

## Finite Model Theory

*Qing Wang*

qing.wang@anu.edu.au

Logic and Computation  
Research School of Computer Science

## Finite Model Theory

- Classical model theory concentrates on **all structures** - the origin is in mathematics.
  - Boolean algebras, random graphs, algebraically closed fields, various models of arithmetic, etc.
- Finite model theory studies logics over **finite structures** - the origin is in computer science.
  - Finite relations
  - Finite graphs
  - Finite strings
  - Finite classes of arithmetic structures
  - ...

## Overview of Topics

- **Part 1: – Introduction**
  - 1 What is finite model theory?
  - 2 Connections to some areas in CS
    - Database theory
    - Complexity theory
  - 3 Basic definitions and terminology
  - 4 Inexpressibility proofs
  - 5 Classical results over finite structures
    - Failures
    - Successes
    - Open questions

## Connections to Database Theory

- Finite model theory plays a central role in the development of database theory.

- A database can be naturally viewed as a finite structure, e.g.,

Database	Structure
Relational databases	finite relations
XML databases	finite trees
Graph databases	finite graphs

- A query language is often measured in terms of logic, e.g.,

Query language	Logic
Relational Calculus	FO
Datalog $\neg$	$\exists$ LFP
Basic SQL + aggregation	FO + counting extension
Core XPath	MSO

- Database theory supplies finite model theory with key motivations and problems.

## Connections to Database Theory

### ● Example: Reachability queries

- **Q1.** Find pairs of cities  $(s, d)$  such that one can fly from  $s$  to  $d$  with at most one stop.
- **Q2.** Find pairs of cities  $(s, d)$  such that one can fly from  $s$  to  $d$  with at most two stops.
- **Q3.** Find pairs of cities  $(s, d)$  such that one can fly from  $s$  to  $d$ .

FLIGHTS	
Source	Destination
Berlin	Beijing
Beijing	Auckland
Auckland	Sydney

## Connections to Database Theory

### ● Example: Reachability queries

- **Q1.** Find pairs of cities  $(s, d)$  such that one can fly from  $s$  to  $d$  with at most one stop.

$$\{(x_s, x_d) \mid \text{FLIGHTS}(x_s, x_d) \vee \exists x_1. (\text{FLIGHTS}(x_s, x_1) \wedge \text{FLIGHTS}(x_1, x_d))\}$$

FLIGHTS	
Source	Destination
Berlin	Beijing
Beijing	Auckland
Auckland	Sydney

ANSWER	
Berlin	Beijing
Beijing	Auckland
Auckland	Sydney
Berlin	Auckland
Beijing	Sydney

## Connections to Database Theory

### ● Example: Reachability queries

- **Q2.** Find pairs of cities  $(s, d)$  such that one can fly from  $s$  to  $d$  with at most two stops.

$$\{(x_s, x_d) \mid \text{FLIGHTS}(x_s, x_d) \vee \exists x_1. (\text{FLIGHTS}(x_s, x_1) \wedge \text{FLIGHTS}(x_1, x_d)) \vee \exists x_1, x_2. (\text{FLIGHTS}(x_s, x_1) \wedge \text{FLIGHTS}(x_1, x_2) \wedge \text{FLIGHTS}(x_2, x_d))\}$$

FLIGHTS	
Source	Destination
Berlin	Beijing
Beijing	Auckland
Auckland	Sydney

ANSWER	
Berlin	Beijing
Beijing	Auckland
Auckland	Sydney
Berlin	Auckland
Beijing	Sydney
Berlin	Sydney

## Connections to Database Theory

### ● Example: Reachability queries

- **Q3.** Find pairs of cities  $(s, d)$  such that one can fly from  $s$  to  $d$ .

## Connections to Database Theory

- **Example: Reachability queries**

- **Q3.** Find pairs of cities  $(s, d)$  such that one can fly from  $s$  to  $d$ .

$$\{(x_s, x_d) \mid \text{FLIGHTS}(x_s, x_d) \vee \exists x_1. (\text{FLIGHTS}(x_s, x_1) \wedge \text{FLIGHTS}(x_1, x_d)) \vee \exists x_1, x_2. (\text{FLIGHTS}(x_s, x_1) \wedge \text{FLIGHTS}(x_1, x_2) \wedge \text{FLIGHTS}(x_2, x_d)) \vee \exists x_1, x_2, x_3. (\text{FLIGHTS}(x_s, x_1) \wedge \dots \wedge \text{FLIGHTS}(x_3, x_d)) \vee \dots\}$$

## Connections to Complexity Theory

- Computational complexity measures the amount of resources (e.g., time and space) that are needed to solve a problem.

