Finite Model Theory

George Voutsadakis[1]

[1]Mathematics and Computer Science
Lake Superior State University

LSSU Math 600

Subsection 1

Structures

Vocabularies

- **Vocabularies** are finite sets that consist of:
  - **Relation symbols** $P, Q, R, \ldots$;
    Every relation symbol is equipped with a natural number ≥ 1, its **arity**.
  - **Constant symbols** (**constants** for short) $c, d, \ldots$.

  We denote vocabularies by $\tau, \sigma, \ldots$.
- A vocabulary is **relational**, if it does not contain constants.

Structures

- A **structure** $\mathcal{A}$ of vocabulary τ, or a τ-**structure**, consists of:
  - A nonempty set A, the **universe** or **domain** of $\mathcal{A}$;
  - An n-ary relation $R^{\mathcal{A}}$ on A for every n-ary relation symbol R in τ;
  - An element $c^{\mathcal{A}}$ of A for every constant c in τ.
- Mostly we use the notations R^A for $R^{\mathcal{A}}$ and c^A for $c^{\mathcal{A}}$.
- An ***n*-ary relation** R^A on A is a subset of A^n, the set of n-tuples of elements of A.
- We mostly write $R^A a_1 \dots a_n$ instead of $(a_1, \dots, a_n) \in R^A$.
- A structure $\mathcal{A}$ is **finite**, if its universe A is a finite set.

Example: Graphs

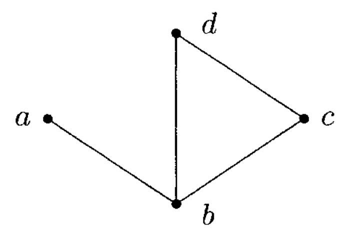
- Let $\tau = \{E\}$ with a binary relation symbol E.

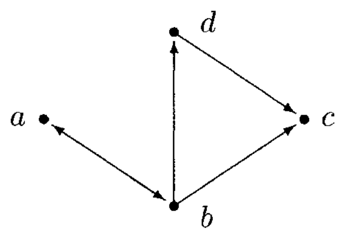
  A **graph**, or **undirected graph**, is a τ-structure $\mathcal{G} = (G, E^G)$ satisfying:
  (1) For all $a \in G$, not $E^G aa$;
  (2) For all $a, b \in G$, if $E^G ab$ then $E^G ba$.
- By GRAPH we denote the class of *finite graphs*.
- If only (1) is required, we speak of a **digraph**, or **directed graph**.
- The elements of G are sometimes called **points**, or **vertices**, and the elements of E^G **edges**.

Examples

- The figure on the left represents the graph

$$(\{a,b,c,d\},\{(a,b),(b,a),(b,c),(c,b),(b,d),(d,b),(c,d),(d,c)\}).$$





- The figure on the right represents the digraph

$$(\{a,b,c,d\},\{(a,b),(b,a),(b,c),(b,d),(d,c)\}).$$

Cliques, Paths, Cycles and Hamiltonian Circuits

- A subset X of the universe of a graph $\mathcal{G}$ is a **clique**, if

$$E^G ab, \text{ for all } a, b \in X, \ a \neq b.$$

- Let $\mathcal{G}$ be a digraph. If $n \geq 1$ and

$$E^G a_0 a_1, E^G a_1 a_2, \ldots, E^G a_{n-1} a_n,$$

  then $a_0, a_1, \ldots, a_n$ is a **path** from a_0 to a_n of **length** n.
- If $a_0 = a_n$ then $a_0, \ldots, a_n$ is a **cycle**.
- $\mathcal{G}$ is **acyclic** if it has no cycle.
- A path $a_0, a_1, \ldots, a_n$ is **Hamiltonian** if $G = \{a_0, \ldots, a_n\}$ and $a_i \neq a_j$, for $i \neq j$.

  If, in addition, $E^G a_n a_0$, we speak of a **Hamiltonian circuit**.

Connected Components

- Let $\mathcal{G}$ be a graph.

  Write $a \sim b$ if $a = b$ or if there is a path from a to b.
- Clearly, $\sim$ is an equivalence relation.
- The equivalence class of a is called the (**connected**) **component** of a.
- $\mathcal{G}$ is **connected** if $a \sim b$, for all $a, b \in G$.

  I.e., $\mathcal{G}$ is connected if there is only one connected component.
- Let CONN be the class of finite connected graphs.

Distance Function

- Denote by $d(a,b)$ the length of a shortest path from a to b.
- More precisely, define the **distance function** $d : G \times G \to \mathbb{N} \cup \{\infty\}$ by

$$d(a,b) = \infty \quad \text{iff} \quad a \nsim b; \quad d(a,b) = 0 \quad \text{iff} \quad a = b;$$

  and otherwise

$$d(a,b) = \min\{n \geq 1 : \text{there is a path from } a \text{ to } b \text{ of length } n\}.$$

- Obviously,

$$d(a,c) \leq d(a,b) + d(b,c),$$

  where we use the natural conventions for ∞.

