every model v of T, $v \models \varphi$ if and only if $v \models \psi$.

Note If all axioms of a theory T are valid (tautologies), e.g for $T = \emptyset$, then all notions with respect to T correspond to the same notions in (pure) logic.

Consequences of a theory

The *consequences* of a theory T over $\mathbb{P}$ is the set $\theta^{\mathbb{P}}(T)$ of all propositions that are valid in T, i.e.

$$\theta^{\mathbb{P}}(T) = \{\varphi \in \mathrm{VF}_{\mathbb{P}} \mid T \models \varphi\}.$$

Proposition[2] *For theories $T \subseteq T'$ and propositions $\varphi, \varphi_1, \ldots, \varphi_n$ over $\mathbb{P}$,*

(1) $T \subseteq \theta^{\mathbb{P}}(T) \;=\; \theta^{\mathbb{P}}(\theta^{\mathbb{P}}(T))$,

(2) $T \subseteq T' \;\Rightarrow\; \theta^{\mathbb{P}}(T) \subseteq \theta^{\mathbb{P}}(T')$,

(3) $\varphi \in \theta^{\mathbb{P}}(\{\varphi_1, \ldots, \varphi_n\}) \;\Leftrightarrow\; \models (\varphi_1 \wedge \ldots \wedge \varphi_n) \rightarrow \varphi$.

Proof Easily from the definitions, since $T \models \varphi \Leftrightarrow M(T) \subseteq M(\varphi)$ and

- $M(\theta(T)) = M(T)$,
- $T \subseteq T' \;\Rightarrow\; M(T') \subseteq M(T)$,
- $\models \psi \rightarrow \varphi \;\Leftrightarrow\; M(\psi) \subseteq M(\varphi)$ and $M(\varphi_1 \wedge \ldots \wedge \varphi_n) = M(\varphi_1, \ldots, \varphi_n)$.

[2] This proposition says that θ is a "*closure operator*".

Properties of theories

A propositional theory T over $\mathbb{P}$ is *(semantically)*

- *inconsistent* (or *unsatisfiable*) if $T \models \bot$, otherwise it is *consistent* (or *satisfiable*),
- *complete* if it is consistent, and $T \models \varphi$ or $T \models \neg\varphi$ for every $\varphi \in \mathrm{VF}_{\mathbb{P}}$, i.e. no proposition over $\mathbb{P}$ is independent in T,
- an *extension* of a theory T' over $\mathbb{P}'$ if $\mathbb{P}' \subseteq \mathbb{P}$ and $\theta^{\mathbb{P}'}(T') \subseteq \theta^{\mathbb{P}}(T)$; we say that an extension T of a theory T' is *simple* if $\mathbb{P} = \mathbb{P}'$; and *conservative* if $\theta^{\mathbb{P}'}(T') = \theta^{\mathbb{P}}(T) \cap \mathrm{VF}_{\mathbb{P}'}$,
- *equivalent* with a theory T' if T is an extension of T' and vice-versa,

Observation *Let T and T' be theories over $\mathbb{P}$. Then T is (semantically)*

(i) *consistent, if and only if it has a model,*

(ii) *complete, if and only if it has a single model,*

(iii) *extension of T', if and only if $M^{\mathbb{P}}(T) \subseteq M^{\mathbb{P}}(T')$,*

(iv) *equivalent with T', if and only if $M^{\mathbb{P}}(T) = M^{\mathbb{P}}(T')$.*

Lindenbaum-Tarski algebra

Let T be a consistent theory over $\mathbb{P}$. On the quotient set $\mathrm{VF}_{\mathbb{P}}/{\sim_T}$ we can naturally define operations $\neg$, $\wedge$, $\vee$, $\bot$, $\top$ using representatives, e.g

$$[\varphi]_{\sim_T} \wedge [\psi]_{\sim_T} = [\varphi \wedge \psi]_{\sim_T}$$

The *Lindenbaum-Tarski algebra* for T is

$$AV^{\mathbb{P}}(T) = \langle \mathrm{VF}_{\mathbb{P}}/{\sim_T}, \neg, \wedge, \vee, \bot, \top \rangle$$

Since $\varphi \sim_T \psi \;\Leftrightarrow\; M(T,\varphi) = M(T,\psi)$, it follows that the mapping $h\colon \mathrm{VF}_{\mathbb{P}}/{\sim_T} \to \mathcal{P}(M(T))$ defined by $h([\varphi]_{\sim_T}) = M(T,\varphi)$ is a (well-defined) injective function, and satisfies the following properties. Moreover, h is *surjective* if $M(T)$ is *finite*.

$$h(\neg[\varphi]_{\sim_T}) = M(T) \setminus M(T,\varphi)$$
$$h([\varphi]_{\sim_T} \wedge [\psi]_{\sim_T}) = M(T,\varphi) \cap M(T,\psi)$$
$$h([\varphi]_{\sim_T} \vee [\psi]_{\sim_T}) = M(T,\varphi) \cup M(T,\psi)$$
$$h([\bot]_{\sim_T}) = \emptyset, \quad h([\top]_{\sim_T}) = M(T)$$

Corollary If T is a consistent theory over a finite $\mathbb{P}$, then $AV^{\mathbb{P}}(T)$ is a Boolean algebra *isomorphic* via h to the (finite) algebra of sets $\underline{\mathcal{P}}(M(T))$.

Analysis of theories over finite languages

Let T be a consistent theory over $\mathbb{P}$ where $|\mathbb{P}| = n \in \mathbb{N}^+$ and $m = |M^{\mathbb{P}}(T)|$. Then the number of (mutually) inequivalent

- propositions (or theories) over $\mathbb{P}$ is 2^{2^n},
- propositions over $\mathbb{P}$ that are valid (contradictory) in T is $2^{2^n - m}$,
- propositions over $\mathbb{P}$ that are independent in T is $2^{2^n} - 2 \cdot 2^{2^n - m}$,
- simple extensions of T is 2^m, out of which 1 is inconsistent,
- complete simple extensions of T is m.

And the number of (mutually) T-inequivalent

- propositions over $\mathbb{P}$ is 2^m,
- propositions over $\mathbb{P}$ that are valid (contradictory) (in T) is 1,
- propositions over $\mathbb{P}$ that are independent (in T) is $2^m - 2$.

Proof Using the bijection of $\mathrm{VF}_{\mathbb{P}}/\!\sim$ resp. $\mathrm{VF}_{\mathbb{P}}/\!\sim_T$ with $\mathcal{P}(M(\mathbb{P}))$ resp. $\mathcal{P}(M^{\mathbb{P}}(T))$ it suffices to count the corresponding sets of models. $\square$

NAIL062 Propositional & Predicate Logic: Lecture 3

Slides by Petr Gregor with minor modifications by Jakub Bulín

October 19, 2020

2-SAT

- A proposition in CNF is in *k-CNF* if every clause has at most k literals.
- *k-SAT* is the following problem (for fixed $k > 0$)

  INSTANCE: *A proposition φ in k-CNF.*

  QUESTION: *Is φ satisfiable?*

The problem k-SAT for $k \geq 3$ is an NP-complete problem. We will show that 2-SAT can be solved in *linear* time (with respect to the length of φ).

We will neglect implementation details (computational model, representation in memory) and use the following fact (see `[ADS I]`):

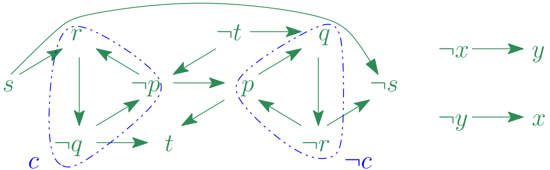
Proposition *A partition of a directed graph (V, E) to strongly connected components can be found in time $\mathcal{O}(|V| + |E|)$.*

- A directed graph G is *strongly connected* if for every two vertices u and v there are directed paths both from u to v and from v to u.
- A strongly connected *component* of a graph G is a maximal strongly connected subgraph of G.

Implication graphs

The *implication graph* G_φ of a 2-CNF proposition φ is the following directed graph:

- vertices are all the propositional letters in φ and their negations,
- a clause $l_1 \vee l_2$ in φ is represented by a pair of edges $\overline{l_1} \to l_2$, $\overline{l_2} \to l_1$,
- a clause l_1 in φ is represented by an edge $\overline{l_1} \to l_1$.



$p \wedge (\neg p \vee q) \wedge (\neg q \vee \neg r) \wedge (p \vee r) \wedge (r \vee \neg s) \wedge (\neg p \vee t) \wedge (q \vee t) \wedge \neg s \wedge (x \vee y)$

Proposition *φ is satisfiable if and only if no strongly connected component of G_φ contains a pair of complementary literals.*

Proof Every satisfying assignment has to assign the same value to all literals in one component; the left-to-right implication (necessity) holds.

Satisfying assignment

For the right-to-left implication (sufficiency), let G_φ^* be the graph obtained from G_φ by contracting strongly connected components to single vertices.

Observation G_φ^* *is acyclic, and therefore has a topological ordering* $<$.

- A directed graph is *acyclic* if it is has no directed *cycles*.
- A linear ordering $<$ of vertices of a directed graph is *topological* if $p < q$ for every edge from p to q.

Now for every unassigned component in an increasing order by $<$, assign 0 to all its literals and 1 to all literals in the complementary component.

It remains to show that such assignment v satisfies φ. If not, then G_φ^* contains edges $p \to q$ and $\overline{q} \to \overline{p}$ with $v(p) = 1$ and $v(q) = 0$. But this contradicts the order of assigning values to components since $p < q$ and $\overline{q} < \overline{p}$. $\square$

Corollary *2-SAT can be solved in linear time.*

Horn-SAT

- A *unit clause* is a clause containing a single literal,
- a *Horn clause* is a clause containing at most one positive literal,

$$\neg p_1 \vee \cdots \vee \neg p_n \vee q \quad \sim \quad (p_1 \wedge \cdots \wedge p_n) \rightarrow q$$

- a *Horn formula* is a conjunction of Horn clauses,
- *Horn-SAT* is the problem of satisfiability of a given Horn formula.

Algorithm

(1) *if φ contains a pair of unit clauses l and $\bar{l}$, then it is not satisfiable,*

(2) *if φ contains a unit clause l, then assign 1 to l, remove all clauses containing l, remove $\bar{l}$ from all clauses, and repeat from the start,*

(3) *if φ does not contain a unit clause, then it is satisfied by assigning 0 to all remaining propositional variables.*

Step (2) is called *unit propagation*.

Unit propagation

$$(\neg p \vee q) \wedge (\neg p \vee \neg q \vee r) \wedge (\neg r \vee \neg s) \wedge (\neg t \vee s) \wedge s \qquad v(s) = 1$$
$$(\neg p \vee q) \wedge (\neg p \vee \neg q \vee r) \wedge \neg r \qquad v(\neg r) = 1$$
$$(\neg p \vee q) \wedge (\neg p \vee \neg q) \qquad v(p) = v(q) = v(t) = 0$$

Observation *Let φ^l be the proposition obtained from φ by unit propagation. Then φ^l is satisfiable if and only if φ is satisfiable.*

Corollary *The algorithm is correct (it solves Horn-SAT).*

Proof The correctness in Step (1) is obvious, in Step (2) it follows from the observation, in Step (3) it follows from the *Horn form* since every remaining clause contains at least one negative literal.

Note A direct implementation requires quadratic time, but with an appropriate representation in memory, one can achieve linear time (w.r.t. the length of φ).

Theory

Informally, a description of the "world" to which we restrict ourselves, i.e., which we want to model.

- A propositional *theory* over the language $\mathbb{P}$ is any set $T \subseteq \mathrm{VF}_{\mathbb{P}}$ if propositions. The propositions in T are *axioms* of the theory T.
- A *model of the theory* T over $\mathbb{P}$ is an assignment $v \in M(\mathbb{P})$ (i.e., a model of the language) in which all axioms of T are true. We write $v \models T$ ("v models T").
- The *class of (all) models* of T is

$$M^{\mathbb{P}}(T) = \{v \in M(\mathbb{P}) \mid v \models \varphi \ \text{ for all } \ \varphi \in T\}.$$

  For example, for $T = \{p,\ \neg p \vee \neg q,\ q \to r\}$ over $\mathbb{P} = \{p, q, r\}$:

$$M^{\mathbb{P}}(T) = \{(1,0,0), (1,0,1)\}$$

- If a theory is finite, it can be replaced by a *conjunction* of its axioms.
- We write $M(T, \varphi)$ as a shortcut for $M(T \cup \{\varphi\})$.

Semantics with respect to a theory

Semantic notions can be defined relative to a theory (more precisely, its models). Let T be a theory over $\mathbb{P}$. A proposition φ over $\mathbb{P}$ is

- *valid in T* (*true in T*) if it is true in every model of T, denoted by $T \models \varphi$, we also say that φ is a (semantic) *consequence* of T,
- *unsatisfiable* (*contradictory*) *in T* (*inconsistent with T*) if it is false in every model of T,
- *independent (or contingency) in T* if it is true in some model of T and false in some other,
- *satisfiable in T* (*consistent with T*) if it is true in some model of T.

Propositions φ and ψ are *equivalent in T* (*T-equivalent*), denoted by $\varphi \sim_T \psi$, if for every model v of T, $v \models \varphi$ if and only if $v \models \psi$.

Note If all axioms of a theory T are valid (tautologies), e.g for $T = \emptyset$, then all notions with respect to T correspond to the same notions in (pure) logic.

Consequences of a theory

The *consequences* of a theory T over $\mathbb{P}$ is the set $\theta^{\mathbb{P}}(T)$ of all propositions that are valid in T, i.e.

$$\theta^{\mathbb{P}}(T) = \{\varphi \in \mathrm{VF}_{\mathbb{P}} \mid T \models \varphi\}.$$

Proposition[1] *For theories $T \subseteq T'$ and propositions $\varphi, \varphi_1, \dots, \varphi_n$ over $\mathbb{P}$,*

(1) $T \subseteq \theta^{\mathbb{P}}(T) \ = \ \theta^{\mathbb{P}}(\theta^{\mathbb{P}}(T))$,

(2) $T \subseteq T' \ \Rightarrow \ \theta^{\mathbb{P}}(T) \subseteq \theta^{\mathbb{P}}(T')$,

(3) $\varphi \in \theta^{\mathbb{P}}(\{\varphi_1, \dots, \varphi_n\}) \ \Leftrightarrow \ \models (\varphi_1 \wedge \dots \wedge \varphi_n) \rightarrow \varphi$.

Proof Easily from the definitions, since $T \models \varphi \Leftrightarrow M(T) \subseteq M(\varphi)$ and

- $M(\theta(T)) = M(T)$,
- $T \subseteq T' \ \Rightarrow \ M(T') \subseteq M(T)$,
- $\models \psi \rightarrow \varphi \ \Leftrightarrow \ M(\psi) \subseteq M(\varphi)$ and $M(\varphi_1 \wedge \dots \wedge \varphi_n) = M(\varphi_1, \dots, \varphi_n)$.

[1] This proposition says that θ is a "*closure operator*".

Properties of theories

A propositional theory T over $\mathbb{P}$ is *(semantically)*

- *inconsistent* (or *unsatisfiable*) if $T \models \bot$, otherwise it is *consistent* (or *satisfiable*),
- *complete* if it is consistent, and $T \models \varphi$ or $T \models \neg\varphi$ for every $\varphi \in \mathrm{VF}_{\mathbb{P}}$, i.e. no proposition over $\mathbb{P}$ is independent in T,
- an *extension* of a theory T' over $\mathbb{P}'$ if $\mathbb{P}' \subseteq \mathbb{P}$ and $\theta^{\mathbb{P}'}(T') \subseteq \theta^{\mathbb{P}}(T)$; we say that an extension T of a theory T' is *simple* if $\mathbb{P} = \mathbb{P}'$; and *conservative* if $\theta^{\mathbb{P}'}(T') = \theta^{\mathbb{P}}(T) \cap \mathrm{VF}_{\mathbb{P}'}$,
- *equivalent* with a theory T' if T is an extension of T' and vice-versa,

Observation *Let T and T' be theories over $\mathbb{P}$. Then T is (semantically)*

(i) *consistent, if and only if it has a model,*

(ii) *complete, if and only if it has a single model,*

(iii) *extension of T', if and only if $M^{\mathbb{P}}(T) \subseteq M^{\mathbb{P}}(T')$,*

(iv) *equivalent with T', if and only if $M^{\mathbb{P}}(T) = M^{\mathbb{P}}(T')$.*

NAIL062 Propositional & Predicate Logic: Lecture 4

Slides by Petr Gregor with minor modifications by Jakub Bulín

October 26, 2020

Lindenbaum-Tarski algebra

Let T be a consistent theory over $\mathbb{P}$. On the quotient set $\mathrm{VF}_{\mathbb{P}}/{\sim_T}$ we can naturally define operations $\neg$, $\wedge$, $\vee$, $\perp$, $\top$ using representatives, e.g

$$[\varphi]_{\sim_T} \wedge [\psi]_{\sim_T} = [\varphi \wedge \psi]_{\sim_T}$$

The *Lindenbaum-Tarski algebra* for T is

$$AV^{\mathbb{P}}(T) = \langle \mathrm{VF}_{\mathbb{P}}/{\sim_T}, \neg, \wedge, \vee, \perp, \top \rangle$$

Since $\varphi \sim_T \psi \;\Leftrightarrow\; M(T,\varphi) = M(T,\psi)$, it follows that the mapping $h\colon \mathrm{VF}_{\mathbb{P}}/{\sim_T} \to \mathcal{P}(M(T))$ defined by $h([\varphi]_{\sim_T}) = M(T,\varphi)$ is a (well-defined) injective function, and satisfies the following properties. Moreover, h is *surjective* if $M(T)$ is *finite*.

$$h(\neg[\varphi]_{\sim_T}) = M(T) \setminus M(T,\varphi)$$
$$h([\varphi]_{\sim_T} \wedge [\psi]_{\sim_T}) = M(T,\varphi) \cap M(T,\psi)$$
$$h([\varphi]_{\sim_T} \vee [\psi]_{\sim_T}) = M(T,\varphi) \cup M(T,\psi)$$
$$h([\perp]_{\sim_T}) = \emptyset, \quad h([\top]_{\sim_T}) = M(T)$$

Corollary If T is a consistent theory over a finite $\mathbb{P}$, then $AV^{\mathbb{P}}(T)$ is a Boolean algebra *isomorphic* via h to the (finite) algebra of sets $\underline{\mathcal{P}}(M(T))$.

Analysis of theories over finite languages

Let T be a consistent theory over $\mathbb{P}$ where $|\mathbb{P}| = n \in \mathbb{N}^+$ and $m = |M^{\mathbb{P}}(T)|$. Then the number of (mutually) inequivalent

- propositions (or theories) over $\mathbb{P}$ is 2^{2^n},
- propositions over $\mathbb{P}$ that are valid (contradictory) in T is $2^{2^n - m}$,
- propositions over $\mathbb{P}$ that are independent in T is $2^{2^n} - 2 \cdot 2^{2^n - m}$,
- simple extensions of T is 2^m, out of which 1 is inconsistent,
- complete simple extensions of T is m.

And the number of (mutually) T-inequivalent

- propositions over $\mathbb{P}$ is 2^m,
- propositions over $\mathbb{P}$ that are valid (contradictory) (in T) is 1,
- propositions over $\mathbb{P}$ that are independent (in T) is $2^m - 2$.

Proof Using the bijection of $\mathrm{VF}_{\mathbb{P}}/\!\sim$ resp. $\mathrm{VF}_{\mathbb{P}}/\!\sim_T$ with $\mathcal{P}(M(\mathbb{P}))$ resp. $\mathcal{P}(M^{\mathbb{P}}(T))$ it suffices to count the corresponding sets of models. $\square$

Formal proof systems

We formalize precisely the notion of proof as a syntactical procedure.

In *(standard)* formal proof systems,

- a proof is a finite object, built from axioms of a given theory,
- $T \vdash \varphi$ denotes that φ is *provable* from a theory T,
- if a formula has a proof, it can be found *"algorithmically"* [assuming that T is *"given algorithmically"*],

We usually require that a formal proof system is

- *sound*, i.e., every formula provable from a theory T is also valid in T,
- *complete*, i.e., every formula valid in T is also provable from T.

Examples of formal proof systems (calculi): tableaux methods, *Hilbert systems, Gentzen systems, natural deduction systems*.

Table of Contents

Method of analytic tableaux

We assume that the language is fixed and countable, i.e. the set $\mathbb{P}$ of propositional letters is countable. Then every theory over $\mathbb{P}$ is countable.

Main features of the tableau method *(informally)*

- a tableau for a formula φ is a binary labeled tree representing systematic search for *counterexample* to φ, i.e. a model of theory is which φ is false,
- a formula is proved if every branch in tableau 'fails', i.e counterexample was not found. In this case the (systematic) tableau will be finite,
- if a counterexample exists, there will be a branch in a (finished) tableau that provides us with this counterexample, but this branch can be infinite.

Introductory examples

- $F(((p \to q) \to p) \to p)$
  - $T((p \to q) \to p)$
    - Fp
      - $T((p \to q) \to p)$
        - $F(p \to q)$
          - $F(p \to q)$
            - Tp
              - Fq
                - $\otimes$
        - Tp
          - $\otimes$

- $F((\neg q \vee p) \to p)$
  - $T(\neg q \vee p)$
    - Fp
      - $T(\neg q \vee p)$
        - $T(\neg q)$
          - $T(\neg q)$
            - Fq
        - Tp
          - $\otimes$

Explanation of the examples

Nodes in tableaux are labeled by *entries*. An entry is a formula with a *sign* T / F representing an assumption that the formula is true / false in some model. If this assumption is correct, then it is correct also for all the entries *in some branch below* this entry.

In both examples we have finished (systematic) tableaux from no axioms.

- On the left, there is a *tableau proof* for $((p \to q) \to p) \to p$. All branches *"failed"*, denoted by $\otimes$, as each contains a pair $T\varphi$, $F\varphi$ for some φ *(counterexample was not found)*. Thus the formula is provable, we write:
$$\vdash ((p \to q) \to p) \to p$$
- On the right, there is a (finished) tableau for $(\neg q \vee p) \to p$. The left branch did not *"fail"* and is finished (all its entries were considered), *it provides us with a counterexample* $v(p) = v(q) = 0$.

Atomic tableaux

An *atomic tableau* is one of the following trees (labeled by entries), where p is any propositional letter and φ, ψ are any propositions.

<table>
<tr>
<td>Tp</td>
<td>Fp</td>
<td>$\begin{array}{c} T(\varphi \wedge \psi) \\ | \\ T\varphi \\ | \\ T\psi \end{array}$</td>
<td>$\begin{array}{cc} \multicolumn{2}{c}{F(\varphi \wedge \psi)} \\ / & \backslash \\ F\varphi & F\psi \end{array}$</td>
<td>$\begin{array}{cc} \multicolumn{2}{c}{T(\varphi \vee \psi)} \\ / & \backslash \\ T\varphi & T\psi \end{array}$</td>
<td>$\begin{array}{c} F(\varphi \vee \psi) \\ | \\ F\varphi \\ | \\ F\psi \end{array}$</td>
</tr>
<tr>
<td>$\begin{array}{c} T(\neg\varphi) \\ | \\ F\varphi \end{array}$</td>
<td>$\begin{array}{c} F(\neg\varphi) \\ | \\ T\varphi \end{array}$</td>
<td>$\begin{array}{cc} \multicolumn{2}{c}{T(\varphi \to \psi)} \\ / & \backslash \\ F\varphi & T\psi \end{array}$</td>
<td>$\begin{array}{c} F(\varphi \to \psi) \\ | \\ T\varphi \\ | \\ F\psi \end{array}$</td>
<td>$\begin{array}{cc} \multicolumn{2}{c}{T(\varphi \leftrightarrow \psi)} \\ / & \backslash \\ T\varphi & F\varphi \\ | & | \\ T\psi & F\psi \end{array}$</td>
<td>$\begin{array}{cc} \multicolumn{2}{c}{F(\varphi \leftrightarrow \psi)} \\ / & \backslash \\ T\varphi & F\varphi \\ | & | \\ F\psi & T\psi \end{array}$</td>
</tr>
</table>

All tableaux will be formally defined using atomic tableaux and rules how to expand them.

Tableaux

A *finite tableau* is a binary tree labeled with entries defined inductively:

(i) every atomic tableau is a finite tableau,

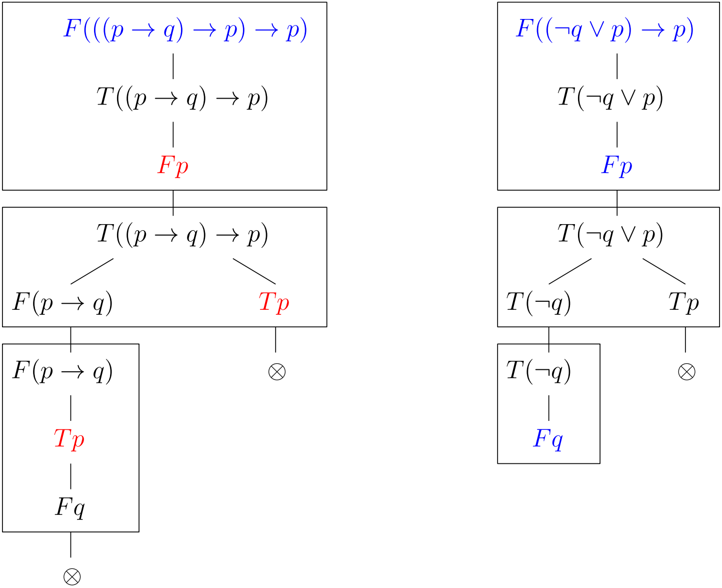
(ii) if E is an entry on a branch B in a finite tableau τ and τ' is obtained from τ by adjoining the atomic tableaux for E at the end of the branch B, then τ' is also a finite tableau,

(iii) every finite tableau is formed by a finite number of steps (i), (ii).

A *tableau* is a sequence $\tau_0, \tau_1, \ldots, \tau_n, \ldots$ (finite or infinite) of finite tableaux such that τ_{n+1} is formed from τ_n by an application of (ii), formally $\tau = \cup \tau_n$.

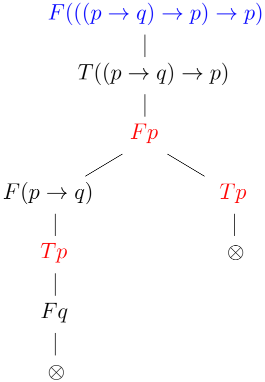
Remark It is not specified how to choose the entry E and the branch B for expansion. This will be specified in systematic tableaux.

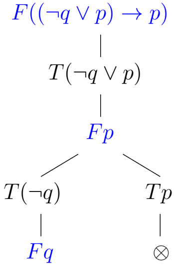
Construction of tableaux



Convention

We will not write the entry that is expanded again on the branch.





Beware *We cannot use this convention later in tableau method for predicate logic; the repeated entries will be needed again.*

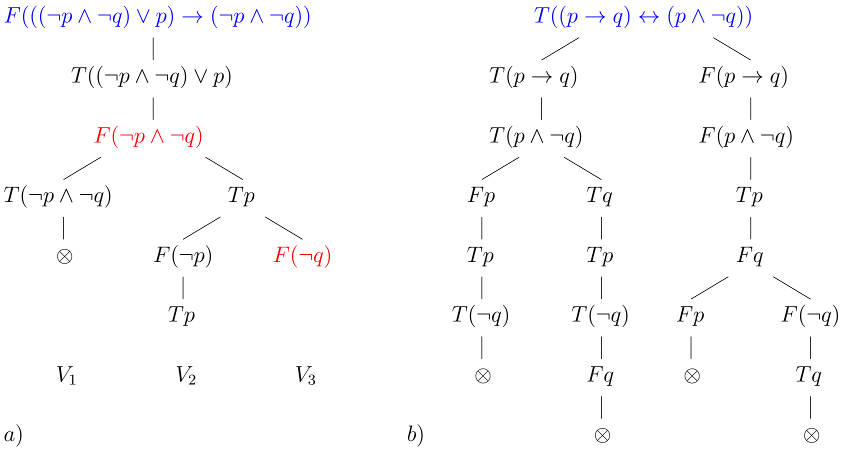
Tableau proofs

Let E be an entry on a branch B in a tableau τ. We say that

- the entry E is *reduced* on B if it occurs on B as the root of an atomic tableau, i.e., it was already expanded on B during the construction of τ,
- the branch B is *contradictory* if it contains entries $T\varphi$ and $F\varphi$ for some proposition φ, otherwise B is *noncontradictory*. The branch B is *finished* if it is contradictory or every entry on B is already reduced on B,
- the tableau τ is *finished* if every branch in τ is finished, and τ is *contradictory* if every branch in τ is contradictory.

A *tableau proof* (*proof by tableau*) of φ is a contradictory tableau with the root entry $F\varphi$; φ is *(tableau) provable*, denoted by $\vdash \varphi$, if it has a tableau proof. Similarly, a *refutation* of φ by *tableau* is a contradictory tableau with the root entry $T\varphi$; φ is *(tableau) refutable* if it has a refutation by tableau, i.e. $\vdash \neg\varphi$.

Examples



a) $F(\neg p \wedge \neg q)$ not reduced on V_1, V_1 contradictory, V_2 finished, V_3 unfinished,

b) a (tableau) refutation of φ: $(p \rightarrow q) \leftrightarrow (p \wedge \neg q)$, i.e. $\vdash \neg\varphi$.

Tableau from a theory

How to add axioms of a given theory into a proof? A *finite tableau from a theory* T is given by an additional rule

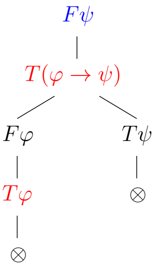
(ii)' if B is a branch of a finite tableau (from T) and $\varphi \in T$, then by adjoining $T\varphi$ at the end of B we get (again) a finite tableau from T.

We generalize other definitions by appending *"from T"*.

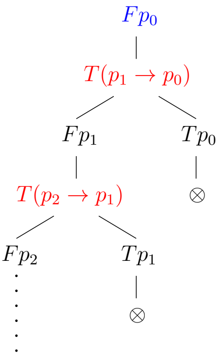
- a *tableau from T* is a sequence $\tau_0, \tau_1, \ldots, \tau_n, \ldots$ of finite tableaux from T such that τ_{n+1} is formed from τ_n applying (ii) or (ii)', formally $\tau = \cup\tau_n$,
- a *tableau proof* of φ *from T* is a contradictory tableaux from T with $F\varphi$ in the root. $T \vdash \varphi$ denotes that φ is *(tableau) provable from T*.
- a *refutation* of φ by a *tableau from T* is a contradictory tableau from T with the root entry $T\varphi$.

Unlike in previous definitions, a branch B of a tableau from T is *finished*, if it is contradictory, or every entry on B is already reduced on B and, moreover, B contains $T\varphi$ for every axiom $\varphi \in T$.

Examples of tableaux from theories



$a)$



$b)$

a) A tableau proof of ψ from $T = \{\varphi, \varphi \to \psi\}$, so $T \vdash \psi$.

b) A finished tableau with the root Fp_0 from $T = \{p_{n+1} \to p_n \mid n \in \mathbb{N}\}$. All branches are finished, the leftmost branch is noncontradictory and infinite. It provides us the (only) model of T in which p_0 is false.

Systematic tableaux

We describe a systematic construction that leads to a finished tableau.

Let R be an entry and $T = \{\varphi_0, \varphi_1, \dots\}$ be a (possibly infinite) theory.

(1) We take the atomic tableau for R as τ_0. Proceed as follows:

(2) Let E be the leftmost entry in the smallest level as possible of the tableau τ_n s.t. E is not reduced on some noncontradictory branch through E.

(3) Let τ_n' be the tableau obtained from τ_n by adjoining the atomic tableau for E to every noncontradictory branch through E. (If E does not exist, we take $\tau_n' = \tau_n$.)

(4) Let τ_{n+1} be the tableau obtained from τ_n' by adjoining $T\varphi_n$ to every noncontradictory branch that does not contain $T\varphi_n$ yet. (If φ_n does not exist, we take $\tau_{n+1} = \tau_n'$.)