- Many models of computation have been invented, e.g.,

Models of Computation	
Lambda calculus	Church
Recursive functions	Gödel
Turing machines	Turing

- Two key questions:

- What can be automatically computed (i.e., computability)?  
 ↪ Turing/Church thesis
- How difficult it is to solve a problem (i.e., complexity)?  
 ↪ Complexity classes (P, NP, Pspace, ...)

## Connections to Database Theory

- **Example: Reachability queries**

- **Q3.** Find pairs of cities  $(s, d)$  such that one can fly from  $s$  to  $d$ .

$$\{(x_s, x_d) \mid \text{FLIGHTS}(x_s, x_d) \vee \exists x_1. (\text{FLIGHTS}(x_s, x_1) \wedge \text{FLIGHTS}(x_1, x_d)) \vee \exists x_1, x_2. (\text{FLIGHTS}(x_s, x_1) \wedge \text{FLIGHTS}(x_1, x_2) \wedge \text{FLIGHTS}(x_2, x_d)) \vee \exists x_1, x_2, x_3. (\text{FLIGHTS}(x_s, x_1) \wedge \dots \wedge \text{FLIGHTS}(x_3, x_d)) \vee \dots\}$$

**Cannot be expressed in Relational Calculus (FO)**

- How can we tell whether a query CAN or CANNOT be expressed in a query language?

## Connections to Complexity Theory

- The development of descriptive complexity is one of the most striking results in finite model theory.

- **How are different logics and complexity classes related?**

- 1 What logic can be used to express a query?
- 2 What is the complexity of evaluating a query?

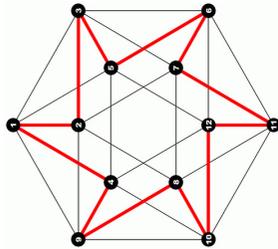
- Some known results:

Query	Logic	Complexity class
Transitive closure	IFP + <	P
Connectivity	IFP + <	P
Evenness	IFP + <	P
Hamiltonicity	$\exists$ SO	NP
3-colorability	$\exists$ SO	NP
Clique	$\exists$ SO	NP
Quantified SAT	PFP + <	Pspace

## Connections to Complexity Theory

### ● Example: **Hamiltonicity**

- A **Hamiltonian cycle** is a cycle that visits each vertex exactly once. A graph that contains a Hamiltonian cycle is called a **Hamiltonian graph**.



## Connections to Complexity Theory

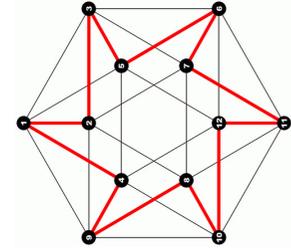
### ● Example: **Hamiltonicity**

- A **Hamiltonian cycle** is a cycle that visits each vertex exactly once. A graph that contains a Hamiltonian cycle is called a **Hamiltonian graph**.

$\exists L \exists S$  linear order( $L$ )  $\wedge$   
 $S$  is the successor relation of  $L \wedge$   
 $\forall x \exists y (L(x, y) \vee L(y, x)) \wedge$   
 $\forall x \forall y (S(x, y) \Rightarrow E(x, y))$

$L$  is a linear ordering relation.

$S$  is a circular successor relation:  
 $\forall x \forall y S(x, y) \Leftrightarrow$   
 $((L(x, y) \wedge \neg \exists z (L(x, z) \wedge L(z, y))) \wedge$   
 $(\neg \exists z L(x, z) \wedge \neg \exists z L(z, y)))$



## Connections to Complexity Theory

### ● Example: **Hamiltonicity**

- Hamiltonicity can be specified in *existential second-order logic* ( $\exists SO$ ).

$\exists L \exists S$  linear order( $L$ )  $\wedge$   
 $S$  is the successor relation of  $L \wedge$   
 $\forall x \exists y (L(x, y) \vee L(y, x)) \wedge$   
 $\forall x \forall y (S(x, y) \Rightarrow E(x, y))$

- Testing Hamiltonicity is an NP-complete problem.
- So, is there any connection between  $\exists SO$  and NP?

## Connections to Complexity Theory

### ● Example: **Hamiltonicity**

- Hamiltonicity can be specified in *existential second-order logic* ( $\exists SO$ ).

