Database Theory and Knowledge Representation

2nd Lecture

David Carral

University of Montpellier

October 5, 2023

Summary and Outlook

We have covered the following topics:

- The relational data model
- Relational queries
- First-order queries

Future Content:

- FO-Queries: exercises
- Complexity of query answering
- Query expressivity: comparing RA and FO queries

First-order Logic Queries

Definition: FO Queries

An n-ary first-order query q is an expression $\varphi[x_1, \ldots, x_n]$ where $x_1, \ldots, x_n$ are exactly the free variables of φ (in a specific order).

Definition: FO Query Answering

An answer to $q = \varphi[x_1, \ldots, x_n]$ over an interpretation $\mathcal{I}$ is a tuple $\langle a_1, \ldots, a_n \rangle$ of constants such that

$$\mathcal{I} \models \varphi[x_1/a_1, \ldots, x_n/a_n]^1$$

where $\varphi[x_1/a_1, \ldots, x_n/a_n]$ is φ with each free x_i replaced by a_i.

The result of q over $\mathcal{I}$ is the set of all answers of q over $\mathcal{I}$.

[1] Note that $\varphi[x_1/a_1, \ldots, x_n/a_n]$ does not feature answer variables.

Boolean Queries

A Boolean query is a query of arity 0
$\rightsquigarrow$ we simply write φ instead of $\varphi[]$
$\rightsquigarrow$ φ is a closed formula (a.k.a. sentence)

What does a Boolean query return?

Two possible cases:

- $\mathcal{I} \not\models \varphi$, then the result of φ over $\mathcal{I}$ is $\emptyset$ (the empty table)
- $\mathcal{I} \models \varphi$, then the result of φ over $\mathcal{I}$ is $\{\langle\rangle\}$ (the unit table)

Interpreted as Boolean check with result true or false (i.e., match or no match)

Exercises

Films

Title	**Director**	**Actor**
The Imitation Game	Tyldum	Cumberbatch
The Imitation Game	Tyldum	Knightley
...	...	...
Internet's Own Boy	Knappenberger	Swartz
Internet's Own Boy	Knappenberger	Lessig
Internet's Own Boy	Knappenberger	Berners-Lee
...	...	...
Dogma	Smith	Damon
Dogma	Smith	Affleck

Venues

Cinema	**Address**	**Phone**
UFA	St. Peter St. 24	4825825
Diagon	King St. 55	8032185
...	...	...

Program

Cinema	**Title**	**Time**
Diagon	The Imitation Game	19:30
Diagon	Dogma	20:45
UFA	The Imitation Game	22:45

1. Who is the director of "The Imitation Game"?

$$\exists y_A.\ \text{Films}(\texttt{"The Imitation Game"}, x_D, y_A)[x_D]$$

2. Which cinemas feature "The Imitation Game"?

$$\exists y_T.\ \text{Program}(x_C, \texttt{"The Imitation Game"}, y_T)[x_C]$$

Exercises

Films

Title	**Director**	**Actor**
The Imitation Game	Tyldum	Cumberbatch
The Imitation Game	Tyldum	Knightley
...	...	...
Internet's Own Boy	Knappenberger	Swartz
Internet's Own Boy	Knappenberger	Lessig
Internet's Own Boy	Knappenberger	Berners-Lee
...	...	...
Dogma	Smith	Damon
Dogma	Smith	Affleck

Venues

Cinema	**Address**	**Phone**
UFA	St. Peter St. 24	4825825
Diagon	King St. 55	8032185
...	...	...

Program

Cinema	**Title**	**Time**
Diagon	The Imitation Game	19:30
Diagon	Dogma	20:45
UFA	The Imitation Game	22:45

3. What are the address and phone number of "UFA"?

$$\text{Venues}(\texttt{"UFA'"}, x_A, x_P)[x_A, x_P]$$

4. *Boolean query:* Is a film directed by "Smith" playing?

$$\exists y_T, y_A, y_C, z_T.\ \text{Films}(y_T, \texttt{"Smith"}, y_A) \wedge \text{Program}(y_C, y_T, z_T)$$

Exercises

Films

Title	**Director**	**Actor**
The Imitation Game	Tyldum	Cumberbatch
The Imitation Game	Tyldum	Knightley
...	...	...
Internet's Own Boy	Knappenberger	Swartz
Internet's Own Boy	Knappenberger	Lessig
Internet's Own Boy	Knappenberger	Berners-Lee
...	...	...
Dogma	Smith	Damon
Dogma	Smith	Affleck

Venues

Cinema	**Address**	**Phone**
UFA	St. Peter St. 24	4825825
Diagon	King St. 55	8032185
...	...	...

