Updated 1/26/2021

Topic 1: Data models and query languages
Unit 2: Logic &relational calculus
Lecture 3

Wolfgang Gatterbauer

CS7240 Principles of scalable data management (sp21)

https://northeastern-datalab.github.io/cs7240/sp21/

1/26/2021

Topic 1: Data models and query languages

PRELIMINARY

- **Lecture 1 (Tue 1/19):** Course introduction, SQL refresher, PostgreSQL setup, SQL files
- **Lecture 2 (Fri 1/22):** SQL continued, Logic & relational calculus
- **Lecture 3 (Tue 1/26):** Relational calculus & relational algebra
- **Lecture 4 (Fri 1/29):** Relational algebra & Codd's theorem, Datalog
- **Lecture 5 (Tue 2/2):** Datalog and more expressive variations
- **Lecture 6 (Fri 2/5): (A1 due)** Datalog vs. stable model semantics
- **Lecture 7 (Tue 2/9):** Datalog evaluation strategies, NoSQL

Pointers to relevant concepts & supplementary material:

- **1. SQL**: SQL refresher [SAMS'12], [Cow'03] Ch3 & Ch5, [Complete'08] Ch6
- **2. Logic**: First-order logic, relational calculus: [Barland+'08] 4.1.2 & 4.2.1 & 4.4, [Genesereth+] Ch6, [Halpern+'01], [Cow'03] Ch4.3 & 4.4, [Elmasri, Navathe'15] Ch8.6 & Ch8.7, [Silberschatz+'10] Ch6.2 & Ch6.3 [Alice'95] Ch3.1-3.3 & Ch4.2 & Ch4.4 & Ch5.3-5.4
- **3. Algebra**: Relational algebra, Codd's theorem: [Cow'03] Ch4.2, [Complete'08] Ch2.4 & Ch5.1-5.2, [Elmasri, Navathe'15] Ch8, [Silberschatz+'10] Ch6.1, [Alice'95] Ch4.4 & Ch5.4
- **4. Datalog**: Datalog, stable model semantics: [Complete'08] Ch5.3, [Cow'03] Ch 24, [Koutris'19] L9 & L10, [Gatterbauer, Suciu'10]
- **5. Data models**: Alternative data models, NoSQL: [Hellerstein, Stonebraker'05], [Sadalage, Fowler'12], [Harrison'16]

Source: https://northeastern-datalab.github.io/cs7240/sp21/

Queries and the connection to logic and algebra

- Why logic?
  - A crash course on FOL
- Relational Calculus
  - Syntax and Semantics
  - Domain Independence and Safety

Logic in Computer Science and Databases

- Logic has had an immense impact on CS
- Computing has strongly driven one particular branch of logic: finite model theory
  - That is, First-order logic (FOL) restricted to finite models
  - Very strong connections to *complexity theory*
  - The basis of various branches in Artificial Intelligence
- It is a natural tool to capture and attack fundamental problems in database management
  - Relations as first-class citizens
  - Inference for assuring data integrity
  - *Inference for question answering (queries)*
- It has been used for developing and analyzing the relational model from the early days [Codd'72]

See also: Halpern, Harper, Immerman, Kolaitis, Vardi, Vianu. "*On the unusual effectiveness of logic in computer science*", 2001. https://doi.org/10.2307/2687775
A play on: Wigner. "*The unreasonable effectiveness of mathematics in the natural sciences*", 1960. https://doi.org/10.1142/9789814503488_0018

Why has Logics turned out to be so powerful?

- Basic Question: What on earth does an obscure, old intellectual discipline have to do with the youngest intellectual discipline?
- Cosma R. Shalizi, CMU:
  - "If, in 1901, a talented and sympathetic outsider had been called upon (say, by a granting-giving agency) to survey the sciences and name the branch that would be least fruitful in century ahead, his choice might well have settled upon mathematical logic, an exceedingly recondite field whose practitioners could all have fit into a small auditorium. It had no practical applications, and not even that much mathematics to show for itself: its crown was an exceedingly obscure definition of cardinal numbers."

Source: Moshe Vardi: Database Queries - Logic and Complexity

Back to The Future

- M. Davis (1988): Influences of Mathematical Logic on Computer Science:
  - “When I was a student, even the topologists regarded mathematical logicians as living in outer space. Today the connections between logic and computers are a matter of engineering practice at every level of computer organization.”

- Question: Why on earth?