The *systematic tableau* from T for the entry R is the result of the above construction, i.e. $\tau = \bigcup_{n\geq 0} \tau_n$.

Systematic tableau is finished

Proposition *Every systematic tableau is finished.*

Proof Let $\tau = \cup \tau_n$ be a systematic tableau from $T = \{\varphi_0, \varphi_1, \dots\}$ with root entry R.

- If a branch is noncontradictory in τ, its prefix in every τ_n is noncontradictory as well.
- If an entry E is unreduced on some branch in τ, it is unreduced on its prefix in every τ_n as well (assuming E occurs in this prefix).
- There are only finitely many entries in τ in levels up to the level of E.
- Thus, if E was unreduced on some noncontradictory branch in τ, it would be considered in some step (2) and reduced by step (3).
- By step (4) every $\varphi_n \in T$ will be (no later than) in τ_{n+1} on every noncontradictory branch.
- Hence in the systematic tableau τ, all branches are finished. $\square$

Finiteness of proofs

König's Lemma *Every infinite, finitely branching tree contains an infinite branch.*

Proposition *For every contradictory tableau $\tau = \cup\tau_n$ there is some n such that τ_n is a contradictory finite tableau.*

Proof Let S be the set of nodes in τ that have no pair of contradictory entries $T\varphi$, $F\varphi$ amongst their predecessors.

- If S was infinite, then by König's lemma, the subtree of τ induced by S would contain an infinite brach, and thus τ would not be contradictory.
- Since S is finite, for some m all nodes of S belong to levels up to m.
- Thus every node in level $m+1$ has a pair of contradictory entries amongst its predecessors.
- Let n be such that τ_n agrees with τ at least up to the level $m+1$.
- Then every branch in τ_n is contradictory. □

Corollary *If a systematic tableau (from a theory) is a proof, it is finite.*

Proof In its construction, we extend only noncontradictory branches. □

NAIL062 Propositional & Predicate Logic: Lecture 5

Slides by Petr Gregor with minor modifications by Jakub Bulín

November 2, 2020

Table of Contents

Soundness

We say the an entry E *agrees* with an assignment v, if E is $T\varphi$ and $\overline{v}(\varphi) = 1$, or if E is $F\varphi$ and $\overline{v}(\varphi) = 0$. A branch B *agrees* with v, if every entry on V agrees with v.

Lemma *Let v be a model of a theory T that agrees with the root entry of a tableau $\tau = \cup\tau_n$ from T. Then τ contains a branch that agrees with v.*

Proof By induction we find a sequence $B_0, B_1, \ldots$ so that for every n, B_n is a branch in τ_n agreeing with v and B_n is contained in B_{n+1}.

- By considering all atomic tableaux we verify the base of induction.
- If τ_{n+1} is obtained from τ_n without extending B_n, we put $B_{n+1} = B_n$.
- If τ_{n+1} is obtained from τ_n by adjoining $T\varphi$ to B_n for some $\varphi \in T$, then let B_{n+1} be this branch. Since v is a model of φ, B_{n+1} agrees with v.
- Otherwise τ_{n+1} is obtained from τ_n by adjoining the atomic tableau for some entry E on B_n to the end of B_n. As E agrees with v and atomic tableaux are verified, B_n we can extend to B_{n+1} as well. $\square$

Theorem on soundness

We will show that the tableau method in propositional logic is sound.

Theorem *For every theory T and proposition φ, if φ is tableau provable from T, then φ is valid in T, i.e. $T \vdash \varphi \;\Rightarrow\; T \models \varphi$.*

Proof

- Let φ be tableau provable from a theory T, i.e. there is a contradictory tableau τ from T with the root entry $F\varphi$.
- Suppose for a contradiction that φ is not valid in T, i.e. there exists a model v of the theory T if which φ is false (a counterexample).
- Since the root entry $F\varphi$ agrees with v, by the previous lemma, there is a branch in the tableau τ that agrees with v.
- But this is impossible, since every branch of τ is contradictory, i.e. it contains a pair of entries $T\psi$, $F\psi$ for some ψ. $\square$

Completeness

A noncontradictory branch in a finished tableau gives us a counterexample.

Lemma *A noncontradictory branch B of a finished tableau τ agrees with the following assignment:*

$$v(p) = \begin{cases} 1 & \text{if } Tp \text{ occurs on } B \\ 0 & \text{otherwise} \end{cases}$$

Proof By induction on the structure of formulas in entries occurring on B.

- For an entry Tp on B, where p is a letter, we have $\overline{v}(p) = 1$ by defn.
- For an entry Fp on B, Tp in not on B since B is noncontradictory, thus $\overline{v}(p) = 0$ by definition of v.
- For an entry $T(\varphi \wedge \psi)$ on B, we have $T\varphi$ and $T\psi$ on B since τ is finished. By induction, we have $\overline{v}(\varphi) = \overline{v}(\psi) = 1$, and thus $\overline{v}(\varphi \wedge \psi) = 1$.
- For an entry $F(\varphi \wedge \psi)$ on B, we have $F\varphi$ or $F\psi$ on B since τ is finished. By induction, we have $\overline{v}(\varphi) = 0$ or $\overline{v}(\psi) = 0$, and thus $\overline{v}(\varphi \wedge \psi) = 0$.
- For other entries similarly as in previous two cases. $\square$

Theorem on completeness

We will show that the tableau method in propositional logic is complete.

Theorem *For every theory T and proposition φ, if φ is valid in T, then φ is tableau provable from T, i.e. $T \models \varphi \;\Rightarrow\; T \vdash \varphi$.*

Proof Let φ be valid in T. We will show that an arbitrary finished tableau (e.g. *systematic*) τ from theory T with the root entry $F\varphi$ is contradictory.

- If not, let B be some noncontradictory branch in τ.
- By the previous lemma, there exists an assignment v such that B agrees with v, in particular in the root entry $F\varphi$, i.e. $\overline{v}(\varphi) = 0$.
- Since B is finished, it contains $T\psi$ for every $\psi \in T$.
- Thus v is a model of theory T (since B agrees with v).
- But this contradicts the assumption that φ is valid in T.

Hence the tableau τ is a proof of φ from T. $\square$

Properties of theories

We introduce syntactic variants of previous semantically defined notions.

Let T be a theory over $\mathbb{P}$. If φ is provable from T, we say that φ is a *theorem* of T. The set of theorems of T is denoted by

$$\mathrm{Thm}^{\mathbb{P}}(T) = \{\varphi \in \mathrm{VF}_{\mathbb{P}} \mid T \vdash \varphi\}.$$

We say that a theory T is

- *inconsistent* if $T \vdash \perp$, otherwise T is *consistent*,
- *complete* if it is consistent and every proposition is provable or refutable from T, i.e. $T \vdash \varphi$ or $T \vdash \neg\varphi$ for every $\varphi \in \mathrm{VF}_{\mathbb{P}}$,
- an *extension* of a theory T' over $\mathbb{P}'$ if $\mathbb{P}' \subseteq \mathbb{P}$ and $\mathrm{Thm}^{\mathbb{P}'}(T') \subseteq \mathrm{Thm}^{\mathbb{P}}(T)$; we say that an extension T of a theory T' is *simple* if $\mathbb{P} = \mathbb{P}'$; and *conservative* if $\mathrm{Thm}^{\mathbb{P}'}(T') = \mathrm{Thm}^{\mathbb{P}}(T) \cap \mathrm{VF}_{\mathbb{P}'}$,
- *equivalent* with a theory T' if T is an extension of T' and vice-versa.

Corollaries

From the soundness and completeness of the tableau method it follows that these syntactic definitions agree with their semantic variants.

Corollary *For every theory T and propositions φ, ψ over $\mathbb{P}$,*

- $T \vdash \varphi$ *if and only if* $T \models \varphi$,
- $\mathrm{Thm}^{\mathbb{P}}(T) = \theta^{\mathbb{P}}(T)$,
- *T is inconsistent if and only if T is unsatisfiable, i.e. it has no model,*
- *T is complete if and only if T is semantically complete, i.e. it has a single model,*
- $T, \varphi \vdash \psi$ *if and only if* $T \vdash \varphi \to \psi$ *(Deduction theorem).*

Remark Deduction theorem can be proved directly by transformations of tableaux.

Theorem on compactness

Theorem *A theory T has a model iff every finite subset of T has a model.*

Proof 1 The implication from left to right is obvious. If T has no model, then it is inconsistent, i.e. $\perp$ is provable by a systematic tableau τ from T. Since τ is finite, $\perp$ is provable from some finite $T' \subseteq T$, i.e. T' has no model. $\square$

Remark *This proof is based on finiteness of proofs, soundness, and completeness. We present an alternative proof (applying König's lemma).*

Proof 2 Let $T = \{\varphi_i \mid i \in \mathbb{N}\}$. Consider a tree S on (certain) finite binary strings σ ordered by being a prefix. We put $\sigma \in S$ if and only if there exists an assignment v with prefix σ such that $v \models \varphi_i$ for every $i \leq \mathrm{lth}(\sigma)$.

Observation *S has an infinite branch if and only if T has a model.*

Since $\{\varphi_i \mid i \in n\} \subseteq T$ has a model for every $n \in \mathbb{N}$, every level in S is nonempty. Thus S is infinite and moreover binary, hence by König's lemma, S contains an infinite branch. $\square$

Application of compactness

A graf (V, E) is *k-colorable* if there exists $c\colon V \to k$ such that $c(u) \neq c(v)$ for every edge $\{u, v\} \in E$.

Theorem *A countably infinite graph $G = (V, E)$ is k-colorable if and only if every finite subgraph of G is k-colorable.*

Proof The implication $\Rightarrow$ is obvious. Assume that every finite subgraph of G is k-colorable. Consider $\mathbb{P} = \{p_{u,i} \mid u \in V, i \in k\}$ and a theory T with axioms

$$
\begin{aligned}
p_{u,0} \vee \cdots \vee p_{u,k-1} \quad & \text{for every } u \in V,\\
\neg(p_{u,i} \wedge p_{u,j}) \quad & \text{for every } u \in V, i < j < k,\\
\neg(p_{u,i} \wedge p_{v,i}) \quad & \text{for every } \{u, v\} \in E, i < k.
\end{aligned}
$$

Then G is k-colorable if and only if T has a model. By compactness, it suffices to show that every finite $T' \subseteq T$ has a model. Let G' be the subgraph of G induced by vertices u such that $p_{u,i}$ appears in T' for some i. Since G' is k-colorable by the assumption, the theory T' has a model. $\square$

Table of Contents

Hilbert's calculus

- basic connectives: $\neg$, $\rightarrow$ (others can be defined from them)
- *logical axioms* (schemes of axioms):

$$
\begin{array}{ll}
(i) & \varphi \rightarrow (\psi \rightarrow \varphi) \\
(ii) & (\varphi \rightarrow (\psi \rightarrow \chi)) \rightarrow ((\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \chi)) \\
(iii) & (\neg\varphi \rightarrow \neg\psi) \rightarrow (\psi \rightarrow \varphi)
\end{array}
$$

  where φ, ψ, χ are any propositions (of a given language).
- *a rule of inference*:

$$\frac{\varphi,\ \varphi \rightarrow \psi}{\psi} \qquad \text{(modus ponens)}$$

A *proof* (in *Hilbert-style*) of a formula φ from a theory T is a finite sequence $\varphi_0, \ldots, \varphi_n = \varphi$ of formulas such that for every $i \leq n$

- φ_i is a logical axiom or $\varphi_i \in T$ (an axiom of the theory), or
- φ_i can be inferred from the previous formulas applying a rule of inference.

Remark *Choice of axioms and inference rules differs in various Hilbert-style proof systems.*

Example and soundness

A formula φ is *provable* from T if it has a proof from T, denoted by $T \vdash_H \varphi$. If $T = \emptyset$, we write $\vdash_H \varphi$.

Example: for $T = \{\neg\varphi\}$ we have $T \vdash_H \varphi \to \psi$ for every ψ.

1)	$\neg\varphi$	an axiom of T
2)	$\neg\varphi \to (\neg\psi \to \neg\varphi)$	a logical axiom (*i*)
3)	$\neg\psi \to \neg\varphi$	by modus ponens from 1), 2)
4)	$(\neg\psi \to \neg\varphi) \to (\varphi \to \psi)$	a logical axiom (*iii*)
5)	$\varphi \to \psi$	by modus ponens from 3), 4)

Theorem *For every theory T and formula φ, $T \vdash_H \varphi \;\Rightarrow\; T \models \varphi$.*

Proof

- If φ is an axiom (logical or from T), then $T \models \varphi$ (logical axioms are tautologies),
- if $T \models \varphi$ and $T \models \varphi \to \psi$, then $T \models \psi$, i.e. modus ponens is sound,
- thus every formula in a proof from T is valid in T. □

Remark *The completeness theorem holds as well: $T \models \varphi \Rightarrow T \vdash_H \varphi$.*

Table of Contents

Resolution method - introduction

Main features of the resolution method (*informally*)

- the underlying method of many systems, e.g. Prolog interpreters, SAT solvers, automated reasoning (deduction/verification) systems, ...
- assumes input in CNF (in general, *"expensive"* transformation),
- works under set representation (clausal form) of formulas,
- has a single rule, so called resolution rule,
- has no explicit axioms (or atomic tableaux) but certain axioms are incorporated *"inside"* via various formatting rules,
- is a *refutation* procedure, similarly as the tableau method; that is, it tries to show that a given formula (or theory) is unsatisfiable,
- has several refinements, e.g. with specific conditions on when the resolution rule may be applied.

Set representation (clausal from) of CNF formulas

- A *literal* l is a prop. letter or its negation. $\bar{l}$ is its *complementary* literal.
- A *clause* C is a finite set of literals (*"forming disjunction"*). The empty clause, denoted by $\square$, is never satisfied (has no satisfied literal).
- A *formula* S is a (possibly infinite) set of clauses (*"forming conjunction"*). An empty formula $\emptyset$ is always satisfied (it has no unsatisfied clause). Infinite formulas represent infinite theories (as conjunction of axioms).
- A (*partial*) *assignment* $\mathcal{V}$ is a consistent set of literals, i.e. not containing any pair of complementary literals. An assignment $\mathcal{V}$ is *total* if it contains a positive or negative literal for each prop. letter.
- $\mathcal{V}$ *satisfies* S, denoted by $\mathcal{V} \models S$, if $C \cap \mathcal{V} \neq \emptyset$ for every $C \in S$.

$((\neg p \vee q) \wedge (\neg p \vee \neg q \vee r) \wedge (\neg r \vee \neg s) \wedge (\neg t \vee s) \wedge s)$ is represented by

$$S = \{\{\neg p, q\}, \{\neg p, \neg q, r\}, \{\neg r, \neg s\}, \{\neg t, s\}, \{s\}\} \quad \text{and}$$

$$\mathcal{V} \models S \quad \text{for} \quad \mathcal{V} = \{s, \neg r, \neg p\}$$

Resolution rule

Let C_1, C_2 be clauses with $l \in C_1$, $\bar{l} \in C_2$ for some literal l. Then from C_1 and C_2 infer through the literal l the clause C, called a *resolvent*, where

$$C = (C_1 \setminus \{l\}) \cup (C_2 \setminus \{\bar{l}\}).$$

Equivalently, if $\sqcup$ means union of disjoint sets,

$$\frac{C_1' \sqcup \{l\}, C_2' \sqcup \{\bar{l}\}}{C_1' \cup C_2'}$$

For example, from $\{p, q, r\}$ and $\{\neg p, \neg q\}$ we can infer $\{q, \neg q, r\}$ or $\{p, \neg p, r\}$.

Observation *The resolution rule is sound; that is, for every assignment $\mathcal{V}$*

$$\mathcal{V} \models C_1 \text{ and } \mathcal{V} \models C_2 \quad \Rightarrow \quad \mathcal{V} \models C.$$

Remark *The resolution rule is a special case of the (so called) cut rule*

$$\frac{\varphi \vee \psi, \ \neg\varphi \vee \chi}{\psi \vee \chi}$$

where φ, ψ, χ are arbitrary formulas.

Resolution proof

- A *resolution proof* (*deduction*) of a clause C from a formula S is a finite sequence $C_0, \ldots, C_n = C$ such that for every $i \leq n$, we have $C_i \in S$ or C_i is a resolvent of some previous clauses,
- a clause C is (resolution) *provable* from S, denoted by $S \vdash_R C$, if it has a resolution proof from S,
- a (resolution) *refutation* of a formula S is a resolution proof of $\square$ from S,
- S is (resolution) *refutable* if $S \vdash_R \square$.

Theorem (soundness) *If S is resolution refutable, then S is unsatisfiable.*

Proof Let $S \vdash_R \square$. If it was $\mathcal{V} \models S$ for some assignment $\mathcal{V}$, from the soundness of the resolution rule we would have $\mathcal{V} \models \square$, impossible. $\square$

Resolution trees and closures

A *resolution tree* of a clause C from formula S is finite binary tree with nodes labeled by clauses so that

- (i) the root is labeled C,
- (ii) the leaves are labeled with clauses from S,
- (iii) every inner node is labeled with a resolvent of the clauses in his sons.

Observation *C has a resolution tree from S if and only if $S \vdash_R C$.*

A *resolution closure* $\mathcal{R}(S)$ of a formula S is the smallest set satisfying

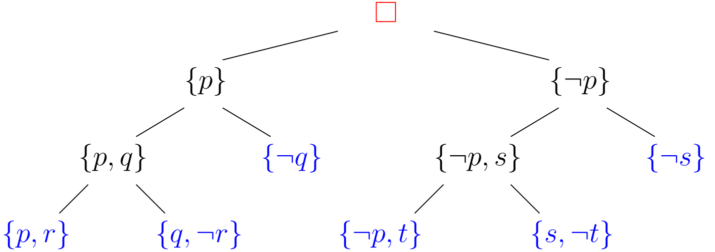
- (i) $C \in \mathcal{R}(S)$ for every $C \in S$,
- (ii) if $C_1, C_2 \in \mathcal{R}(S)$ and C is a resolvent of C_1, C_2, then $C \in \mathcal{R}(S)$.

Observation *$C \in \mathcal{R}(S)$ if and only if $S \vdash_R C$.*

Remark *All notions on resolution proofs can therefore be equivalently introduced in terms of resolution trees or resolution closures.*

Example

Formula $((p \vee r) \wedge (q \vee \neg r) \wedge (\neg q) \wedge (\neg p \vee t) \wedge (\neg s) \wedge (s \vee \neg t))$ is unsatisfiable since for $S = \{\{p, r\}, \{q, \neg r\}, \{\neg q\}, \{\neg p, t\}, \{\neg s\}, \{s, \neg t\}\}$ we have $S \vdash_R \square$.



The resolution closure of S (*the closure of S under resolution*) is

$$\mathcal{R}(S) = \{\{p, r\}, \{q, \neg r\}, \{\neg q\}, \{\neg p, t\}, \{\neg s\}, \{s, \neg t\}, \{p, q\}, \{\neg r\}, \{r, t\}, \{q, t\}, \{\neg t\}, \{\neg p, s\}, \{r, s\}, \{t\}, \{q\}, \{q, s\}, \square, \{\neg p\}, \{p\}, \{r\}, \{s\}\}$$

NAIL062 Propositional & Predicate Logic: Lecture 6

Slides by Petr Gregor with minor modifications by Jakub Bulín

November 9, 2020

Reduction by substitution

For a formula S and a literal l, we define $S^l = \{C \setminus \{\bar{l}\} \mid l \notin C \in S\}$ (cf. unit propagation)

Observation

- S^l is equivalent to a formula obtained from S by substituting the constant $\top$ (true, 1) for all literals l and the constant $\bot$ (false, 0) for all literals $\bar{l}$ in S,
- Neither l nor $\bar{l}$ occurs in (the clauses of) S^l.
- if $\{\bar{l}\} \in S$, then $\square \in S^l$.

Lemma *S is satisfiable if and only if S^l or $S^{\bar{l}}$ is satisfiable.*

Proof ($\Rightarrow$) Let $\mathcal{V} \models S$ for some $\mathcal{V}$ and assume (w.l.o.g.) that $\bar{l} \notin \mathcal{V}$. Then $\mathcal{V} \models S^l$ as for $l \notin C \in S$ we have $\mathcal{V} \setminus \{l, \bar{l}\} \models C$ and thus $\mathcal{V} \models C \setminus \{\bar{l}\}$.

($\Leftarrow$) On the other hand, assume (w.l.o.g.) that $\mathcal{V} \models S^l$ for some $\mathcal{V}$. Since neither l nor $\bar{l}$ occurs in S^l, we have $\mathcal{V}' \models S^l$ for $\mathcal{V}' = (\mathcal{V} \setminus \{\bar{l}\}) \cup \{l\}$. Then $\mathcal{V}' \models S$ since for $C \in S$ containing l we have $l \in \mathcal{V}'$ and for $C \in S$ not containing l we have $\mathcal{V}' \models (C \setminus \{\bar{l}\}) \in S^l$. $\square$

Tree of reductions

Step by step reductions of literals can be represented in a binary tree.

$$\begin{array}{ccc} & S = \{\{p\}, \{\neg q\}, \{\neg p, \neg q\}\} & \\ S^{p} = \{\{\neg q\}\} & & S^{\overline{p}} = \{\square, \{\neg q\}\} \\ S^{pq} = \{\square\} \quad S^{p\overline{q}} = \emptyset & & \end{array}$$

Corollary *S is unsatisfiable if and only if every branch contains $\square$.*

Remarks *Since S can be infinite over a countable language, this tree can be infinite. However, if S is unsatisfiable, by the compactness theorem there is a finite $S' \subseteq S$ that is unsatisfiable. Thus after reduction of all literals occurring in S', there will be $\square$ in every branch after finitely many steps.*

Completeness of resolution

Theorem *If a finite S is unsatisfiable, then it is resolution refutable, i.e. $S \vdash_R \square$.*

Proof Show that $S \vdash_R \square$ by induction on the number of variables in S.

- If unsatisfiable S has no variable, it is $S = \{\square\}$ and thus $S \vdash_R \square$,
- Let l be a literal occurring in S. By Lemma, S^l and $S^{\bar{l}}$ are unsatisfiable.
- Since S^l and $S^{\bar{l}}$ have less variables than S, by induction there exist resolution trees T^l and $T^{\bar{l}}$ for derivation of $\square$ from S^l resp. $S^{\bar{l}}$.
- If every leaf of T^l is in S, then T^l is a resolution tree of $\square$ from S, $S \vdash_R \square$.
- Otherwise, by appending the literal $\bar{l}$ to every leaf of T^l that is not in S, (and to all predecessors) we obtain a resolution tree of $\{\bar{l}\}$ from S.
- Similarly, we get a resolution tree $\{l\}$ from S by appending l in the tree $T^{\bar{l}}$.
- By resolution of roots $\{\bar{l}\}$ and $\{l\}$ we get a res. tree of $\square$ from S. $\square$

Corollary *If S is unsatisfiable, then it is resolution refutable, i.e. $S \vdash_R \square$.*

Proof Follows from the previous theorem by compactness.

Table of Contents

Linear resolution - introduction

The resolution method can be significantly refined.

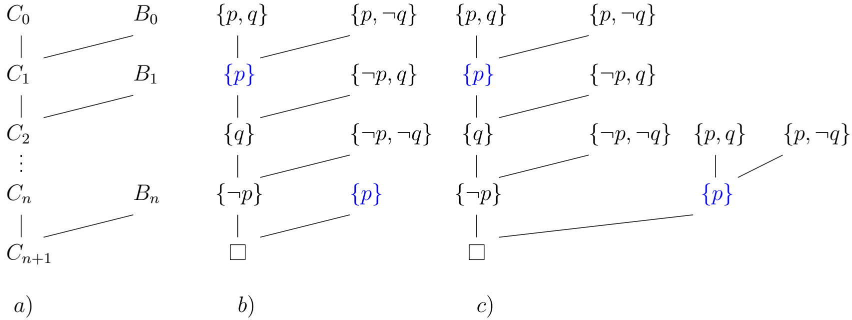
- A *linear proof* of a clause C from a formula S is a finite sequence of pairs $(C_0, B_0), \ldots, (C_n, B_n)$ such that $C_0 \in S$ and for every $i \leq n$
  - *i*) $B_i \in S$ or $B_i = C_j$ for some $j < i$, and
  - *ii*) C_{i+1} is a resolvent of C_i and B_i where $C_{n+1} = C$.
- C_0 is called a *starting* clause, C_i a *central* clause, B_i a *side* clause.
- C is *linearly provable* from S, $S \vdash_L C$, if it has a linear proof from S.
- A *linear refutation* of S is a linear proof of $\square$ from S.
- S is *linearly refutable* if $S \vdash_L \square$.

Observation (soundness) *If S is linearly refutable, it is unsatisfiable.*

Proof Every linear proof can be transformed to a (general) resolution proof.

Remark *The completeness is preserved as well (proof omitted here).*

Example of linear resolution



- (a) a general form of linear resolution,
- (b) for $S = \{\{p,q\},\{p,\neg q\},\{\neg p,q\},\{\neg p,\neg q\}\}$ we have $S \vdash_L \square$,
- (c) a transformation of a linear proof to a (general) resolution proof.

Table of Contents

LI-resolution

Linear resolution can be further refined for Horn formulas as follows.

- a *Horn clause* is a clause containing at most one positive literal,
- a *Horn formula* is a (possibly infinite) set of Horn clauses,
- a *fact* is a (Horn) clause $\{p\}$ where p is a positive literal,
- a *rule* is a (Horn) clause with exactly one positive literal and at least one negative literal. Rules and facts are *program clauses*,
- a *goal* is a nonempty (Horn) clause with only negative literals.

Observation *If a Horn formula S is unsatisfiable and $\square \notin S$, it contains some fact and some goal.*

Proof If S does not contain any fact (goal), it is satisfied by the assignment of all propositional variables to 0 (resp. to 1). $\square$

A *linear input resolution* (*LI-resolution*) from a formula S is a linear resolution from S in which every side clause B_i is from the (input) formula S. We write $S \vdash_{LI} C$ to denote that C is provable by LI-resolution from S.

Completeness of LI-resolution for Horn formulas

Theorem *If T is a satisfiable Horn formula but $T \cup \{G\}$ is unsat. for some goal G, then $\square$ has a LI-resolution from $T \cup \{G\}$ with starting clause G.*

Proof By the compactness theorem we may assume that T is finite.

- We proceed by induction on the number of variables in T.
- By Observation, T contains a fact $\{p\}$ for some variable p.
- By Lemma, $T' = (T \cup \{G\})^p = T^p \cup \{G^p\}$ is unsatisfiable where $G^p = G \setminus \{\overline{p}\}$.
- If $G^p = \square$, we have $G = \{\overline{p}\}$ and thus $\square$ is a resolvent of G and $\{p\} \in T$.
- Otherwise, since T^p is satisfiable (by the assignment satisfying T) and has less variables than T, by induction assumption, there is an LI-resolution of $\square$ from T' starting with G^p.
- By appending the literal $\overline{p}$ to all leaves that are not in $T \cup \{G\}$ (and nodes below) we obtain an LI-resolution of $\{\overline{p}\}$ from $T \cup \{G\}$ that starts with G.
- By an additional resolution step with the fact $\{p\} \in T$ we resolve $\square$.

$\square$

Example of LI-resolution

$$T = \{\{p, \neg r, \neg s\}, \{r, \neg q\}, \{q, \neg s\}, \{s\}\}, \qquad G = \{\neg p, \neg q\}$$

$$T^s = \{\{p, \neg r\}, \{r, \neg q\}, \{q\}\}$$

$$T^{sq} = \{\{p, \neg r\}, \{r\}\}$$

$$T^{sqr} = \{\{p\}\}$$

$$\begin{array}{ll} G^{sqr} = \{\neg p\} & \{p\} \\ G^{sqrp} = \square & \end{array}$$

$$\begin{array}{ll} G^{sq} = \{\neg p\} & \{p, \neg r\} \\ \{\neg r\} & \{r\} \\ \square & \end{array}$$

$$\begin{array}{ll} G^{s} = \{\neg p, \neg q\} & \{p, \neg r\} \\ \{\neg q, \neg r\} & \{r, \neg q\} \\ \{\neg q\} & \{q\} \\ \square & \end{array}$$

$$\begin{array}{ll} G = \{\neg p, \neg q\} & \{p, \neg r, \neg s\} \\ \{\neg q, \neg r, \neg s\} & \{r, \neg q\} \\ \{\neg q, \neg s\} & \{q, \neg s\} \\ \{\neg s\} & \{s\} \\ \square & \end{array}$$

$$T^{sqr}, G^{sqr} \vdash_{LI} \square \qquad T^{sq}, G^{sq} \vdash_{LI} \square \qquad T^{s}, G^{s} \vdash_{LI} \square \qquad T, G \vdash_{LI} \square$$

Program in Prolog

A (propositional) *program* (in Prolog) is a Horn formula containing only program clauses, i.e. facts or rules.

a rule	p :– q, r.	$q \wedge r \rightarrow p$	$\{p, \neg q, \neg r\}$	
	p :– s.	$s \rightarrow p$	$\{p, \neg s\}$	
	q :– s.	$s \rightarrow q$	$\{q, \neg s\}$	
a fact	r.	r	$\{r\}$	
	s.	s	$\{s\}$	*a program*
a query	?– p, q.		$\{\neg p, \neg q\}$	*a goal*

We want to know whether a given query follows from a given program.

Corollary *For every program P and query $(p_1 \wedge \ldots \wedge p_n)$, the following are equivalent:*

1. $P \models p_1 \wedge \ldots \wedge p_n$,
2. $P \cup \{\neg p_1, \ldots, \neg p_n\}$ is unsatisfiable,
3. $\square$ has LI-resolution from $P \cup \{G\}$ starting with the goal $G = \{\neg p_1, \ldots, \neg p_n\}$.

Resolution in Prolog

(1) Interpreter stores clauses as sequences of literals (definite clauses).

An *LD-resolution* (*linear definite*) is an *LI*-resolution in which in each step the resolvent of the present goal $(\neg p_1, \ldots, \neg p_{i-1}, \neg p_i, \neg p_{i+1}, \ldots, \neg p_n)$ and the side clause $(p_i, \neg q_1, ..., \neg q_m)$ is:

$$(\neg p_1, ..., \neg p_{i-1}, \neg q_1, ..., \neg q_m, \neg p_{i+1}, ..., \neg p_n)$$

Observation *Every LI-proof can be transformed into an LD-proof of the same clause from the same formula with the same starting clause (goal).*

(2) The choice of literal from the present goal for resolution is determined by a given selection rule $\mathcal{R}$. Typically, "choose the first literal".

An *SLD-resolution* (*selection*) via $\mathcal{R}$ is an *LD*-resolution in which each step (C_i, B_i) we resolve through the literal $\mathcal{R}(C_i)$.

Observation *Every LD-proof can be transformed into an SLD-proof of the same clause from the same formula with the same starting clause (goal).*

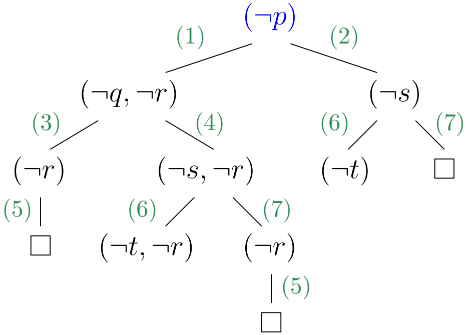
Corollary *SLD-resolution is complete for queries over programs in Prolog.*

SLD-tree

Which program clause will be used for resolution with the present goal?

An *SLD-tree* of a program P and a goal G via a selection rule $\mathcal{R}$ is a tree with nodes labeled by goals so that the root has label G and if a node has label G', his sons correspond to all possibilities of resolving G' with program clauses of P through literal $\mathcal{R}(G')$ and are labeled by the corresponding resolvents.

$$
\begin{array}{ll}
p :- \ q, r. & (1) \\
p :- \ s. & (2) \\
q. & (3) \\
q :- \ s. & (4) \\
r. & (5) \\
s :- \ t. & (6) \\
s. & (7) \\
\quad ?- \ p. &
\end{array}
$$

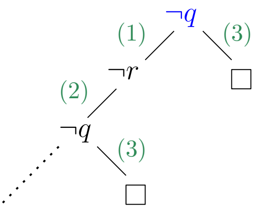


Concluding remarks

- Prolog interpreters search the SLD-tree, the order is not specified.
- Implementations that are based on DFS may not preserve completeness.

$$
\begin{array}{ll}
q :- r. & (1)\\
r :- q. & (2)\\
q. & (3)
\end{array}
$$

$?- \ q.$



- A certain control over the search is provided by !, the cut operation.
- If we allow negation, we may have troubles with semantics of programs.