Program

Cinema	**Title**	**Time**
Diagon	The Imitation Game	19:30
Diagon	Dogma	20:45
UFA	The Imitation Game	22:45

5. List the pairs of persons such that the first directed the second in a film, and vice versa.

$$\exists y_T, z_T.\ \mathsf{Films}(y_T, x_D, x_A) \wedge \mathsf{Films}(z_T, x_A, x_D)[x_D, x_A]$$

6. List the names of directors who have acted in a film they directed.

$$\exists y_T.\ \mathsf{Films}(y_T, x_D, x_D)[x_D]$$

Exercises

Films

Title	**Director**	**Actor**
The Imitation Game	Tyldum	Cumberbatch
The Imitation Game	Tyldum	Knightley
...	...	...
Internet's Own Boy	Knappenberger	Swartz
Internet's Own Boy	Knappenberger	Lessig
Internet's Own Boy	Knappenberger	Berners-Lee
...	...	...
Dogma	Smith	Damon
Dogma	Smith	Affleck

Venues

Cinema	**Address**	**Phone**
UFA	St. Peter St. 24	4825825
Diagon	King St. 55	8032185
...	...	...

Program

Cinema	**Title**	**Time**
Diagon	The Imitation Game	19:30
Diagon	Dogma	20:45
UFA	The Imitation Game	22:45

7. Return $\{\text{Title} \mapsto \texttt{"Ap.\ Now"}, \text{Director} \mapsto \texttt{"Coppola"}\}$ as the answer.

$$\{\text{DirectedBy}(\texttt{"Apocalypse Now"}, \texttt{"Coppola"})\}$$

8. Find the actors cast in at least one film by "Smith".

$$\exists y_T.\ \text{Films}(y_T, \texttt{"Smith"}, x_A)[x_A]$$

Exercises

Films

Title	**Director**	**Actor**
The Imitation Game	Tyldum	Cumberbatch
The Imitation Game	Tyldum	Knightley
...	...	...
Internet's Own Boy	Knappenberger	Swartz
Internet's Own Boy	Knappenberger	Lessig
Internet's Own Boy	Knappenberger	Berners-Lee
...	...	...
Dogma	Smith	Damon
Dogma	Smith	Affleck

Venues

Cinema	**Address**	**Phone**
UFA	St. Peter St. 24	4825825
Diagon	King St. 55	8032185
...	...	...

Program

Cinema	**Title**	**Time**
Diagon	The Imitation Game	19:30
Diagon	Dogma	20:45
UFA	The Imitation Game	22:45

9. Find the actors that are NOT cast in a movie by "Smith."

$$\exists y_T, y_D.\text{Films}(y_T, y_D, x_A) \wedge$$
$$\forall x_T, x_D.(\text{Films}(x_T, x_D, x_A) \rightarrow x_D \not\approx \texttt{"Smith"}[x_A]$$

10. Find all pairs of actors who act together in at least one film.

$$\exists \vec{y}.\ \text{Films}(y_T, y_D, x_A) \wedge \text{Films}(y_T, y'_D, x_{A'}) \wedge x_A \not\approx x_{A'}[x_A, x_{A'}]$$

Review: The Relational Calculus

What we have learned so far:

- There are many ways to describe databases:
  $\rightsquigarrow$ set of tables, set of facts, (hyper)graphs
- We have studied two different languages:
  $\rightsquigarrow$ relational algebra and FO queries

Later we show that the above languages are equivalent:
The Relational Calculus

Outlook:

- Next question: How hard is it to answer such queries?
- Related question: Are you familiar with computational complexity theory?

How to Measure Complexity of Queries?

- Complexity classes often for decision problems
  $\rightsquigarrow$ database queries return many results
- The size of a query result can be very large
  $\rightsquigarrow$ it would not be fair to measure this as "complexity"
- In practice, database instances are much larger than queries
  $\rightsquigarrow$ can we take this into account?

Query Answering as Decision Problem

We consider the following decision problems:

- Boolean Query Entailment: given a Boolean query q and a database instance $\mathcal{I}$, does $\mathcal{I} \models q$ hold?
- Query Answering Problem: given an n-ary query q, a database instance $\mathcal{I}$ and a tuple $\langle c_1, \ldots, c_n \rangle$, does $\langle c_1, \ldots, c_n \rangle \in M[q](\mathcal{I})$ hold?
- Query Emptiness Problem: given a query q and a database instance $\mathcal{I}$, does $M[q](\mathcal{I}) \neq \emptyset$ hold?

Discussion

These problems are computationally equivalent.

The Size of the Input

Definition: Combined Complexity

Input: Boolean query q and database instance $\mathcal{I}$
Output: Does $\mathcal{I} \models q$ hold?

$\rightsquigarrow$ "2KB query/2TB database" $=$ "2TB query/2KB database"

Study worst-case complexity of algorithms for fixed queries:

Definition: Data Complexity

Input: database instance $\mathcal{I}$
Output: Does $\mathcal{I} \models q$ hold? (for fixed q)

We can also fix the database and vary the query:

Definition: Query Complexity

Input: Boolean query q
Output: Does $\mathcal{I} \models q$ hold? (for fixed $\mathcal{I}$)

The Turing Machine (1)

Computation is usually modelled with Turing Machines (TMs)
$\rightsquigarrow$ "algorithm" = "something implemented on a TM"

A TM is an automaton with (unlimited) working memory:

- It has a finite set of states Q
- Q includes a start state q_{start} and an accept state q_{acc}
- The memory is a tape with numbered cells $0, 1, 2, \ldots$
- Each tape cell holds one symbol from the set of tape symbols Γ
- There is a special symbol $\sqcup$ for empty tape cells
- The TM has a transition relation $\Delta \subseteq (Q \times \Gamma) \times (Q \times \Gamma \times \{l, r, s\})$
- Δ might be a partial function $(Q \times \Gamma) \to (Q \times \Gamma \times \{l, r, s\})$
  $\rightsquigarrow$ deterministic TM (DTM); otherwise nondeterministic TM

There are many different but equivalent ways of defining TMs.