$\exists L \exists S$  linear order( $L$ )  $\wedge$   
 $S$  is the successor relation of  $L \wedge$   
 $\forall x \exists y (L(x, y) \vee L(y, x)) \wedge$   
 $\forall x \forall y (S(x, y) \Rightarrow E(x, y))$

- Testing Hamiltonicity is an NP-complete problem.
- So, is there any connection between  $\exists SO$  and NP?

$\exists SO$	Existential SO quantifiers	+	FO formula
NP	Guess stage	+	Verify stage

- **Descriptive complexity** aims to characterize complexity classes by means of logics.

- Apparently, finite structures are a subclass of “all structures”. Is finite model theory just a subfield of classical model theory?
- Some history:
  - Before 1970s, some problems about FO over finite structures were studied.
  - In 1970s,
    - Fagin published several papers relating to finite model theory.
    - Hájek proposed to “develop logic (classical and generalized) modified by allowing only finite models”.
    - ...
  - Since 1980s, finite model theory becomes an active line of research.

- A **vocabulary**  $\sigma$  is a *finite* set of relation symbols, each with a fixed arity.  
Note that, we restrict vocabularies to be relational, and there are *no function symbols* in  $\sigma$ .
- A **structure** (also called a **model**) of vocabulary  $\sigma$  is  $\mathfrak{A} = \langle A, (R^{\mathfrak{A}})_{R \in \sigma} \rangle$ , where
  - $A$ , called the **universe** of  $\mathfrak{A}$ , is a nonempty set, and
  - each  $R^{\mathfrak{A}} \subseteq A^n$  is an interpretation of a  $n$ -ary relation symbol from  $\sigma$ .
- A structure  $\mathfrak{A}$  is called **finite** if its universe  $A$  is a finite set.
- **Conventions:**
  - We denote universes by using Roman letters corresponding to their structures, e.g., the universe of  $\mathfrak{A}$  is  $A$ , the universe of  $\mathfrak{B}$  is  $B$ , etc.
  - We use the same symbol  $R$  for both a relation symbol in  $\sigma$ , and its interpretation  $R^{\mathfrak{A}}$ .

- **Logic and expressiveness**
  - Ehrenfeucht-Fraïssé games
  - Locality
- **Logic and complexity**
  - Descriptive complexity
  - Data complexity and expression complexity
- **Logic and combinatorics**
  - Zero-one laws
  - Asymptotic probabilities
- **Various logics**
  - Finite variable logics
  - Fixed point logics
  - Logics with counting
  - ...

- Recall terms and formulas of first-order logic (FO) (we assume a countably infinite set of variables).
- **Syntax:**
  - **Terms** of FO are defined by:
    - Each variable  $x$  and constant  $c$  is a term.
    - $f(t_1, \dots, t_n)$  is a term, where  $f$  is a relation symbol, and  $t_1, \dots, t_n$  are terms.
  - **Formulas** of FO can be inductively defined by:
    - atomic formulas:  $R(t_1, \dots, t_n)$ ,  $t_1 = t_2$ ;
    - Boolean operations:  $\varphi_1 \wedge \varphi_2$ ,  $\varphi_1 \vee \varphi_2$ ,  $\neg\varphi$ ;
    - first-order quantifiers:  $\exists\varphi$ ,  $\forall\varphi$ .
- **Semantics:** we skip the details as it has been covered in the last week

## First-order Logic

- A **sentence** is a formula without free variables.
- A sentence  $\varphi$  is **satisfiable** if it has a model, and is **valid** if it is true in every structure.
  - $\varphi$  is not valid iff  $\neg\varphi$  is satisfiable.
- A sentence  $\varphi$  is **finitely satisfiable** if it has a finite model, and is **finitely valid** if it is true in every finite structure.
  - $\varphi$  is not finitely valid iff  $\neg\varphi$  is finitely satisfiable.
- We use  $\mathfrak{A} \models \varphi(a_1, \dots, a_n)$  to denote that  $\varphi(a_1, \dots, a_n)$  is true in  $\mathfrak{A}$ .