Degrees

- The following definitions apply only for *finite* digraphs.
- A vertex b is a **successor** of a vertex a (and a a **predecessor** of b) if $E^G ab$.
- The **in-degree** of a vertex is the number of its predecessors.
- The **out-degree** of a vertex is the number of its successors.
- In graphs the in-degree and the out-degree of a vertex a coincide and are called the **degree** of a.
- A **root** of a digraph is a vertex with in-degree 0.
- A **leaf** is a vertex with out-degree 0.

Trees and Forests

- A **forest** is an acyclic digraph where each vertex has in-degree at most 1.
- A **tree** is a forest with connected underlying graph, i.e., a forest (G, E^G), such that $(G, \{(a, b) : E^G ab \text{ or } E^G ba\})$ is connected.
- Note that a finite tree has exactly one root.
- Let TREE be the class of finite trees.

Example: Orderings

- Let $\tau = \{<\}$, with a binary relation symbol $<$.
- A τ-structure $\mathcal{A} = (A, <^A)$ is called an **ordering** if, for all $a, b, c \in A$:
  (1) not $a <^A a$;
  (2) $a <^A b$ or $b <^A a$ or $a = b$;
  (3) if $a <^A b$ and $b <^A c$, then $a <^A c$.

Second View of Orderings

- Sometimes we consider finite orderings also as $\{<, S, \min, \max\}$-structures where:
  - S is a binary relation symbol representing the **successor relation**;
  - min and max are constants for the **first** and the **last element** of the ordering.

  When considering the natural ordering on $\{0, \ldots, n\}$ we often refer to min as the **zero-th element**.
- Thus, a finite $\{<, S, \min, \max\}$-structure $\mathcal{A}$ is an ordering if, in addition to Conditions (1), (2), (3), for all $a, b \in A$:
  (4) $S^A ab$ iff ($a <^A b$ and for all c, if $a <^A c$, then $b <^A c$ or $b = c$);
  (5) $\min^A <^A a$ or $\min^A = a$;
  (6) $a <^A \max^A$ or $a = \max^A$.

Finite Ordered Structures

- It might be advantageous to consider finite orderings as $\{<, \min, \max\}$-structures.
- Suppose that τ_0 is a vocabulary with $\{<\} \subseteq \tau_0 \subseteq \{<, S, \min, \max\}$.
- Let σ be an arbitrary vocabulary with $\tau_0 \subseteq \sigma$.
- A finite σ-structure $\mathcal{A}$ is said to be **ordered** if the reduct $A|_{\tau_0}$, i.e., the τ_0-structure obtained from $\mathcal{A}$ by forgetting the interpretations of the symbols in $\sigma \backslash \tau_0$, is an ordering.
- The class of finite ordered σ-structures is denoted $\mathcal{O}[\sigma]$.

Isomorphic Structures

- Let $\mathcal{A}$ and $\mathcal{B}$ be two τ-structures.
- An **isomorphism** from $\mathcal{A}$ to $\mathcal{B}$, is a bijection $\pi : A \to B$ preserving relations and constants, that is:
  - For n-ary $R \in \tau$ and $a_1, \ldots, a_n \in A$,
  $$R^A a_1 \ldots a_n \quad \text{iff} \quad R^B \pi(a_1) \ldots \pi(a_n);$$
  - For $c \in \tau$,
  $$\pi(c^A) = c^B.$$
- Two τ-structures $\mathcal{A}$ and $\mathcal{B}$ are **isomorphic**, written $\mathcal{A} \cong \mathcal{B}$, if there is an isomorphism from $\mathcal{A}$ to $\mathcal{B}$.

Products of Structures

- For τ-structures $\mathcal{A}$ and $\mathcal{B}$, the **product** $\mathcal{A} \times \mathcal{B}$ of $\mathcal{A}$ and $\mathcal{B}$ is the τ-structure with domain

$$A \times B := \{(a, b) : a \in A, b \in B\},$$

  which is given by:
  - For n-ary R in τ and $(a_1, b_1), \ldots, (a_n, b_n) \in A \times B$,

  $$R^{A \times B}(a_1, b_1) \ldots (a_n, b_n) \quad \text{iff} \quad R^A a_1 \ldots a_n \quad \text{and} \quad R^B b_1 \ldots b_n;$$

  - For c in τ,

  $$c^{A \times B} := (c^A, c^B).$$

Union of Structures

- For relational τ, we introduce the **union** (or **disjoint union**) of structures.
- Assume that $\mathcal{A}$ and $\mathcal{B}$ are τ-structures with $A \cap B = \varnothing$.
- Then $\mathcal{A} \mathbin{\dot\cup} \mathcal{B}$, the **union** of $\mathcal{A}$ and $\mathcal{B}$, is the τ-structure with domain $A \cup B$ and for any R in τ

$$R^{\mathcal{A} \dot\cup \mathcal{B}} := R^{\mathcal{A}} \cup R^{\mathcal{B}}.$$

- In case $\mathcal{A}$ and $\mathcal{B}$ are structures with $A \cap B \neq \varnothing$, we take isomorphic copies A' of A and B' of B with disjoint universes (e.g., with universes $A \times \{1\}$ and $B \times \{2\}$) and set

$$\mathcal{A} \mathbin{\dot\cup} \mathcal{B} := \mathcal{A}' \mathbin{\dot\cup} \mathcal{B}'.$$

- Note that the union of ordered structures is not an ordered structure.