Table of Contents

Predicate logic

Deals with statements about objects, their properties and relations.

"She is intelligent and her father knows the rector." $I(x) \wedge K(f(x), r)$

- x is a variable, representing an object,
- r is a constant symbol, representing a particular object,
- f is a function symbol, representing a function,
- I, K are relation (predicate) symbols, representing relations (the property of *"being intelligent"* and the relation *"to know"*).

"Everybody has a father." $(\forall x)(\exists y)(y = f(x))$

- $(\forall x)$ is the universal quantifier (*for every* x),
- $(\exists y)$ is the existential quantifier (*there exists* y),
- $=$ is a (binary) relation symbol, representing the identity relation.

NAIL062 Propositional & Predicate Logic: Lecture 6

Slides by Petr Gregor with minor modifications by Jakub Bulín

November 16, 2020

Table of Contents

Language

A first-order language consists of

- variables $x, y, z, \ldots, x_0, x_1, \ldots$ (countable many), the set of all variables is denoted by Var,
- function symbols $f, g, h, \ldots$, including constant symbols $c, d, \ldots$, which are nullary function symbols,
- relation (predicate) symbols $P, Q, R, \ldots$, eventually the symbol $=$ (equality) as a special relation symbol,
- quantifiers $(\forall x)$, $(\exists x)$ for every variable $x \in \mathrm{Var}$,
- logical connectives $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$
- parentheses (,)

Every function and relation symbol S has an associated *arity* $\mathrm{ar}(S) \in \mathbb{N}$.

Remark *Compared to propositional logic we have no (explicit) propositional variables, but they can be introduced as nullary relation symbols.*

Signatures

- *Symbols of logic* are variables, quantifiers, connectives and parentheses.
- *Non-logical symbols* are function and relation symbols except the equality symbol. The equality is (usually) considered separately.
- A *signature* is a pair $\langle \mathcal{R}, \mathcal{F} \rangle$ of disjoint sets of relation and function symbols with associated arities, whereas none of them is the equality symbol. A signature lists all non-logical symbols.
- A *language* is determined by a signature $L = \langle \mathcal{R}, \mathcal{F} \rangle$ and by specifying whether it is a language with equality or not. A language must contain at least one relation symbol (non-logical or the equality).

Remark *The meaning of symbols in a language is not assigned, e.g. the symbol $+$ does not have to represent the standard addition.*

Examples of languages

We describe a language by a list of all non-logical symbols with eventual clarification of arity and whether they are relation or function symbols.

The following examples of languages are all with equality.

- $L = \langle\ \rangle$ is the language of pure equality,
- $L = \langle c_i \rangle_{i \in \mathbb{N}}$ is the language of countable many constants,
- $L = \langle \leq \rangle$ is the language of orderings,
- $L = \langle E \rangle$ is the language of the graph theory,
- $L = \langle +, -, 0 \rangle$ is the language of the group theory,
- $L = \langle +, -, \cdot, 0, 1 \rangle$ is the language of the field theory,
- $L = \langle -, \wedge, \vee, 0, 1 \rangle$ is the language of Boolean algebras,
- $L = \langle S, +, \cdot, 0, \leq \rangle$ is the language of arithmetic,

where c_i, 0, 1 are constant symbols, S, $-$ are unary function symbols, $+$, $\cdot$, $\wedge$, $\vee$ are binary function symbols, E, $\leq$ are binary relation symbols.

Terms

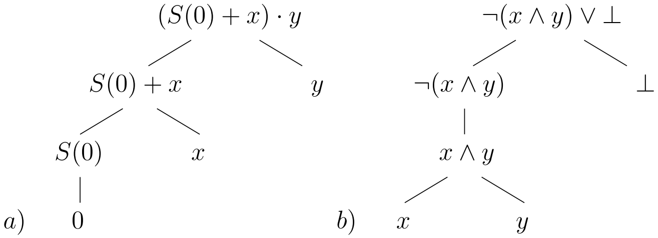
Are expressions representing values of (composed) functions.

Terms of a language L are defined inductively by

1. every variable or constant symbol in L is a term,
2. if f is a function symbol in L of arity $n > 0$ and $t_0, \ldots, t_{n-1}$ are terms, then also the expression $f(t_0, \ldots, t_{n-1})$ is a term,
3. every term is formed by a finite number of steps (i), (ii).

- A *ground term* is a term with no variables.
- The set of all terms of a language L is denoted by Term_L.
- A term that is a part of another term t is called a *subterm* of t.
- The structure of terms can be represented by their formation trees.
- For binary function symbols we often use infix notation, e.g. we write $(x + y)$ instead of $+(x, y)$.

Examples of terms



- (a) The formation tree of the term $(S(0)+x)\cdot y$ of the language of arithmetic.
- (b) Propositional formulas only with connectives $\neg$, $\wedge$, $\vee$, eventually with constants $\top$, $\bot$ can be viewed as terms of the language of Boolean algebras.

Atomic formulas

Are the simplest formulas.

- An *atomic formula* of a language L is an expression $R(t_0, \ldots, t_{n-1})$ where

  R is an n-ary relation symbol in L and $t_0, \ldots, t_{n-1}$ are terms of L.
- The set of all atomic formulas of a language L is denoted by AFm_L.
- The structure of an atomic formula can be represented by a formation tree from the formation subtrees of its terms.
- For binary relation symbols we often use infix notation, e.g. $t_1 = t_2$ instead of $=(t_1, t_2)$ or $t_1 \leq t_2$ instead of $\leq(t_1, t_2)$.
- *Examples of atomic formulas*

$$K(f(x), r), \quad x \cdot y \leq (S(0) + x) \cdot y, \quad \neg(x \wedge y) \vee \bot = \bot.$$

Formula

Formulas of a language L are defined inductively by

- every atomic formula is a formula,
- if φ, ψ are formulas, then also the following expressions are formulas

$$(\neg\varphi)\ ,(\varphi \wedge \psi)\ ,(\varphi \vee \psi)\ ,(\varphi \rightarrow \psi)\ ,(\varphi \leftrightarrow \psi),$$

- if φ is a formula and x is a variable, then also the expressions $((\forall x)\varphi)$ and $((\exists x)\varphi)$ are formulas.
- every formula is formed by a finite number of steps (i), (ii), (iii).

- The set of all formulas of a language L is denoted by Fm_L.
- A formula that is a part of another formula φ is called a *subformula* of φ.
- The structure of formulas can be represented by their formation trees.

Conventions

- After introducing *priorities* for binary function symbols e.g. $+$, $\cdot$ we are in infix notation allowed to omit parentheses that are around a subterm formed by a symbol of higher priority, e.g. $x \cdot y + z$ instead of $(x \cdot y) + z$.
- After introducing *priorities* for connectives and quantifiers we are allowed to omit parentheses that are around subformulas formed by connectives of higher priority.

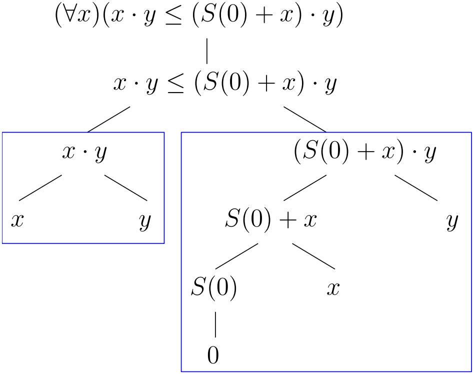
$$(1) \quad \to,\ \leftrightarrow \qquad\qquad (2) \quad \wedge,\ \vee \qquad\qquad (3) \quad \neg,\ (\forall x),\ (\exists x)$$

- They can be always omitted around subformulas formed by $\neg$, $(\forall x)$, $(\exists x)$.
- We may also omit parentheses in $(\forall x)$ and $(\exists x)$ for every $x \in \mathrm{Var}$.
- The outer parentheses may be omitted as well.

$$(((\neg((\forall x)R(x))) \wedge ((\exists y)P(y))) \to (\neg(((\forall x)R(x)) \vee (\neg((\exists y)P(y))))))$$

$$\neg\forall x R(x) \wedge \exists y P(y) \to \neg(\forall x R(x) \vee \neg\exists y P(y))$$

An example of a formula



The formation tree of the formula $(\forall x)(x \cdot y \leq (S(0) + x) \cdot y)$.

Occurrences of variables

Let φ be a formula and x be a variable.

- An *occurrence* of x in φ is a leaf labeled by x in its formation tree.
- An occurrence of x in φ is *bound* if it is in some subformula ψ that starts with $(\forall x)$ or $(\exists x)$. An occurrence of x in φ is *free* if it is not bound.
- A variable x is *free* in φ if it has at least one free occurrence in φ. It is *bound* in φ if it has at least one bound occurrence in φ.
- A variable x can be both free and bound in φ. For example in
$$(\forall x)(\exists y)(x \leq y) \vee x \leq z.$$
- We write $\varphi(x_1, \ldots, x_n)$ to denote that $x_1, \ldots, x_n$ are all free variables in the formula φ. *(φ states something about these variables.)*

Remark *We will see that the truth value of a formula (in a given interpretation of symbols) depends only on the assignment of free variables.*

Open and closed formulas

- A formula is *open* if it is without quantifiers. For the set OFm_L of all open formulas in a language L it holds that $\mathrm{AFm}_L \subsetneq \mathrm{OFm}_L \subsetneq \mathrm{Fm}_L$.
- A formula is *closed* (a *sentence*) if it has no free variable; that is, all occurrences of variables are bound.
- A formula can be both open and closed. In this case, all its terms are ground terms.

$$
\begin{array}{rl}
x + y \leq 0 & \textit{open}, \varphi(x, y) \\
(\forall x)(\forall y)(x + y \leq 0) & \textit{a sentence}, \\
(\forall x)(x + y \leq 0) & \textit{neither open nor a sentence}, \varphi(y) \\
1 + 0 \leq 0 & \textit{open sentence}
\end{array}
$$

Remark *We will see that in a fixed interpretation of symbols a sentence has a fixed truth value; that is, it does not depend on the assignment of variables.*

Instances

After substituting a term t for a free variable x in a formula φ, we would expect that the new formula (newly) says about t "the same" as φ did about x.

$\varphi(x)$	$(\exists y)(x + y = 1)$	*"there is an element $1 - x$"*
for $t = 1$ we can $\varphi(x/t)$	$(\exists y)(1 + y = 1)$	*"there is an element $1 - 1$"*
for $t = y$ we cannot	$(\exists y)(y + y = 1)$	*"1 is divisible by 2"*

- A term t is *substitutable* for a variable x in a formula φ if substituting t for all free occurrences of x in φ does not introduce a new bound occurrence of a variable from t.
- Then we denote the obtained formula $\varphi(x/t)$ and we call it an *instance* of the formula φ after a *substitution* of a term t for a variable x.
- t is not substitutable for x in φ if and only if x has a free occurrence in some subformula that starts with $(\forall y)$ or $(\exists y)$ for some variable y in t.
- Ground terms are always substitutable.

Variants

Quantified variables can be (under certain conditions) renamed so that we obtain an equivalent formula.

Let $(Qx)\psi$ be a subformula of φ where Q means $\forall$ or $\exists$ and y is a variable such that the following conditions hold.

1) y is substitutable for x in ψ, and
2) y does not have a free occurrence in ψ.

Then by replacing the subformula $(Qx)\psi$ with $(Qy)\psi(x/y)$ we obtain a *variant* of φ *in subformula* $(Qx)\psi$. After variation of one or more subformulas in φ we obtain a *variant* of φ. *For example,*

$(\exists x)(\forall y)(x \leq y)$	*is a formula* φ,
$(\exists u)(\forall v)(u \leq v)$	*is a variant of* φ,
$(\exists y)(\forall y)(y \leq y)$	*is not a variant of* φ, 1) *does not hold,*
$(\exists x)(\forall x)(x \leq x)$	*is not a variant of* φ, 2) *does not hold.*

Table of Contents

Structures

- $\underline{S} = \langle S, \leq \rangle$ is an ordered set where $\leq$ is reflexive, antisymmetric, transitive binary relation on S,
- $G = \langle V, E \rangle$ is an undirected graph without loops where V is the set of *vertices* and E is irreflexive, symmetric binary relation on V *(adjacency)*,
- $\underline{\mathbb{Z}_p} = \langle \mathbb{Z}_p, +, -, 0 \rangle$ is the additive group of integers modulo p,
- $\underline{\mathbb{Q}} = \langle \mathbb{Q}, +, -, \cdot, 0, 1 \rangle$ is the field of rational numbers,
- $\underline{\mathcal{P}(X)} = \langle \mathcal{P}(X), -, \cap, \cup, \emptyset, X \rangle$ is the set algebra over X,
- $\underline{\mathbb{N}} = \langle \mathbb{N}, S, +, \cdot, 0, \leq \rangle$ is the standard model of arithmetic,
- finite automata and other models of computation,
- relational databases, ...

A structure for a language

Let $L = \langle \mathcal{R}, \mathcal{F} \rangle$ be a signature of a language and A be a nonempty set.

- A *realization* (*interpretation*) of a *relation symbol* $R \in \mathcal{R}$ on A is any relation $R^A \subseteq A^{\mathrm{ar}(R)}$. A realization of $=$ on A is the relation Id_A (identity).
- A *realization* (*interpretation*) of a *function symbol* $f \in \mathcal{F}$ on A is any function $f^A \colon A^{\mathrm{ar}(f)} \to A$. Thus a realization of a constant symbol is some element of A.

A *structure* for the language L (*L-structure*) is a triple $\mathcal{A} = \langle A, \mathcal{R}^A, \mathcal{F}^A \rangle$, where

- A is nonempty set, called the *domain* of the structure $\mathcal{A}$,
- $\mathcal{R}^A = \langle R^A \mid R \in \mathcal{R} \rangle$ is a collection of realizations of relation symbols,
- $\mathcal{F}^A = \langle f^A \mid f \in \mathcal{F} \rangle$ is a collection of realizations of function symbols.

A structure for the language L is also called a *model of the language* L. The class of all models of L is denoted by $M(L)$. *Examples for* $L = \langle \leq \rangle$ *are*

$$\langle \mathbb{N}, \leq \rangle,\ \langle \mathbb{Q}, > \rangle,\ \langle X, E \rangle,\ \langle \mathcal{P}(X), \subseteq \rangle.$$

Value of terms

Let t be a term of $L = \langle \mathcal{R}, \mathcal{F} \rangle$ and $\mathcal{A} = \langle A, \mathcal{R}^A, \mathcal{F}^A \rangle$ be an L-structure.

- A *variable assignment* over the domain A is a function $e \colon \mathrm{Var} \to A$.
- The *value* $t^A[e]$ of the term t in the structure $\mathcal{A}$ with respect to the assignment e is defined by

  $$x^A[e] = e(x) \quad \text{for every } x \in \mathrm{Var},$$
  $$(f(t_0, \ldots, t_{n-1}))^A[e] = f^A(t_0^A[e], \ldots, t_{n-1}^A[e]) \quad \text{for every } f \in \mathcal{F}.$$

- In particular, for a constant symbol c we have $c^A[e] = c^A$.
- If t is a ground term, its value in $\mathcal{A}$ is independent of the assignment e.
- The value of t in $\mathcal{A}$ depends only on the assignment of variables in t.

For example, the value of the term $x + 1$ in the structure $\mathcal{N} = \langle \mathbb{N}, +, 1 \rangle$ with respect to the assignment e with $e(x) = 2$ is $(x + 1)^N[e] = 3$.

Values of atomic formulas

Let φ be an atomic formula of $L = \langle \mathcal{R}, \mathcal{F} \rangle$ in the form $R(t_0, \ldots, t_{n-1})$, $\mathcal{A} = \langle A, \mathcal{R}^A, \mathcal{F}^A \rangle$ be an L-structure, and e be a variable assignment over A.

- The *value* $H^A_{at}(\varphi)[e]$ of the formula φ in the structure $\mathcal{A}$ with respect to e is

$$H^A_{at}(R(t_0, \ldots, t_{n-1}))[e] = \begin{cases} 1 & \text{if } (t_0^A[e], \ldots, t_{n-1}^A[e]) \in R^A, \\ 0 & \text{otherwise.} \end{cases}$$

  where $=^A$ is Id_A; that is, $H^A_{at}(t_0 = t_1)[e] = 1$ if $t_0^A[e] = t_1^A[e]$, and $H^A_{at}(t_0 = t_1)[e] = 0$ otherwise.

- If φ is a sentence; that is, all its terms are ground, then its value in $\mathcal{A}$ is independent on the assignment e.
- The value of φ in $\mathcal{A}$ depends only on the assignment of variables in φ.

For example, the value of φ in form $x + 1 \leq 1$ in $\mathcal{N} = \langle \mathbb{N}, +, 1, \leq \rangle$ with respect to the assignment e is $H^N_{at}(\varphi)[e] = 1$ if and only if $e(x) = 0$.

Values of formulas

The *value* $H^A(\varphi)[e]$ of the formula φ in the structure $\mathcal{A}$ wrt. e is

$$\begin{aligned}
H^A(\varphi)[e] &= H^A_{at}(\varphi)[e] \ \text{ if } \varphi \text{ is atomic,}\\
H^A(\neg\varphi)[e] &= -_1(H^A(\varphi)[e])\\
H^A(\varphi \wedge \psi)[e] &= \wedge_1(H^A(\varphi)[e], H^A(\psi)[e])\\
H^A(\varphi \vee \psi)[e] &= \vee_1(H^A(\varphi)[e], H^A(\psi)[e])\\
H^A(\varphi \rightarrow \psi)[e] &= \rightarrow_1(H^A(\varphi)[e], H^A(\psi)[e])\\
H^A(\varphi \leftrightarrow \psi)[e] &= \leftrightarrow_1(H^A(\varphi)[e], H^A(\psi)[e])\\
H^A((\forall x)\varphi)[e] &= \min_{a \in A}(H^A(\varphi)[e(x/a)])\\
H^A((\exists x)\varphi)[e] &= \max_{a \in A}(H^A(\varphi)[e(x/a)])
\end{aligned}$$

where $-_1$, $\wedge_1$, $\vee_1$, $\rightarrow_1$, $\leftrightarrow_1$ are the Boolean functions given by the tables and $e(x/a)$ for $a \in A$ denotes the assignment obtained from e by setting $e(x) = a$.

Observation $H^A(\varphi)[e]$ *depends only on assignment of free variables in* φ.

Satisfiability with respect to assignments

The structure $\mathcal{A}$ satisfies the formula φ with assignment e if $H^A(\varphi)[e] = 1$. Then we write $\mathcal{A} \models \varphi[e]$, and $\mathcal{A} \not\models \varphi[e]$ otherwise. It holds that

$$
\begin{array}{lcl}
\mathcal{A} \models \neg\varphi[e] & \Leftrightarrow & \mathcal{A} \not\models \varphi[e] \\
\mathcal{A} \models (\varphi \wedge \psi)[e] & \Leftrightarrow & \mathcal{A} \models \varphi[e] \text{ and } \mathcal{A} \models \psi[e] \\
\mathcal{A} \models (\varphi \vee \psi)[e] & \Leftrightarrow & \mathcal{A} \models \varphi[e] \text{ or } \mathcal{A} \models \psi[e] \\
\mathcal{A} \models (\varphi \rightarrow \psi)[e] & \Leftrightarrow & \mathcal{A} \models \varphi[e] \text{ implies } \mathcal{A} \models \psi[e] \\
\mathcal{A} \models (\varphi \leftrightarrow \psi)[e] & \Leftrightarrow & \mathcal{A} \models \varphi[e] \text{ if and only if } \mathcal{A} \models \psi[e] \\
\mathcal{A} \models (\forall x)\varphi[e] & \Leftrightarrow & \mathcal{A} \models \varphi[e(x/a)] \text{ for every } a \in A \\
\mathcal{A} \models (\exists x)\varphi[e] & \Leftrightarrow & \mathcal{A} \models \varphi[e(x/a)] \text{ for some } a \in A
\end{array}
$$

Observation *Let term t be substitutable for x in φ and ψ be a variant of φ. Then for every structure $\mathcal{A}$ and assignment e*

1. $\mathcal{A} \models \varphi(x/t)[e]$ *if and only if* $\mathcal{A} \models \varphi[e(x/a)]$ *where* $a = t^A[e]$,
2. $\mathcal{A} \models \varphi[e]$ *if and only if* $\mathcal{A} \models \psi[e]$.

Validity in a structure

Let φ be a formula of a language L and $\mathcal{A}$ be an L-structure.

- φ is *valid* (*true*) *in the structure* $\mathcal{A}$, denoted by $\mathcal{A} \models \varphi$, if $\mathcal{A} \models \varphi[e]$ for every $e\colon \mathrm{Var} \to A$. We say that $\mathcal{A}$ *satisfies* φ. Otherwise, we write $\mathcal{A} \not\models \varphi$.
- φ is *contradictory in* $\mathcal{A}$ if $\mathcal{A} \models \neg\varphi$; that is, $\mathcal{A} \not\models \varphi[e]$ for every $e\colon \mathrm{Var} \to A$.
- For every formulas φ, ψ, variable x, and structure $\mathcal{A}$

$$\begin{array}{llcl}
(1) & \mathcal{A} \models \varphi & \Rightarrow & \mathcal{A} \not\models \neg\varphi \\
(2) & \mathcal{A} \models \varphi \wedge \psi & \Leftrightarrow & \mathcal{A} \models \varphi \ \text{ and } \ \mathcal{A} \models \psi \\
(3) & \mathcal{A} \models \varphi \vee \psi & \Leftarrow & \mathcal{A} \models \varphi \ \text{ or } \ \mathcal{A} \models \psi \\
(4) & \mathcal{A} \models \varphi & \Leftrightarrow & \mathcal{A} \models (\forall x)\varphi
\end{array}$$

- If φ is a sentence, it is valid or contradictory in $\mathcal{A}$, and thus (1) holds also in $\Leftarrow$. If moreover ψ is a sentence, also (3) holds in $\Rightarrow$.
- By (4), $\mathcal{A} \models \varphi$ if and only if $\mathcal{A} \models \psi$ where ψ is a *universal closure* of φ, i.e. a formula $(\forall x_1)\cdots(\forall x_n)\varphi$ where $x_1, \ldots, x_n$ are all free variables in φ.

Validity in a theory

- A *theory* of language L is any set T of formulas of L (so called *axioms*).
- A *model of a theory* T is an L-structure $\mathcal{A}$ such that $\mathcal{A} \models \varphi$ for every $\varphi \in T$. Then we write $\mathcal{A} \models T$ and we say that $\mathcal{A}$ *satisfies* T.
- The *class of models* of a theory T is $M(T) = \{\mathcal{A} \in M(L) \mid \mathcal{A} \models T\}$.
- A formula φ is *valid in T* (*true in T*), denoted by $T \models \varphi$, if $\mathcal{A} \models \varphi$ for every model $\mathcal{A}$ of T. Otherwise, we write $T \not\models \varphi$.
- φ is *contradictory in T* if $T \models \neg\varphi$, i.e. φ is contradictory in all models of T.
- φ is *independent in T* if it is neither valid nor contradictory in T.
- If $T = \emptyset$, we have $M(T) = M(L)$ and we omit T, eventually we say *"in logic"*. Then $\models \varphi$ means that φ is (*universally*) *valid* (a *tautology*).
- A *consequence* of T is the set $\theta^L(T)$ of all sentences of L valid in T, i.e. $$\theta^L(T) = \{\varphi \in \mathrm{Fm}_L \mid T \models \varphi \text{ and } \varphi \text{ is a sentence}\}.$$

Example of a theory

A *theory of orderings* T in language $L = \langle \leq \rangle$ with equality has axioms

$$x \leq x \qquad \text{(reflexivity)}$$
$$x \leq y \quad \wedge \quad y \leq x \quad \rightarrow \quad x = y \qquad \text{(antisymmetry)}$$
$$x \leq y \quad \wedge \quad y \leq z \quad \rightarrow \quad x \leq z \qquad \text{(transitivity)}$$

Models of T are L-structures $\langle S, \leq_S \rangle$, so called ordered sets, that satisfy the axioms of T, for example $\mathcal{A} = \langle \mathbb{N}, \leq \rangle$ or $\mathcal{B} = \langle \mathcal{P}(X), \subseteq \rangle$ for $X = \{0, 1, 2\}$.

- *A formula* φ: $x \leq y \vee y \leq x$ *is valid in* $\mathcal{A}$ *but not in* $\mathcal{B}$ *since* $\mathcal{B} \not\models \varphi[e]$ *for the assignment* $e(x) = \{0\}, e(y) = \{1\}$, *thus* φ *is independent in* T.
- *A sentence* ψ: $(\exists x)(\forall y)(y \leq x)$ *is valid in* $\mathcal{B}$ *and contradictory in* $\mathcal{A}$, *hence it is independent in* T *as well. We write* $\mathcal{B} \models \psi$, $\mathcal{A} \models \neg\psi$.
- *A formula* χ: $(x \leq y \wedge y \leq z \wedge z \leq x) \rightarrow (x = y \wedge y = z)$ *is valid in* T, *denoted by* $T \models \chi$, *the same holds for its universal closure.*

Properties of theories

A theory T of a language L is *(semantically)*

- *inconsistent* if $T \models \perp$, otherwise T is *consistent* (*satisfiable*),
- *complete* if it is consistent and every sentence of L is valid in T or contradictory in T,
- an *extension* of a theory T' of language L' if $L' \subseteq L$ and $\theta^{L'}(T') \subseteq \theta^{L}(T)$, we say that an extension T of a theory T' is *simple* if $L = L'$; and *conservative* if $\theta^{L'}(T') = \theta^{L}(T) \cap \mathrm{Fm}_{L'}$,
- *equivalent* with a theory T' if T is an extension of T' and vice-versa,

Structures $\mathcal{A}$, $\mathcal{B}$ for a language L are *elementarily equivalent*, denoted by $\mathcal{A} \equiv \mathcal{B}$, if they satisfy the same sentences of L.

Observation *Let T and T' be theories of a language L. T is (semantically)*

1. *consistent if and only if it has a model,*
2. *complete iff it has a single model, up to elementarily equivalence,*
3. *an extension of T' if and only if $M(T) \subseteq M(T')$,*
4. *equivalent with T' if and only if $M(T) = M(T')$.*

Unsatisfiability and validity

The problem of validity in a theory can be transformed to the problem of satisfiability of (another) theory.

Proposition *For every theory T and sentence φ (of the same language)*

$$T, \neg\varphi \ \textit{ is unsatisfiable} \quad \Leftrightarrow \quad T \models \varphi.$$

Proof By definitions, it is equivalent that

1. $T, \neg\varphi$ is unsatisfiable (i.e. it has no model),
2. $\neg\varphi$ is not valid in any model of T,
3. φ is valid in every model of T,
4. $T \models \varphi$. $\square$

Remark *The assumption that φ is a sentence is necessary for (2) $\Rightarrow$ (3).*

For example, the theory $\{P(c), \neg P(x)\}$ is unsatisfiable, but $P(c) \not\models P(x)$, where P is a unary relation symbol and c is a constant symbol.

Substructures

Let $\mathcal{A} = \langle A, \mathcal{R}^A, \mathcal{F}^A \rangle$ and $\mathcal{B} = \langle B, \mathcal{R}^B, \mathcal{F}^B \rangle$ be structures for $L = \langle \mathcal{R}, \mathcal{F} \rangle$.

We say that $\mathcal{B}$ is an (induced) *substructure* of $\mathcal{A}$, denoted by $\mathcal{B} \subseteq \mathcal{A}$, if

(i) $B \subseteq A$,

(ii) $R^B = R^A \cap B^{\mathrm{ar}(R)}$ for every $R \in \mathcal{R}$,

(iii) $f^B = f^A \cap (B^{\mathrm{ar}(f)} \times B)$; that is, $f^B = f^A \restriction B^{\mathrm{ar}(f)}$, for every $f \in \mathcal{F}$.

A set $C \subseteq A$ is a domain of some substructure of $\mathcal{A}$ if and only if C is closed under all functions of $\mathcal{A}$. Then the respective substructure, denoted by $\mathcal{A} \restriction C$, is said to be the *restriction* of the structure $\mathcal{A}$ to C.

- A set $C \subseteq A$ is *closed* under a function $f\colon A^n \to A$ if
  $f(x_0, \ldots, x_{n-1}) \in C$
  for every $x_0, \ldots, x_{n-1} \in C$.

Example: $\underline{\mathbb{Z}} = \langle \mathbb{Z}, +, \cdot, 0 \rangle$ *is a substructure of* $\underline{\mathbb{Q}} = \langle \mathbb{Q}, +, \cdot, 0 \rangle$ *and* $\underline{\mathbb{Z}} = \underline{\mathbb{Q}} \restriction \mathbb{Z}$. *Furthermore,* $\underline{\mathbb{N}} = \langle \mathbb{N}, +, \cdot, 0 \rangle$ *is their substructure and* $\underline{\mathbb{N}} = \underline{\mathbb{Q}} \restriction \mathbb{N} = \underline{\mathbb{Z}} \restriction \mathbb{N}$.

Validity in a substructure

Let $\mathcal{B}$ be a substructure of a structure $\mathcal{A}$ for a (fixed) language L.

Proposition *For every open formula φ and assignment $e\colon \mathrm{Var} \to B$,*

$$\mathcal{A} \models \varphi[e] \quad \textit{if and only if} \quad \mathcal{B} \models \varphi[e].$$

Proof For atomic φ it follows from the definition of the truth value with respect to an assignment. Otherwise by induction on the structure of the formula. □

Corollary *For every open formula φ and structure $\mathcal{A}$,*

$$\mathcal{A} \models \varphi \quad \textit{if and only if} \quad \mathcal{B} \models \varphi \textit{ for every substructure } \mathcal{B} \subseteq \mathcal{A}.$$

- A theory T is *open* if all axioms of T are open.

Corollary *Every substr. of a model of an open theory T is a model of T. For example, every substructure of a graph, i.e. a model of theory of graphs, is a graph, called a subgraph. Similarly subgroups, Boolean subalgebras, etc.*

Generated substructure, expansion, reduct

Let $\mathcal{A} = \langle A, \mathcal{R}^A, \mathcal{F}^A \rangle$ be a structure and $X \subseteq A$. Let B be the smallest subset of A containing X that is closed under all functions of the structure $\mathcal{A}$ (including constants). Then the structure $\mathcal{A} \restriction B$ is denoted by $\mathcal{A}\langle X \rangle$ and is called the substructure of $\mathcal{A}$ *generated* by the set X.

Example: for $\underline{\mathbb{Q}} = \langle \mathbb{Q}, +, \cdot, 0 \rangle$, $\underline{\mathbb{Z}} = \langle \mathbb{Z}, +, \cdot, 0 \rangle$, $\underline{\mathbb{N}} = \langle \mathbb{N}, +, \cdot, 0 \rangle$ *it is* $\underline{\mathbb{Q}}\langle \{1\} \rangle = \underline{\mathbb{N}}$, $\underline{\mathbb{Q}}\langle \{-1\} \rangle = \underline{\mathbb{Z}}$, *and* $\underline{\mathbb{Q}}\langle \{2\} \rangle$ *is the substructure on all even natural numbers.*

Let $\mathcal{A}$ be a structure for a language L and $L' \subseteq L$. By omitting realizations of symbols that are not in L' we obtain from $\mathcal{A}$ a structure $\mathcal{A}'$ called the *reduct* of $\mathcal{A}$ to the language L'. Conversely, $\mathcal{A}$ is an *expansion* of $\mathcal{A}'$ into L.