## Queries Definable in FO

- Consider  $\sigma = \{E\}$  and  $G = \langle V, E \rangle$ .
  - 1 Graphs whose edges are antireflexive and symmetric:  
$$\forall x \neg E(x, x) \wedge \forall x \forall y (E(x, y) \Rightarrow E(y, x)).$$
  - 2 Graphs that contain at least one triangle:  
$$\exists x \exists y \exists z (x \neq y \wedge y \neq z \wedge x \neq z \wedge E(x, y) \wedge E(x, z) \wedge E(y, z)).$$
  - 3 Graphs that contain at least  $n$  vertices:  
$$\exists x_1 \dots \exists x_n \bigwedge_{1 \leq i < j \leq n} x_i \neq x_j.$$

## Queries

- A formula  $\varphi(x_1, \dots, x_n)$  with free variables  $x_1, \dots, x_n$  defines a **mapping**  $Q$  that associates to every structure  $\mathfrak{A}$  a ( $n$ -ary) relation on  $\mathfrak{A}$ :  
$$Q(\mathfrak{A}) = \{(a_1, \dots, a_n) \mid \mathfrak{A} \models \varphi(a_1, \dots, a_n)\}.$$
- A  **$n$ -ary query**  $Q$  is such a mapping **closed under isomorphism**, i.e., if  $h: A \rightarrow B$  is an isomorphism between  $\mathfrak{A}$  and  $\mathfrak{B}$ , it is also an isomorphism between  $Q(\mathfrak{A})$  and  $Q(\mathfrak{B})$ .
- If  $n = 0$ , a 0-ary query is a mapping from structures to  $\{true, false\}$ , which is called a **Boolean query**.
- A query  $Q$  is **definable** in a logic  $L$  if there is a formula  $\varphi(x_1, \dots, x_n)$  of  $L$  that defines  $Q$ . A Boolean query  $Q$  is **definable** in a logic  $L$  if there is a formula  $\varphi$  of  $L$  such that  $Q(\mathfrak{A}) = true$  iff  $\mathfrak{A} \models \varphi$  for all  $\mathfrak{A}$ .

## Limitations of FO

- The expressive power of FO on finite structures is limited:
  - Cannot express counting properties, e.g.,
    - 1 **Evenness**: Given a graph  $G$ , is the number of vertices in  $G$  even?  
$$even(V) = \begin{cases} 1 & \text{if } |V| \text{ is even} \\ 0 & \text{otherwise.} \end{cases}$$
  - Cannot express properties that require iterative algorithms, e.g.,
    - 1 **Connectivity**: Given a graph  $G$ , is it connected?  
i.e., there exists a path between any two nodes  $a$  and  $b$  in  $G$ .
  - Cannot express properties that require to quantify relations, e.g.,
    - 1 **3-Colorability**
    - 2 **Clique**
    - 3 ...

## Inexpressibility Proofs

- How can one prove that a property is inexpressible in a logic, e.g., FO?
- Some techniques are available for inexpressibility proofs in FO.
  - Compactness theorem
  - Ehrenfeucht-Fraïssé games
  - ...

## Inexpressibility Proofs

- **Compactness theorem**

Let  $\Phi$  be a set of first-order sentences. If every finite subset of  $\Phi$  is satisfiable, then  $\Phi$  is satisfiable.
- Main ideas of using the compactness theorem to prove inexpressibility of a property  $P$ :
  - Assume  $P$  is expressible by a FO-sentence  $\varphi$ .
  - Construct a set of sentences  $\Psi$  so that each model of  $\Psi$  does not satisfy  $P$ , but each finite subset of  $\Psi$  has a model satisfying  $P$ .
  - By compactness we would know that  $\Psi \cup \{\varphi\}$  has a model.
  - Contradiction with the assumption.

## Inexpressibility Proofs

- **Connectivity** of arbitrary graphs is not FO-definable.

### Proof:

- Assume that connectivity is definable by  $\phi$ .
- Expand the vocabulary with two constants  $c_1$  and  $c_2$ , and let  $T = \{\psi_n | n > 0\} \cup \{\phi\}$ , where

$$\psi_n = \neg(\exists x_1 \dots \exists x_n (x_1 = c_1 \wedge x_n = c_2 \wedge \bigwedge_{1 \leq i \leq n-1} E(x_i, x_{i+1})))$$

i.e., there is no path of length  $n+1$  from  $c_1$  to  $c_2$ .

- Every finite subset  $T' \subseteq T$  is satisfiable, because there exists  $N$ , s.t. for all  $\psi_n \in T'$ ,  $n < N$ , and  $T'$  has a model with a path of length  $N+1$ . By compactness,  $T$  is satisfiable.
- However,  $T$  has no model. *Contradiction.*

## Inexpressibility Proofs

- Does the previous proof tell us that FO cannot express connectivity over finite graphs?