Ordered Sum of Structures

- Let τ, with $< \in \tau$, be a relational vocabulary.
- Let $\mathcal{A}$ and $\mathcal{B}$ be ordered τ-structures.
- Assume that $A \cap B = \varnothing$.
- Define $\mathcal{A} \triangleleft \mathcal{B}$, the **ordered sum** of $\mathcal{A}$ and $\mathcal{B}$, as $\mathcal{A} \uplus \mathcal{B}$ but setting

$$<^{\mathcal{A} \triangleleft \mathcal{B}} := <^{\mathcal{A}} \cup <^{\mathcal{B}} \cup \{(a, b) : a \in A, b \in B\},$$

  that is, in $\mathcal{A} \triangleleft \mathcal{B}$ all elements of A precede all elements of B.
- We have:
  - $\mathcal{A} \times \mathcal{B} \cong \mathcal{B} \times \mathcal{A}$;
  - $\mathcal{A} \uplus \mathcal{B} \cong \mathcal{B} \uplus \mathcal{A}$;
  - However, in general, $\mathcal{A} \triangleleft \mathcal{B} \ncong \mathcal{B} \triangleleft \mathcal{A}$.

Operations on Multiple Structures

- The definition of product, union and ordered sum can easily be extended to more than two structures.
- For example, we set
$$\mathcal{A} \lhd \mathcal{B} \lhd \mathcal{C} := ((\mathcal{A} \lhd \mathcal{B}) \lhd \mathcal{C}).$$
- For a finite nonempty set I, we denote by
$$\mathcal{A}^I, \quad \dot{\bigcup_I} \mathcal{A} \quad \text{and} \quad \lhd_I \mathcal{A}$$
the product, the union and the ordered sum, respectively, of $\|I\|$ copies of $\mathcal{A}$, where $\|I\|$ denotes the cardinality of I.

Subsection 2

Syntax and Semantics of First-Order Logic

Symbols and Terms of First-Order Logic

- Fix a vocabulary τ.
- Each formula of first-order logic will be a string of symbols taken from the alphabet consisting of:
  - $v_1, v_2, v_3, \ldots$ (the **variables**);
  - $\neg, \vee$ (the **connectives not**, **or**);
  - $\exists$ (the **existential quantifier**);
  - $=$ (the **equality symbol**);
  -), (;
  - the symbols in τ.
- A **term** of vocabulary τ is a variable or a constant in τ.
- Henceforth, we shall often use the letters $x, y, z, \ldots$ for variables and $t, t_1, \ldots$ for terms.

Formulas of First-Order Logic

- The **formulas** of first-order logic of vocabulary τ are those strings which are obtained by finitely many applications of the following rules:
  - (F1) If t_0 and t_1 are terms, then $t_0 = t_1$ is a formula.
  - (F2) If R in τ is n-ary and $t_1, \ldots, t_n$ are terms, then $Rt_1 \ldots t_n$ is a formula.
  - (F3) If φ is a formula then $\neg\varphi$ is a formula.
  - (F4) If φ and ψ are formulas, then $(\varphi \vee \psi)$ is a formula.
  - (F5) If φ is a formula and x a variable, then $\exists x\varphi$ is a formula.
- $\mathrm{FO}[\tau]$ denotes the set of formulas of first-order logic of vocabulary τ.
- Formulas obtained by (F1) or (F2) are called **atomic formulas**.

Abbreviations and Conventions

- For formulas φ and ψ we use the following abbreviations:
  - $(\varphi \wedge \psi)$ for $\neg(\neg\varphi \vee \neg\psi)$;
  - $(\varphi \rightarrow \psi)$ for $(\neg\varphi \vee \psi)$;
  - $(\varphi \leftrightarrow \psi)$ for $((\neg\varphi \vee \psi) \wedge (\neg\psi \vee \varphi))$.
- We shall often omit parentheses in formulas if they are not essential such as the outermost parentheses in disjunctions $(\varphi \vee \psi)$.
- In examples, different letters $x, y, z, \ldots$ will always stand for different variables.

Axioms for Graphs and Orderings

- The axioms for graphs stated above have the following formalizations in $\mathrm{FO}[\{E\}]$:
  - $\forall x \neg Exx$;
  - $\forall x \forall y (Exy \rightarrow Eyx)$.
- The axioms for orderings have the following formalizations in $\mathrm{FO}[\{<\}]$:
  - $\forall x \neg x < x$;
  - $\forall x \forall y (x < y \vee y < x \vee x = y)$;
  - $\forall x \forall y \forall z ((x < y \wedge y < z) \rightarrow x < z)$.
- For orderings as $\{<, S, \min, \max\}$-structures we need in addition:
  - $\forall x \forall y (Sxy \leftrightarrow (x < y \wedge \forall z (x < z \rightarrow (y < z \vee y = z))))$;
  - $\forall x (\min < x \vee \min = x)$;
  - $\forall x (x < \max \vee x = \max)$.