For example, $\langle \mathbb{N}, + \rangle$ *is a reduct of* $\langle \mathbb{N}, +, \cdot, 0 \rangle$*. On the other hand, the structure* $\langle \mathbb{N}, +, c_i \rangle_{i \in \mathbb{N}}$ *with* $c_i = i$ *for every* $i \in \mathbb{N}$ *is the expansion of* $\langle \mathbb{N}, + \rangle$ *by names of elements from* $\mathbb{N}$*.*

Theorem on constants

Theorem *Let φ be a formula in a language L with free variables $x_1, \ldots, x_n$ and let T be a theory in L. Let L' be the extension of L with new constant symbols $c_1, \ldots, c_n$ and let T' denote the theory T in L'. Then*

$$T \models \varphi \quad \textit{if and only if} \quad T' \models \varphi(x_1/c_1, \ldots, x_n/c_n).$$

Proof ($\Rightarrow$) If $\mathcal{A}'$ is a model of T', let $\mathcal{A}$ be the reduct of $\mathcal{A}'$ to L. Since $\mathcal{A} \models \varphi[e]$ for every assignment e, we have in particular

$$\mathcal{A} \models \varphi[e(x_1/c_1^{A'}, \ldots, x_n/c_n^{A'})], \quad \text{i.e.} \;\; \mathcal{A}' \models \varphi(x_1/c_1, \ldots, x_n/c_n).$$

($\Leftarrow$) If $\mathcal{A}$ is a model of T and e an assignment, let $\mathcal{A}'$ be the expansion of A into L' by setting $c_i^{A'} = e(x_i)$ for every i. Since $\mathcal{A}' \models \varphi(x_1/c_1, \ldots, x_n/c_n)[e']$ for every assignment e', we have

$$\mathcal{A}' \models \varphi[e(x_1/c_1^{A'}, \ldots, x_n/c_n^{A'})], \quad \text{i.e.} \;\; \mathcal{A} \models \varphi[e]. \quad \square$$

Boolean algebras

The theory of *Boolean algebras* has the language $L = \langle -, \wedge, \vee, 0, 1\rangle$ with equality and the following axioms.

$$x \wedge (y \wedge z) = (x \wedge y) \wedge z \qquad \text{(asociativity of } \wedge\text{)}$$

$$x \vee (y \vee z) = (x \vee y) \vee z \qquad \text{(asociativity of } \vee\text{)}$$

$$x \wedge y = y \wedge x \qquad \text{(commutativity of } \wedge\text{)}$$

$$x \vee y = y \vee x \qquad \text{(commutativity of } \vee\text{)}$$

$$x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z) \qquad \text{(distributivity of } \wedge \text{ over } \vee\text{)}$$

$$x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z) \qquad \text{(distributivity of } \vee \text{ over } \wedge\text{)}$$

$$x \wedge (x \vee y) = x, \quad x \vee (x \wedge y) = x \qquad \text{(absorption)}$$

$$x \vee (-x) = 1, \quad x \wedge (-x) = 0 \qquad \text{(complementation)}$$

$$0 \neq 1 \qquad \text{(non-triviality)}$$

The smallest model is $\underline{2} = \langle 2, -_1, \wedge_1, \vee_1, 0, 1\rangle$. Finite Boolean algebras are (up to isomorphism) exactly $\underline{^n2} = \langle {}^n2, -_n, \wedge_n, \vee_n, 0_n, 1_n\rangle$ for $n \in \mathbb{N}^+$, where the operations *(on binary n-tuples)* are the coordinate-wise operations of $\underline{2}$.

Relations of propositional and predicate logic

- Propositional formulas over connectives $\neg$, $\wedge$, $\vee$ (eventually with $\top$, $\bot$) can be viewed as Boolean terms. Then the truth value of φ in a given assignment is the value of the term in the Boolean algebra $\underline{2}$.
- Lindenbaum-Tarski algebra over $\mathbb{P}$ is Boolean algebra (also for $\mathbb{P}$ infinite).
- If we represent atomic subformulas in an open formula φ (without equality) with propositional letters, we obtain a proposition that is valid if and only if φ is valid.
- Propositional logic can be introduced as a fragment of predicate logic using nullary relation symbols (*syntax*) and nullary relations (*semantics*) since $A^0 = \{\emptyset\} = 1$, so $R^A \subseteq A^0$ is either $R^A = \emptyset = 0$ or $R^A = \{\emptyset\} = 1$.

NAIL062 Propositional & Predicate Logic: Lecture 8

Slides by Petr Gregor with minor modifications by Jakub Bulín

November 23, 2020

Substructures

Let $\mathcal{A} = \langle A, \mathcal{R}^A, \mathcal{F}^A \rangle$ and $\mathcal{B} = \langle B, \mathcal{R}^B, \mathcal{F}^B \rangle$ be structures for $L = \langle \mathcal{R}, \mathcal{F} \rangle$.

We say that $\mathcal{B}$ is an (induced) *substructure* of $\mathcal{A}$, denoted by $\mathcal{B} \subseteq \mathcal{A}$, if

(i) $B \subseteq A$,

(ii) $R^B = R^A \cap B^{\mathrm{ar}(R)}$ for every $R \in \mathcal{R}$,

(iii) $f^B = f^A \cap (B^{\mathrm{ar}(f)} \times B)$; that is, $f^B = f^A \restriction B^{\mathrm{ar}(f)}$, for every $f \in \mathcal{F}$.

A set $C \subseteq A$ is a domain of some substructure of $\mathcal{A}$ if and only if C is closed under all functions of $\mathcal{A}$. Then the respective substructure, denoted by $\mathcal{A} \restriction C$, is said to be the *restriction* of the structure $\mathcal{A}$ to C.

- A set $C \subseteq A$ is *closed* under a function $f \colon A^n \to A$ if
  $f(x_0, \ldots, x_{n-1}) \in C$
  for every $x_0, \ldots, x_{n-1} \in C$.

Example: $\underline{\mathbb{Z}} = \langle \mathbb{Z}, +, \cdot, 0 \rangle$ *is a substructure of* $\underline{\mathbb{Q}} = \langle \mathbb{Q}, +, \cdot, 0 \rangle$ *and* $\underline{\mathbb{Z}} = \underline{\mathbb{Q}} \restriction \mathbb{Z}$. *Furthermore,* $\underline{\mathbb{N}} = \langle \mathbb{N}, +, \cdot, 0 \rangle$ *is their substructure and* $\underline{\mathbb{N}} = \underline{\mathbb{Q}} \restriction \mathbb{N} = \underline{\mathbb{Z}} \restriction \mathbb{N}$.

Validity in a substructure

Let $\mathcal{B}$ be a substructure of a structure $\mathcal{A}$ for a (fixed) language L.

Proposition *For every open formula φ and assignment $e\colon \mathrm{Var} \to B$,*

$$\mathcal{A} \models \varphi[e] \quad \text{if and only if} \quad \mathcal{B} \models \varphi[e].$$

Proof For atomic φ it follows from the definition of the truth value with respect to an assignment. Otherwise by induction on the structure of the formula. $\square$

Corollary *For every open formula φ and structure $\mathcal{A}$,*

$$\mathcal{A} \models \varphi \quad \text{if and only if} \quad \mathcal{B} \models \varphi \text{ for every substructure } \mathcal{B} \subseteq \mathcal{A}.$$

- A theory T is *open* if all axioms of T are open.

Corollary *Every substr. of a model of an open theory T is a model of T. For example, every substructure of a graph, i.e. a model of theory of graphs, is a graph, called a subgraph. Similarly subgroups, Boolean subalgebras, etc.*

Generated substructure, expansion, reduct

Let $\mathcal{A} = \langle A, \mathcal{R}^A, \mathcal{F}^A \rangle$ be a structure and $X \subseteq A$. Let B be the smallest subset of A containing X that is closed under all functions of the structure $\mathcal{A}$ (including constants). Then the structure $\mathcal{A} \restriction B$ is denoted by $\mathcal{A}\langle X \rangle$ and is called the substructure of $\mathcal{A}$ *generated* by the set X.

Example: for $\underline{\mathbb{Q}} = \langle \mathbb{Q}, +, \cdot, 0 \rangle$, $\underline{\mathbb{Z}} = \langle \mathbb{Z}, +, \cdot, 0 \rangle$, $\underline{\mathbb{N}} = \langle \mathbb{N}, +, \cdot, 0 \rangle$ it is $\underline{\mathbb{Q}}\langle \{1\} \rangle = \underline{\mathbb{N}}$, $\underline{\mathbb{Q}}\langle \{-1\} \rangle = \underline{\mathbb{Z}}$, and $\underline{\mathbb{Q}}\langle \{2\} \rangle$ is the substructure on all even natural numbers.

Let $\mathcal{A}$ be a structure for a language L and $L' \subseteq L$. By omitting realizations of symbols that are not in L' we obtain from $\mathcal{A}$ a structure $\mathcal{A}'$ called the *reduct* of $\mathcal{A}$ to the language L'. Conversely, $\mathcal{A}$ is an *expansion* of $\mathcal{A}'$ into L.

For example, $\langle \mathbb{N}, + \rangle$ is a reduct of $\langle \mathbb{N}, +, \cdot, 0 \rangle$. On the other hand, the structure $\langle \mathbb{N}, +, c_i \rangle_{i \in \mathbb{N}}$ with $c_i = i$ for every $i \in \mathbb{N}$ is the expansion of $\langle \mathbb{N}, + \rangle$ by names of elements from $\mathbb{N}$.

Theorem on constants

Theorem *Let φ be a formula in a language L with free variables $x_1, \dots, x_n$ and let T be a theory in L. Let L' be the extension of L with new constant symbols $c_1, \dots, c_n$ and let T' denote the theory T in L'. Then*

$$T \models \varphi \quad \textit{if and only if} \quad T' \models \varphi(x_1/c_1, \dots, x_n/c_n).$$

Proof ($\Rightarrow$) If $\mathcal{A}'$ is a model of T', let $\mathcal{A}$ be the reduct of $\mathcal{A}'$ to L. Since $\mathcal{A} \models \varphi[e]$ for every assignment e, we have in particular

$$\mathcal{A} \models \varphi[e(x_1/c_1^{A'}, \dots, x_n/c_n^{A'})], \quad \text{i.e. } \mathcal{A}' \models \varphi(x_1/c_1, \dots, x_n/c_n).$$

($\Leftarrow$) If $\mathcal{A}$ is a model of T and e an assignment, let $\mathcal{A}'$ be the expansion of A into L' by setting $c_i^{A'} = e(x_i)$ for every i. Since $\mathcal{A}' \models \varphi(x_1/c_1, \dots, x_n/c_n)[e']$ for every assignment e', we have

$$\mathcal{A}' \models \varphi[e(x_1/c_1^{A'}, \dots, x_n/c_n^{A'})], \quad \text{i.e. } \mathcal{A} \models \varphi[e]. \quad \square$$

Boolean algebras

The theory of *Boolean algebras* has the language $L = \langle -, \wedge, \vee, 0, 1 \rangle$ with equality and the following axioms.

$x \wedge (y \wedge z) = (x \wedge y) \wedge z$ (asociativity of $\wedge$)

$x \vee (y \vee z) = (x \vee y) \vee z$ (asociativity of $\vee$)

$x \wedge y = y \wedge x$ (commutativity of $\wedge$)

$x \vee y = y \vee x$ (commutativity of $\vee$)

$x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$ (distributivity of $\wedge$ over $\vee$)

$x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$ (distributivity of $\vee$ over $\wedge$)

$x \wedge (x \vee y) = x, \quad x \vee (x \wedge y) = x$ (absorption)

$x \vee (-x) = 1, \quad x \wedge (-x) = 0$ (complementation)

$0 \neq 1$ (non-triviality)

The smallest model is $\underline{2} = \langle 2, -_1, \wedge_1, \vee_1, 0, 1 \rangle$. Finite Boolean algebras are (up to isomorphism) exactly $\underline{^n 2} = \langle ^n 2, -_n, \wedge_n, \vee_n, 0_n, 1_n \rangle$ for $n \in \mathbb{N}^+$, where the operations *(on binary n-tuples)* are the coordinate-wise operations of $\underline{2}$.

Relations of propositional and predicate logic

- Propositional formulas over connectives $\neg$, $\wedge$, $\vee$ (eventually with $\top$, $\bot$) can be viewed as Boolean terms. Then the truth value of φ in a given assignment is the value of the term in the Boolean algebra $\underline{2}$.
- Lindenbaum-Tarski algebra over $\mathbb{P}$ is Boolean algebra (also for $\mathbb{P}$ infinite).
- If we represent atomic subformulas in an open formula φ (without equality) with propositional letters, we obtain a proposition that is valid if and only if φ is valid.
- Propositional logic can be introduced as a fragment of predicate logic using nullary relation symbols (*syntax*) and nullary relations (*semantics*) since $A^0 = \{\emptyset\} = 1$, so $R^A \subseteq A^0$ is either $R^A = \emptyset = 0$ or $R^A = \{\emptyset\} = 1$.

Definable sets

Which sets [or relations] can be defined by [first-order] properties in a given structure?

- The set defined by a formula $\varphi(x_1, \ldots, x_n)$ in the structure $\mathcal{A}$ is

$$\varphi^{\mathcal{A}}(x_1, \ldots, x_n) = \{(a_1, \ldots, a_n) \in A^n \mid \mathcal{A} \models \varphi[e(x_1/a_1, \ldots, x_n/a_n)]\}$$

  For brevity, we write $\varphi^{\mathcal{A}}(\bar{x}) = \{\bar{a} \in A^{|\bar{x}|} \mid \mathcal{A} \models \varphi[e(\bar{x}/\bar{a})]\}$ where $|\bar{x}| = n$.

- The set defined by $\varphi(\bar{x}, \bar{y})$ with parameters $\bar{b} \in A^{|\bar{y}|}$ in $\mathcal{A}$ is

$$\varphi^{\mathcal{A},\bar{b}}(\bar{x}, \bar{y}) = \{\bar{a} \in A^{|\bar{x}|} \mid \mathcal{A} \models \varphi[e(\bar{x}/\bar{a}, \bar{y}/\bar{b})]\}$$

  E.g. for $\varphi = E(x, y)$, $\varphi^{\mathcal{G},b}(x, y)$ is the set of all neighbours of the vertex b in the graph $\mathcal{G}$.

- Given a structure $\mathcal{A}$, a set $B \subseteq A$ and $n \in \mathbb{N}$, we denote by $\mathrm{Df}^m(\mathcal{A}, B)$ the set of all relations $D \subseteq A^n$ definable in $\mathcal{A}$ with parameters from B

Observation $\mathrm{Df}^m(\mathcal{A}, B)$ is closed under complement, union, intersection, and contains $\emptyset, A^n$, i.e., it is a *subalgebra* of the set algebra $\underline{\mathcal{P}}(A^n)$.

Application: Database queries

title	director	year
Avengers: Endgame	Russo	2019
Avatar	Cameron	2009
Titanic	Cameron	1997
...		

Table 1: Movies

cinema	title	time
Atlas	Avengers: Endgame	19:30
Světozor	Avengers: Endgame	18:30
Světozor	Titanic	21:00
...		

Table 2: Program

Where and when can I see a Cameron movie?

```
SELECT Program.cinema, Program.time FROM Movies, Pogram
WHERE Movies.title = Program.title AND director = 'Cameron';
```

This is equivalent to $\varphi^{\mathcal{D}}(x_{cin}, x_{time})$ where

$$\varphi(x_{cin}, x_{time}) = (\exists x_{title})(\exists x_{year})(M(x_{title}, c_{\text{Cameron}}, x_{year}) \wedge P(x_{cin}, x_{title}, x_{time}))$$

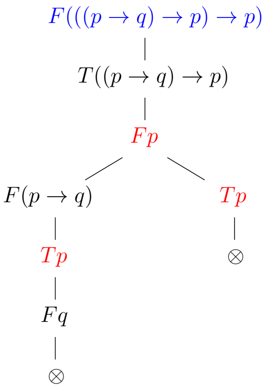
in the structure $\mathcal{D} = \langle D, M^D, P^D, \{c_d \mid d \in D\}\rangle$ where $D = \{$'Avengers: Endgame', 'Russo', '2019', 'Avatar',...,'21:00'$\}$, M^D and P^D are given by rows of the tables, and $c_d^{\mathcal{D}} = d$ for all $d \in D$.

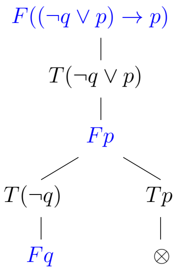
Table of Contents

Tableau method in propositional logic - a review

- A tableau is a binary tree that represents a search for a *counterexample*.
- Nodes are labeled by entries, i.e. formulas with a sign T / F that represents an assumption that the formula is true / false in some model.
- If this assumption is correct, then it is correct also for all the entries in some branch below that came from this entry.
- A branch is contradictory (it fails) if it contains $T\psi$, $F\psi$ for some ψ.
- A proof of formula φ is a contradictory tableau with root $F\varphi$, i.e. a tableau in which every branch is contradictory. If φ has a proof, it is valid.
- If a counterexample exists, there will be a branch in a finished tableau that provides us with this counterexample, but this branch can be infinite.
- We can construct a systematic tableau that is always finished.
- If φ is valid, the systematic tableau for φ is contradictory, i.e. it is a proof of φ; and in this case, it is also finite.

Tableau method in propositional logic - examples





- (a) A tableau proof of the formula $((p \to q) \to p) \to p$.
- (b) A finished tableau for $(\neg q \vee p) \to p$. The left branch provides us with a counterexample $v(p) = v(q) = 0$.

Tableau method in predicate logic - what is different

- Formulas in entries will always be sentences (closed formulas), i.e. formulas without free variables.
- We add new atomic tableaux for quantifiers.
- In these tableaux we substitute ground terms for quantified variables following certain rules.
- We extend the language by new (auxiliary) constant symbols (countably many) to represent *"witnesses"* of entries $T(\exists x)\varphi(x)$ and $F(\forall x)\varphi(x)$.
- In a finished noncontradictory branch containing an entry $T(\forall x)\varphi(x)$ or $F(\exists x)\varphi(x)$ we have instances $T\varphi(x/t)$ resp. $F\varphi(x/t)$ for every ground term t (of the extended language).

Assumptions

1. *The formula φ that we want to prove (or refute) is a sentence.* If not, we replace φ with its universal closure φ', since for every theory T,
$$T \models \varphi \quad \text{if and only if} \quad T \models \varphi'.$$

2. *We prove from a theory in a closed form, i.e. every axiom is a sentence.*
By replacing every axiom ψ with its universal closure ψ' we obtain an equivalent theory as for every structure $\mathcal{A}$ (of the given language L),
$$\mathcal{A} \models \psi \quad \text{if and only if} \quad \mathcal{A} \models \psi'.$$

3. *The language L is countable.* Then every theory of L is countable.
We denote by L_C the extension of L by new constant symbols $c_0, c_1, \ldots$ (countably many). Then there are countable many ground terms of L_C.
Let t_i denote the i-th ground term (in some fixed enumeration).

4. *First, we assume that the language is without equality.*

Tableaux in predicate logic - examples

$$
\begin{array}{c}
F((\exists x)\neg P(x) \to \neg(\forall x)P(x)) \\
| \\
T(\exists x)\neg P(x) \\
| \\
F(\neg(\forall x)P(x)) \\
| \\
T(\forall x)P(x) \\
| \\
T(\neg P(c)) \quad c \ \text{new} \\
| \\
FP(c) \\
| \\
T(\forall x)P(x) \\
| \\
TP(c) \\
| \\
\otimes
\end{array}
$$

$$
\begin{array}{c}
F(\neg(\forall x)P(x) \to (\exists x)\neg P(x)) \\
| \\
T(\neg(\forall x)P(x)) \\
| \\
F(\exists x)\neg P(x) \\
| \\
F(\forall x)P(x) \\
| \\
FP(d) \quad d \ \text{new} \\
| \\
F(\exists x)\neg P(x) \\
| \\
F(\neg P(d)) \\
| \\
TP(d) \\
| \\
\otimes
\end{array}
$$

Atomic tableaux - original

An *atomic tableau* is one of the following trees (labeled by entries), where α is any atomic sentence and φ, ψ are any sentences, all of language L_C.

$T\alpha$	$F\alpha$	$T(\varphi \land \psi)$ | $T\varphi$ | $T\psi$	$F(\varphi \land \psi)$ / \\ $F\varphi$ $F\psi$	$T(\varphi \lor \psi)$ / \\ $T\varphi$ $T\psi$	$F(\varphi \lor \psi)$ | $F\varphi$ | $F\psi$
$T(\neg\varphi)$ | $F\varphi$	$F(\neg\varphi)$ | $T\varphi$	$T(\varphi \to \psi)$ / \\ $F\varphi$ $T\psi$	$F(\varphi \to \psi)$ | $T\varphi$ | $F\psi$	$T(\varphi \leftrightarrow \psi)$ / \\ $T\varphi$ $F\varphi$ | | $T\psi$ $F\psi$	$F(\varphi \leftrightarrow \psi)$ / \\ $T\varphi$ $F\varphi$ | | $F\psi$ $T\psi$

Atomic tableaux - new

Atomic tableaux are also the following trees (labeled by entries), where φ is any formula of the language L_C with a free variable x, t is any ground term of L_C and c is a new constant symbol from $L_C \setminus L$.

$\sharp$ $T(\forall x)\varphi(x)$ $\vert$ $T\varphi(x/t)$ for any ground term t of L_C	$*$ $F(\forall x)\varphi(x)$ $\vert$ $F\varphi(x/c)$ for a *new* constant c	$*$ $T(\exists x)\varphi(x)$ $\vert$ $T\varphi(x/c)$ for a *new* constant c	$\sharp$ $F(\exists x)\varphi(x)$ $\vert$ $F\varphi(x/t)$ for any ground term t of L_C

Remark *The constant symbol c represents a "witness" of the entry $T(\exists x)\varphi(x)$ or $F(\forall x)\varphi(x)$. Since we need that no prior demands are put on c, we specify (in the definition of a tableau) which constant symbols c may be used.*

Tableau

A *finite tableau* from a theory T is a binary tree labeled with entries described

1. every atomic tableau is a finite tableau from T, whereas in case $(*)$ we may use any constant symbol $c \in L_C \setminus L$,
2. if E is an entry on a branch B in a finite tableau from T, then by adjoining the atomic tableau for E at the end of branch B we obtain (again) a finite tableau from T, whereas in case $(*)$ we may use only a constant symbol $c \in L_C \setminus L$ that does not appear on B,
3. if B is a branch in a finite tableau from T and $\varphi \in T$, then by adjoining $T\varphi$ at the end of branch B we obtain (again) a finite tableau from T.
4. every finite tableau from T is formed by finitely many steps (i), (ii), (iii).

A *tableau* from T is a sequence $\tau_0, \tau_1, \ldots, \tau_n, \ldots$ of finite tableaux from T such that τ_{n+1} is formed from τ_n by (ii) or (iii), formally $\tau = \cup\tau_n$.

Construction of tableaux

$$F((\exists x)\neg P(x) \to \neg(\forall x)P(x))$$
$$T(\exists x)\neg P(x)$$
$$F(\neg(\forall x)P(x))$$

$$F(\neg(\forall x)P(x))$$
$$T(\forall x)P(x)$$

$$T(\exists x)\neg P(x)$$
$$T(\neg P(c)) \quad c \text{ new}$$

$$T(\neg P(c))$$
$$FP(c)$$

$$T(\forall x)P(x)$$
$$TP(c) \quad \text{choose } t = c$$

$$\otimes$$

$$F(\neg(\forall x)P(x) \to (\exists x)\neg P(x))$$
$$T(\neg(\forall x)P(x))$$
$$F(\exists x)\neg P(x)$$

$$T(\neg(\forall x)P(x))$$
$$F(\forall x)P(x)$$

$$F(\forall x)P(x)$$
$$FP(d) \quad d \text{ new}$$

$$F(\exists x)\neg P(x)$$
$$F(\neg P(d)) \quad \text{choose } t = d$$

$$F(\neg P(d))$$
$$TP(d)$$

$$\otimes$$

Convention

$$F((\exists x)\neg P(x) \to \neg(\forall x)P(x))$$
$$T(\exists x)\neg P(x)$$
$$F(\neg(\forall x)P(x))$$
$$T(\forall x)P(x)$$
$$T(\neg P(c))\ c\ \text{ new}$$
$$FP(c)$$
$$T(\forall x)P(x)$$
$$TP(c)$$
$$\otimes$$

$$F(\neg(\forall x)P(x) \to (\exists x)\neg P(x))$$
$$T(\neg(\forall x)P(x))$$
$$F(\exists x)\neg P(x)$$
$$F(\forall x)P(x)$$
$$FP(d) \quad d\ \text{ new}$$
$$F(\exists x)\neg P(x)$$
$$F(\neg P(d))$$
$$TP(d)$$
$$\otimes$$

We will not write the entry that is expanded again on the branch, except in cases when the entry is in the form of $T(\forall x)\varphi(x)$ or $F(\exists x)\varphi(x)$.

Tableau proof

- A branch B in a tableau τ is *contradictory* if it contains entries $T\varphi$ and $F\varphi$ for some sentence φ, otherwise B is *noncontradictory*.
- A tableau τ is *contradictory* if every branch in τ is contradictory.
- A *tableau proof* (*proof by tableau*) of a sentence φ from a theory T is a contradictory tableau from T with $F\varphi$ in the root.
- A sentence φ is *(tableau) provable* from T, denoted by $T \vdash \varphi$, if it has a tableau proof from T.
- A *refutation* of a sentence φ by *tableau* from a theory T is a contradictory tableau from T with the root entry $T\varphi$.
- A sentence φ is *(tableau) refutable* from T if it has a refutation by tableau from T, i.e. $T \vdash \neg\varphi$.

Examples

- $F((\forall x)(P(x) \to Q(x)) \to ((\forall x)P(x) \to (\forall x)Q(x)))$
  - $T(\forall x)(P(x) \to Q(x))$
  - $F((\forall x)P(x) \to (\forall x)Q(x))$
  - $T(\forall x)P(x)$
  - $F(\forall x)Q(x)$
  - $FQ(c)$ $\quad c$ new
  - $T(\forall x)P(x)$
  - $TP(c)$
  - $T(\forall x)(P(x) \to Q(x))$
  - $T(P(c) \to Q(c))$
    - $FP(c)$
      - $\otimes$
    - $TQ(c)$
      - $\otimes$

- $F((\forall x)(\varphi(x) \land \psi(x)) \leftrightarrow ((\forall x)\varphi(x) \land (\forall x)\psi(x)))$
  - $T((\forall x)(\varphi(x) \land \psi(x)))$
    - $F((\forall x)\varphi(x) \land (\forall x)\psi(x))$
      - $F(\forall x)\varphi(x)$
        - $F\varphi(c)$ $\quad c$ new
        - $T((\forall x)(\varphi(x) \land \psi(x)))$
        - $T(\varphi(c) \land \psi(c))$
        - $T\varphi(c)$
        - $T\psi(c)$
        - $\otimes$
      - $F(\forall x)\psi(x)$
        - $F\psi(d)$ $\quad d$ new
        - $T((\forall x)(\varphi(x) \land \psi(x)))$
        - $T(\varphi(d) \land \psi(d))$
        - $T\varphi(d)$
        - $T\psi(d)$
        - $\otimes$
  - $F((\forall x)(\varphi(x) \land \psi(x)))$
    - $T((\forall x)\varphi(x) \land (\forall x)\psi(x))$
    - $T(\forall x)\varphi(x)$
    - $T(\forall x)\psi(x)$
    - $F(\varphi(e) \land \psi(e))$ $\quad e$ new
      - $F\varphi(e)$
        - $T(\forall x)\varphi(x)$
        - $T\varphi(e)$
        - $\otimes$
      - $F\psi(e)$
        - $T(\forall x)\psi(x)$
        - $T\psi(e)$
        - $\otimes$

Finished tableau

A finished noncontradictory branch should provide us with a counterexample.

An occurrence of an entry E in a node B of a tableau τ is *i-th* if B has exactly $i-1$ predecessors labeled by E; and is *reduced* on a branch B through B if

a. E is neither in form of $T(\forall x)\varphi(x)$ nor $F(\exists x)\varphi(x)$ and E occurs on B as a root of an atomic tableau, i.e. it was already expanded on B, or
b. E is in form of $T(\forall x)\varphi(x)$ or $F(\exists x)\varphi(x)$, E has an $(i+1)$-th occurrence on B, and B contains an entry $T\varphi(x/t_i)$ resp. $F\varphi(x/t_i)$ where t_i is the i-th ground term (of the language L_C).

Let B be a branch in a tableau τ from a theory T. We say that

- B is *finished* if it is contradictory, or every occurrence of an entry on B is reduced on B and, moreover, B contains $T\varphi$ for every $\varphi \in T$,
- τ is *finished* if every branch in τ is finished.

Systematic tableau - construction

Let R be an entry and $T = \{\varphi_0, \varphi_1, \dots\}$ be a (possibly infinite) theory.

(1) We take the atomic tableau for R as τ_0. In case $(*)$ we choose any $c \in L_C \setminus L$, in case $(\sharp)$ we take t_1 for t. Proceed as follows:

(2) Let B be the leftmost node in the smallest possible level in τ_n containing an occurrence of an entry E that is not reduced on some noncontradictory branch through B. (If B doesn't exist, set $\tau'_n = \tau_n$.)

(3a) If E is neither $T(\forall x)\varphi(x)$ nor $F(\exists x)\varphi(x)$, let τ'_n be the tableau obtained from τ_n by adjoining the atomic tableau for E to every noncontr. branch through B. In case $(*)$, choose c_i with smallest i.

(3b) If E is $T(\forall x)\varphi(x)$ or $F(\exists x)\varphi(x)$ and it has i-th occurrence in B, let τ'_n be the tableau obtained from τ_n by adjoining atomic tableau for E to every noncontr. branch through B, where we take the term t_i for t.

(4) Let τ_{n+1} be the tableau obtained from τ'_n by adjoining $T\varphi_n$ to every noncontradictory branch that does not contain $T\varphi_n$ yet. (If φ_n does not exist, we take $\tau_{n+1} = \tau'_n$.)

The *systematic tableau* for R from T is the result of this process: $\tau = \cup \tau_n$

Systematic tableau - an example

$$T((\exists y)(\neg R(y,y) \vee P(y,y)) \wedge (\forall x)R(x,x))$$

$$T(\exists y)(\neg R(y,y) \vee P(y,y))$$

$$T(\forall x)R(x,x)$$

$$T(\neg R(c_0,c_0) \vee P(c_0,c_0)) \qquad c_0 \text{ new}$$

$$T(\forall x)R(x,x)$$

$$TR(c_0,c_0) \qquad \text{(assuming that } t_1 = c_0\text{)}$$

Left branch:

$$T(\neg R(c_0,c_0))$$

$$T(\forall x)R(x,x)$$

$$TR(t_2,t_2)$$

$$FR(c_0,c_0)$$

$$\otimes$$

Right branch:

$$TP(c_0,c_0)$$

$$T(\forall x)R(x,x)$$

$$TR(t_2,t_2)$$

$$T(\forall x)R(x,x)$$

$$TR(t_3,t_3)$$

$$\vdots$$

Systematic tableau - being finished

Proposition *Every systematic tableau is finished.*

Proof Let $\tau = \cup \tau_n$ be a systematic tableau from $T = \{\varphi_0, \varphi_1, \dots\}$ with root R and let E be an entry in a node B of the tableau τ.

- There are only finitely many entries in τ in levels up to the level of B.
- If the occurrence of E in B was unreduced on some noncontradictory branch in τ, it would be found in some step (2) and reduced by ($3a$), ($3b$).
- By step (4) every $\varphi_n \in T$ will be (no later than) in τ_{n+1} on every noncontradictory branch.
- Hence the systematic tableau τ has all branches finished. $\square$

Proposition *If a systematic tableau τ is a proof (from a theory T), it is finite.*

Proof Suppose that τ is infinite. Then by König's lemma, τ contains an infinite branch. This branch is noncontradictory since in the construction only noncontradictory branches are prolonged. But this contradicts the assumption that τ is a contradictory tableau. $\square$

Equality

Axioms of equality for a language L with equality are

1. $x = x$
2. $x_1 = y_1 \ \wedge \ \cdots \ \wedge \ x_n = y_n \quad \rightarrow \quad f(x_1, \ldots, x_n) = f(y_1, \ldots, y_n)$
   for each n-ary function symbol f of the language L.
3. $x_1 = y_1 \ \wedge \ \cdots \ \wedge \ x_n = y_n \quad \rightarrow \quad (R(x_1, \ldots, x_n) \rightarrow R(y_1, \ldots, y_n))$
   for each n-ary relation symbol R of the language L including $=$.