The Turing Machine (2)

TMs operate step-by-step:

- At every moment, the TM is in one state $q \in Q$ with its read/write head at a certain tape position $p \in \mathbb{N}$, and the tape has a certain contents $\sigma_0\sigma_1\sigma_2\cdots$ with all $\sigma_i \in \Gamma$
  $\rightsquigarrow$ current configuration of the TM
- The TM starts in state q_{start} and at tape position 0.
- Transition $\langle q, \sigma, q', \sigma', d\rangle \in \Delta$ means:
  if in state q and the tape symbol at its current position is σ,
  then change to state q', write symbol σ' to tape, move head by d (left/right/stay)
- If there is more than one possible transition, the TM picks one nondeterministically
- The TM halts when there is no possible transition for the current configuration (possibly never)

A computation path (or run) of a TM is a sequence of configurations that can be obtained by some choice of transition.

The Turing Machine (3)

A Turing machine can be described with different levels of precision:

- Formal level: define the states, transition function, alphabet, etc; can be done via diagram (see example in the board).
- Implementational level: describe how the machine works at an implementational level; e.g., describe encodings precisely as well as how the different tapes will be used.
- High level: give an intuitive description of how the Turing machine works.

Example

Discuss how to implement a Turing machine that computes the result of the join operator.

Solving Computation Problems with TMs

A decision problem is a language $\mathcal{L}$ of words over $\Sigma = \Gamma \setminus \{\sqcup\}$
$\rightsquigarrow$ the set of all inputs for which the answer is "yes"

A TM decides a decision problem $\mathcal{L}$ if it halts on all inputs and accepts exactly the words in $\mathcal{L}$

TMs take time (number of steps):

- $\textsc{Time}(f(n))$: Problems that can be decided by a DTM in $O(f(n))$ steps, where f is a function of the input length n
- $\textsc{Space}(f(n))$: Problems that can be decided by a TM that uses at most $O(f(n))$ tape cells

Reminder

Given some functions f and g defined over the natural numbers, we write $f(x) = O(g(x))$ to indicate that there are some $n, x_0 > 0$ such that $|f(x)| \leq n \cdot g(x)$ for all $x \geq x_0$.

Some Common Complexity Classes

$$\mathrm{P} = \mathrm{PTime} = \bigcup_{k \geq 1} \mathrm{Time}(n^k)$$

$$\mathrm{Exp} = \mathrm{ExpTime} = \bigcup_{k \geq 1} \mathrm{Time}(2^{n^k})$$

$$2\mathrm{Exp} = 2\mathrm{ExpTime} = \bigcup_{k \geq 1} \mathrm{Time}(2^{2^{n^k}})$$

$$\mathrm{ETime} = \bigcup_{k \geq 1} \mathrm{Time}(2^{nk})$$

$$\mathrm{L} = \mathrm{LogSpace} = \mathrm{Space}(\log n)$$

$$\mathrm{PSpace} = \bigcup_{k \geq 1} \mathrm{Space}(n^k)$$

$$\mathrm{ExpSpace} = \bigcup_{k \geq 1} \mathrm{Space}(2^{n^k})$$

How to Measure Query Answering Complexity

Query answering as decision problem
$\rightsquigarrow$ consider Boolean queries

Various notions of complexity:

- Combined complexity (complexity w.r.t. size of query and database instance)
- Data complexity (worst case complexity for any fixed query)
- Query complexity (worst case complexity for any fixed database instance)

Various common complexity classes:

$$\mathrm{P} \subseteq \mathrm{NP} \subseteq \mathrm{PSpace} \subseteq \mathrm{ExpTime}$$

An Algorithm for Evaluating FO Queries

```
function Eval(φ, I)

01  switch (φ) {
02      case p(c_1, ..., c_n) : return p(c_1, ..., c_n) ∈ I
03      case ¬ψ : return ¬Eval(ψ, I)
04      case ψ_1 ∧ ψ_2 : return Eval(ψ_1, I) ∧ Eval(ψ_2, I)
05      case ∃x.ψ :
06          for c ∈ Δ^I {
07              if Eval(ψ[x ↦ c], I) then return true
08          }
09          return false
10  }
```

Remark

The formula φ is a Boolean FO query. How can we extend the above procedure to solve query answering?

Summary and Outlook

We have covered the following topics:

- First-order queries: exercises
- Complexity of query entailment

Future Content:

- Query expressivity: comparing RA and FO queries
- More efficient (and less expressive) query languages