Sentences, Free and Bound Variables

- Formulas in which every variable in an atomic subformula is in the scope of a corresponding quantifier are called **sentences**.
- Such occurrences of a variable are called **bound** occurrences.

  Example: The last occurrence of x in $(\forall x \neg Exx \wedge \exists y Exy)$ is not in the scope of a quantifier binding it.
- Such occurrences are called **free**.
- The notion of a **free variable** of a formula φ is made precise by the following definition by induction on (the length of) φ.
- The set $\text{free}(\varphi)$ of **free variables** of a formula φ is defined by:
  - If φ is atomic then the set $\text{free}(\varphi)$ of free variables of φ is the set of variables occurring in φ;
  - $\text{free}(\neg\varphi) := \text{free}(\varphi)$;
  - $\text{free}(\varphi \vee \psi) := \text{free}(\varphi) \cup \text{free}(\psi)$;
  - $\text{free}(\exists x \varphi) := \text{free}(\varphi) \backslash \{x\}$.

Notation for Formulas with Free Variables

- It is common practice to use the notation $\varphi(x_1, \ldots, x_n)$ to indicate that:
  - $x_1, \ldots, x_n$ are distinct;
  - $\text{free}(\varphi) \subseteq \{x_1, \ldots, x_n\}$ without implying that all x_i are actually free in φ.
- In some chapters we give another meaning to this notation.
- Often we abbreviate an n-tuple $x_1, \ldots, x_n$ of variables by $\overline{x}$.

  Example: Writing $\varphi(\overline{x})$ for $\varphi(x_1, \ldots, x_n)$.
- Usually we do not make explicit the length of $\overline{x}$ (here n), its size either being inessential or clear from the context.
- We often omit commas writing, for example, $\overline{x} = x_1 \ldots x_n$

Assignments

- Let $\mathcal{A}$ be a τ-structure.
- An **assignment** in $\mathcal{A}$ is a function α with domain the set of variables and with values in A,

$$\alpha : \{v_n : n \geq 1\} \to A.$$

- Think of α as assigning the meaning $\alpha(x)$ to the variable x.
- Extend α to a function defined for all terms by setting

$$\alpha(c) := c^{\mathcal{A}}, \quad \text{for } c \text{ in } \tau.$$

- Denote by $\alpha\frac{a}{x}$ the assignment that agrees with α except that $\alpha\frac{a}{x}(x) = a$.

Satisfaction

- We define the relation $\mathcal{A} \vDash \varphi[\alpha]$ ("the assignment α **satisfies** the formula φ **in** $\mathcal{A}$" or "φ is **true in** $\mathcal{A}$ **under** α") as follows:
  - $\mathcal{A} \vDash t_1 = t_2[\alpha]$ iff $\alpha(t_1) = \alpha(t_2)$;
  - $\mathcal{A} \vDash Rt_1 \dots t_n[\alpha]$ iff $R^A\alpha(t_1)\dots\alpha(t_n)$;
  - $\mathcal{A} \vDash \neg\varphi[\alpha]$ iff not $\mathcal{A} \vDash \varphi[\alpha]$;
  - $\mathcal{A} \vDash (\varphi \vee \psi)[\alpha]$ iff $\mathcal{A} \vDash \varphi[\alpha]$ or $\mathcal{A} \vDash \psi[\alpha]$;
  - $\mathcal{A} \vDash \exists x\varphi[\alpha]$ iff there is an $a \in A$, such that $\mathcal{A} \vDash \phi[\alpha\frac{a}{x}]$.

Satisfaction and Free Variables

- Note that the truth or falsity of $\mathcal{A} \vDash \varphi[\alpha]$ depends only on the values of α for those variables x which are free in φ.
- That is, if $\alpha_1(x) = \alpha_2(x)$, for all $x \in \text{free}(\varphi)$, then
$$\mathcal{A} \vDash \varphi[a_1] \quad \text{iff} \quad \mathcal{A} \vDash \varphi[\alpha_2].$$
- Thus, if $\varphi = \varphi(x_1, \ldots, x_n)$, and $a_1 = \alpha(x_1), \ldots, a_n = \alpha(x_n)$, then we may write
$$\mathcal{A} \vDash \varphi[a_1, \ldots, a_n]$$
for $\mathcal{A} \vDash \varphi[\alpha]$.

Models and Satisfiability

- If φ is a sentence, then the truth or falsity of $\mathcal{A} \vDash \varphi[\alpha]$ is completely independent of α.
- Thus, we may write
$$\mathcal{A} \vDash \varphi,$$
read $\mathcal{A}$ is a **model** of φ, or $\mathcal{A}$ **satisfies** φ, if, for some (hence every) assignment α, $\mathcal{A} \vDash \varphi[\alpha]$.
- For a set Φ of formulas, $\mathcal{A} \vDash \Phi[\alpha]$ means that $\mathcal{A} \vDash \varphi[\alpha]$, for all $\varphi \in \Phi$.
- Φ is **satisfiable** if, there exists a structure $\mathcal{A}$ and an assignment α in $\mathcal{A}$, such that
$$\mathcal{A} \vDash \Phi[\alpha].$$

Consequence, Validity and Equivalence

- A formula ψ is a **consequence** of Φ, written $\Phi \vDash \psi$, if
$$\mathcal{A} \vDash \Phi[\alpha] \quad \text{implies} \quad \mathcal{A} \vDash \psi[\alpha].$$
- The formula ψ is **logically valid**, written $\vDash \psi$, if $\varnothing \vDash \psi$.