A *tableau proof* from a theory T in a language L *with equality* is a tableau proof from T^* where T^* denotes the extension of T by adding axioms of equality for L *(resp. their universal closures).*

Remark *In context of logic programming the equality often has other meaning than in mathematics (identity). For example in Prolog, $t_1 = t_2$ means that t_1 and t_2 are unifiable.*

Congruence and quotient structure

An equivalence $\sim$ on A, $f : A^n \to A$, and $R \subseteq A^n$, where $n \in \mathbb{N}$, is:

- a *congruence for the function* f if for every $x_1, \dots, x_n, y_1, \dots, y_n \in A$
$$x_1 \sim y_1 \ \wedge \ \cdots \ \wedge \ x_n \sim y_n \quad \Rightarrow \quad f(x_1, \dots, x_n) \sim f(y_1, \dots, y_n),$$
- a *congruence for the relation* R if for every $x_1, \dots, x_n, y_1, \dots, y_n \in A$
$$x_1 \sim y_1 \ \wedge \ \cdots \ \wedge \ x_n \sim y_n \quad \Rightarrow \quad (R(x_1, \dots, x_n) \Leftrightarrow R(y_1, \dots, y_n)).$$

Let an equivalence $\sim$ on A be a congruence for every function and relation in a structure $\mathcal{A} = \langle A, \mathcal{F}^A, \mathcal{R}^A \rangle$ of language $L = \langle \mathcal{F}, \mathcal{R} \rangle$. The *quotient* (*structure*) of $\mathcal{A}$ by $\sim$ is the structure $\mathcal{A}/\sim\, = \langle A/\sim, \mathcal{F}^{A/\sim}, \mathcal{R}^{A/\sim} \rangle$ where

$$f^{A/\sim}([x_1]_\sim, \dots, [x_n]_\sim) = [f^A(x_1, \dots, x_n)]_\sim$$
$$R^{A/\sim}([x_1]_\sim, \dots, [x_n]_\sim) \Leftrightarrow R^A(x_1, \dots, x_n)$$

for each $f \in \mathcal{F}$, $R \in \mathcal{R}$, and $x_1, \dots, x_n \in A$, i.e. the functions and relations are defined from $\mathcal{A}$ using representatives.

Example: $\underline{\mathbb{Z}}_p$ is the quotient of $\underline{\mathbb{Z}} = \langle \mathbb{Z}, +, -, 0 \rangle$ by the congruence modulo E.

Role of axioms of equality

Let $\mathcal{A}$ be a structure of a language L in which the equality is interpreted as a relation $=^A$ satisfying the axioms of equality for L, i.e. not necessarily the identity relation.

1. From axioms (i) and (iii) it follows that the relation $=^A$ is an equivalence.
2. Axioms (ii) and (iii) express that the relation $=^A$ is a congruence for every function and relation in $\mathcal{A}$.
3. If $\mathcal{A} \models T^*$ then also $(\mathcal{A}/{=^A}) \models T^*$ where $\mathcal{A}/{=^A}$ is the quotient of $\mathcal{A}$ by $=^A$. Moreover, the equality is interpreted in $\mathcal{A}/{=^A}$ as the identity relation.

On the other hand, in every model in which the equality is interpreted as the identity relation, all axioms of equality evidently hold.

Soundness

We say that a model $\mathcal{A}$ *agrees* with an entry E, if E is $T\varphi$ and $\mathcal{A} \models \varphi$ or if E is $F\varphi$ and $\mathcal{A} \models \neg\varphi$, i.e. $\mathcal{A} \not\models \varphi$. Moreover, $\mathcal{A}$ *agrees* with a branch B if $\mathcal{A}$ agrees with every entry on B.

Lemma *Let $\mathcal{A}$ be a model of a theory T of a language L that agrees with the root entry R in a tableau $\tau = \cup\tau_n$ from T. Then $\mathcal{A}$ can be expanded to the language L_C so that it agrees with some branch B in τ.*

Remark *It suffices to expand $\mathcal{A}$ only by constants c^A such that $c \in L_C \setminus L$ occurs on B, other constants may be defined arbitrarily.*

Proof By induction on n we find a branch B_n in τ_n and an expansion $\mathcal{A}_n$ of $\mathcal{A}$ by constants c^A for all $c \in L_C \setminus L$ on B_n s.t. $\mathcal{A}_n$ agrees with B_n and $B_{n-1} \subseteq B_n$. Assume we have a branch B_n in τ_n and an expansion $\mathcal{A}_n$ that agrees with B_n.

- If τ_{n+1} is formed from τ_n without extending the branch B_n, we take $B_{n+1} = B_n$ and $\mathcal{A}_{n+1} = \mathcal{A}_n$.
- If τ_{n+1} is formed from τ_n by appending $T\varphi$ to B_n for some $\varphi \in T$, let B_{n+1} be this branch and $\mathcal{A}_{n+1} = \mathcal{A}_n$. Since $\mathcal{A} \models \varphi$, $\mathcal{A}_{n+1}$ agrees with B_{n+1}.

Soundness - proof (cont.)

- Otherwise τ_{n+1} is formed from τ_n by appending an atomic tableau to B_n for some entry E on B_n. By induction we know that $\mathcal{A}_n$ agrees with E.

(i) If E is formed by a logical connective, we take $\mathcal{A}_{n+1} = \mathcal{A}_n$ and verify that B_n can always be extended to a branch B_{n+1} agreeing with $\mathcal{A}_{n+1}$.

(ii) If E is in form $T(\forall x)\varphi(x)$, let B_{n+1} be the (unique) extension of B_n to a branch in τ_{n+1}, i.e. by the entry $T\varphi(x/t)$. Let $\mathcal{A}_{n+1}$ be any expansion of by new constants from t. Since $\mathcal{A}_n \models (\forall x)\varphi(x)$, we have $\mathcal{A}_{n+1} \models \varphi(x/t)$. Analogously for E in form $F(\exists x)\varphi(x)$.

(iii) If E is in form $T(\exists x)\varphi(x)$, let B_{n+1} be the (unique) extension of B_n to a branch in τ_{n+1}, i.e. by the entry $T\varphi(x/c)$. Since $\mathcal{A}_n \models (\exists x)\varphi(x)$, there is some $a \in A$ with $\mathcal{A}_n \models \varphi(x)[e(x/a)]$ for every assignment e. Let $\mathcal{A}_{n+1}$ be the expansion of $\mathcal{A}_n$ by a new constant $c^A = a$. Then $\mathcal{A}_{n+1} \models \varphi(x/c)$. Analogously for E in form $F(\forall x)\varphi(x)$.

The base step for $n = 0$ follows from similar analysis of atomic tableaux for the root entry R applying the assumption that $\mathcal{A}$ agrees with R. $\square$

Theorem on soundness

We will show that the tableau method in predicate logic is sound.

Theorem *For every theory T and sentence φ, if φ is tableau provable from T, then φ is valid in T, i.e. $T \vdash \varphi \;\Rightarrow\; T \models \varphi$.*

Proof

- Let φ be tableau provable from a theory T, i.e. there is a contradictory tableau τ from T with the root entry $F\varphi$.
- Suppose for a contradiction that φ is not valid in T, i.e. there exists a model $\mathcal{A}$ of the theory T in which φ is not true (a counterexample).
- Since $\mathcal{A}$ agrees with the root entry $F\varphi$, by the previous lemma, $\mathcal{A}$ can be expanded to the language L_C so that it agrees with some branch in τ.
- But this is impossible, since every branch of τ is contradictory, i.e. it contains a pair of entries $T\psi$, $F\psi$ for some sentence ψ. $\square$

The canonical model

From a noncontradictory branch B of a finished tableau we build a model that agrees with B. We build it on available (syntactical) objects - ground terms.

Let B be a noncontradictory branch of a finished tableau from a theory T of a language $L = \langle \mathcal{F}, \mathcal{R} \rangle$. The *canonical model* from B is the L_C-structure $\mathcal{A} = \langle A, \mathcal{F}^A, \mathcal{R}^A \rangle$ where

1. A is the set of all ground terms of the language L_C,
2. $f^A(t_{i_1}, \ldots, t_{i_n}) = f(t_{i_1}, \ldots, t_{i_n})$ for every n-ary function symbol $f \in \mathcal{F} \cup (L_C \setminus L)$ a $t_{i_1}, \ldots, t_{i_n} \in A$.
3. $R^A(t_{i_1}, \ldots, t_{i_n}) \Leftrightarrow TR(t_{i_1}, \ldots, t_{i_n})$ is an entry on B for every n-ary relation symbol $R \in \mathcal{R}$ or equality and $t_{i_1}, \ldots, t_{i_n} \in A$.

Remark *The expression $f(t_{i_1}, \ldots, t_{i_n})$ on the right side of (2) is a ground term of L_C, i.e. an element of A. Informally, to indicate that it is a syntactical object*

$$f^A(t_{i_1}, \ldots, t_{i_n}) = \text{“}f(t_{i_1}, \ldots, t_{i_n})\text{”}$$

The canonical model - an example

Let $T = \{(\forall x)R(f(x))\}$ be a theory of a language $L = \langle R, f, d\rangle$. The systematic tableau for $F\neg R(d)$ from T contains a single branch B, which is noncontradictory.

The canonical model $\mathcal{A} = \langle A, R^A, f^A, d^A, c_i^A\rangle_{i\in\mathbb{N}}$ from B is for language L_C and

$$A = \{d, f(d), f(f(d)), \ldots, c_0, f(c_0), f(f(c_0)), \ldots, c_1, f(c_1), f(f(c_1)), \ldots\},$$
$$d^A = d, \quad c_i^A = c_i \text{ for } i \in \mathbb{N},$$
$$f^A(d) = \text{“}f(d)\text{”}, \quad f^A(f(d)) = \text{“}f(f(d))\text{”}, \quad f^A(f(f(d))) = \text{“}f(f(f(d)))\text{”}, \ \ldots$$
$$R^A = \{d, f(d), f(f(d)), \ldots, f(c_0), f(f(c_0)), \ldots, f(c_1), f(f(c_1)), \ldots\}.$$

The reduct of $\mathcal{A}$ to the language L is $\mathcal{A}' = \langle A, R^A, f^A, d^A\rangle$.

The canonical model with equality

If L is with equality, T^* is the extension of T by axioms of equality for L.

If we require that the equality is interpreted as the identity, we have to take the quotient of the canonical model $\mathcal{A}$ by the congruence $=^A$.

By (3), for the relation $=^A$ in $\mathcal{A}$ from B it holds that for every $t_{i_1}, t_{i_2} \in A$,

$$t_{i_1} =^A t_{i_2} \quad \Leftrightarrow \quad T(t_{i_1} = t_{i_2}) \text{ is an entry on } V.$$

Since B is finished and contains the axioms of equality, the relation $=^A$ is a congruence for all functions and relations in $\mathcal{A}$.

The *canonical model with equality* from B is the quotient $\mathcal{A}/{=^A}$.

Observation *For every formula φ,*

$$\mathcal{A} \models \varphi \quad \Leftrightarrow \quad (\mathcal{A}/{=^A}) \models \varphi,$$

where $=$ is interpreted in $\mathcal{A}$ by the relation $=^A$, while in $\mathcal{A}/{=^A}$ by the identity.

Remark *$\mathcal{A}$ is a countably infinite model, but $\mathcal{A}/{=^A}$ can be finite.*

The canonical model with equality - an example

Let $T = \{(\forall x)R(f(x)),\ (\forall x)(x = f(f(x)))\}$ be of $L = \langle R, f, d\rangle$ with equality. The systematic tableau for $F\neg R(d)$ from T^* contains a noncontradictory B.

In the canonical model $\mathcal{A} = \langle A, R^A, =^A, f^A, d^A, c_i^A\rangle_{i\in\mathbb{N}}$ from B we have that

$$s =^A t \quad \Leftrightarrow \quad t = f(\cdots(f(s)\cdots) \ \text{ or } \ s = f(\cdots(f(t)\cdots),$$

where f is applied $2i$-times for some $i \in \mathbb{N}$.

The canonical model with equality from B is

$\mathcal{B} = (\mathcal{A}/=^A) = \langle A/=^A, R^B, f^B, d^B, c_i^B\rangle_{i\in\mathbb{N}}$ where

$$(A/=^A) = \{[d]_{=^A}, [f(d)]_{=^A}, [c_0]_{=^A}, [f(c_0)]_{=^A}, [c_1]_{=^A}, [f(c_1)]_{=^A}, \ldots\},$$

$$d^B = [d]_{=^A}, \quad c_i^B = [c_i]_{=^A} \ \text{ for } \ i \in \mathbb{N},$$

$$f^B([d]_{=^A}) = [f(d)]_{=^A}, \quad f^B([f(d)]_{=^A}) = [f(f(d))]_{=^A} = [d]_{=^A}, \quad \ldots$$

$$R^B = (A/=^A).$$

The reduct of $\mathcal{B}$ to the language L is $\mathcal{B}' = \langle A/=^A, R^B, f^B, d^B\rangle$.

Completeness

Lemma *Canonical model $\mathcal{A}$ from a noncontr. finished B agrees with B.*

Proof By induction on the structure of a sentence in an entry on B.

- For atomic φ, if $T\varphi$ is on B, then $\mathcal{A} \models \varphi$ by (3). If $F\varphi$ is on B, then $T\varphi$ is not on B since B is noncontradictory, so $\mathcal{A} \models \neg\varphi$ by (3).
- If $T(\varphi \wedge \psi)$ is on B, then $T\varphi$ and $T\psi$ are on B since B is finished. By induction, $\mathcal{A} \models \varphi$ and $\mathcal{A} \models \psi$, and thus $\mathcal{A} \models \varphi \wedge \psi$.
- If $F(\varphi \wedge \psi)$ is on B, then $F\varphi$ or $F\psi$ is on B since B is finished. By induction, $\mathcal{A} \models \neg\varphi$ or $\mathcal{A} \models \neg\psi$, and thus $\mathcal{A} \models \neg(\varphi \wedge \psi)$.
- For other connectives similarly as in previous two cases.
- If $T(\forall x)\varphi(x)$ is on B, then $T\varphi(x/t)$ is on B for every $t \in A$ since B is finished. By induction, $\mathcal{A} \models \varphi(x/t)$ for every $t \in A$, and thus $\mathcal{A} \models (\forall x)\varphi(x)$. Similarly for $F(\exists x)\varphi(x)$ on B.
- If $T(\exists x)\varphi(x)$ is on B, then $T\varphi(x/c)$ is on B for some $c \in A$ since B is finished. By induction, $\mathcal{A} \models \varphi(x/c)$, and thus $\mathcal{A} \models (\exists x)\varphi(x)$. Similarly for $F(\forall x)\varphi(x)$ on B. $\square$

Theorem on completeness

We will show that the tableau method in predicate logic is complete.

Theorem *For every theory T and sentence φ, if φ is valid in T, then φ is tableau provable from T, i.e. $T \models \varphi \;\Rightarrow\; T \vdash \varphi$.*

Proof Let φ be valid in T. We will show that an arbitrary finished tableau (e.g. systematic) τ from a theory T with the root entry $F\varphi$ is contradictory.

- If not, then there is some noncontradictory branch B in τ.
- By the previous lemma, there is a structure $\mathcal{A}$ for L_C that agrees with B, in particular with the root entry $F\varphi$, i.e. $\mathcal{A} \models \neg\varphi$.
- Let $\mathcal{A}'$ be the reduct of $\mathcal{A}$ to the language L. Then $\mathcal{A}' \models \neg\varphi$.
- Since B is finished, it contains $T\psi$ for every $\psi \in T$.
- Thus $\mathcal{A}'$ is a model of T (as $\mathcal{A}'$ agrees with $T\psi$ for every $\psi \in T$).
- But this contradicts the assumption that φ is valid in T.

Therefore the tableau τ is a proof of φ from T. $\square$

Properties of theories

We introduce syntactic variants of previous semantical definitions.

Let T be a theory of a language L. If a sentence φ is provable from T, we say that φ is a *theorem* of T. The set of theorems of T is denoted by

$$\mathrm{Thm}^L(T) = \{\varphi \in \mathrm{Fm}_L \mid T \vdash \varphi\}.$$

We say that a theory T is

- *inconsistent* if $T \vdash \perp$, otherwise T is *consistent*,
- *complete* if it is consistent and every sentence is provable or refutable from T, i.e. $T \vdash \varphi$ or $T \vdash \neg\varphi$.
- an *extension* of a theory T' of L' if $L' \subseteq L$ and $\mathrm{Thm}^{L'}(T') \subseteq \mathrm{Thm}^L(T)$, we say that an extension T of a theory T' is *simple* if $L = L'$; and
  conservative if $\mathrm{Thm}^{L'}(T') = \mathrm{Thm}^L(T) \cap \mathrm{Fm}_{\mathrm{L}'}$,
- *equivalent* with a theory T' if T is an extension of T' and vice-versa.

Corollaries

From the soundness and completeness of the tableau method it follows that these syntactic definitions agree with their semantic variants.

Corollary *For every theory T and sentences φ, ψ of a language L,*

- $T \vdash \varphi$ *if and only if* $T \models \varphi$,
- $\mathrm{Thm}^L(T) = \theta^L(T)$,
- T *is inconsistent if and only if* T *is unsatisfiable, i.e. it has no model,*
- T *is complete if and only if* T *is semantically complete, i.e. it has a single model, up to elementarily equivalence,*
- $T, \varphi \vdash \psi$ *if and only if* $T \vdash \varphi \rightarrow \psi$ *(Deduction theorem).*

Remark Deduction theorem can be proved directly by transformations of tableaux.

Existence of a countable model and compactness

Theorem *Every consistent theory T of a countable language L without equality has a countably infinite model.*

Proof Let τ be the systematic tableau from T with $F\bot$ in the root. Since τ is finished and contains a noncontradictory branch B as $\bot$ is not provable from T, there exists a canonical model $\mathcal{A}$ from B. Since $\mathcal{A}$ agrees with B, its reduct to the language L is a desired countably infinite model of T. $\square$

Remark *This is a weak version of so called Löwenheim-Skolem theorem.*

In a countable language with equality the canonical model with equality is countable (i.e. finite or countably infinite).

Theorem *A theory T has a model iff every finite subset of T has a model.*

Proof The implication from left to right is obvious. If T has no model, then it is inconsistent, i.e. $\bot$ is provable by a systematic tableau τ from T. Since τ is finite, $\bot$ is provable from some finite $T' \subseteq T$, i.e. T' has no model. $\square$

NAIL062 Propositional & Predicate Logic: Lecture 9

Slides by Petr Gregor with minor modifications by Jakub Bulín

November 30, 2020

Finished tableau

A finished noncontradictory branch should provide us with a counterexample.

An occurrence of an entry E in a node B of a tableau τ is *i-th* if B has exactly $i-1$ predecessors labeled by E; and is *reduced* on a branch B through B if

a. E is neither in form of $T(\forall x)\varphi(x)$ nor $F(\exists x)\varphi(x)$ and E occurs on B as a root of an atomic tableau, i.e. it was already expanded on B, or

b. E is in form of $T(\forall x)\varphi(x)$ or $F(\exists x)\varphi(x)$, E has an $(i+1)$-th occurrence on B, and B contains an entry $T\varphi(x/t_i)$ resp. $F\varphi(x/t_i)$ where t_i is the i-th ground term (of the language L_C).

Let B be a branch in a tableau τ from a theory T. We say that

- B is *finished* if it is contradictory, or every occurrence of an entry on B is reduced on B and, moreover, B contains $T\varphi$ for every $\varphi \in T$,
- τ is *finished* if every branch in τ is finished.

Systematic tableau - construction

Let R be an entry and $T = \{\varphi_0, \varphi_1, \dots\}$ be a (possibly infinite) theory.

(1) We take the atomic tableau for R as τ_0. In case $(*)$ we choose any $c \in L_C \setminus L$, in case $(\sharp)$ we take t_1 for t. Proceed as follows:

(2) Let B be the leftmost node in the smallest possible level in τ_n containing an occurrence of an entry E that is not reduced on some noncontradictory branch through B. (If B doesn't exist, set $\tau'_n = \tau_n$.)

(3a) If E is neither $T(\forall x)\varphi(x)$ nor $F(\exists x)\varphi(x)$, let τ'_n be the tableau obtained from τ_n by adjoining the atomic tableau for E to every noncontr. branch through B. In case $(*)$, choose c_i with smallest i.

(3b) If E is $T(\forall x)\varphi(x)$ or $F(\exists x)\varphi(x)$ and it has i-th occurrence in B, let τ'_n be the tableau obtained from τ_n by adjoining atomic tableau for E to every noncontr. branch through B, where we take the term t_i for t.

(4) Let τ_{n+1} be the tableau obtained from τ'_n by adjoining $T\varphi_n$ to every noncontradictory branch that does not contain $T\varphi_n$ yet. (If φ_n does not exist, we take $\tau_{n+1} = \tau'_n$.)

The *systematic tableau* for R from T is the result of this process: $\tau = \cup\tau_n$

Systematic tableau - an example

$$
\begin{array}{c}
T((\exists y)(\neg R(y,y) \vee P(y,y)) \wedge (\forall x)R(x,x)) \\
| \\
T(\exists y)(\neg R(y,y) \vee P(y,y)) \\
| \\
T(\forall x)R(x,x) \\
| \\
T(\neg R(c_0,c_0) \vee P(c_0,c_0)) \qquad c_0 \ \text{new} \\
| \\
T(\forall x)R(x,x) \\
| \\
TR(c_0,c_0) \qquad \text{(assuming that } t_1 = c_0\text{)} \\
\swarrow \qquad \searrow \\
\begin{array}{cc}
T(\neg R(c_0,c_0)) & TP(c_0,c_0) \\
| & | \\
T(\forall x)R(x,x) & T(\forall x)R(x,x) \\
| & | \\
TR(t_2,t_2) & TR(t_2,t_2) \\
| & | \\
FR(c_0,c_0) & T(\forall x)R(x,x) \\
| & | \\
\otimes & TR(t_3,t_3) \\
 & \vdots
\end{array}
\end{array}
$$

Systematic tableau - being finished

Proposition *Every systematic tableau is finished.*

Proof Let $\tau = \cup\tau_n$ be a systematic tableau from $T = \{\varphi_0, \varphi_1, \dots\}$ with root R and let E be an entry in a node V of the tableau τ.

- There are only finitely many entries in τ in levels up to the level of V.
- If the occurrence of E in V was unreduced on some noncontradictory branch in τ, it would be found in some step (2) and reduced by $(3a)$, $(3b)$.
- By step (4) every $\varphi_n \in T$ will be (no later than) in τ_{n+1} on every noncontradictory branch.
- Hence the systematic tableau τ has all branches finished. $\square$

Proposition *If a systematic tableau τ is a proof (from a theory T), it is finite.*

Proof Suppose that τ is infinite. Then by König's lemma, τ contains an infinite branch. This branch is noncontradictory since in the construction only noncontradictory branches are prolonged. But this contradicts the assumption that τ is a contradictory tableau. $\square$

Equality

Axioms of equality for a language L with equality are

1. $x = x$
2. $x_1 = y_1 \ \wedge \ \cdots \ \wedge \ x_n = y_n \quad \rightarrow \quad f(x_1, \dots, x_n) = f(y_1, \dots, y_n)$
   for each n-ary function symbol f of the language L.
3. $x_1 = y_1 \ \wedge \ \cdots \ \wedge \ x_n = y_n \quad \rightarrow \quad (R(x_1, \dots, x_n) \rightarrow R(y_1, \dots, y_n))$
   for each n-ary relation symbol R of the language L including $=$.

A *tableau proof* from a theory T in a language L *with equality* is a tableau proof from T^* where T^* denotes the extension of T by adding axioms of equality for L *(resp. their universal closures).*

Remark *In context of logic programming the equality often has other meaning than in mathematics (identity). For example in Prolog, $t_1 = t_2$ means that t_1 and t_2 are unifiable.*

Congruence and quotient structure

An equivalence $\sim$ on A, $f : A^n \to A$, and $R \subseteq A^n$, where $n \in \mathbb{N}$, is:

- a *congruence for the function* f if for every $x_1, \dots, x_n, y_1, \dots, y_n \in A$
$$x_1 \sim y_1 \ \wedge \ \cdots \ \wedge \ x_n \sim y_n \quad \Rightarrow \quad f(x_1, \dots, x_n) \sim f(y_1, \dots, y_n),$$
- a *congruence for the relation* R if for every $x_1, \dots, x_n, y_1, \dots, y_n \in A$
$$x_1 \sim y_1 \ \wedge \ \cdots \ \wedge \ x_n \sim y_n \quad \Rightarrow \quad (R(x_1, \dots, x_n) \Leftrightarrow R(y_1, \dots, y_n)).$$

Let an equivalence $\sim$ on A be a congruence for every function and relation in a structure $\mathcal{A} = \langle A, \mathcal{F}^A, \mathcal{R}^A \rangle$ of language $L = \langle \mathcal{F}, \mathcal{R} \rangle$. The *quotient* (*structure*) of $\mathcal{A}$ by $\sim$ is the structure $\mathcal{A}/\sim \, = \langle A/\sim, \mathcal{F}^{A/\sim}, \mathcal{R}^{A/\sim} \rangle$ where

$$f^{A/\sim}([x_1]_\sim, \dots, [x_n]_\sim) = [f^A(x_1, \dots, x_n)]_\sim$$
$$R^{A/\sim}([x_1]_\sim, \dots, [x_n]_\sim) \Leftrightarrow R^A(x_1, \dots, x_n)$$

for each $f \in \mathcal{F}$, $R \in \mathcal{R}$, and $x_1, \dots, x_n \in A$, i.e. the functions and relations are defined from $\mathcal{A}$ using representatives.

Example: $\underline{\mathbb{Z}}_p$ is the quotient of $\underline{\mathbb{Z}} = \langle \mathbb{Z}, +, -, 0 \rangle$ by the congruence modulo E.

Role of axioms of equality

Let $\mathcal{A}$ be a structure of a language L in which the equality is interpreted as a relation $=^A$ satisfying the axioms of equality for L, i.e. not necessarily the identity relation.

1. From axioms (i) and (iii) it follows that the relation $=^A$ is an equivalence.
2. Axioms (ii) and (iii) express that the relation $=^A$ is a congruence for every function and relation in $\mathcal{A}$.
3. If $\mathcal{A} \models T^*$ then also $(\mathcal{A}/{=^A}) \models T^*$ where $\mathcal{A}/{=^A}$ is the quotient of $\mathcal{A}$ by $=^A$. Moreover, the equality is interpreted in $\mathcal{A}/{=^A}$ as the identity relation.

On the other hand, in every model in which the equality is interpreted as the identity relation, all axioms of equality evidently hold.

Soundness

We say that a model $\mathcal{A}$ *agrees* with an entry E, if E is $T\varphi$ and $\mathcal{A} \models \varphi$ or if E is $F\varphi$ and $\mathcal{A} \models \neg\varphi$, i.e. $\mathcal{A} \not\models \varphi$. Moreover, $\mathcal{A}$ *agrees* with a branch B if $\mathcal{A}$ agrees with every entry on B.

Lemma *Let $\mathcal{A}$ be a model of a theory T of a language L that agrees with the root entry R in a tableau $\tau = \cup\tau_n$ from T. Then $\mathcal{A}$ can be expanded to the language L_C so that it agrees with some branch B in τ.*

Remark *It suffices to expand $\mathcal{A}$ only by constants c^A such that $c \in L_C \setminus L$ occurs on B, other constants may be defined arbitrarily.*

Proof By induction on n we find a branch B_n in τ_n and an expansion $\mathcal{A}_n$ of $\mathcal{A}$ by constants c^A for all $c \in L_C \setminus L$ on B_n s.t. $\mathcal{A}_n$ agrees with B_n and $B_{n-1} \subseteq B_n$. Assume we have a branch B_n in τ_n and an expansion $\mathcal{A}_n$ that agrees with B_n.

- If τ_{n+1} is formed from τ_n without extending the branch B_n, we take $B_{n+1} = B_n$ and $\mathcal{A}_{n+1} = \mathcal{A}_n$.
- If τ_{n+1} is formed from τ_n by appending $T\varphi$ to B_n for some $\varphi \in T$, let B_{n+1} be this branch and $\mathcal{A}_{n+1} = \mathcal{A}_n$. Since $\mathcal{A} \models \varphi$, $\mathcal{A}_{n+1}$ agrees with B_{n+1}.

Soundness - proof (cont.)

- Otherwise τ_{n+1} is formed from τ_n by appending an atomic tableau to B_n for some entry E on B_n. By induction we know that $\mathcal{A}_n$ agrees with E.

(i) If E is formed by a logical connective, we take $\mathcal{A}_{n+1} = \mathcal{A}_n$ and verify that B_n can always be extended to a branch B_{n+1} agreeing with $\mathcal{A}_{n+1}$.

(ii) If E is in form $T(\forall x)\varphi(x)$, let B_{n+1} be the (unique) extension of B_n to a branch in τ_{n+1}, i.e. by the entry $T\varphi(x/t)$. Let $\mathcal{A}_{n+1}$ be any expansion of by new constants from t. Since $\mathcal{A}_n \models (\forall x)\varphi(x)$, we have $\mathcal{A}_{n+1} \models \varphi(x/t)$. Analogously for E in form $F(\exists x)\varphi(x)$.

(iii) If E is in form $T(\exists x)\varphi(x)$, let B_{n+1} be the (unique) extension of B_n to a branch in τ_{n+1}, i.e. by the entry $T\varphi(x/c)$. Since $\mathcal{A}_n \models (\exists x)\varphi(x)$, there is some $a \in A$ with $\mathcal{A}_n \models \varphi(x)[e(x/a)]$ for every assignment e. Let $\mathcal{A}_{n+1}$ be the expansion of $\mathcal{A}_n$ by a new constant $c^A = a$. Then $\mathcal{A}_{n+1} \models \varphi(x/c)$. Analogously for E in form $F(\forall x)\varphi(x)$.

The base step for $n = 0$ follows from similar analysis of atomic tableaux for the root entry R applying the assumption that $\mathcal{A}$ agrees with R. $\square$

Theorem on soundness

We will show that the tableau method in predicate logic is sound.

Theorem *For every theory T and sentence φ, if φ is tableau provable from T, then φ is valid in T, i.e. $T \vdash \varphi \;\Rightarrow\; T \models \varphi$.*

Proof

- Let φ be tableau provable from a theory T, i.e. there is a contradictory tableau τ from T with the root entry $F\varphi$.
- Suppose for a contradiction that φ is not valid in T, i.e. there exists a model $\mathcal{A}$ of the theory T in which φ is not true (a counterexample).
- Since $\mathcal{A}$ agrees with the root entry $F\varphi$, by the previous lemma, $\mathcal{A}$ can be expanded to the language L_C so that it agrees with some branch in τ.
- But this is impossible, since every branch of τ is contradictory, i.e. it contains a pair of entries $T\psi$, $F\psi$ for some sentence ψ. □

The canonical model

From a noncontradictory branch B of a finished tableau we build a model that agrees with B. We build it on available (syntactical) objects - ground terms.

Let B be a noncontradictory branch of a finished tableau from a theory T of a language $L = \langle \mathcal{F}, \mathcal{R} \rangle$. The *canonical model* from B is the L_C-structure $\mathcal{A} = \langle A, \mathcal{F}^A, \mathcal{R}^A \rangle$ where

1. A is the set of all ground terms of the language L_C,
2. $f^A(t_{i_1}, \ldots, t_{i_n}) = f(t_{i_1}, \ldots, t_{i_n})$ for every n-ary function symbol $f \in \mathcal{F} \cup (L_C \setminus L)$ a $t_{i_1}, \ldots, t_{i_n} \in A$.
3. $R^A(t_{i_1}, \ldots, t_{i_n}) \Leftrightarrow TR(t_{i_1}, \ldots, t_{i_n})$ is an entry on B for every n-ary relation symbol $R \in \mathcal{R}$ or equality and $t_{i_1}, \ldots, t_{i_n} \in A$.