### The previous proof:

- Assume that connectivity is definable by  $\phi$ .
- Expand the vocabulary with two constants  $c_1$  and  $c_2$ , and let  $T = \{\psi_n | n > 0\} \cup \{\phi\}$ , where

$$\psi_n = \neg(\exists x_1 \dots \exists x_n (x_1 = c_1 \wedge x_n = c_2 \wedge \bigwedge_{1 \leq i \leq n-1} E(x_i, x_{i+1})))$$

i.e., there is no path of length  $n+1$  from  $c_1$  to  $c_2$ .

- Every finite subset  $T' \subseteq T$  is satisfiable, because there exists  $N$ , s.t. for all  $\psi_n \in T'$ ,  $n < N$ , and  $T'$  has a model with a path of length  $N+1$ . By compactness,  $T$  is satisfiable.
- However,  $T$  has no model. *Contradiction.*

## Failure of Compactness Theorem

- Does the previous proof tell us that FO cannot express connectivity over finite graphs?
- To modify the previous proof for finite models, one would be to use compactness over finite models.
- But compactness fails over finite models.

**Proposition:** There is a set  $T$  of FO-sentences s.t.

- 1  $T$  has no finite models, and
- 2 every finite subset of  $T$  has a finite model.

## Inexpressibility Proofs

- For the techniques available for inexpressibility proofs in FO:
  - **Compactness theorem**  
↔ fails over finite structures.
  - **Ehrenfeucht-Fraïssé games**  
↔ used as a central tool on classes of finite structures.

## Failure of Compactness Theorem

- **Proposition:** There is a set  $T$  of FO-sentences s.t.
  - 1  $T$  has no finite models, and
  - 2 every finite subset of  $T$  has a finite model.

**Proof:**

- Assume that  $\sigma = \emptyset$ , and define

$$\psi_n = \exists x_1 \dots \exists x_n \bigwedge_{i \neq j} \neg(x_i = x_j).$$

i.e., the universe has at least  $n$  distinct elements.

- Let  $T = \{\psi_n \mid n > 0\}$ .
- $T$  has no finite model. But for each finite subset of  $T$ , a set whose cardinality exceeds the maximal  $n$  is a model.

## Classical Model Theory

- Main topics:
  - Logical definability of structural properties
  - Classification of models of theories
  - Algebraic properties in axiomatic theories
- Some key results:
  - 1 Completeness theorem (and Compactness theorem)
  - 2 Löwenheim-Skolem theorem
  - 3 Beth's definability theorem
  - 4 Craig's interpolation theorem
  - 5 Los-Tarski preservation theorem and Lyndon's positivity theorem

## Classical Model Theory

- Some key results:
  - 1 Completeness theorem (and Compactness theorem)
  - 2 Löwenheim-Skolem theorem
  - 3 Beth's definability theorem
  - 4 Craig's interpolation theorem
  - 5 Los-Tarski preservation theorem and Lyndon's positivity theorem
- *Can these results of classical model theory hold over finite structures?*
- Unfortunately, **all the above results fail when we restrict to finite structures.**

## Failure of Completeness Theorem

- Some consequences of Trakhtenbrot's Theorem:
  - Unsatisfiability of FO is semi-decidable, but finite unsatisfiability is not semi-decidable.
  - Satisfiability of FO is not semi-decidable, but finite satisfiability is semi-decidable.

## Failure of Completeness Theorem

### • Trakhtenbrot's Theorem<sup>1</sup>

For every relational vocabulary  $\sigma$  with at least one binary relation symbol, it is undecidable whether a sentence  $\Phi$  of  $\sigma$  is finitely satisfiable.

Basic idea:

- For any Turing machine  $M$ , there is a FO sentence  $\varphi_M$  such that  $M$  halts iff  $\varphi_M$  has a finite model.

### • Corollary

For every relational vocabulary  $\sigma$  with at least one binary relation symbol, **the set of finitely valid FO sentences is not recursively enumerable.**

Basic idea:

- Because the set of non-halting Turing machines is not recursively enumerable,  $\{\varphi \mid \neg\varphi \text{ has no finite models}\}$  is also not recursively enumerable.

<sup>1</sup>B. Trakhtenbrot, The Impossibility of an Algorithm for the Decidability Problem on Finite Classes. Proceedings of the USSR Academy of Sciences (in Russian) 70 (4): 569572, 1950.