  I.e., ψ is logically valid if ψ is true in all structures under all assignments.
- Formulas φ and ψ are **logically equivalent** if $\vDash \varphi \leftrightarrow \psi$.
- When only taking into consideration finite structures, we use the notations $\Phi \vDash_{\text{fin}} \psi$ and $\vDash_{\text{fin}} \psi$, and speak of **equivalent formulas**.
- Hence, φ and ψ are **equivalent** if $\vDash_{\text{fin}} \varphi \leftrightarrow \psi$.

  I.e., φ and ψ are equivalent if $\varphi \leftrightarrow \psi$ holds in all finite structures under all assignments.

Truth and Falsity

- At some places it will be convenient to assume that first-order logic contains two zero-ary relation symbols T, F.
- In every structure, T and F are interpreted as TRUE (i.e., as being true) and FALSE, respectively.
- Hence:
  - The atomic formula T is logically equivalent to $\exists x x = x$;
  - The atomic formula F is logically equivalent to $\neg \exists x x = x$.
- If $\Phi = \{\varphi_1, \ldots, \varphi_n\}$, we sometimes write:
  - $\bigwedge \Phi$ for $\varphi_1 \wedge \cdots \wedge \varphi_n$;
  - $\bigvee \Phi$ for $\varphi_1 \vee \cdots \vee \varphi_n$.
- In case $\Phi = \varnothing$, we set $\bigwedge \Phi = T$ and $\bigvee \Phi = F$.
- Then, for arbitrary finite Φ,

$$A \vDash \bigwedge \Phi \quad \text{iff} \quad \text{for all } \varphi \in \Phi,\ \mathcal{A} \vDash \varphi.$$

Quantifier Rank

- The **quantifier rank** $\mathrm{qr}(\varphi)$ of a formula φ is the maximum number of nested quantifiers occurring in it:
  - $\mathrm{qr}(\varphi) := 0$, if φ is atomic;
  - $\mathrm{qr}(\neg\varphi) := \mathrm{qr}(\varphi)$;
  - $\mathrm{qr}(\varphi \vee \psi) := \max\{\mathrm{qr}(\varphi), \mathrm{qr}(\psi)\}$;
  - $\mathrm{qr}(\exists x\varphi) := \mathrm{qr}(\varphi) + 1$.

Σ_n, Π_n and Δ_n Formulas

- It can be shown that every first-order formula is logically equivalent to a formula in **prenex normal form**, i.e., to a formula of the form

$$Q_1 x_1 \cdots Q_s x_s \psi,$$

  where $Q_1, \ldots, Q_s \in \{\forall, \exists\}$ and where ψ is quantifier-free.
- Such a formula is called a Σ_n **formula**, if the string of quantifiers consists of n consecutive blocks, where in each block all quantifiers are of the same type (i.e., all universal or all existential), adjacent blocks contain quantifiers of different type, and the first block is existential.
- Π_n **formulas** are defined in the same way, but now we require that the first block consists of universal quantifiers.
- A Δ_n-**formula** is a formula logically equivalent to both a Σ_n-formula and a Π_n-formula.

Permissible Substitutions

- If $\varphi(x_1, \ldots, x_n)$ is a formula and $t_1, \ldots, t_n$ are terms, then

$$\varphi\frac{t_1 \ldots t_n}{x_1 \ldots x_n} \text{ or, more simply, } \varphi(t_1, \ldots, t_n)$$

  denotes the result of simultaneously replacing all free occurrences of $x_1, \ldots, x_n$ by $t_1, \ldots, t_n$, respectively.
- This presupposes that none of the variables in $t_1, \ldots, t_n$ gets into the scope of a corresponding quantifier.
- If the latter happens, the bound variables in φ must be renamed in some canonical fashion before replacing.

Numerical Existential Abbreviations

- Given a formula $\varphi(x, \overline{z})$ and $n \geq 1$,

$$\exists^{\geq n} x \varphi(x, \overline{z})$$

  is an abbreviation for the formula

$$\exists x_1 \cdots \exists x_n \left(\bigwedge_{1 \leq i \leq n} \varphi(x_i, \overline{z}) \wedge \bigwedge_{1 \leq i < j \leq n} \neg x_i = x_j \right).$$

- It expresses that there are at least n elements x with $\varphi(x, \overline{z})$.
- We define similarly the abbreviations:
  - $\exists^{=n} x \varphi(x, \overline{z})$;
  - $\exists^{\leq n} x \varphi(x, \overline{z})$.