Remark *The expression $f(t_{i_1}, \ldots, t_{i_n})$ on the right side of (2) is a ground term of L_C, i.e. an element of A. Informally, to indicate that it is a syntactical object*

$$f^A(t_{i_1}, \ldots, t_{i_n}) = \text{“}f(t_{i_1}, \ldots, t_{i_n})\text{”}$$

The canonical model - an example

Let $T = \{(\forall x)R(f(x))\}$ be a theory of a language $L = \langle R, f, d\rangle$. The systematic tableau for $F\neg R(d)$ from T contains a single branch B, which is noncontradictory.

The canonical model $\mathcal{A} = \langle A, R^A, f^A, d^A, c_i^A\rangle_{i\in\mathbb{N}}$ from B is for language L_C and

$$\begin{aligned} A &= \{d, f(d), f(f(d)), \dots, c_0, f(c_0), f(f(c_0)), \dots, c_1, f(c_1), f(f(c_1)), \dots\}, \\ d^A &= d, \quad c_i^A = c_i \ \text{ for } \ i \in \mathbb{N}, \\ f^A(d) &= \text{“}f(d)\text{”}, \ \ f^A(f(d)) = \text{“}f(f(d))\text{”}, \ \ f^A(f(f(d))) = \text{“}f(f(f(d)))\text{”}, \ \dots \\ R^A &= \{d, f(d), f(f(d)), \dots, f(c_0), f(f(c_0)), \dots, f(c_1), f(f(c_1)), \dots\}. \end{aligned}$$

The reduct of $\mathcal{A}$ to the language L is $\mathcal{A}' = \langle A, R^A, f^A, d^A\rangle$.

The canonical model with equality

If L is with equality, T^* is the extension of T by axioms of equality for L.

If we require that the equality is interpreted as the identity, we have to take the quotient of the canonical model $\mathcal{A}$ by the congruence $=^A$.

By (3), for the relation $=^A$ in $\mathcal{A}$ from B it holds that for every $t_{i_1}, t_{i_2} \in A$,

$$t_{i_1} =^A t_{i_2} \quad \Leftrightarrow \quad T(t_{i_1} = t_{i_2}) \text{ is an entry on } V.$$

Since B is finished and contains the axioms of equality, the relation $=^A$ is a congruence for all functions and relations in $\mathcal{A}$.

The *canonical model with equality* from B is the quotient $\mathcal{A}/{=^A}$.

Observation *For every formula φ,*

$$\mathcal{A} \models \varphi \quad \Leftrightarrow \quad (\mathcal{A}/{=^A}) \models \varphi,$$

where $=$ is interpreted in $\mathcal{A}$ by the relation $=^A$, while in $\mathcal{A}/{=^A}$ by the identity.

Remark *$\mathcal{A}$ is a countably infinite model, but $\mathcal{A}/{=^A}$ can be finite.*

The canonical model with equality - an example

Let $T = \{(\forall x)R(f(x)),\ (\forall x)(x = f(f(x)))\}$ be of $L = \langle R, f, d\rangle$ with equality. The systematic tableau for $F\neg R(d)$ from T^* contains a noncontradictory B.

In the canonical model $\mathcal{A} = \langle A, R^A, =^A, f^A, d^A, c_i^A\rangle_{i\in\mathbb{N}}$ from B we have that

$$s =^A t \quad \Leftrightarrow \quad t = f(\cdots(f(s)\cdots) \ \text{ or } \ s = f(\cdots(f(t)\cdots),$$

where f is applied $2i$-times for some $i \in \mathbb{N}$.

The canonical model with equality from B is

$\mathcal{B} = (\mathcal{A}/{=^A}) = \langle A/{=^A}, R^B, f^B, d^B, c_i^B\rangle_{i\in\mathbb{N}}$ where

$$(A/{=^A}) = \{[d]_{=^A}, [f(d)]_{=^A}, [c_0]_{=^A}, [f(c_0)]_{=^A}, [c_1]_{=^A}, [f(c_1)]_{=^A}, \dots\},$$

$$d^B = [d]_{=^A}, \quad c_i^B = [c_i]_{=^A} \ \text{ for } \ i \in \mathbb{N},$$

$$f^B([d]_{=^A}) = [f(d)]_{=^A}, \ \ f^B([f(d)]_{=^A}) = [f(f(d))]_{=^A} = [d]_{=^A}, \ \ \dots$$

$$R^B = (R^A/{=^A}).$$

The reduct of $\mathcal{B}$ to the language L is $\mathcal{B}' = \langle A/{=^A}, R^B, f^B, d^B\rangle$.

Completeness

Lemma *Canonical model $\mathcal{A}$ from a noncontr. finished B agrees with B.*

Proof By induction on the structure of a sentence in an entry on B.

- For atomic φ, if $T\varphi$ is on B, then $\mathcal{A} \models \varphi$ by (3). If $F\varphi$ is on B, then $T\varphi$ is not on B since B is noncontradictory, so $\mathcal{A} \models \neg\varphi$ by (3).
- If $T(\varphi \wedge \psi)$ is on B, then $T\varphi$ and $T\psi$ are on B since B is finished. By induction, $\mathcal{A} \models \varphi$ and $\mathcal{A} \models \psi$, and thus $\mathcal{A} \models \varphi \wedge \psi$.
- If $F(\varphi \wedge \psi)$ is on B, then $F\varphi$ or $F\psi$ is on B since B is finished. By induction, $\mathcal{A} \models \neg\varphi$ or $\mathcal{A} \models \neg\psi$, and thus $\mathcal{A} \models \neg(\varphi \wedge \psi)$.
- For other connectives similarly as in previous two cases.
- If $T(\forall x)\varphi(x)$ is on B, then $T\varphi(x/t)$ is on B for every $t \in A$ since B is finished. By induction, $\mathcal{A} \models \varphi(x/t)$ for every $t \in A$, and thus $\mathcal{A} \models (\forall x)\varphi(x)$. Similarly for $F(\exists x)\varphi(x)$ on B.
- If $T(\exists x)\varphi(x)$ is on B, then $T\varphi(x/c)$ is on B for some $c \in A$ since B is finished. By induction, $\mathcal{A} \models \varphi(x/c)$, and thus $\mathcal{A} \models (\exists x)\varphi(x)$. Similarly for $F(\forall x)\varphi(x)$ on B. $\square$

Theorem on completeness

We will show that the tableau method in predicate logic is complete.

Theorem *For every theory T and sentence φ, if φ is valid in T, then φ is tableau provable from T, i.e. $T \models \varphi \;\Rightarrow\; T \vdash \varphi$.*

Proof Let φ be valid in T. We will show that an arbitrary finished tableau (e.g. systematic) τ from a theory T with the root entry $F\varphi$ is contradictory.

- If not, then there is some noncontradictory branch B in τ.
- By the previous lemma, there is a structure $\mathcal{A}$ for L_C that agrees with B, in particular with the root entry $F\varphi$, i.e. $\mathcal{A} \models \neg\varphi$.
- Let $\mathcal{A}'$ be the reduct of $\mathcal{A}$ to the language L. Then $\mathcal{A}' \models \neg\varphi$.
- Since B is finished, it contains $T\psi$ for every $\psi \in T$.
- Thus $\mathcal{A}'$ is a model of T (as $\mathcal{A}'$ agrees with $T\psi$ for every $\psi \in T$).
- But this contradicts the assumption that φ is valid in T.

Therefore the tableau τ is a proof of φ from T. $\square$

Properties of theories

We introduce syntactic variants of previous semantical definitions.

Let T be a theory of a language L. If a sentence φ is provable from T, we say that φ is a *theorem* of T. The set of theorems of T is denoted by

$$\mathrm{Thm}^{L}(T) = \{\varphi \in \mathrm{Fm}_{L} \mid T \vdash \varphi\}.$$

We say that a theory T is

- *inconsistent* if $T \vdash \perp$, otherwise T is *consistent*,
- *complete* if it is consistent and every sentence is provable or refutable from T, i.e. $T \vdash \varphi$ or $T \vdash \neg\varphi$.
- an *extension* of a theory T' of L' if $L' \subseteq L$ and $\mathrm{Thm}^{L'}(T') \subseteq \mathrm{Thm}^{L}(T)$, we say that an extension T of a theory T' is *simple* if $L = L'$; and

  conservative if $\mathrm{Thm}^{L'}(T') = \mathrm{Thm}^{L}(T) \cap \mathrm{Fm}_{\mathrm{L}'}$,
- *equivalent* with a theory T' if T is an extension of T' and vice-versa.

Corollaries

From the soundness and completeness of the tableau method it follows that these syntactic definitions agree with their semantic variants.

Corollary *For every theory T and sentences φ, ψ of a language L,*

- $T \vdash \varphi$ *if and only if* $T \models \varphi$,
- $\mathrm{Thm}^L(T) = \theta^L(T)$,
- *T is inconsistent if and only if T is unsatisfiable, i.e. it has no model,*
- *T is complete if and only if T is semantically complete, i.e. it has a single model, up to elementarily equivalence,*
- $T, \varphi \vdash \psi$ *if and only if* $T \vdash \varphi \to \psi$ *(Deduction theorem).*

Remark Deduction theorem can be proved directly by transformations of tableaux.

NAIL062 Propositional & Predicate Logic: Lecture 10

Slides by Petr Gregor with minor modifications by Jakub Bulín

December 7, 2020

Existence of a countable model and compactness

Theorem *Every consistent theory T of a countable language L without equality has a countably infinite model.*

Proof Let τ be the systematic tableau from T with $F\bot$ in the root. Since τ is finished and contains a noncontradictory branch B as $\bot$ is not provable from T, there exists a canonical model $\mathcal{A}$ from B. Since $\mathcal{A}$ agrees with B, its reduct to the language L is a desired countably infinite model of T. $\square$

Remark *This is a weak version of so called Löwenheim-Skolem theorem.* *In a countable language with equality the canonical model with equality is countable (i.e. finite or countably infinite).*

Theorem *A theory T has a model iff every finite subset of T has a model.*

Proof The implication from left to right is obvious. If T has no model, then it is inconsistent, i.e. $\bot$ is provable by a systematic tableau τ from T. Since τ is finite, $\bot$ is provable from some finite $T' \subseteq T$, i.e. T' has no model. $\square$

Non-standard model of natural numbers

Let $\underline{\mathbb{N}} = \langle \mathbb{N}, S, +, \cdot, 0, \leq \rangle$ be the standard model of natural numbers.

Let $\mathrm{Th}(\underline{\mathbb{N}})$ denote the set of all sentences that are valid in $\underline{\mathbb{N}}$. For $n \in \mathbb{N}$ let $\underline{n}$ denote the term $S(S(\cdots(S(0))\cdots))$, so called the *n-th numeral*, where S is applied n-times.

Consider the following theory T where c is a new constant symbol.

$$T = \mathrm{Th}(\underline{\mathbb{N}}) \cup \{\underline{n} < c \mid n \in \mathbb{N}\}$$

Observation *Every finite subset of T has a model.*

Thus by the compactness theorem, T has a model $\mathcal{A}$. It is a non-standard model of natural numbers. Every sentence from $\mathrm{Th}(\underline{\mathbb{N}})$ is valid in $\mathcal{A}$ but it contains an element c^A that is greater then every $n \in \mathbb{N}$ (i.e. the value of the term $\underline{n}$ in $\mathcal{A}$).

Table of Contents

Extensions of theories

We show that introducing new definitions has only an "auxiliary character".

Proposition *Let T be an L-theory and T' an L'-theory, where $L \subseteq L'$.*

1. *T' is an extension of T if and only if the reduct $\mathcal{A}$ of every model $\mathcal{A}'$ of T' to the language L is a model of T,*
2. *T' is a conservative extension of T if T' is an extension of T and every model $\mathcal{A}$ of T can be expanded to the language L' on a model $\mathcal{A}'$ of T'.*

Proof

$(i)a)$ If T' is an extension of T and φ is any axiom of T, then $T' \models \varphi$. Thus $\mathcal{A}' \models \varphi$ and also $\mathcal{A} \models \varphi$, which implies that $\mathcal{A}$ is a model of T.

$(i)b)$ If $\mathcal{A}$ is a model of T and $T \models \varphi$ where φ is of L, then $\mathcal{A} \models \varphi$ and also $\mathcal{A}' \models \varphi$. This implies that $T' \models \varphi$ and thus T' is an extension of T.

(ii) If $T' \models \varphi$ where φ is of L and $\mathcal{A}$ is a model of T, then in its expansion $\mathcal{A}'$ that models T' we have $\mathcal{A}' \models \varphi$. Thus also $\mathcal{A} \models \varphi$, and hence $T \models \varphi$. Therefore T' is conservative. $\square$

Extensions by definition of a relation symbol

Let T be a theory of L, $\psi(x_1, \ldots, x_n)$ be a formula of L in free variables $x_1, \ldots, x_n$ and L' denote the language L with a new n-ary relation symbol R.

The *extension* of T *by definition of R* with the formula ψ is the theory T' of L' obtained from T by adding the axiom

$$R(x_1, \ldots, x_n) \;\leftrightarrow\; \psi(x_1, \ldots, x_n)$$

Observation *Every model of T can be uniquely expanded to a model of T'.*

Corollary *T' is a conservative extension of T.*

Proposition *For every formula φ' of L' there is φ of L s.t. $T' \models \varphi' \leftrightarrow \varphi$.*

Proof Replace each subformula $R(t_1, \ldots, t_n)$ in φ with $\psi'(x_1/t_1, \ldots, x_n/t_n)$, where ψ' is a suitable variant of ψ allowing all substitutions. $\square$

For example, the symbol $\leq$ can be defined in arithmetics by the axiom

$$x \leq y \quad\leftrightarrow\quad (\exists z)(x + z = y)$$

Extensions by definition of a function symbol

Let T be a theory of a language L and $\psi(x_1, \ldots, x_n, y)$ be a formula of L in free variables $x_1, \ldots, x_n, y$ such that

$$T \models (\exists y)\psi(x_1, \ldots, x_n, y) \qquad \text{(existence)}$$

$$T \models \psi(x_1, \ldots, x_n, y) \ \wedge \ \psi(x_1, \ldots, x_n, z) \ \rightarrow \ y = z \qquad \text{(uniqueness)}$$

Let L' denote the language L with a new n-ary function symbol f.

The *extension* of T *by definition of f* with the formula ψ is the theory T' of L' obtained from T by adding the axiom

$$f(x_1, \ldots, x_n) = y \ \leftrightarrow \ \psi(x_1, \ldots, x_n, y)$$

Remark *In particular, if ψ is $t(x_1, \ldots, x_n) = y$ where t is a term and $x_1, \ldots, x_n$ are the variables in t, both the conditions of existence and uniqueness hold.*

For example binary $-$ can be defined using $+$ and unary $-$ by the axiom

$$x - y = z \quad \leftrightarrow \quad x + (-y) = z$$

Extensions by definition of a function symbol (cont.)

Observation *Every model of T can be uniquely expanded to a model of T'.*

Corollary *T' is a conservative extension of T.*

Proposition *For every formula φ' of L' there is φ of L s.t. $T' \models \varphi' \leftrightarrow \varphi$.*

Proof It suffices to consider φ' with a single occurrence of f. If φ' has more, we may proceed inductively. Let φ^* denote the formula obtained from φ' by replacing the term $f(t_1, \ldots, t_n)$ with a new variable z. Let φ be the formula

$$(\exists z)(\varphi^* \ \wedge \ \psi'(x_1/t_1, \ldots, x_n/t_n, y/z)),$$

where ψ' is a suitable variant of ψ allowing all substitutions.

Let $\mathcal{A}$ be a model of T', e be an assignment, and $a = f^A(t_1, \ldots, t_n)[e]$. By the two conditions, $\mathcal{A} \models \psi'(x_1/t_1, \ldots, x_n/t_n, y/z)[e]$ if and only if $e(z) = a$. Thus

$$\mathcal{A} \models \varphi[e] \quad \Leftrightarrow \quad \mathcal{A} \models \varphi^*[e(z/a)] \quad \Leftrightarrow \quad \mathcal{A} \models \varphi'[e]$$

for every assignment e, i.e. $\mathcal{A} \models \varphi' \leftrightarrow \varphi$ and so $T' \models \varphi' \leftrightarrow \varphi$. $\square$

Extensions by definitions

A theory T' of L' is called an *extension* of a theory T of L *by definitions* if it is obtained from T by successive definitions of relation and function symbols.

Corollary *Let T' be an extension of a theory T by definitions. Then*

- *every model of T can be uniquely expanded to a model of T',*
- *T' is a conservative extension of T,*
- *for every formula φ' of L' there is a formula φ of L such that $T' \models \varphi' \leftrightarrow \varphi$.*

For example, in $T = \{(\exists y)(x + y = 0), (x + y = 0) \wedge (x + z = 0) \rightarrow y = z\}$ of $L = \langle +, 0, \leq \rangle$ with equality we can define $<$ and unary $-$ by the axioms

$$\begin{aligned} -x = y \quad &\leftrightarrow \quad x + y = 0 \\ x < y \quad &\leftrightarrow \quad x \leq y \ \wedge \ \neg(x = y) \end{aligned}$$

Then the formula $-x < y$ is equivalent in this extension to a formula

$$(\exists z)((z \leq y \ \wedge \ \neg(z = y)) \ \wedge \ x + z = 0).$$

Table of Contents

Equisatisfiability

We will see that the problem of satisfiability can be reduced to open theories.

- Theories T, T' are *equisatisfiable* if T has a model $\Leftrightarrow$ T' has a model.
- A formula φ is in the *prenex (normal) form (PNF)* if it is written as
$$(Q_1 x_1) \dots (Q_n x_n) \varphi',$$
where Q_i denotes $\forall$ or $\exists$, variables $x_1, \dots, x_n$ are all distinct and φ' is an open formula, called the *matrix*. $(Q_1 x_1) \dots (Q_n x_n)$ is called the *prefix*.
- In particular, if all quantifiers are $\forall$, then φ is a *universal* formula.

To find an open theory equisatisfiable with T we proceed as follows.

1. We replace axioms of T by equivalent formulas in the prenex form.
2. We transform them, using new function symbols, to equisatisfiable universal formulas, so called Skolem variants.
3. We take their matrices as axioms of a new theory.

Conversion rules for quantifiers

Let Q denote $\forall$ or $\exists$ and let $\overline{Q}$ denote the complementary quantifier.

For every formulas φ, ψ such that x is not free in the formula ψ,

$$\begin{aligned}
&\models \quad \neg(Qx)\varphi \leftrightarrow (\overline{Q}x)\neg\varphi \\
&\models \quad ((Qx)\varphi \wedge \psi) \leftrightarrow (Qx)(\varphi \wedge \psi) \\
&\models \quad ((Qx)\varphi \vee \psi) \leftrightarrow (Qx)(\varphi \vee \psi) \\
&\models \quad ((Qx)\varphi \rightarrow \psi) \leftrightarrow (\overline{Q}x)(\varphi \rightarrow \psi) \\
&\models \quad (\psi \rightarrow (Qx)\varphi) \leftrightarrow (Qx)(\psi \rightarrow \varphi)
\end{aligned}$$

The above equivalences can be verified semantically or proved by the tableau method (*by taking the universal closure if it is not a sentence*).

Remark *The assumption that x is not free in ψ is necessary in each rule above (except the first one) for some quantifier Q. For example,*

$$\not\models \; ((\exists x)P(x) \wedge P(x)) \leftrightarrow (\exists x)(P(x) \wedge P(x))$$

Conversion to the prenex normal form

Proposition *Let φ' be the formula obtained from φ by replacing some occurrences of a subformula ψ with ψ'. If $T \models \psi \leftrightarrow \psi'$, then $T \models \varphi \leftrightarrow \varphi'$.*

Proof Easily by induction on the structure of the formula φ. □

Proposition *For every formula φ there is an equivalent formula φ' in the prenex normal form, i.e. $\models \varphi \leftrightarrow \varphi'$.*

Proof By induction on the structure of φ applying the conversion rules for quantifiers, replacing subformulas with their variants if needed, and applying the above proposition on equivalent transformations. □

For example,

$$\begin{aligned}
((\forall z)P(x,z) \wedge P(y,z)) &\rightarrow \neg(\exists x)P(x,y) \\
((\forall u)P(x,u) \wedge P(y,z)) &\rightarrow (\forall x)\neg P(x,y) \\
(\forall u)(P(x,u) \wedge P(y,z)) &\rightarrow (\forall v)\neg P(v,y) \\
(\exists u)((P(x,u) \wedge P(y,z)) &\rightarrow (\forall v)\neg P(v,y)) \\
(\exists u)(\forall v)((P(x,u) \wedge P(y,z)) &\rightarrow \neg P(v,y))
\end{aligned}$$

Skolem variants

Let φ be a sentence of a language L in the prenex normal form, let $y_1, \dots, y_n$ be the existentially quantified variables in φ (in this order), and for every $i \le n$ let $x_1, \dots, x_{n_i}$ be the variables that are universally quantified in φ before y_i. Let L' be an extension of L with new n_i-ary function symbols f_i for all $i \le n$.

Let φ_S denote the formula of L' obtained from φ by removing all $(\exists y_i)$'s from the prefix and by replacing each occurrence of y_i with the term $f_i(x_1, \dots, x_{n_i})$. Then φ_S is called a *Skolem variant* of φ.

For example, for the sentence φ

$$(\exists y_1)(\forall x_1)(\forall x_2)(\exists y_2)(\forall x_3)R(y_1, x_1, x_2, y_2, x_3)$$

the following formula φ_S is a Skolem variant of φ

$$(\forall x_1)(\forall x_2)(\forall x_3)R(f_1, x_1, x_2, f_2(x_1, x_2), x_3),$$

where f_1 is a new constant symbol and f_2 is a new binary function symbol.

Properties of Skolem variants

Lemma *Let φ be a sentence $(\forall x_1)\dots(\forall x_n)(\exists y)\psi$ of L and φ' be a sentence $(\forall x_1)\dots(\forall x_n)\psi(y/f(x_1,\dots,x_n))$ where f is a new function symbol. Then*

1. *the reduct $\mathcal{A}$ of every model $\mathcal{A}'$ of φ' to L is a model of φ,*
2. *every model $\mathcal{A}$ of φ can be expanded into a model $\mathcal{A}'$ of φ'.*

Remark *Compared to extensions by definition of a function symbol, the expansion in* (2) *does not need to be unique now.*

Proof (1) Let $\mathcal{A}' \models \varphi'$ and $\mathcal{A}$ be the reduct of $\mathcal{A}'$ to L. Since $\mathcal{A} \models \psi[e(y/a)]$ for every assignment e where $a = (f(x_1,\dots,x_n))^{A'}[e]$, we have also $\mathcal{A} \models \varphi$.

(2) Let $\mathcal{A} \models \varphi$. There exists a function $f^A\colon A^n \to A$ such that for every assignment e it holds $\mathcal{A} \models \psi[e(y/a)]$ where $a = f^A(e(x_1),\dots,e(x_n))$, and thus the expansion $\mathcal{A}'$ of $\mathcal{A}$ by the function f^A is a model of φ'. $\square$

Corollary *If φ' is a Skolem variant of φ, then both statements* (1) *and* (2) *hold for φ, φ' as well. Hence φ, φ' are equisatisfiable.*

Skolem's theorem

Theorem *Every theory T has an open conservative extension T^*.*

Proof We may assume that T is in a closed form. Let L be its language.

- By replacing each axiom of T with an equivalent formula in the prenex normal form we obtain an equivalent theory T°.
- By replacing each axiom of T° with its Skolem variant we obtain a theory T' in an extended language $L' \supseteq L$.
- Since the reduct of every model of T' to the language L is a model of T, the theory T' is an extension of T.
- Furthermore, since every model of T can be expanded to a model of T', it is a conservative extension.
- Since every axiom of T' is a universal sentence, by replacing them with their matrices we obtain an open theory T^* equivalent to T'. □

Corollary *For every theory there is an equisatisfiable open theory.*

Table of Contents

Reduction of unsatisfiability to propositional logic

If an open theory is unsatisfiable, we can demonstrate it "via ground terms".

For example, in the language $L = \langle P, R, f, c \rangle$ the theory

$$T = \{P(x, y) \vee R(x, y),\ \neg P(c, y),\ \neg R(x, f(x))\}$$

is unsatisfiable, and this can be demonstrated by an unsatisfiable conjunction of finitely many instances of (some) axioms of T in ground terms

$$(P(c, f(c)) \vee R(c, f(c)))\ \wedge\ \neg P(c, f(c))\ \wedge\ \neg R(c, f(c)),$$

which may be seen as an unsatisfiable propositional formula

$$(p \vee r)\ \wedge\ \neg p\ \wedge\ \neg r.$$

An instance $\varphi(x_1/t_1, \ldots, x_n/t_n)$ of an open formula φ in free variables $x_1, \ldots, x_n$ is a *ground instance* if all terms $t_1, \ldots, t_n$ are ground terms (i.e. terms without variables).

Herbrand model

Let $L = \langle \mathcal{R}, \mathcal{F} \rangle$ be a language with at least one constant symbol. *(If needed, we add a new constant symbol to L.)*

- The *Herbrand universe* for L is the set of all ground terms of L.

  For example, for $L = \langle P, f, c \rangle$ with f a binary function symbol, c a constant symbol:

  $$A = \{c, f(c,c), f(f(c,c),c), f(c, f(c,c)), f(f(c,c), f(c,c)), \dots\}$$

- An L-structure $\mathcal{A}$ is a *Herbrand structure* if its domain A is the Herbrand universe for L and for each n-ary function symbol $f \in \mathcal{F}$, $t_1, \dots, t_n \in A$,

  $$f^A(t_1, \dots, t_n) = f(t_1, \dots, t_n)$$

  (including $n = 0$, i.e. $c^A = c$ for every constant symbol c).

  Remark *Compared to a canonical model, the relations are not specified.*

  E.g. $\mathcal{A} = \langle A, P^A, f^A, c^A \rangle$ *with* $P^A = \emptyset$, $c^A = c$, $f^A(c,c) = f(c,c)$, $\dots$

- A *Herbrand model* of a theory T is a Herbrand structure that models T.

Herbrand's theorem

Theorem *Let T be an open theory of a language L without equality and with at least one constant symbol. Then*

- (a) *either T has a Herbrand model, or*
- (b) *there are finitely many ground instances of axioms of T whose conjunction is unsatisfiable, and thus T has no model.*

Proof Let T' be the set of all ground instances of axioms of T. Consider a finished (e.g. systematic) tableau τ from T' in the language L (without adding new constant symbols) with the root entry $F\bot$.

- If the tableau τ contains a noncontradictory branch B, the canonical model from B is a Herbrand model of T.
- Else, τ is contradictory, i.e. $T' \vdash \bot$. Moreover, τ is finite, so $\bot$ is provable from finitely many formulas of T', i.e. their conjunction is unsatisfiable. $\square$

Remark *If the language L is with equality, we extend T to T^* by axioms of equality for L and if T^* has a Herbrand model $\mathcal{A}$, we take its quotient by $=^A$.*

Corollaries of Herbrand's theorem

Let L be a language containing at least one constant symbol.

Corollary *For every open $\varphi(x_1, \ldots, x_n)$ of L, the formula $(\exists x_1) \ldots (\exists x_n)\varphi$ is valid if and only if there exist mn ground terms t_{ij} of L for some m such that*

$$\varphi(x_1/t_{11}, \ldots, x_n/t_{1n}) \vee \cdots \vee \varphi(x_1/t_{m1}, \ldots, x_n/t_{mn})$$

is a (propositional) tautology.

Proof $(\exists x_1) \ldots (\exists x_n)\varphi$ is valid $\Leftrightarrow$ $(\forall x_1) \ldots (\forall x_n)\neg\varphi$ is unsatisfiable $\Leftrightarrow$ $\neg\varphi$ is unsatisfiable. The rest follows from Herbrand's theorem for $\{\neg\varphi\}$. □

Corollary *An open theory T of L is satisfiable if and only if the theory T' of all ground instances of axioms of T is satisfiable.*

Proof If T has a model $\mathcal{A}$, every instance of each axiom of T is valid in $\mathcal{A}$, thus $\mathcal{A}$ is a model of T'. If T is unsatisfiable, by H. theorem there are (finitely many) formulas of T' whose conjunction is unsatisfiable, thus T' is unsatisfiable. □

NAIL062 Propositional & Predicate Logic: Lecture 11

Slides by Petr Gregor with minor modifications by Jakub Bulín

December 14, 2020

Table of Contents

Resolution method in predicate logic - introduction

- A refutation procedure - its aim is to show that a given formula (or theory) is unsatisfiable.
- It assumes open formulas in CNF (and in clausal form).

  A *literal* is *(now)* an atomic formula or its negation.

  A *clause* is a finite set of literals, $\square$ denotes the empty clause.

  A *formula (in clausal form)* is a (possibly infinite) set of clauses.

  Remark *Every formula (theory) can be converted to an equisatisfiable open formula (theory) in CNF, and then to a formula in clausal form.*
- The resolution rule is more general - it allows to resolve through literals that are unifiable.
- Resolution in predicate logic is based on resolution in propositional logic and unification.

Local scope of variables

Variables can be renamed locally within clauses.

Let φ be an *(input)* open formula in CNF.

- φ is satisfiable if and only if its universal closure φ' is satisfiable.
- For every two formulas ψ, χ and a variable x

$$\models \quad (\forall x)(\psi \wedge \chi) \ \leftrightarrow\ (\forall x)\psi \wedge (\forall x)\chi$$

  (also in the case that x is free both in ψ and χ).
- Every clause in φ can thus be replaced by its universal closure.
- We can then take any variants of clauses (to rename variables apart).

For example, by renaming variables in the second clause of (1) *we obtain an equisatisfiable formula* (2).

1. $\{\{P(x), Q(x, y)\}, \{\neg P(x), \neg Q(y, x)\}\}$
2. $\{\{P(x), Q(x, y)\}, \{\neg P(v), \neg Q(u, v)\}\}$

Reduction to propositional level (grounding)

Herbrand's theorem gives us the following (inefficient) method.

- Let S be the *(input)* formula in clausal form.
- We can assume that the language contains at least one constant symbol.
- Let S' be the set of all ground instances of all clauses from S.
- By introducing propositional letters representing atomic sentences we may view S' as a (possibly infinite) propositional formula in clausal form.
- We may verify that it is unsatisfiable by resolution on propositional level.

E.g. for $S = \{\{P(x,y), R(x,y)\}, \{\neg P(c,y)\}, \{\neg R(x, f(x))\}\}$ *the set*

$$S' = \{\{P(c,c), R(c,c)\}, \{P(c,f(c)), R(c,f(c))\}, \{P(f(c),f(c)), R(f(c),f(c))\} \ldots$$
$$\{\neg P(c,c)\}, \{\neg P(c,f(c))\}, \ldots, \{\neg R(c,f(c))\}, \{\neg R(f(c),f(f(c)))\}, \ldots\}$$

is unsatisfiable since on propositional level

$$S' \supseteq \{\{P(c,f(c)), R(c,f(c))\}, \{\neg P(c,f(c))\}, \{\neg R(c,f(c))\}\} \vdash_R \square.$$

Substitutions - examples

It is more efficient to use suitable substitutions. For example, in

a) $\{P(x), Q(x,a)\}$, $\{\neg P(y), \neg Q(b,y)\}$ substituting x/b, y/a gives $\{P(b), Q(b,a)\}$, $\{\neg P(a), \neg Q(b,a)\}$, which resolves to $\{P(b), \neg P(a)\}$.

