Foundations of Data and Knowledge Bases

First-Order Logics for Databases

Joachim Niehren

Links: Linking Dynamic Data
Inria Lille

September 7, 2022

Outline

1. Databases as relational structures
2. FO logics for relational structures
   - FO-Queries
   - Query answering
3. FO logics for XML databases
   - Unranked trees
   - FO logic for unranked trees

Other Classes of Structures

Relational Structures

- trees (binary, unranked, rooted, ordered, unordered, ...)
- undirected graphs

Structures with Functions and Relations

- groups
- vector spaces
- integers
- reals

Outline

1. Databases as relational structures

2. FO logics for relational structures
   - FO-Queries
   - Query answering

3. FO logics for XML databases
   - Unranked trees
   - FO logic for unranked trees

FO Logic for Relational Structures

Parameters

- relational signature $\Sigma = \mathrm{Consts} \uplus \mathrm{Rels}$ (for syntax)
- class C of Σ-structures (for semantics)

Fixed

infinite set of variables $\mathrm{Vars} = \mathbb{N}$ ranged over by x, y, z

Syntax of FO (where $a \in \mathrm{Consts}$, $r \in \mathrm{Rels}$):

$$\begin{array}{ll} \text{terms} & t ::= a \mid x \\ \text{formulas} & \phi ::= r(t_1, \ldots, t_{\mathrm{ar}(r)}) \mid \phi \wedge \phi' \mid \neg\phi \mid \exists x.\phi \end{array}$$

Syntax of FO with Equality (where $a \in \mathrm{Consts}$, $r \in \mathrm{Rels}$):

$$\text{formulas} \quad \phi ::= r(t_1, \ldots, t_{\mathrm{ar}(r)}) \mid \phi \wedge \phi' \mid \neg\phi \mid \exists x.\phi \mid t_1 = t_2$$

Shortcuts for FO Formulas

for all FO formulas ϕ and ϕ' define new FO formulas:

$$\phi \vee \phi' \;=_{df}\; \neg(\neg\phi \wedge \neg\phi')$$
$$\phi \rightarrow \phi' \;=_{df}\; \neg\phi \vee \phi'$$
$$\forall x.\phi \;=_{df}\; \neg\exists x.\neg\phi$$

Definition of Semantics

Types

$eval^{S,\alpha}(t) \subseteq \mathrm{Dom}(S)$ where $S \in C$ structure

$eval^{S,\alpha}(\phi) \subseteq \mathbb{B}$ $\alpha : \mathrm{Vars} \to \mathrm{Dom}(S)$ variable assignment

Values

$$eval^{S,\alpha}(x) = \alpha(x)$$

$$eval^{S,\alpha}(a) = a^S$$

$$eval^{S,\alpha}(r(t_1,\ldots,t_n)) = \begin{cases} 1 & \text{if } (eval^{S,\alpha}(t_1),\ldots eval^{S,\alpha}(t_n)) \in r^S \\ 0 & \text{else} \end{cases}$$

$$eval^{S,\alpha}(\phi \wedge \phi') = eval^{S,\alpha}(\phi) \wedge^{\mathbb{B}} eval^{S,\alpha}(\phi')$$

$$eval^{S,\alpha}(\neg\phi) = \neg^{\mathbb{B}}(eval^{S,\alpha}(\phi))$$

$$eval^{S,\alpha}(\exists x.\phi) = \begin{cases} 1 & \text{if } eval^{S,\alpha[d/x]}(\phi) = 1 \text{ for some } d \in \mathrm{Dom}(S) \\ 0 & \text{else} \end{cases}$$

where

$$\neg^{\mathbb{B}}(1) = 0, \quad \neg^{\mathbb{B}}(0) = 1, \quad \wedge^{\mathbb{B}}(1,1) = 1, \quad \alpha[d/x](y) = \begin{cases} d & \text{if } y = x \\ \alpha(y) & \text{else} \end{cases}$$

Excercise [FO Equivalences]

Two FO formulas ϕ and ϕ' are called equivalent if $\mathit{eval}^{S,\alpha}(\phi) = \mathit{eval}^{S,\alpha}(\phi')$ for all relational structures S and all variable assignements $\alpha : \mathrm{EleVars} \to \mathit{nod}(S)$.