Cardinality Formulas

- We set
$$\begin{aligned} \varphi_{\geq n} &:= \exists^{\geq n} x x = x; \\ \varphi_{=n} &:= \exists^{=n} x x = x; \\ \varphi_{\leq n} &:= \exists^{\leq n} x x = x. \end{aligned}$$
- Clearly, we have:
  - $\mathcal{A} \vDash \varphi_{\geq n}$ iff $\|A\| \geq n$;
  - $\mathcal{A} \vDash \varphi_{=n}$ iff $\|A\| = n$;
  - $\mathcal{A} \vDash \varphi_{\leq n}$ iff $\|A\| \leq n$.

Subsection 3

Some Classical Results of First-Order Logic

Gödel's Completeness Theorem

- We assume given a finite system of formal rules for first-order logic.
- Let τ be a vocabulary, ψ a τ-formula and Φ a set of τ-formulas.
- A **formal proof of ψ from Φ** consists of a sequence of applications of the rules leading from the formulas in Φ to ψ.
- We say ψ is **formally provable from Φ** if there exists a formal proof of ψ from Φ.

Theorem (Gödel's Completeness Theorem)

Let τ be a vocabulary, ψ a τ-formula and Φ a set of τ-formulas. ψ is a consequence of Φ iff ψ is formally provable from Φ.

Consequences

Theorem

The set of logically valid sentences of first-order logic is recursively enumerable.

Theorem (Compactness Theorem)

(a) If ψ is a consequence of Φ, then ψ is already a consequence of a finite subset of Φ.

(b) If every finite subset of Φ is satisfiable, then Φ is satisfiable.

- The proof of the Completeness Theorem often leads to a proof of

Theorem (Löwenheim-Skolem Theorem)

If Φ has a model, then Φ has an at most countable model.

The Case of Finite Structures

- Neither Recursive Enumerability nor Compactness remain valid if one only considers finite structures.

  Example: Consider the set

  $$\Phi_\infty = \{\varphi_{\geq n} : n \geq 1\}.$$

  Each finite subset of Φ_∞ has a finite model.

  However, Φ_∞ has no finite model.

- The failure of recursive enumerability is documented by

Theorem (Trahtenbrot's Theorem)

The set of sentences of first-order logic valid in all finite structures is not recursively enumerable.

- A proof of this result will be presented later.

A Consequence of Compactness

Lemma

Let $\varphi \in \mathrm{FO}[\tau]$ and for $i \in I$, let $\Phi^i \subseteq \mathrm{FO}[\tau]$. Assume that $\vDash \varphi \leftrightarrow \bigvee_{i \in I} \bigwedge \Phi^i$. Then there is a finite $I_0 \subseteq I$ and, for every $i \in I_0$, a finite $\Phi_0^i \subseteq \Phi^i$ such that $\vDash \varphi \leftrightarrow \bigvee_{i \in I_0} \bigwedge \Phi_0^i$.

- For simplicity we assume that φ is a sentence and that every Φ^i is a set of sentences. By hypothesis, for $i \in I$ we have $\Phi^i \vDash \varphi$. Hence, by the Compactness Theorem, $\Phi_0^i \vDash \varphi$ for some finite $\Phi_0^i \subseteq \Phi^i$. Therefore, $\vDash \bigvee_{i \in I_0} \bigwedge \Phi_0^i \rightarrow \varphi$, for each finite subset $I_0 \subseteq I$.

  Suppose there is no such I_0, with $\vDash \varphi \rightarrow \bigvee_{i \in I_0} \bigwedge \Phi_0^i$.

  Then each finite subset of $\{\varphi\} \cup \{\neg \bigwedge \Phi_0^i : i \in I\}$ has a model.

  Hence, by the Compactness Theorem, there is a model of φ which, for all $i \in I$, satisfies $\neg \bigwedge \Phi_0^i$. But this contradicts the hypothesis $\vDash \varphi \leftrightarrow \bigvee_{i \in I} \bigwedge \Phi^i$.

Elementary Equivalence

- Structures $\mathcal{A}$ and $\mathcal{B}$ (of the same vocabulary) are said to be **elementarily equivalent**, written $A \equiv B$, if they satisfy the same first-order sentences.

Corollary

Let Φ be a set of first-order sentences. Assume that any two structures that satisfy the same sentences of Φ are elementarily equivalent.
Then any first-order sentence is equivalent to a boolean combination of sentences of Φ (that is, is equivalent to a sentence obtainable by closing Φ under $\neg$ and $\vee$).

- For any structure $\mathcal{A}$, set

$$\Phi(\mathcal{A}) := \{\psi : \psi \in \Phi, \mathcal{A} \vDash \psi\} \cup \{\neg\psi : \psi \in \Phi, \mathcal{A} \vDash \neg\psi\}.$$

Let φ be any first-order sentence. By the preceding lemma it suffices to show that $\vDash \varphi \leftrightarrow \bigvee_{\mathcal{A} \vDash \varphi} \bigwedge \Phi(\mathcal{A})$.