## Failure of Löwenheim-Skolem Theorem

### • Corollary

There is no recursive function  $f$  s.t. if a FO sentence  $\varphi$  has a finite model, then it has a model of size at most  $f(|\varphi|)$ .

- Recall the Löwenheim-Skolem theorem in classical model theory:

If a countable first-order theory has an infinite model, then for every infinite cardinal number  $k$  it has a model of size  $k$ .

## Trakhtenbrot Theorem

- **Theorem** (Trakhtenbrot, 1950)

For every relational vocabulary  $\sigma$  with at least one binary relation symbol, it is undecidable whether a sentence  $\Phi$  of  $\sigma$  is finitely satisfiable.

- **Proof:**

For every Turing machine  $M = (S, \Sigma, \Delta, \delta, q_0, S_a, S_r)$ , construct a sentence  $\varphi_M$  of  $\sigma$  s.t.  $\varphi_M$  **is finitely satisfiable iff  $M$  halts on the empty input.**

- Let  $\sigma = \{<, min, T_0, T_1, (H_q)_{q \in S}\}$ , and  $\varphi_M$  states that each relation symbol in  $\sigma$  is interpreted below, and  $M$  eventually halts.
  - $<$  is a linear order, and  $min$  is the minimal element w.r.t.  $<$ ;
  - $T_0(p, t)$  and  $T_1(p, t)$  are *tape predicates*: indicate that position  $p$  at time  $t$  contains 0 or 1;
  - $H_q(p, t)$ 's are *head predicates*: indicate that at time  $t$ ,  $M$  is in state  $q$  and its head is in position  $p$ .
- If  $\varphi_M$  has a finite model, then such a model represents a computation of  $M$  that halts on an empty input, and vice versa.

## Classical Result – Revisited for Finite Models

- Are there any results of classical model theory that survive on finite models?
- Rosen wrote:<sup>2</sup>

*“there seems to be no example of a theorem [of classical model theory] that remains true when relativized to finite structures but for which there are entirely different proofs for the two cases. It would be interesting to find a theorem proved using the compactness theorem that can be established using a new method over finite structures.”*

<sup>2</sup>Rosen, E. Some aspects of model theory and finite structures. Bull. Symbolic Logic 8, 380-403, 2002.

## Classical Result – Revisited for Finite Models

- A FO sentence  $\varphi(\vec{x})$  is **preserved under homomorphisms** if  $\mathfrak{A} \models \varphi(\vec{a})$  implies  $\mathfrak{B} \models \varphi(h(\vec{a}))$  whenever  $h : \mathfrak{A} \rightarrow \mathfrak{B}$  is a homomorphism.
- Recall the homomorphism preservation theorem in classical model theory:

The following statements are equivalent for any FO sentence  $\varphi$ :

- 1  $\varphi$  is preserved under homomorphisms on all structures;
  - 2  $\varphi \equiv \varphi^*$  on all structures for some existential positive FO-sentence  $\varphi^*$ .
- **Theorem** (Rossman, 2005)  
If a FO sentence  $\varphi$  is preserved under homomorphisms on all finite structures, then there is an existential positive FO-sentence  $\varphi^*$  that is equivalent to  $\varphi$  on all finite structures.

## Classical Result – Revisited for Finite Models

- First-order logic has a special place in classical model theory.
- **Theorem** (Lindström, 1969)  
First-order logic is a maximal logic possessing both the Compactness Theorem and the Löwenheim-Skolem Theorem, i.e., no logic that is compact and satisfies the Löwenheim-Skolem property can properly extend FO.
- **Question**
  - Is there a similar characterization of first-order logic/least fixed-point logic on finite structures?

## Useful References

- ① L. Libkin. Elements of Finite Model Theory. Springer, 2004.
- ② M. Grohe. Descriptive Complexity, Canonisation, and Definable Graph Structure Theory, 2011.
- ③ H.-D. Ebbinghaus and J. Flum. Finite Model Theory. Springer, 2005.
- ④ N. Immerman. Descriptive Complexity. Springer, 1999.
- ⑤ M. Otto. Finite Model Theory, available at <http://www.mathematik.tu-darmstadt.de/otto/LEHRE/FMT0809.ps>
- ⑥ Abiteboul, Hull, Vianu. Foundations of Databases. 1995
- ⑦ Grädel et al. Finite Model Theory and Its Applications. 2007