   Or, substituting x/y and resolving through $P(y)$ gives $\{Q(y,a), \neg Q(b,y)\}$.

b) $\{P(x), Q(x,a), Q(b,y)\}$, $\{\neg P(v), \neg Q(u,v)\}$ substituting x/b, y/a, u/b, v/a gives $\{P(b), Q(b,a)\}$, $\{\neg P(a), \neg Q(b,a)\}$, resolving to $\{P(b), \neg P(a)\}$.

c) $\{P(x), Q(x,z)\}$, $\{\neg P(y), \neg Q(f(y),y)\}$ substituting $x/f(z)$, y/z gives $\{P(f(z)), Q(f(z),z)\}$, $\{\neg P(z), \neg Q(f(z),z)\}$, resolving to $\{P(f(z)), \neg P(z)\}$.

   Alternatively, substituting $x/f(a)$, y/a, z/a gives $\{P(f(a)), Q(f(a),a)\}$, $\{\neg P(a), \neg Q(f(a),a)\}$, which resolves to $\{P(f(a)), \neg P(a)\}$. But the previous substitution is more general.

Substitutions

- A *substitution* is a (finite) set $\sigma = \{x_1/t_1, \ldots, x_n/t_n\}$, where x_i's are distinct variables, t_i's are terms, and the term t_i is not x_i.
- If all t_i's are ground terms, then σ is a *ground substitution*.
- If all t_i's are distinct variables, then σ is a *renaming of variables*.
- An *expression* is a literal or a term.
- An *instance* of an expression E *by substitution* $\sigma = \{x_1/t_1, \ldots, x_n/t_n\}$ is the expression $E\sigma$ obtained from E by simultaneous replacing all occurrences of all x_i's for t_i's, respectively.
- For a set S of expressions, let $S\sigma = \{E\sigma \mid E \in S\}$.

Remark *Since we substitute for all variables simultaneously, a possible occurrence of x_i in t_j does not lead to a chain of substitutions.*

For example, for $S = \{P(x), R(y, z)\}$ and $\sigma = \{x/f(y, z), y/x, z/c\}$ we have

$$S\sigma = \{P(f(y, z)), R(x, c)\}.$$

Composing substitutions

For substitutions $\sigma = \{x_1/t_1, \ldots, x_n/t_n\}$ and $\tau = \{y_1/s_1, \ldots, y_n/s_n\}$ we define the *composition* of σ and τ to be

$$\sigma\tau = \{x_i/t_i\tau \mid x_i \in X,\ \ t_i\tau \text{ is not } x_i\} \cup \{y_j/s_j \mid y_j \in Y \setminus X\}$$

where $X = \{x_1, \ldots, x_n\},\ Y = \{y_1, \ldots, y_m\}$.

For example, for $\sigma = \{x/f(y), w/v\}$, $\tau = \{x/a, y/g(x), v/w, u/c\}$ we have $\sigma\tau = \{x/f(g(x)), y/g(x), v/w, u/c\}$.

Proposition (without proof) *For every expression E and subst. σ, τ, ϱ,*

1. $(E\sigma)\tau = E(\sigma\tau)$,
2. $(\sigma\tau)\varrho = \sigma(\tau\varrho)$.

Remark *Composition of substitutions is not commutative, for the above:*

$$\tau\sigma = \{x/a, y/g(f(y)), u/c, w/v\} \neq \sigma\tau.$$

Unification

Let $S = \{E_1, \ldots, E_n\}$ be a (finite) set of expressions.

- A *unification* of S is a substitution σ such that $E_1\sigma = E_2\sigma = \cdots = E_n\sigma$, i.e. $S\sigma$ is a singleton.
- S is *unifiable* if it has a unification.
- A unification σ of S is a *most general unification (mgu)* if for every unification τ of S there is a substitution λ such that $\tau = \sigma\lambda$.

For example, $S = \{P(f(x), y), P(f(a), w)\}$ is unifiable by a most general unification $\sigma = \{x/a, y/w\}$. A unification $\tau = \{x/a, y/b, w/b\}$ is obtained as $\sigma\lambda$ for $\lambda = \{w/b\}$. τ is not mgu, it cannot give us $\varrho = \{x/a, y/c, w/c\}$.

Observation *If σ, τ are two most general unifications of S, they differ only in renaming of variables.*

Unification algorithm

Let S be a (finite) nonempty set of expressions and p be the leftmost position in which some expressions of S differ. Then *the difference* in S is the set $D(S)$ of subexpressions of all expressions from S starting at the position p.

For example, $S = \{P(x, y), P(f(x), z), P(z, f(x))\}$ *has* $D(S) = \{x, f(x), z\}$.

Input Nonempty (finite) set of expressions S.

Output A most general unification σ of S or *"S is not unifiable"*.

(0) Let $S_0 := S$, $\sigma_0 := \emptyset$, $k := 0$. *(initialization)*

1. If S_k is a singleton, output $\sigma = \sigma_0\sigma_1 \cdots \sigma_k$. *(mgu of S)*
2. Check if $D(S_k)$ contains a variable x and a term t with no occurrence of x.
3. If not, output *"S is not unifiable"*.
4. Otherwise, $\sigma_{k+1} := \{x/t\}$, $S_{k+1} := S_k\sigma_{k+1}$, $k := k + 1$, GOTO(1).

Remark *The occurrence check of x in t in step* (2) *can be "expensive".*

Unification algorithm - an example

$S = \{P(f(y, g(z)), h(b)),\ P(f(h(w), g(a)), t),\ P(f(h(b), g(z)), y)\}$

1. $S_0 = S$ is not singleton, $D(S_0) = \{y, h(w), h(b)\}$ has a term $h(w)$ and a var. y not occurring in $h(w)$. Then $\sigma_1 = \{y/h(w)\}$, $S_1 = S_0\sigma_1$:
   $S_1 = \{P(f(h(w), g(z)), h(b)),\ P(f(h(w), g(a)), t),\ P(f(h(b), g(z)), h(w))\}$
2. $D(S_1) = \{w, b\}$, $\sigma_2 = \{w/b\}$, $S_2 = S_1\sigma_2$, i.e.
   $S_2 = \{P(f(h(b), g(z)), h(b)),\ P(f(h(b), g(a)), t)\}$
3. $D(S_2) = \{z, a\}$, $\sigma_3 = \{z/a\}$, $S_3 = S_2\sigma_3$, i.e.
   $S_3 = \{P(f(h(b), g(a)), h(b)),\ P(f(h(b), g(a)), t)\}$
4. $D(S_3) = \{h(b), t\}$, $\sigma_4 = \{t/h(b)\}$, $S_4 = S_3\sigma_4$, i.e.
   $S_4 = \{P(f(h(b), g(a)), h(b))\}$
5. S_4 is a singleton and a most general unification of S is
   $\sigma = \{y/h(w)\}\{w/b\}\{z/a\}\{t/h(b)\} = \{y/h(b), w/b, z/a, t/h(b)\}$

Unification algorithm - correctness

Proposition *The unification algorithm outputs a correct answer in finite time for any input S, i.e. a most general unification σ of S or it detects that S is not unifiable. $(*)$ Moreover, for every unification τ of S it holds that $\tau = \sigma\tau$.*

Proof It eliminates one variable in each round, so it ends in finite time.

- If it ends negatively, $D(S_k)$ is not unifiable, and neither is S.
- If it outputs $\sigma = \sigma_0\sigma_1 \cdots \sigma_k$, clearly σ is a unification of S.
- If we show the property $(*)$ for σ, then σ is a most general unification of S.

(1) Let τ be a unification of S. We show $\tau = \sigma_0\sigma_1 \cdots \sigma_i\tau$ for all $i \leq k$.
(2) For $i = 0$ it holds. Let $\sigma_{i+1} = \{x/t\}$ and assume $\tau = \sigma_0\sigma_1 \cdots \sigma_i\tau$.
(3) It suffices to show that $v\sigma_{i+1}\tau = v\tau$ for every variable v.
(4) If $v \neq x$, $v\sigma_{i+1} = v$, so (3) holds. Otherwise $v = x$ and $v\sigma_{i+1} = x\sigma_{i+1} = t$.
(5) Since τ unifies $S_i = S\sigma_0\sigma_1 \cdots \sigma_i$ and both the variable x and the term t are in $D(S_i)$, τ has to unify x and t, i.e. $t\tau = x\tau$, as required for (3). □

The general resolution rule

Let C_1, C_2 be clauses with distinct variables such that

$$C_1 = C_1' \sqcup \{A_1, \dots, A_n\}, \quad C_2 = C_2' \sqcup \{\neg B_1, \dots, \neg B_m\},$$

where $S = \{A_1, \dots, A_n, B_1, \dots, B_m\}$ is unifiable and $n, m \geq 1$. Then the clause

$$C = C_1'\sigma \cup C_2'\sigma,$$

where σ is a most general unification of S, is the *resolvent* of C_1 and C_2.

For example, in clauses $\{P(x), Q(x,z)\}$ and $\{\neg P(y), \neg Q(f(y), y)\}$ we can unify $S = \{Q(x,z), Q(f(y), y)\}$ applying a most general unification $\sigma = \{x/f(y), z/y\}$, and then resolve to a clause $\{P(f(y)), \neg P(y)\}$.

Remark *The condition on distinct variables can be satisfied by renaming variables apart. This is sometimes necessary, e.g. from*
$\{\{P(x)\}, \{\neg P(f(x))\}\}$
after renaming we can get $\square$*, but $\{P(x), P(f(x))\}$ is not unifiable.*

Resolution proof

We have the same notions as in propositional logic, up to renaming variables.

- *Resolution proof (deduction)* of a clause C from a formula S is a finite sequence $C_0, \ldots, C_n = C$ such that for every $i \leq n$, we have $C_i = C_i'\sigma$ for some $C_i' \in S$ and a renaming of variables σ, or C_i is a resolvent of some previous clauses.
- A clause C is (resolution) *provable* from S, denoted by $S \vdash_R C$, if it has a resolution proof from S.
- A (resolution) *refutation* of a formula S is a resolution proof of $\square$ from S.
- S is (resolution) *refutable* if $S \vdash_R \square$.

Remark *Elimination of several literals at once is sometimes necessary, e.g.* $S = \{\{P(x), P(y)\}, \{\neg P(x), \neg P(y)\}\}$ *is resolution refutable, but it has no refutation that eliminates only a single literal in each resolution step.*

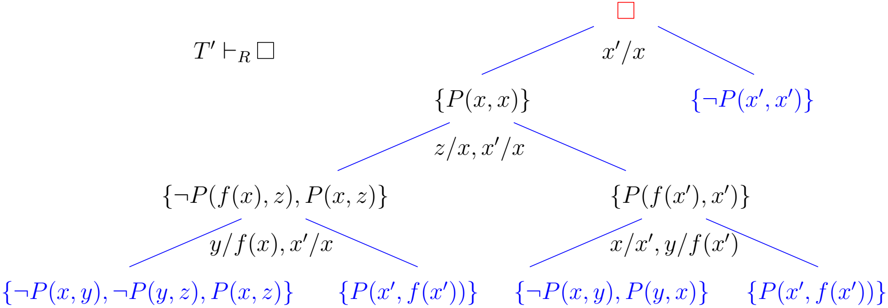
Resolution in predicate logic - an example

Consider $T = \{\neg P(x,x),\ P(x,y) \rightarrow P(y,x),\ P(x,y) \wedge P(y,z) \rightarrow P(x,z)\}$.

Is $T \models (\exists x)\neg P(x, f(x))$? Equivalently, is the following T' unsatisfiable?

$T' = \{\{\neg P(x,x)\}, \{\neg P(x,y), P(y,x)\}, \{\neg P(x,y), \neg P(y,z), P(x,z)\}, \{P(x, f(x))\}\}$

$T' \vdash_R \square$



Soundness of resolution

First we show soundness of the general resolution rule.

Proposition Let C be a resolvent of clauses C_1, C_2. Then for every L-structure $\mathcal{A}$:

$$\mathcal{A} \models C_1 \text{ and } \mathcal{A} \models C_2 \quad \Rightarrow \quad \mathcal{A} \models C.$$

Proof Let $C_1 = C_1' \sqcup \{A_1, \dots, A_n\}$, $C_2 = C_2' \sqcup \{\neg B_1, \dots, \neg B_m\}$, σ be a most general unification for $S = \{A_1, \dots, A_n, B_1, \dots, B_m\}$, and $C = C_1'\sigma \cup C_2'\sigma$.

- Since C_1, C_2 are open, it holds also $\mathcal{A} \models C_1\sigma$ and $\mathcal{A} \models C_2\sigma$.
- We have $C_1\sigma = C_1'\sigma \cup \{S\sigma\}$ and $C_2\sigma = C_2'\sigma \cup \{\neg(S\sigma)\}$.
- We show $\mathcal{A} \models C[e]$ for every e. If $\mathcal{A} \models S\sigma[e]$, then $\mathcal{A} \models C_2'\sigma[e]$, and thus $\mathcal{A} \models C[e]$. Otherwise $\mathcal{A} \not\models S\sigma[e]$, so $\mathcal{A} \models C_1'\sigma[e]$, and thus $\mathcal{A} \models C[e]$. $\square$

Theorem (soundness) *If S is resolution refutable, then S is unsatisfiable.*

Proof Let $S \vdash_R \square$. Suppose $\mathcal{A} \models S$ for some structure $\mathcal{A}$. By soundness of the general resolution rule we have $\mathcal{A} \models \square$, which is impossible. $\square$

NAIL062 Propositional & Predicate Logic: Lecture 12

Slides by Petr Gregor with minor modifications by Jakub Bulín

December 21, 2020

Lifting lemma

A resolution proof on propositional level can be "lifted" to predicate level.

Lemma *Let $C_1^* = C_1\tau_1$, $C_2^* = C_2\tau_2$ be ground instances of clauses C_1, C_2 with distinct variables and C^* be a resolvent of C_1^* a C_2^*. Then there exists a resolvent C of C_1 and C_2 such that $C^* = C\tau_1\tau_2$ is a ground instance of C.*

Proof Let C^* be a resolvent of C_1^*, C_2^* through a literal $P(t_1, \ldots, t_k)$.

- We have $C_1 = C_1' \sqcup \{A_1, \ldots, A_n\}$ and $C_2 = C_2' \sqcup \{\neg B_1, \ldots, \neg B_m\}$, where $\{A_1, \ldots, A_n\}\tau_1 = \{P(t_1, \ldots, t_k)\}$ & $\{\neg B_1, \ldots, \neg B_m\}\tau_2 = \{\neg P(t_1, \ldots, t_k)\}$
- Thus $(\tau_1\tau_2)$ unifies $S = \{A_1, \ldots, A_n, B_1, \ldots, B_m\}$ and if σ is mgu of S from the unif. algorithm, then $C = C_1'\sigma \cup C_2'\sigma$ is a resolvent of C_1, C_2.
- Moreover, $(\tau_1\tau_2) = \sigma(\tau_1\tau_2)$ by the property $(*)$ for σ, and hence

$$\begin{aligned} C\tau_1\tau_2 &= (C_1'\sigma \cup C_2'\sigma)\tau_1\tau_2 = C_1'\sigma\tau_1\tau_2 \cup C_2'\sigma\tau_1\tau_2 = C_1'\tau_1 \cup C_2'\tau_2 \\ &= (C_1 \setminus \{A_1, \ldots, A_n\})\tau_1 \cup (C_2 \setminus \{\neg B_1, \ldots, \neg B_m\})\tau_2 \\ &= (C_1^* \setminus \{P(t_1, \ldots, t_k)\}) \cup (C_2^* \setminus \{\neg P(t_1, \ldots, t_k)\}) = C^*. \quad \square \end{aligned}$$

Completeness

Corollary *Let S' be the set of all ground instances of clauses of a formula S. If $S' \vdash_R C'$ (on propositional level) where C' is a ground clause, then $C' = C\sigma$ for some clause C and a ground substitution σ such that $S \vdash_R C$ (on pred. level).*

Proof By induction on the length of resolution proof using lifting lemma. $\square$

Theorem (completeness) *If S is unsatisfiable, then $S \vdash_R \square$.*

Proof If S is unsatisfiable, then by the (corollary of) Herbrand's theorem, also the set S' of all ground instances of clauses of S is unsatisfiable.

- By completeness of resolution in prop. logic, $S' \vdash_R \square$ (on prop. level).
- By the above corollary, there is a clause C and a ground substitution σ such that $\square = C\sigma$ and $S \vdash_R C$ (on pred. level).
- The only clause that has $\square$ as a ground instance is the clause $C = \square$. $\square$

Linear resolution

As in propositional logic, the resolution method can be significantly refined (without using completeness).

- A *linear proof* of a clause C from a formula S is a finite sequence of pairs $(C_0, B_0), \dots, (C_n, B_n)$ such that $C_0 \in S$ and for every $i \leq n$
  - *i)* $B_i \in S$ or $B_i = C_j$ for some $j < i$, and
  - *ii)* C_{i+1} is a resolvent of C_i and B_i where $C_{n+1} = C$.
- C_0 is called a *starting* clause, C_i a *central* clause, B_i a *side* clause.
- C is *linearly provable* from S, $S \vdash_L C$, if it has a linear proof from S.
- A *linear refutation* of S is a linear proof of $\square$ from S.
- S is *linearly refutable* if $S \vdash_L \square$.

Theorem *S is linearly refutable, if and only if it is unsatisfiable.*

Proof ($\Rightarrow$) Every linear proof can be transformed to a (general) resolution proof. ($\Leftarrow$) Follows from completeness of propositional resolution, the lifting lemma preserves linearity of proofs. $\square$

LI-resolution

As in prop. logic, for Horn formulas we can further refine linear resolution.

- *LI-resolution* ("linear input") from S is a linear resolution from S in which every side clause B_i is a variant of a clause from S. We write $S \vdash_{LI} C$ to denote that C is provable by LI-resolution from S.
- a *Horn clause* is a clause containing at most one positive literal,
- a *Horn formula* is a (possibly infinite) set of Horn clauses,
- a *fact* is a (Horn) clause $\{p\}$ where p is a positive literal,
- a *rule* is a (Horn) clause with exactly one positive literal and at least one negative literal. Rules and facts are *program clauses*,
- a *goal* is a nonempty (Horn) clause with only negative literals.

Theorem *If T is a satisfiable Horn formula but $T \cup \{G\}$ is unsat. for some goal G, then $\square$ has a LI-resolution from $T \cup \{G\}$ with starting clause G.*

Proof Follows from Herbrand's Theorem, the same theorem in propositional logic, and the lifting lemma.

A program in Prolog

A (Prolog) program is a Horn formula containing only program clauses, i.e. facts and rules.

$son(X,Y) :- father(Y,X), man(X)$	$\{son(X,Y), \neg father(Y,X), \neg man(X)\}$
$son(X,Y) :- mother(Y,X), man(X)$	$\{son(X,Y), \neg mother(Y,X), \neg man(X)\}$
$man(john).$	$\{man(john)\}$
$father(george, john).$	$\{father(george, john)\}$
$mother(julie, john).$	$\{mother(julie, john)\}$

$? - son(john, X)$ $\quad P \models (\exists X)son(john, X)$ $\quad \{\neg son(john, X)\}$

We want to know if the given query follows from the program.

Theorem *Let P be a program and $G = \{\neg A_1, \ldots, \neg A_n\}$ a goal in variables $X_1, \ldots, X_m$. TFAE:*

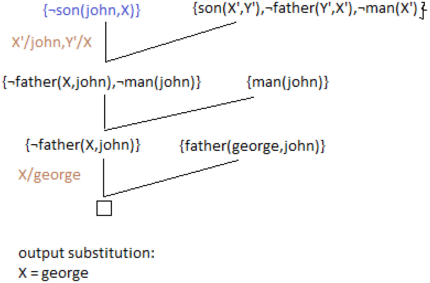
(1) $P \models (\exists X_1) \ldots (\exists X_m)(A_1 \wedge \cdots \wedge A_n)$, *if and only if*

(2) $\square$ *has a LI-resolution from $P \cup \{G\}$ staring with (a variant of) the goal G.*

LI-resolution over the program

If the answer to the query is positive, we also want to know the output substitution. The output substitution σ for LI-resolution of $\square$ from $P \cup \{G\}$ starting from $G = \{\neg A_1, \ldots, \neg A_n\}$ is the composition of mgu from individual steps (only for variables of G. Note that:

$$P \models (A_1 \wedge \ldots A_n)\sigma$$



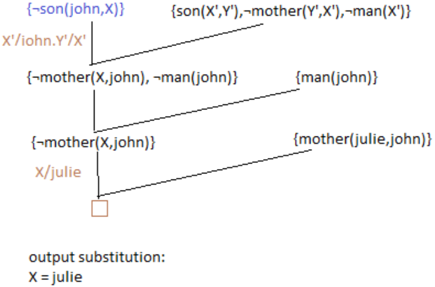


Table of Contents

Hilbert's calculus in predicate logic

- basic connectives and quantifier: $\neg$, $\rightarrow$, $(\forall x)$ (others are derived)
- allows to prove any formula (not just sentences)
- *logical axioms* (schemes of axioms):

$$
\begin{array}{rll}
(i) & \varphi \rightarrow (\psi \rightarrow \varphi) & \\
(ii) & (\varphi \rightarrow (\psi \rightarrow \chi)) \rightarrow ((\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \chi)) & \\
(iii) & (\neg\varphi \rightarrow \neg\psi) \rightarrow (\psi \rightarrow \varphi) & \\
(iv) & (\forall x)\varphi \rightarrow \varphi(x/t) & \text{if } t \text{ is substitutable for } x \text{ to } \varphi \\
(v) & (\forall x)(\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow (\forall x)\psi) & \text{if } x \text{ is not free in } \varphi
\end{array}
$$

  where φ, ψ, χ are any formulas (of a given language), t is any term, and x is any variable
- in a language with equality we include also the axioms of equality
- *rules of inference*

$$\frac{\varphi,\ \varphi \rightarrow \psi}{\psi} \quad \text{(modus ponens)}, \qquad \frac{\varphi}{(\forall x)\varphi} \quad \text{(generalization)}$$

Hilbert-style proofs

A *proof* (in *Hilbert-style*) of a formula φ from a theory T is a finite sequence $\varphi_0, \ldots, \varphi_n = \varphi$ of formulas such that for every $i \leq n$

- φ_i is a logical axiom or $\varphi_i \in T$ (an axiom of the theory), or
- φ_i can be inferred from the previous formulas applying a rule of inference.

A formula φ is *provable* from T if it has a proof from T, denoted by $T \vdash_H \varphi$.

Theorem (soundness) *For every T and φ, $T \vdash_H \varphi \;\Rightarrow\; T \models \varphi$.*

Proof

- If φ is an axiom (logical or from T), then $T \models \varphi$ (l. axioms are tautologies),
- if $T \models \varphi$ and $T \models \varphi \rightarrow \psi$, then $T \models \psi$, i.e. modus ponens is sound,
- if $T \models \varphi$, then $T \models (\forall x)\varphi$, i.e. generalization is sound,
- thus every formula in a proof from T is valid in T. $\square$

Remark *The completeness holds as well, i.e. $T \models \varphi \Rightarrow T \vdash_H \varphi$.*

Table of Contents

Basic algebraic theories

- theory of *groups* in the language $L = \langle +, -, 0 \rangle$ with equality:

$$x + (y + z) = (x + y) + z \qquad \text{(associativity of } +\text{)}$$

$$0 + x = x = x + 0 \qquad \text{(0 is neutral to } +\text{)}$$

$$x + (-x) = 0 = (-x) + x \qquad (-x \text{ is inverse of } x)$$

- theory of *Abelian groups* has moreover ax. $x + y = y + x$ (commutativity)
- theory of *rings* in $L = \langle +, -, \cdot, 0, 1 \rangle$ with equality has additionally

$$1 \cdot x = x = x \cdot 1 \qquad \text{(1 is neutral to } \cdot\text{)}$$

$$x \cdot (y \cdot z) = (x \cdot y) \cdot z \qquad \text{(associativity of } \cdot\text{)}$$

$$x \cdot (y + z) = x \cdot y + x \cdot z,\ (x + y) \cdot z = x \cdot z + y \cdot z \qquad \text{(distributivity)}$$

- theory of *commutative rings* has moreover the axiom $x \cdot y = y \cdot x$ (commutativity)
- theory of *fields* in the same language has additionally the axioms

$$x \neq 0 \rightarrow (\exists y)(x \cdot y = 1) \qquad \text{(existence of inverses to } \cdot\text{)}$$

$$0 \neq 1 \qquad \text{(nontriviality)}$$

Theories of structures

What properties hold in particular structures?

The *theory of a structure* $\mathcal{A}$ is the set $\mathrm{Th}(\mathcal{A})$ of all sentences (of the same language) that are valid in $\mathcal{A}$.

Observation *For every structure $\mathcal{A}$ and a theory T of a language L,*

- (i) $\mathrm{Th}(\mathcal{A})$ *is a complete theory,*
- (ii) *if $\mathcal{A} \models T$, then $\mathrm{Th}(\mathcal{A})$ is a simple (complete) extension of T,*
- (iii) *if $\mathcal{A} \models T$ and T is complete, then $\mathrm{Th}(\mathcal{A})$ is equivalent with T, i.e. $\theta^L(T) = \mathrm{Th}(\mathcal{A})$.*

E.g. $\mathrm{Th}(\underline{\mathbb{N}})$ where $\underline{\mathbb{N}} = \langle \mathbb{N}, S, +, \cdot, 0, \leq \rangle$ is the arithmetics of natural numbers.

Remark *Later, we will see that $\mathrm{Th}(\underline{\mathbb{N}})$ is (algorithmically) undecidable although it is complete.*

Elementary equivalence

- Structures $\mathcal{A}$ and $\mathcal{B}$ of a language L are *elementarily equivalent*, denoted $\mathcal{A} \equiv \mathcal{B}$, if they satisfy the same sentences (of L), i.e. $\mathrm{Th}(\mathcal{A}) = \mathrm{Th}(\mathcal{B})$.

  For example, $\langle \mathbb{R}, \leq \rangle \equiv \langle \mathbb{Q}, \leq \rangle$ *and* $\langle \mathbb{Q}, \leq \rangle \not\equiv \langle \mathbb{Z}, \leq \rangle$ *since every element has an immediate successor in* $\langle \mathbb{Z}, \leq \rangle$ *but not in* $\langle \mathbb{Q}, \leq \rangle$*.*

- T is complete iff it has a single model, up to elementary equivalence.

  For example, the theory of dense linear orders without ends (DeLO).

How to describe models of a given theory (up to elementary equivalence)?

Observation *For every models* $\mathcal{A}$, $\mathcal{B}$ *of a theory* T, $\mathcal{A} \equiv \mathcal{B}$ *if and only if* $\mathrm{Th}(\mathcal{A})$, $\mathrm{Th}(\mathcal{B})$ *are equivalent (simple complete extensions of* T*).*

Remark *If we can describe effectively (recursively) for a given theory* T *all simple complete extensions of* T*, then* T *is (algorithmically) decidable.*

Simple complete extensions - an example

The theory $DeLO^*$ of dense linear orders of $L = \langle \leq \rangle$ with equality:

$$x \leq x \qquad \text{(reflexivity)}$$

$$x \leq y \;\wedge\; y \leq x \;\rightarrow\; x = y \qquad \text{(antisymmetry)}$$

$$x \leq y \;\wedge\; y \leq z \;\rightarrow\; x \leq z \qquad \text{(transitivity)}$$

$$x \leq y \;\vee\; y \leq x \qquad \text{(dichotomy)}$$

$$x < y \;\rightarrow\; (\exists z)\,(x < z \;\wedge\; z < y) \qquad \text{(density)}$$

$$(\exists x)(\exists y)(x \neq y) \qquad \text{(nontriviality)}$$

where '$x < y$' is a shortcut for '$x \leq y \;\wedge\; x \neq y$'.

Let φ, ψ be the sentences $(\exists x)(\forall y)(x \leq y)$, resp. $(\exists x)(\forall y)(y \leq x)$. We will show that the following are all (inequivalent) simple complete extensions of the theory $DeLO^*$:

$$DeLO \;= DeLO^* \cup \{\neg\varphi, \neg\psi\}, \qquad DeLO^{\pm} = DeLO^* \cup \{\varphi, \psi\},$$

$$DeLO^{+} = DeLO^* \cup \{\neg\varphi, \psi\}, \qquad DeLO^{-} = DeLO^* \cup \{\varphi, \neg\psi\}$$

Corollary of the Löwenheim-Skolem theorem

We already know the following theorem, by a canonical model (with $=$).

Theorem *Let T be a consistent theory of a countable language L. If L is without equality, then T has a countably infinite model. If L is with equality, then T has a model that is countable (finite or countably infinite).*

Corollary *For every structure $\mathcal{A}$ of a countable language without equality there exists a countably infinite structure $\mathcal{B}$ with $\mathcal{A} \equiv \mathcal{B}$.*

Proof $\mathrm{Th}(\mathcal{A})$ is consistent since it has a model $\mathcal{A}$. By the previous theorem, it has a countably inf. model $\mathcal{B}$. Since $\mathrm{Th}(\mathcal{A})$ is complete, we have $\mathcal{A} \equiv \mathcal{B}$. $\square$

Corollary *For every infinite structure $\mathcal{A}$ of a countable language with equality there exists a countably infinite structure $\mathcal{B}$ with $\mathcal{A} \equiv \mathcal{B}$.*

Proof Similarly as above. Since the sentence *"there is exactly n elements"* is false in $\mathcal{A}$ for all n and $\mathcal{A} \equiv \mathcal{B}$, it follows that B is infinite. $\square$

A countable algebraically closed field

We say that a field $\mathcal{A}$ is *algebraically closed* if every polynomial (of nonzero degree) has a root in $\mathcal{A}$; that is, for every $n \geq 1$ we have

$$\mathcal{A} \models (\forall x_{n-1}) \ldots (\forall x_0)(\exists y)(y^n + x_{n-1} \cdot y^{n-1} + \cdots + x_1 \cdot y + x_0 = 0)$$

where y^k is a shortcut for the term $y \cdot y \cdot \, \cdots \, \cdot y$ ($\cdot$ applied $(k-1)$-times).

For example, the field $\underline{\mathbb{C}} = \langle \mathbb{C}, +, -, \cdot, 0, 1 \rangle$ *is algebraically closed, whereas the fields* $\underline{\mathbb{R}}$ *and* $\underline{\mathbb{Q}}$ *are not (since the polynomial* $x^2 + 1$ *has no root in them).*

Corollary *There exists a countable algebraically closed field.*

Proof By the previous corollary, there is a countable structure elementarily equivalent with the field $\underline{\mathbb{C}}$. Hence it is algebraically closed as well. $\square$

Isomorphisms of structures

Let $\mathcal{A}$ and $\mathcal{B}$ be structures of a language $L = \langle \mathcal{F}, \mathcal{R} \rangle$.

- A bijection $h \colon A \to B$ is an *isomorphism* of structures $\mathcal{A}$ and $\mathcal{B}$ if

  (i) $h(f^A(a_1, \ldots, a_n)) = f^B(h(a_1), \ldots, h(a_n))$
  for every n-ary function symbol $f \in \mathcal{F}$ and every $a_1, \ldots, a_n \in A$,

  (ii) $R^A(a_1, \ldots, a_n) \quad \Leftrightarrow \quad R^B(h(a_1), \ldots, h(a_n))$
  for every n-ary relation symbol $R \in \mathcal{R}$ and every $a_1, \ldots, a_n \in A$.

- $\mathcal{A}$ and $\mathcal{B}$ are *isomorphic* (via h), denoted $\mathcal{A} \simeq \mathcal{B}$ ($\mathcal{A} \simeq_h \mathcal{B}$), if there is an isomorphism h of $\mathcal{A}$ and $\mathcal{B}$. We also say $\mathcal{A}$ is *isomorphic with* $\mathcal{B}$.

- An *automorphism* of a structure $\mathcal{A}$ is an isomorphism of $\mathcal{A}$ with $\mathcal{A}$.

For example, the power set algebra $\underline{\mathcal{P}(X)} = \langle \mathcal{P}(X), -, \cap, \cup, \emptyset, X \rangle$ with $X = n$ is isomorphic to the Boolean algebra $\underline{{}^n2} = \langle {}^n2, -_n, \wedge_n, \vee_n, 0_n, 1_n \rangle$ via $h : A \mapsto \chi_A$ where χ_A is the characteristic function of the set $A \subseteq X$.

Isomorphisms and semantics

We will see that isomorphism preserves semantics.