Elementary Equivalence (Cont'd)

- We must show $\vDash \varphi \leftrightarrow \bigvee_{\mathcal{A} \vDash \varphi} \bigwedge \Phi(\mathcal{A})$.

  Suppose $\mathcal{B} \vDash \varphi$.

  Then $\mathcal{B} \in \{\mathcal{A} : \mathcal{A} \vDash \varphi\}$ and $\mathcal{B} \vDash \Phi(\mathcal{B})$.

  Thus, $\mathcal{B} \vDash \bigvee_{\mathcal{A} \vDash \varphi} \bigwedge \Phi(\mathcal{A})$.

  For the converse, suppose $\mathcal{B} \vDash \bigvee_{\mathcal{A} \vDash \varphi} \bigwedge \Phi(\mathcal{A})$.

  Then, for some model $\mathcal{A}$ of φ, $\mathcal{B} \vDash \Phi(\mathcal{A})$.

  By definition of $\Phi(\mathcal{A})$, $\mathcal{A}$ and $\mathcal{B}$ satisfy the same sentences of Φ.

  Hence, by hypothesis, $\mathcal{A}$ and $\mathcal{B}$ are elementarily equivalent.

  Therefore, $\mathcal{B} \vDash \varphi$.

Subsection 4

Model Classes and Global Relations

Invariance of Class of Finite Models Under Isomorphisms

- Fix a vocabulary τ.
- Let φ be a sentence of $\mathrm{FO}[\tau]$.
- The class of *finite models* of φ is denoted by

$$\mathrm{Mod}(\varphi).$$

- If π is an isomorphism from $\mathcal{A}$ to $\mathcal{B}$, $\varphi(x_1,\ldots,x_n) \in \mathrm{FO}[\tau]$, and $a_1,\ldots,a_n \in A$, then an easy induction on formulas shows

$$\mathcal{A} \vDash \varphi[a_1,\ldots,a_n] \quad \text{iff} \quad \mathcal{B} \vDash \varphi[\pi(a_1),\ldots,\pi(a_n)].$$

- In particular, if φ is a sentence then $\mathcal{A} \vDash \varphi$ iff $\mathcal{B} \vDash \varphi$.
- Hence, $\mathrm{Mod}(\varphi)$ is closed under isomorphisms, i.e.,

$$\mathcal{A} \in \mathrm{Mod}(\varphi) \quad \text{and} \quad \mathcal{A} \cong \mathcal{B} \quad \text{imply} \quad \mathcal{B} \in \mathrm{Mod}(\varphi).$$

Definable Tuples in Structures

- For $\varphi(x_1,\dots,x_n) \in \mathrm{FO}[\tau]$ and a structure $\mathcal{A}$ let
$$\varphi^{\mathcal{A}}(-) := \{(a_1,\dots,a_n) : \mathcal{A} \vDash \varphi[a_1,\dots,a_n]\}$$
be the set of n-tuples **defined by** φ **in** $\mathcal{A}$.
- For $n = 0$ this should be read as
$$\varphi^{\mathcal{A}} := \begin{cases} \mathrm{TRUE}, & \text{if } \mathcal{A} \vDash \varphi, \\ \mathrm{FALSE}, & \text{if } \mathcal{A} \nvDash \varphi. \end{cases}$$
- Using this notation we can rewrite, for $\mathcal{A} \cong \mathcal{B}$, the equivalence $\mathcal{A} \vDash \varphi[a_1,\dots,a_n]$ iff $\mathcal{B} \vDash \varphi[\pi(a_1),\dots,\pi(a_n)]$ as
$$\text{if } \pi : \mathcal{A} \cong \mathcal{B}, \text{ then } \pi(\varphi^{\mathcal{A}}(-)) = \varphi^{\mathcal{B}}(-),$$
where for $X \subseteq A^n$, $\pi(X) := \{(\pi(a_1),\dots,\pi(a_n)) : (a_1,\dots,a_n) \in X\}$.

Convention of Classes of Structures

- We saw that only classes of structures closed under isomorphisms can be axiomatizable in FO$[\tau]$.
- Only such classes will be of interest.
- We agree that, throughout the slides, all classes K of structures considered will tacitly be assumed to be closed under isomorphisms, i.e.,

$$\mathcal{A} \in K \quad \text{and} \quad \mathcal{A} \cong \mathcal{B} \quad \text{imply} \quad \mathcal{B} \in K.$$

Global Relations

- We saw that properties expressible in logics correspond to so-called global relations.

Definition

Let K be a class of τ-structures. An n-ary **global relation Γ on K** is a mapping assigning to each $\mathcal{A} \in K$ an n-ary relation $\Gamma(\mathcal{A})$ on $\mathcal{A}$ satisfying

$$\Gamma(\mathcal{A})a_1 \ldots a_n \quad \text{iff} \quad \Gamma(\mathcal{B})\pi(a_1) \ldots \pi(a_n),$$

for every isomorphism $\pi : \mathcal{A} \cong \mathcal{B}$ and every $a_1, \ldots, a_n \in A$.
If K is the class of all finite τ-structures, then we just speak of an n-ary **global relation**.