1. Show that any formula $\phi_1 \to (\phi_2 \to \phi_3)$ is equivalent to $(\phi_1 \wedge \phi_2) \to \phi_3$.

How to Define n-ary Queries

How to define queries a class C of Σ-structures S?
Use FO formulas ϕ with n-free variables $x_1, \ldots, x_n \in \mathrm{Vars}$:

$$Q_{\phi(x_1,\ldots,x_n)}(S) = \{(\alpha(x_1), \ldots, \alpha(x_n)) \mid eval^{S,\alpha}(\phi) = 1\}$$

All other variables in ϕ are implicitely existentially quantified

Movie Database

Movies	*Title*	*Director*	*Actor*
	The Trouble with Harry	Hitchcock	Gwenn
	The Trouble with Harry	Hitchcock	Forsythe
	The Trouble with Harry	Hitchcock	MacLaine
	The Trouble with Harry	Hitchcock	Hitchcock
	...	...	...
	Cries and Whispers	Bergman	Andersson
	Cries and Whispers	Bergman	Sylwan
	Cries and Whispers	Bergman	Thulin
	Cries and Whispers	Bergman	Ullman

Location	*Theater*	*Address*	*Phone Number*
	Gaumont Opéra	31 bd. des Italiens	47 42 60 33
	Saint André des Arts	30 rue Saint André des Arts	43 26 48 18
	Le Champo	51 rue des Ecoles	43 54 51 60
	...	...	...
	Georges V	144 av. des Champs-Elysées	45 62 41 46
	Les 7 Montparnassiens	98 bd. du Montparnasse	43 20 32 20

Pariscope	*Theater*	*Title*	*Schedule*
	Gaumont Opéra	Cries and Whispers	20:30
	Saint André des Arts	The Trouble with Harry	20:15
	Georges V	Cries and Whispers	22:15
	...	...	...

Exercises

2. Define an FO query on the movie database that selects all the other theaters that play all films of Bergman played at Georges V.

Classes of FO Queries

Conjunctive Queries are Defined by Conjunctive Formulas

$$\phi ::= r(t_1, \ldots, t_n) \mid \phi \wedge \phi'$$

Positive Propositional Queries are Defined by such Formulas

$$\phi ::= r(t_1, \ldots, t_n) \mid \phi \wedge \phi' \mid \phi \vee \phi'$$

recall that existential quantifiers are implicit in query definitions.

Query Answering Problems

Parameters

relational signature Σ and class C of Σ-structures

Enumeration problem

input: query definition $\phi(x_1, \ldots, x_n)$, structure S of class C
output: answer set $Q_{\phi(x_1,\ldots,x_n)}(S)$

Decision problem

input: query definition $\phi(x_1, \ldots, x_n)$, structure S of class C
output: answer set $Q_{\phi(x_1,\ldots,x_n)}(S) \neq \emptyset$

Hardness for Simple Structures

A single structure $\mathbb{B}$ with domain $\{0, 1\}$ and constants 1 and 0 and relation $=^{\mathbb{B}}$.

- Answering positive propositional queries was an exercise. How difficult is it?
- Answering FO-queries over $\mathbb{B}$ is PSpace-complete.

Which cases are easy?

- acyclic conjunctive queries (Yannakakis 81)
- FO-queries by formulas with a bounded number of reusable variables (Immerman & Kozen 89)

Exercises

3. Define a relational structure for the Boolean functions "And", "Or", "Not".
4. Can you express the propositional formula:

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

   by a conjunctive query in this database? Which of the Boolean functions in the database do you really need for this?
5. Let P be a set of propositional variables. Given a formula of the form:

$$\begin{aligned} t &::= p \mid \neg p \\ C &::= \mathit{false} \mid t \mid C \vee C' \\ D &::= \mathit{true} \mid C \mid D \wedge D' \end{aligned}$$

   Show how to define the set of all solutions of a formula D by a FO conjunctive query with equality on the database for the Boolean functions.

Exercises

6. Can you define a closed formula *tree* in the FO-logic of digraphs with relational signature $\Sigma = \{edge^{(2)}\}$, which is satisfied by a digraph if and only if it is a tree? (Be careful, this question is difficult).

Outline

1. Databases as relational structures
2. FO logics for relational structures
   - FO-Queries
   - Query answering
3. FO logics for XML databases
   - Unranked trees
   - FO logic for unranked trees

Unranked Trees

Unranked trees in T_Δ

$t ::= a(t_1, \ldots, t_n)$ where $a \in \Delta$ and $n \in \mathbb{N}_0$

As colored digraphs (naive)

$G_t = (nod(t), (lab_a^t)_{a \in \Delta}, (\mathrm{child}_i^t)_{i \in \mathbb{N}})$
but need an infinite number of edge colors in order to express order of children (whose number may be unbounded).

FO unsuitable!