Proposition *Let $\mathcal{A}$ and $\mathcal{B}$ be structures of a language $L = \langle \mathcal{F}, \mathcal{R} \rangle$. A bijection $h\colon A \to B$ is an isomorphism of $\mathcal{A}$ and $\mathcal{B}$ if and only if both*

(*i*) $h(t^A[e]) = t^B[he]$ *for every term t and $e\colon \mathrm{Var} \to A$,*

(*ii*) $\mathcal{A} \models \varphi[e] \quad \Leftrightarrow \quad \mathcal{B} \models \varphi[he]$ *for every formula φ and $e\colon \mathrm{Var} \to A$.*

Proof ($\Rightarrow$) By induction on the structure of t, resp. φ.

($\Leftarrow$) By applying (*i*) for each term $f(x_1, \ldots, x_n)$ or (*ii*) for each atomic formula $R(x_1, \ldots, x_n)$ and assigning $e(x_i) = a_i$ we verify that h is an isomorphism. $\square$

Corollary *For every structures $\mathcal{A}$ and $\mathcal{B}$ of the same language,*

$$\mathcal{A} \simeq \mathcal{B} \quad \Rightarrow \quad \mathcal{A} \equiv \mathcal{B}.$$

Remark *The other implication ($\Leftarrow$) does not hold in general. For example, $\langle \mathbb{Q}, \leq \rangle \equiv \langle \mathbb{R}, \leq \rangle$ but $\langle \mathbb{Q}, \leq \rangle \not\simeq \langle \mathbb{R}, \leq \rangle$ since $|\mathbb{Q}| = \omega$ and $|\mathbb{R}| = 2^\omega$.*

Finite models in language with equality

Proposition *For every finite structures $\mathcal{A}$, $\mathcal{B}$ of a language with equality,*

$$\mathcal{A} \equiv \mathcal{B} \quad \Rightarrow \quad \mathcal{A} \simeq \mathcal{B}.$$

Proof $|A| = |B|$ since we can express *"there are exactly n elements"*.

- Let $\mathcal{A}'$ be expansion of $\mathcal{A}$ to $L' = L \cup \{c_a\}_{a \in A}$ by names of elements.
- We show that $\mathcal{B}$ has an expansion $\mathcal{B}'$ to L' such that $\mathcal{A}' \equiv \mathcal{B}'$. Then clearly $h\colon a \mapsto c_a^{B'}$ is an isomorfism of $\mathcal{A}'$ to $\mathcal{B}'$, and thus also $\mathcal{A}$ to $\mathcal{B}$.
- If suffices to find $b \in B$ for every $c_a^{A'} = a \in A$ s.t. $\langle \mathcal{A}, a \rangle \equiv \langle \mathcal{B}, b \rangle$.
- Let Ω be set of all formulas $\varphi(x)$ s.t. $\langle \mathcal{A}, a \rangle \models \varphi(x/c_a)$, i.e. $\mathcal{A} \models \varphi[e(x/a)]$
- Since A is finite, there are finitely many formulas $\varphi_0(x), \ldots, \varphi_m(x)$ such that for every $\varphi \in \Omega$ it holds $\mathcal{A} \models \varphi \leftrightarrow \varphi_i$ for some i.
- Since $\mathcal{B} \equiv \mathcal{A} \models (\exists x) \bigwedge_{i \leq m} \varphi_i$, there exists $b \in B$ s.t. $\mathcal{B} \models \bigwedge_{i \leq m} \varphi_i[e(x/b)]$.
- Hence for every $\varphi \in \Omega$ it holds $\mathcal{B} \models \varphi[e(x/b)]$, i.e. $\langle \mathcal{B}, b \rangle \models \varphi(x/c_a)$. $\square$

Corollary *If a complete theory T in a language with equality has a finite model, then all models of T are isomorphic.*

NAIL062 Propositional & Predicate Logic: Lecture 13

Slides by Petr Gregor with minor modifications by Jakub Bulín

January 4, 2021

Definable sets and automorphisms

The set defined by $\varphi(\bar{x}, \bar{y})$ with parameters $\bar{b} \in A^{|\bar{y}|}$ in $\mathcal{A}$ is

$$\varphi^{\mathcal{A},\bar{b}}(\bar{x}, \bar{y}) = \{\bar{a} \in A^{|\bar{x}|} \mid \mathcal{A} \models \varphi[e(\bar{x}/\bar{a}, \bar{y}/\bar{b})]\}$$

Proposition Let $D \subseteq A^n$ be a set definable in a structure $\mathcal{A}$ with parameters $\bar{b}$ and let h be an automorphism of $\mathcal{A}$ which is identical on $\bar{b}$. Then $h[D] = D$.

Proof Let $D = \varphi^{\mathcal{A},\bar{b}}(\bar{x}, \bar{y})$. Then for any $\bar{a} \in A^{|\bar{x}|}$:

$$\begin{aligned} \bar{a} \in D &\Leftrightarrow \mathcal{A} \models \varphi[e(\bar{x}/\bar{a}, \bar{y}/\bar{b})] \\ &\Leftrightarrow \mathcal{A} \models \varphi[(e \circ h)(\bar{x}/\bar{a}, \bar{y}/\bar{b})] \\ &\Leftrightarrow \mathcal{A} \models \varphi[e(\bar{x}/h(\bar{a}), \bar{y}/h(\bar{b}))] \\ &\Leftrightarrow \mathcal{A} \models \varphi[e(\bar{x}/h(\bar{a}), \bar{y}/\bar{b})] \\ &\Leftrightarrow h(\bar{a}) \in D \end{aligned}$$

Example: find automorphisms of a given graph.

Categoricity

- The (isomorphism) *spectrum* of a theory T is given by the number $I(\kappa, T)$ of mutually nonisomorphic models of T for every cardinality κ.
- A theory T is *κ-categorical* if it has exactly one (up to isomorphism) model of cardinality κ, i.e. $I(\kappa, T) = 1$.

Proposition *The theory DeLO (i.e. "without ends") is ω-categorical.*

Proof Let $\mathcal{A}, \mathcal{B} \models DeLO$ with $A = \{a_i\}_{i\in\mathbb{N}}$, $B = \{b_i\}_{i\in\mathbb{N}}$. By induction on n we can find injective partial functions $h_n \subseteq h_{n+1} \subset A \times B$ preserving the ordering s.t. $\{a_i\}_{i<n} \subseteq \mathrm{dom}(h_n)$ and $\{b_i\}_{i<n} \subseteq \mathrm{rng}(h_n)$. Then $\mathcal{A} \simeq \mathcal{B}$ via $h = \bigcup_n h_n$. $\square$

Similarly we obtain that (e.g.) $\mathcal{A} = \langle \mathbb{Q}, \leq \rangle$, $\mathcal{A} \restriction (0,1]$, $\mathcal{A} \restriction [0,1)$, $\mathcal{A} \restriction [0,1]$ are (up to isomorphism) all countable models of $DeLO^*$. Then

$$I(\kappa, DeLO^*) = \begin{cases} 0 & \text{for } \kappa \in \mathbb{N}, \\ 4 & \text{for } \kappa = \omega. \end{cases}$$

ω-categorical criterium of completeness

Theorem *Let L be at most countable language.*

1. *If a theory T in L without equality is ω-categorical, then it is complete.*
2. *If a theory T in L with equality is ω-categorical and without finite models, then it is complete.*

Proof Every model of T is elementarily equivalent with some countably infinite model of T, but such model is unique up to isomorphism. Thus all models of T are elementarily equivalent, i.e. T is complete. □

For example, $DeLO$, $DeLO^+$, $DeLO^-$, $DeLO^\pm$ are complete and they are the all (mutually nonequivalent) simple complete extensions of $DeLO^*$.

Remark *A similar criterium holds also for cardinalities bigger than ω.*

Axiomatizability

Can the given part of the world be "nicely" described?

Let $K \subseteq M(L)$ be a class of L-structures. We say that K is

- axiomatizable if there exists a theory T such that $M(T) = K$,
- finitely axiomatizable if is it axiomatizable by a finite theory, and
- openly axiomatizable if is it axiomatizable by an open theory.
- a theory T is finitely [openly] axiomatizable if $M(T)$ is.

Observation If K is axiomatizable, then it is closed under elementary equivalence.

For example:

- linear orders are finitely and openly axiomatizable,
- fields are finitely but not openly axiomatizable, and
- infinite groups are axiomatiable, but not finitely axiomatizable.

A consequence of compactness

Theorem If a theory T has for every $n > 0$ an at least n-element model, then T has an infinite model.

Proof Obvious for languages without equality, conside L with $=$.

- Consider the extension $T' = T \cup \{c_i \neq c_j \mid i \neq j\}$ of T in the language extended by countably infinitely many new constant symbols c_i.
- By assumption, every finite part of T' has a model.
- By the Compactness theorem, T' has a model $\mathcal{A}'$ but that model is necessarily infinite.
- The redukt of $\mathcal{A}'$ to the original language is an infinite modle of T.

Corollary If a theory T has for every $n > 0$ an at least n-element model, then the class of all finite models of T is not axiomatizable.

For example, finite groups, finite fields etc. are not axiomatizable. But the class of all infinite models of a theory T in a language with equality is axiomatizable.

Finite axiomatizability

Theorem Let $K \subseteq M(L)$ and $\bar{K} = M(L) \setminus K$, where L is a langauge. Then K is finitely axiomatizable, if and only if both K and $\bar{K}$ are axiomatizable.

Proof ($\Rightarrow$) If T is a finite axiomatization of K in closed form, then the theory with a single axiom $\bigvee_{\varphi \in T} \neg\varphi$ axiomatizes $\bar{K}$.

($\Leftarrow$) To prove this implication:

- Let T, S be theories of a language L such that $M(T) = K$ and $M(S) = \bar{K}$.
- Then $M(T \cup S) = M(T) \cap M(S) = \emptyset$ and by compactness, there exist finite $T' \subseteq T$ and finite $S' \subseteq S$ such that $\emptyset = M(T' \cup S') = M(T') \cap M(S')$.
- The finite theory T' axiomatizes K, since

$$M(T) \subseteq M(T') \subseteq \overline{M(S')} \subseteq \overline{M(S)} = M(T).$$

Finite axiomatizability – an example

Let T be the theory of fields. We say that a field $\mathcal{A} = \langle A, +, -, \cdot, 0, 1\rangle$ is

- of characteristic 0 if there is no $p \in \mathbb{N}^+$ such that $\mathcal{A} \models p1 = 0$ where $p1$ denotes the term $1 + 1 + \cdots + 1$ (where $+$ is applied $(p-1)$-times).
- of characteristic p, where p is a prime number, if p is smallest such that $\mathcal{A} \models p1 = 0$
- The class of fields of characteristic p, for a fixed prime p, is finitely axiomatizable by the theory $T \cup \{p1 = 0\}$.
- The class of fields of characteristic 0 is axiomatized by an (infinite) theory $T' = T \cup \{p1 \neq 0 \mid p \in \mathbb{N}^+\}$.

Proposition The class K of field of characteristic 0 is not finitely axiomatizable.

Proof It suffices to show that $\bar{K}$ is not axiomatizable. If $M(S) = \bar{K}$, then $S' = S \cup T'$ has a model $\mathcal{B}$, because every finite $S^* \subseteq S'$ has a model (a field of characteristic p' where p' is a prime greater than any prime p appearing in the axioms of S^*), But then $\mathcal{B} \in M(S) = \bar{K}$ and at the same time $B \in M(T') = K$ which is not possible.

Open axiomatizability

Theorem If a theory T is openly axiomatizable, then every substructure of a model of T is also a model of T.

Proof Let T' be an open axiomatization of $M(T)$, $\mathcal{A} \models T'$ and $\mathcal{B} \subseteq \mathcal{A}$. We know that for every $\varphi \in T'$, $\mathcal{B} \models \varphi$ because φ is open. Therefore $\mathcal{B}$ is a model of T'.

Note The converse is also true: if every substructure of a model of a theory T is a model of T as well, then T is openly axiomatizable.

For example, the theory DeLO is not openly axiomatizable, because for example a finite substructure of a model of DeLO is not a model of DeLO.

As another example, at most n-element groups, for a fixed $n > 1$, are openly axiomatizable:

$$T \cup \{ \bigvee_{i,j \leq n, i \neq j} x_i = x_j \}$$

where T is the (open) theory of groups

Table of Contents

Recursive and recursively enumerable sets

Which problems are algorithmically solvable?

- The notion of *"algorithm"* can be rigorously formalized (e.g. by TM).
- We may encode decision problems into sets of natural numbers corresponding to the positive instances (with answer $\mathtt{yes}$). For example,

$$SAT = \{\lceil\varphi\rceil \mid \varphi \text{ is a satisfiable proposition in CNF}\}.$$

- A set $A \subseteq \mathbb{N}$ is *recursive* if there is an algorithm that for every input $x \in \mathbb{N}$ halts and correctly tells whether or not $x \in A$. We say that such algorithm decides $x \in A$.
- A set $A \subseteq \mathbb{N}$ is *recursively enumerable* (*r. e.*) if there is an algorithm that for every input $x \in \mathbb{N}$ halts if and only if $x \in A$. We say that such algorithm recognizes $x \in A$. Equivalently, A is recursively enumerable if there is an algorithm that generates (i.e. *enumerates*) all elements of A.

Observation *For every $A \subseteq \mathbb{N}$ it holds that A is recursive $\Leftrightarrow$ A, $\overline{A}$ are r. e.*

Decidable theories

Is the truth in a given theory algorithmically decidable?

We (always) assume that the language L is recursive. A theory T of L is *decidable* if $Thm(T)$ is recursive; otherwise, T is *undecidable*.

Proposition *For every theory T of L with recursively enumerable axioms,*

1. *$Thm(T)$ is recursively enumerable,*
2. *if T is complete, then $Thm(T)$ is recursive, i.e. T is decidable.*

Proof The construction of systematic tableau from T with a root $F\varphi$ assumes a given enumeration of axioms of T. Since T has recursively enumerable axioms, the construction provides an algorithm that recognizes $T \vdash \varphi$.

If T is complete, then $T \nvdash \varphi$ if and only if $T \vdash \neg\varphi$ for every sentence φ. Hence, the parallel construction of systematic tableaux from T with roots $F\varphi$ resp. $T\varphi$ provides an algorithm that decides $T \vdash \varphi$. $\square$

Recursively enumerable complete extensions

What happens if we are able to describe all simple complete extensions?

We say that the set of all (up to equivalence) simple complete extensions of a theory T is *recursively enumerable* if there exists an algorithm $\alpha(i, j)$ that generates i-th axiom of j-th extension (in some enumeration) or announces that it (such an axiom or an extension) does not exist.

Proposition *If a theory T has recursively enumerable axioms and the set of all (up to equivalence) simple complete extensions of T is recursively enumerable, then T is decidable.*

Proof By the previous proposition there is an algorithm to recognize $T \vdash \varphi$. On the other hand, if $T \not\vdash \varphi$ then $T' \vdash \neg\varphi$ is some simple complete extension T' of T. This can be recognized by parallel construction of systematic tableaux with root $T\varphi$ from all extensions. In the i-th step we construct tableaux up to i levels for the first i extensions. $\square$

Examples of decidable theories

The following theories are decidable although not complete.

- the theory of pure equality; with no axioms, in $L = \langle\rangle$ with equality,
- the theory of unary predicate; with no axioms, in $L = \langle U \rangle$ with equality, where U is a unary relation symbol,
- the theory of dense linear orders $DeLO^*$,
- the theory of algebraically closed fields in $L = \langle +, -, \cdot, 0, 1 \rangle$ with equality, with the axioms of fields, and the axioms (for all $n \geq 1$)

  $$(\forall x_{n-1}) \dots (\forall x_0)(\exists y)(y^n + x_{n-1} \cdot y^{n-1} + \cdots + x_1 \cdot y + x_0 = 0),$$

  where y^k is a shortcut for the term $y \cdot y \cdot \ \cdots \ \cdot y$ ($\cdot$ applied $(k-1)$-times).
- the theory of Abelian groups,
- the theory of Boolean algebras.

Recursive axiomatizability

Can we "effectively" describe common mathematical structures?

- A class $K \subseteq M(L)$ is *recursively axiomatizable* if there exists a recursive theory T of language L with $M(T) = K$.
- A theory T is recursively axiomatizable if $M(T)$ is recursively axiomatizable, i.e. there is an equivalent recursive theory.

Proposition *For every finite structure $\mathcal{A}$ of a finite language with equality the theory $\mathrm{Th}(\mathcal{A})$ is recursively axiomatizable. Thus, $\mathrm{Th}(\mathcal{A})$ is decidable.*

Proof Let $A = \{a_1, \ldots, a_n\}$. $\mathrm{Th}(\mathcal{A})$ can be axiomatized by a single sentence (thus recursively) that describes $\mathcal{A}$. It is of the form *"there are exactly n elements $a_1, \ldots, a_n$ satisfying exactly those atomic formulas on function values and relations that are valid in the structure $\mathcal{A}$."* □

Examples of recursive axiomatizability

The following structures $\mathcal{A}$ have recursively axiomatizable $\mathrm{Th}(\mathcal{A})$.

- $\langle \mathbb{Z}, \leq \rangle$, by the theory of discrete linear orderings,
- $\langle \mathbb{Q}, \leq \rangle$, by the theory of dense linear orderings without ends (*DeLO*),
- $\langle \mathbb{N}, S, 0 \rangle$, by the theory of successor with zero,
- $\langle \mathbb{N}, S, +, 0 \rangle$, by so called Presburger arithmetic,
- $\langle \mathbb{R}, +, -, \cdot, 0, 1 \rangle$, by the theory of real closed fields,
- $\langle \mathbb{C}, +, -, \cdot, 0, 1 \rangle$, by the theory of algebraically closed fields with characteristic 0.

Corollary *For all the above structures $\mathcal{A}$ the theory $\mathrm{Th}(\mathcal{A})$ is decidable.*

Remark *However, $\underline{\mathbb{N}} = \langle \mathbb{N}, S, +, \cdot, 0, \leq \rangle$ is not recursively axiomatizable. (This follows from the Gödel's incompleteness theorem).*

Robinson arithmetic

How to effectively and "almost" completely axiomatize $\underline{\mathbb{N}} = \langle \mathbb{N}, S, +, \cdot, 0, \leq \rangle$*?*

The language of arithmetic is $L = \langle S, +, \cdot, 0, \leq \rangle$ with equality.

Robinson arithmetic Q has axioms (finitely many)

$$\begin{array}{ll} S(x) \neq 0 & x \cdot 0 = 0 \\ S(x) = S(y) \to x = y & x \cdot S(y) = x \cdot y + x \\ x + 0 = x & x \neq 0 \to (\exists y)(x = S(y)) \\ x + S(y) = S(x + y) & x \leq y \leftrightarrow (\exists z)(z + x = y) \end{array}$$

Remark *Q is quite weak; for example, it does not prove commutativity or associativity of $+$, $\cdot$, or transitivity of $\leq$. However, it suffices to prove, for example, existential sentences on numerals that are true in $\underline{\mathbb{N}}$.*

For example, for $\varphi(x, y)$ in the form $(\exists z)(x + z = y)$ it is

$$Q \vdash \varphi(\underline{1}, \underline{2}), \quad \textit{where } \underline{1} = S(0) \textit{ and } \underline{2} = S(S(0)).$$

Peano arithmetic

Peano arithmetic PA has axioms of

a. Robinson arithmetic Q,
b. scheme of induction; that is, for every formula $\varphi(x, \overline{y})$ of L the axiom

$$(\varphi(0, \overline{y}) \wedge (\forall x)(\varphi(x, \overline{y}) \rightarrow \varphi(S(x), \overline{y}))) \rightarrow (\forall x)\varphi(x, \overline{y}).$$

Remark *PA is quite successful approximation of* $\mathrm{Th}(\underline{\mathbb{N}})$, *it proves all "elementary" properties that are true in* $\underline{\mathbb{N}}$ *(e.g. commutativity of* $+$*). But it is still incomplete, there are sentences that are true in* $\underline{\mathbb{N}}$ *but independent in PA.*

Remark *In the second-order language we can completely axiomatize* $\underline{\mathbb{N}}$ *(up to isomorphism) by taking directly the following (second-order) axiom of induction instead of scheme of induction*

$$(\forall X)\ ((X(0) \wedge (\forall x)(X(x) \rightarrow X(S(x)))) \rightarrow (\forall x)\ X(x)).$$

Hilbert's 10th problem

- Let $p(x_1, \ldots, x_n)$ be a polynomial with integer coefficients. Does the *Diophantine equation* $p(x_1, \ldots, x_n) = 0$ have a solution in integers?
- Hilbert (1900) *"Find an algorithm that determines in finitely many steps whether a given Diophantine equation in an arbitrary number of variables and with integer coefficient has an integer solution."*

Remark *Equivalently, one may ask for an algorithm to determine whether there is a solution in natural numbers.*

Theorem (DPRM, 1970) *The problem of existence of integer solution to a given Diophantine equation with integer coefficients is alg. undecidable.*

Corollary *There is no algorithm to determine for given polynomials* $p(x_1, \ldots, x_n)$, $q(x_1, \ldots, x_n)$ *with natural coefficients whether*

$$\underline{\mathbb{N}} \models (\exists x_1) \ldots (\exists x_n)(p(x_1, \ldots, x_n) = q(x_1, \ldots, x_n)).$$

Undecidability of predicate logic

Is there an algorithm to decide if a given sentence is (logically) true?

- We know that Robinson arithmetic Q has finitely many axioms, model $\underline{\mathbb{N}}$, and proves existential sentences on numerals that are true in $\underline{\mathbb{N}}$.
- Precisely, for every existential formula $\varphi(x_1, \dots, x_n)$ in arithmetic,
$$Q \vdash \varphi(x_1/\underline{a_1}, \dots, x_n/\underline{a_n}) \quad \Leftrightarrow \quad \underline{\mathbb{N}} \models \varphi[e(x_1/a_1, \dots, x_n/a_n)]$$
for every $a_1, \dots, a_n \in \mathbb{N}$ where $\underline{a_i}$ denotes the a_i-th numeral.
- In particular, for φ of the form $(\exists x_1) \dots (\exists x_n)(p(x_1, \dots, x_n) = q(x_1, \dots, x_n))$, where p, q are polynomials with natural coefficients (numerals) we have
$$\underline{\mathbb{N}} \models \varphi \quad \Leftrightarrow \quad Q \vdash \varphi \quad \Leftrightarrow \quad \vdash \psi \to \varphi \quad \Leftrightarrow \quad \models \psi \to \varphi,$$
where ψ is the conjunction of (closures) of all axioms of Q.
- Thus, if there were an algorithm deciding logical truth of sentences, there would be also an algorithm deciding $\underline{\mathbb{N}} \models \varphi$, which is impossible.

Table of Contents

Gödel's incompleteness theorems

Theorem (1st) *For every consistent recursively axiomatized extension T of Robinson arithmetic there is a sentence true in $\underline{\mathbb{N}}$ and unprovable in T.*

Remarks

- *"Recursively axiomatized" means that T is "effectively given".*
- *"Extension of R. arithmetic" means that T is "sufficiently strong".*
- *If, moreover, $\underline{\mathbb{N}} \models T$, the theory T is incomplete.*
- *The sentence constructed in the proof says "I am not provable in T".*
- *The proof is based on two principles:*

  (*a*) *arithmetization of syntax,*

  (*b*) *self-reference.*

Arithmetization - provability predicate

- Finite objects of syntax (symbols of language, terms, formulas, finite tableaux, proofs) can be (effectively) encoded by natural numbers.
- Let $\ulcorner\varphi\urcorner$ denote the code of formula φ and let $\underline{\varphi}$ denote the numeral (a term of arithmetic) representing $\ulcorner\varphi\urcorner$.
- If T has recursive axiomatization, the relation $\mathrm{Prf}_T \subseteq \mathbb{N}^2$ is recursive.

$$\mathrm{Prf}_T(x,y) \quad \Leftrightarrow \quad \textit{a (tableau) } y \textit{ is a proof of (a sentence) } x \textit{ in } T.$$

- If, moreover, T extends Robinson arithmetic Q, the relation Prf_T can be represented by some formula $Prf_T(x,y)$ s.t. for every $x, y \in \mathbb{N}$

$$\begin{aligned} Q &\vdash Prf_T(\underline{x},\underline{y}), && \textit{if} \quad \mathrm{Prf}_T(x,y), \\ Q &\vdash \neg Prf_T(\underline{x},\underline{y}), && \textit{otherwise.} \end{aligned}$$

- $Prf_T(x,y)$ expresses that "*y is a proof of x in T*".
- $(\exists y)Prf_T(x,y)$ expresses that "*x is provable in T*".
- If $T \vdash \varphi$, then $\underline{\mathbb{N}} \models (\exists y)Prf_T(\underline{\varphi},y)$ and moreover $T \vdash (\exists y)Prf_T(\underline{\varphi},y)$.

Self-reference principle

- *This sentence has 24 letters.*

  In formal systems self-reference is not always available straightforwardly.

- *The following sentence has 32 letters "The following sentence has 32 letters".*

  Such direct reference is available, if we can "talk" about sequences of symbols. But the above sentence is not self-referencial.

- *The following sentence written once more and then once again between quotation marks has 116 letters "The following sentence written once more and then once again between quotation marks has 116 letters".*

  With use of direct reference we can have self-reference. Instead of *"it has x letters"* we can have other properties.

Fixed-point theorem

Theorem *Let T be consistent extension of Robinson arithmetic. For every formula $\varphi(x)$ in language of theory T there is a sentence ψ s.t. $T \vdash \psi \leftrightarrow \varphi(\underline{\psi})$.*

Remark ψ is self-referencial, it says "*This formula satisfies condition φ*".

Proof (idea) Consider the *doubling* function d: for every formula $\chi(x)$

$$d(\ulcorner\chi(x)\urcorner) = \ulcorner\chi(\underline{\chi(x)})\urcorner$$

- It can be shown that d is expressible in T. Assume *(for simplicity)* that it is expressible by some term, denoted also by d.
- Then for every formula $\chi(x)$ in language of theory T it holds that
$$T \vdash d(\underline{\chi(x)}) = \underline{\chi(\underline{\chi(x)})} \tag{1}$$
- We take $\varphi(d(\underline{\varphi(d(x))}))$ for ψ. If suffices to verify that $T \vdash d(\underline{\varphi(d(x))}) = \underline{\psi}$.
- This follows from (1) for $\chi(x)$ being $\varphi(d(x))$, since in this case
$$T \vdash d(\underline{\varphi(d(x))}) = \underline{\varphi(d(\underline{\varphi(d(x))}))} \quad \square$$

Undefinability of truth

We say that a formula $\tau(x)$ *defines truth* in theory T of arithmetical language if for every sentence φ it holds that $T \vdash \varphi \leftrightarrow \tau(\underline{\varphi})$.

Theorem *Let T be consistent extension of Robinson arithmetic. Then T has no definition of truth.*

Proof By the fixed-point theorem for $\neg\tau(x)$ there is a sentence φ such that

$$T \vdash \varphi \leftrightarrow \neg\tau(\underline{\varphi}).$$

Supposing that $\tau(x)$ defines truth in T, we would have

$$T \vdash \varphi \leftrightarrow \neg\varphi,$$

which is impossible in a consistent theory T. □

Remark *This is based on the liar paradox, the sentence φ would express "This sentence is not true in T".*

Proof of the first incompleteness theorem

Theorem (Gödel) *For any consistent recursively axiomatized extension T of Robinson arithmetic there is a sentence true in $\underline{\mathbb{N}}$ and unprovable in T.*

Proof Let $\varphi(x)$ be $\neg(\exists y)Prf_T(x, y)$, it says *"x is not provable in T"*.

- By the fixed-point theorem for $\varphi(x)$ there is a sentence ψ_T such that

$$T \vdash \psi_T \leftrightarrow \neg(\exists y)Prf_T(\underline{\psi_T}, y). \tag{2}$$

  ψ_T says *"I am not provable in T"*. More precisely, ψ_T is equivalent to a sentence expressing that ψ_T is not provable T (where the equivalence holds both in $\underline{\mathbb{N}}$ and in T).

- First, we show ψ_T *is not provable in* T. If $T \vdash \psi_T$, i.e. ψ_T is contradictory in $\underline{\mathbb{N}}$, then $\underline{\mathbb{N}} \models (\exists y)Prf_T(\underline{\psi_T}, y)$ and moreover $T \vdash (\exists y)Prf_T(\underline{\psi_T}, y)$. Thus from (2) it follows $T \vdash \neg\psi_T$, which is impossible since T is consistent.

- It remains to show ψ_T is true in $\underline{\mathbb{N}}$. If not, i.e. $\underline{\mathbb{N}} \models \neg\psi_T$, then $\underline{\mathbb{N}} \models (\exists y)Prf_T(\underline{\psi_T}, y)$. Hence $T \vdash \psi_T$, which we already disproved. $\square$

Corollaries and a strengthened version

Corollary *If, moreover, $\underline{\mathbb{N}} \models T$, then the theory T is incomplete.*

Proof Suppose T is complete. Then $T \vdash \neg\psi_T$ and thus $\underline{\mathbb{N}} \models \neg\psi_T$, which contradicts $\underline{\mathbb{N}} \models \psi_T$. □

Corollary $\mathrm{Th}(\underline{\mathbb{N}})$ *is not recursively axiomatizable.*

Proof $\mathrm{Th}(\underline{\mathbb{N}})$ is consistent extension of Robinson arithmetic and has a model $\underline{\mathbb{N}}$. Suppose $\mathrm{Th}(\underline{\mathbb{N}})$ is recursively axiomatizable. Then by previous corollary, $\mathrm{Th}(\underline{\mathbb{N}})$ is incomplete, but $\mathrm{Th}(\underline{\mathbb{N}})$ is clearly complete. □

Gödel's first incompleteness theorem can be strengthened as follows.

Theorem (Rosser) *Every consistent recursively axiomatized extension T of Robinson arithmetic has an independent sentence. Thus T is incomplete.*

Remark *Hence the assumption in the first corollary that $\underline{\mathbb{N}} \models T$ is superfluous.*

Gödel's second incompleteness theorem

Let Con_T denote the sentence $\neg(\exists y)Prf_T(\underline{0=1}, y)$. We have that $\underline{\mathbb{N}} \models Con_T \Leftrightarrow T \nvdash 0 = \underline{1}$. Thus Con_T expresses that *"T is consistent"*.

Theorem (Gödel) *For every consistent recursively axiomatized extension T of Peano arithmetic it holds that Con_T is unprovable in T.*

Proof (idea) Let ψ_T be the Gödel's sentence *"This is not provable in T"*.

- In the first part of the proof of the 1st theorem we showed that

  $$\text{"If } T \text{ is consistent, then } \psi_T \text{ is not provable in } T\text{."} \tag{3}$$

  In other words, we showed it holds $Con_T \rightarrow \psi_T$.
- If T is an extension of Peano arithmetic, the proof of (3) can be formalized within the theory T itself. Hence $T \vdash Con_T \rightarrow \psi_T$.
- Since T is consistent by the assumption, from (3) we have $T \nvdash \psi_T$.
- Therefore from the previous two bullets, it follows that $T \nvdash Con_T$. □

Remark *Hence such a theory T cannot prove its own consistency.*

Corollaries of the second theorem

Corollary *Peano arithmetic has a model $\mathcal{A}$ s.t.* $\mathcal{A} \models (\exists y) Prf_{PA}(\underline{0 = 1}, y)$.

Remark *$\mathcal{A}$ has to be nonstandard model of PA, the witness must be some nonstandard element (other than a value of a numeral).*

Corollary *There is a consistent recursively axiomatized extension T of Peano arithmetic such that $T \vdash \neg Con_T$.*

Proof Let $T = PA \cup \{\neg Con_{PA}\}$. Then T is consistent since $PA \nvdash Con_{PA}$. Moreover, $T \vdash \neg Con_{PA}$, i.e. T proves inconsistency of $PA \subseteq T$, and thus also $T \vdash \neg Con_T$. □

Remark *$\underline{\mathbb{N}}$ cannot be a model of T.*

Corollary *If the set theory ZFC is consistent, then Con_{ZFC} is unprovable in ZFC.*