Examples

(a) Any formula $\varphi(x_1, \ldots, x_n)$ of $\mathrm{FO}[\tau]$ defines the global relation

$$\mathcal{A} \mapsto \varphi^{\mathcal{A}}(-).$$

(b) The "transitive closure relation" TC is the binary global relation on GRAPH with

$$\mathrm{TC}(\mathcal{G}) := \{(a, b) : a, b \in G, \text{ there is a path from } a \text{ to } b\}.$$

(c) For $m \geq 0$, Γ_m is a unary global relation on GRAPH, where

$$\Gamma_m(\mathcal{G}) := \{a : \|\{b \in G : E^G ab\}\| = m\}$$

is the set of elements of $\mathcal{G}$ of degree m.

The Case $n = 0$

- In the definition of a global relation, we also allow the case $n = 0$.
- There are only two 0-ary relations on a structure, TRUE and FALSE.
- One identifies a 0-ary global relation Γ on K with the class

$$\{\mathcal{A} \in K : \Gamma(\mathcal{A}) = \text{TRUE}\}.$$

- By this identification, the global relation associated with a first-order sentence φ is the class $\text{Mod}(\varphi)$ of finite models of φ.

Accommodating Function Symbols

- Nearly all the methods and results we study can directly be extended to vocabularies containing function symbols.
- Moreover, by replacing functions by their graphs, one can always pass to function-free vocabularies.
- Consider a vocabulary τ.
- For every n-ary function symbol $f \in \tau$, we introduce a new $(n+1)$-ary relation symbol F .
- Let the vocabulary τ^r consist of the relation symbols and constants from τ together with the new relation symbols.
- Thus, τ^r is function-free.

Accommodating Function Symbols (Cont'd)

- For a τ-structure $\mathcal{A}$, let $\mathcal{A}^r$ be the τ^r-structure obtained from $\mathcal{A}$ by replacing every n-ary function f^A by its graph F^A,

$$F^A := \{(a_1, \ldots, a_n, f^A(a_1, \ldots, a_n)) : a_1, \ldots, a_n \in A\}.$$

Example: Consider a group $(G, \circ^G, e^G)$.

We can perceive it as an $\{R, e\}$-structure, where R is a ternary relation symbol.

Then, we look at the structure (G, R^G, e^G), where the ternary relation R is interpreted as the graph of $\circ^G$, i.e.,

$$R^G = \{(a, b, a \circ^G b) : a, b \in G\}.$$

Accommodating Function Symbols (Cont'd)

- The class of τ^r-structures of the form $\mathcal{A}^r$ is the class of models of the conjunction of the formulas

$$\forall x_1 \cdots \forall x_n \exists^{=1} y F x_1 \ldots x_n y, \quad f \in \tau.$$

- For every τ-sentence φ, there is a τ^r-sentence φ^r and, for every τ^r-sentence ψ, there is a τ-sentence ψ^{-r}, such that for every τ-structure $\mathcal{A}$, we have

$$\begin{array}{rcl} \mathcal{A} \models \varphi & \text{iff} & \mathcal{A}^r \models \varphi^r, \\ \mathcal{A} \models \psi^{-r} & \text{iff} & \mathcal{A}^r \models \psi. \end{array}$$

Accommodating Function Symbols (Cont'd)

- Example: Consider the sentence
$$\varphi := \exists x \forall y f(g(y)) = x.$$
Then
$$\varphi^r = \exists x \forall y \exists u (Gyu \wedge Fux).$$
Consider now the sentence
$$\psi := \forall x \exists y (Fxc \wedge \neg Gcy).$$
Then
$$\psi^{-r} = \forall x \exists y (f(x) = c \wedge \neg g(c) = y).$$
- Note that, in general, $\mathrm{qr}(\varphi^r) > \mathrm{qr}(\varphi)$.
- A class K of τ-structures is the class of models of a first-order sentence iff $K^r := \{\mathcal{A}^r : \mathcal{A} \in K\}$ is the class of models of a first-order sentence.

Subsection 5

Relational Databases and Query Languages

Relational Databases and Query Languages

- Suppose that a database contains:
  - The names of the main cities in the world;
  - The pairs (a, b) of such cities such that a given airline offers service from a to b without stopover.
- We can view the database as a first-order structure, namely as a digraph $\mathcal{G} = (G, E^G)$, where:
  - G is the set of cities;
  - $E^G ab$ means that there is a flight without stopover from a to b.
- First-order logic can be considered as a query language.

  Example: Let $\varphi(x, y) := Exy \vee \exists z(Exz \wedge Ezy)$.

  If φ is thought of as a query to the database, then the response is the set of pairs (a, b) of cities such that b can be reached from a with at most one stop.
- We obtain a global relation if we assign to any database (digraph) the response corresponding to the query φ.

Limitations of First-Order Logic

- First-order logic provides a rich class of database queries.
- But some plausible queries are not first-order expressible.

  Example: It is impossible to express the query

  "Can one fly from x to y?"

  by a first-order formula such that we get the right answer in all possible databases (digraphs).
- As a consequence, we are led to study stronger logics (or, query languages).