Source: Moshe Vardi: Database Queries - Logic and Complexity

Birth of Computer Science: 1930s

- Church, Gödel, Kleene, Post, Turing: Mathematical proofs have to be “machine checkable” - computation lies at the heart of mathematics!
  - Fundamental Question: What is “machine checkable”?
- Fundamental Concepts:
  - algorithm: a procedure for solving a problem by carrying out a precisely determined sequence of simpler, unambiguous steps
  - distinction between hardware and software
  - a universal machine: a machine that can execute arbitrary programs
  - a programming language: notation to describe algorithms

Source: Moshe Vardi: Database Queries - Logic and Complexity

Leibniz's Dream

An Amazing Dream: a universal mathematical language, lingua characteristica universalis, in which all human knowledge can be expressed, and calculational rules, calculus ratiocinator, carried out by machines, to derive all logical relationships

- "If controversies were to arise, there would be no more need of disputation between two philosophers than between two accountants. For it would suffice to take their pencils in their hands, and say to each other: Calculemus–Let us calculate."

Source: Moshe Vardi: Database Queries - Logic and Complexity

Example: Aristotle' Syllogisms

- "All men are mortal"

?

Source: Moshe Vardi: Database Queries - Logic and Complexity

Example: Aristotle' Syllogisms

- "All men are mortal"

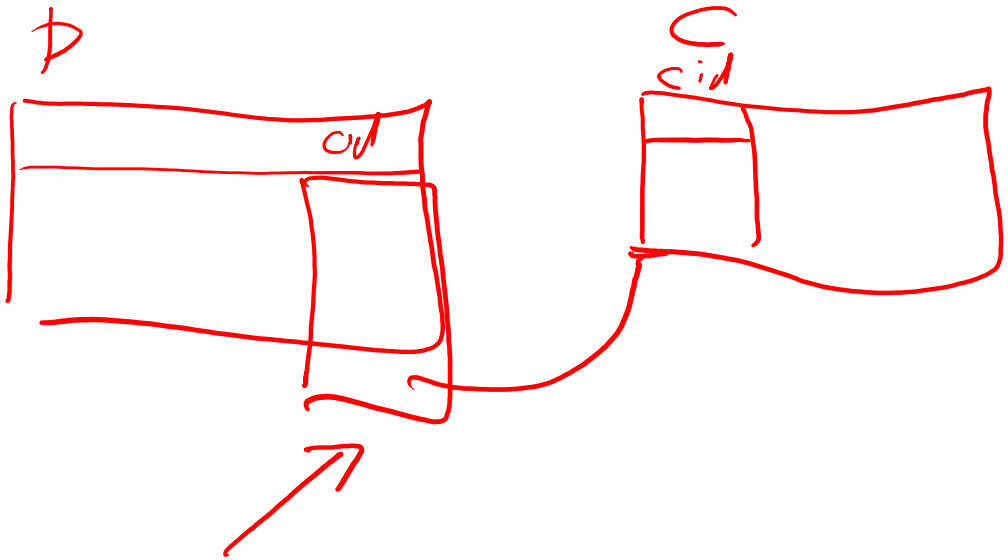
- "For all x, if x is a man, then x is mortal"

?

Source: Moshe Vardi: Database Queries - Logic and Complexity

Example: Aristotle' Syllogisms

- "All men are mortal"
- "For all x, if x is a man, then x is mortal"
- ∀x [Man(x) → Mortal(x)]



Source: Moshe Vardi: Database Queries - Logic and Complexity

Logic and Databases

Two main uses of logic in databases:

- Logic used as a database query language to express questions asked against databases (our main focus)

- Logic used as a specification language to express integrity constraints in databases (example from previous slide)