FO logic of these digraphs cannot express $\mathrm{child}(x, y)$ since this formula is equivalent to an infinite disjunction $\vee_{i \in \mathbb{N}} \mathrm{child}_i(x, y)$ but FO formulas are finite.

Unranked Trees as Relational Structures

Idea: less naive colored digraphs

Add $\mathrm{next\_sib}$ edges and keep all others but $\mathrm{first\_child}$ edges!
This can be understood as an encoding of unranked trees to binary terms:
$a(b, c, d(e)) \Rightarrow a(b(\bot, c(\bot, d(e(\bot, \bot), \bot))), \bot)$

Colors

In addition need reflexive transitive closure $\mathrm{next\_sib}^*$ and child^*. Since $\mathrm{next\_sib}$ and $\mathrm{first\_child}$ are FO definable from these, we set

$$C_{nod} = \{lab_a \mid a \in \Delta\}$$
$$C_{edg} = \{\mathrm{child}^*, \mathrm{next\_sib}^*\}$$

Relations of tree $t \in T_\Delta$ interpreting colors

$$lab_a(t) = \{\pi \in nod(t) \mid lab^t(\pi) = a\}$$
$$\mathrm{child}^*(t) = \{(\pi, \pi \cdot \pi') \mid \pi \cdot \pi' \in nod(t)\}$$
$$\mathrm{next\_sib}^*(t) = \{(\pi \cdot i, \pi \cdot j \mid \pi \cdot j \in nod(t), 1 \leq i \leq j\}$$

FO Logic of Unranked Trees in T_Δ

everything is said already by defining the relational structure.

Syntax (where $a \in \Delta$, $x \in \mathrm{Vars}$):

$$\phi ::= \forall x \phi \mid \neg \phi \mid \phi \wedge \phi \mid lab_a(x) \mid \mathrm{child}^*(x, y) \mid \mathrm{next\_sib}^*(x, y)$$

Semantics

For tree t tree and variable assignement $\alpha : \mathrm{Vars} \to nod(t)$:

$$eval^{\alpha,t}(\phi) \in \mathbb{B}$$

Example on Expressiveness

Express equality $x = y$ by FO formula without equality

$$\mathrm{child}^*(x, y) \wedge \mathrm{child}^*(y, x)$$

A formula that has the same semantics as $\mathrm{child}(x, y)$

$$\mathrm{child}^*(x, y) \wedge \neg\exists z.(\mathrm{child}^*(x, z) \wedge \mathrm{child}^*(z, y) \wedge x \neq z \wedge z \neq y)$$

Formula that has the same semantics as $\mathrm{first\_child}^*(x, y)$

$$\mathrm{first\_child}^* = \mathrm{child}^* \cap \neg(\mathrm{child}^* \circ \mathrm{next\_sib} \circ \mathrm{child}^*)$$

Negative Examples on Expressiveness?

Can you define $\mathrm{first\_child}^*$ from $\mathrm{first\_child}$ in FO? (no)
Can you define $\mathrm{second\_child}^*$ in FO? This is the relation $(\mathrm{first\_child} \circ \mathrm{next\_sib})^*$. (yes)

Exercises [FO in Unranked Trees]

In all the following exercises, we consider the FO logic for unranked trees in T_Δ, with binary relations child^* and next_sib^* and labeling relations for all symbols of Δ.

Cousins

7. Find a FO formula $\text{cousin}(x, y)$ that relates nodes x to their cousins y.

Next siblings

8. Find a FO formula $\text{next\_sib}(x, y)$ that relates nodes x to their nextsibling y. Note that your formulas may only contain atoms $\text{next\_sib}^*(x', y')$.

Exercises [FO in Unranked Trees]

For every tree $t \in T_\Delta$ the leaves of t are totally ordered from the left to the right.

Leaf Order

9. Find a FO formula $\mathrm{following\_leaf}(x, y)$ that relates leaves x to a greater leaf in this order.

Next Leaf

10. Find a FO formula $\mathrm{next\_leaf}(x, y)$ that relates leafs x and y if y succeeds x immediately in this total order.

Exercises [FO in Unranked Trees]

Document Order

For every tree $t \in T_\Delta$, the document order is a total order on $nod(t)$, according to the visit order during the preorder traversal on t, which operates left-first and depth-first. We denote this order by $\leq^t_{\text{following}}$. The same order can be obtained by restricting the lexicographic order on $\mathbb{N}^*$ to $nod(t)$.

11. Define $\leq^t_{\text{following}}$ formally.
12. Find a FO formula $\text{following}(x, y)$ that relates nodes x to greater nodes y in this order.

    Find a FO formula $\text{next}(x, y)$ that relates nodes x to immediate sucessors y in that order.