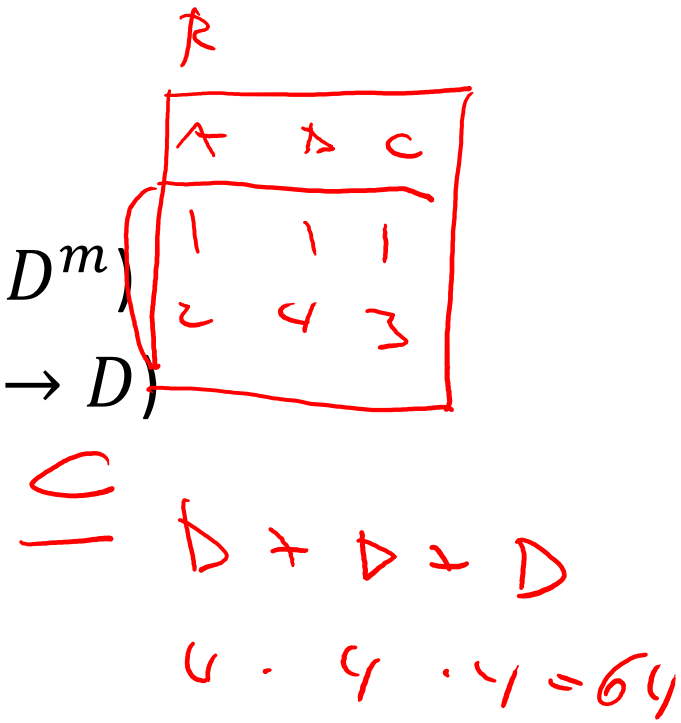
Source: Phokion Kolaitis

Queries and the connection to logic and algebra

- Why logic?
  - A crash course on FOL
- Relational Calculus
  - Syntax and Semantics
  - Domain Independence and Safety

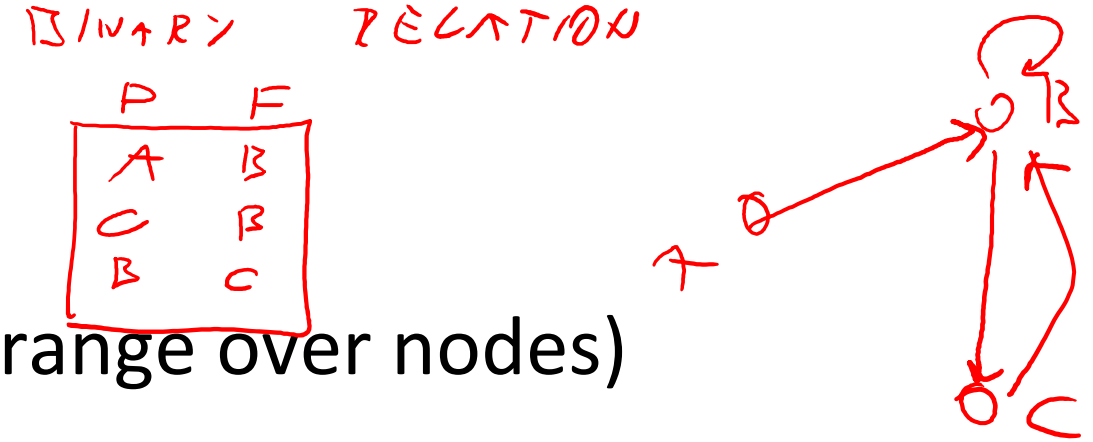
First-Order Logic

- A formalism for specifying properties of mathematical structures, such as graphs, partial orders, groups, rings, fields, . . .
- Mathematical Structure:
  - $A = (D, R_1, \ldots, R_k, f_1, \ldots, f_l)$
  - D is a non-empty set: universe, or domain
  - R_i is an m-ary relation on D, for some m (that is, $R_i \subseteq D^m$)
  - f_j is an n-ary function on D, for some n (that is, $f_i: D^n \to D$)



Source: Moshe Vardi: Database Queries - Logic and Complexity

First-Order Logic on Graphs



Syntax:

- First-order variables: x, y, z, . . . (range over nodes)
- Atomic formulas: E(x, y), x = y
- Formulas: Atomic Formulas + Boolean Connectives (∨, ∧, ¬) + First-Order Quantifiers (∃x, ∀x)

Source: Moshe Vardi: Database Queries - Logic and Complexity

Examples

Assume schema is E(from, to), yet undirected. Thus for every edge E(x,y), we also have E(y,x).

- "node x has at least two distinct neighbors"

?

- "each node has at least two distinct neighbors"

?

Source: Moshe Vardi: Database Queries - Logic and Complexity

Examples

Assume schema is E(from, to), yet undirected. Thus for every edge E(x,y), we also have E(y,x).

- "node x has at least two distinct neighbors"
  - $\exists y \exists z [\neg(y = z) \wedge E(x, y) \wedge E(x, z)]$
  - Notice: x is free in the above formula, which expresses a property of nodes.

QUERY : return x

- "each node has at least two distinct neighbors"

?

Source: Moshe Vardi: Database Queries - Logic and Complexity