CS245: Database Management Systems

Vijaya Saradhi

IIT Guwahati

Fri, 03-Apr-020

Introduction

Datalog

- The contents of today's self study is from the text book **Database Management Systems The Complete Book by Hector Garcia-Molina et. al** `https://bit.ly/2UTjzMU`
- Please refer to chapter 10 - *Logical Query Languages*
- In particular sections 10.1 and 10.2
- In this self study material and the next the idea is to introduce **Recursion** in SQL
- Simplest form of logic devised for relational model
- In its non-recursive form, datalog have the same power as the classical relational algebra
- By allowing recursion, we can express queries in datalog that cannot be expressed in SQL2
- Solution provided by datalog has been used to provide a way to allow meaningful recursion in the most recent SQL-99 standard

A Logic for Relations

Logic

- Alternative to abstract query language based on algebra, one can use a form of logic to express queries
- The logical query language Datalog (database logic) consists of if-then-else rules
- Each of these rules expresses the idea that from certain combinations of tuples in certain relations we may infer the answer to a query (that some other tuple is in some other relation)

Predicates and Atoms

Definitions

- Relations are represented in Datalog by predicates
- Each predicate takes a fixed number of arguments
- A predicate followed by its arguments is called an atom
- Syntax of atoms is just like that of function calls in conventional programming language
- Example: $P(x_1, x_2, \cdots, x_n)$ is an atom
- This atom consists of predicate P with arguments $x_1, x_2, \cdots, x_n$

Predicates and Atoms

Definitions

- A predicate is the name of a function
- The function that returns a boolean value
- If R is a relation with n attributes in some fixed order, then we shall also use R as the name of the predicate
- The atom $R(a_1, a_2, \cdots, a_n$ have value TRUE if $(a_1, a_2, \cdots, a_n)$ is a tuple of R
- The atom have value FALSE otherwise

Predicates and Atoms

Example

R	
A	B
1	2
3	4

- R(1, 2) is true and so is R(3, 4)
- For any other values x, y, $R(x, y)$ is false

Predicates and Atoms

Example

- If R is the predicate of the above example, $R(x, y)$ is the function that tells for x and y whether the tuple (x, y) is in relation R
- R(x, y) returns true if x = 1 and y = 2;
- R(x, y) returns true if x = 3 and y = 4

Arithmetic Atoms

Arithmetic Atoms

- $<, \leq, =, \neq, >, \geq$ are arithmetic atoms
- They perform comparison between two arithmetic expressions
- Both relational atoms and arithmetic atoms take as arguments values of variables
- Both return boolean values

Datalog Rules and Queries

Rules and Queries

- A datalog rule consists of
  - A relational atom call the head followed by
  - the symbol $\leftarrow$ read as "if" followed by
  - A body consisting of one or more atoms known as subgoals

Datalog Rules and Queries

Example Rule

- `LongMovie(t, y)` $\leftarrow$ `Movie(t, y, l, c, s, p) AND l` ≥ 100
- t: title; y: year; l: length; c: inColor; s: studionName; p: producer
- The above defines a set of long movies which are at least 100 minutes long
- The head of the rule is the atom `LongMovie(t, y)`
- Body consists of two subgoals

First	Have predicate `Movie` with six arguments
Arugments	a variable assuming a value;
Interpretation	`Movie(t, y, l, c, s, p)` is true whenever the six variables have values that is in `Movie` relation
Second	$l \geq 100$
Arugments	None
Interpretation	for `(t, y, l, c, s, p)` which is in Movie relation and $l \geq 100$

Datalog Rules and Queries

Example Rule

- When ever these variables all have values that make subgoals true, then we see what value of head is for those variables
- We add the resulting tuple to the relation whose predicate is the head

Datalog Rules and Queries

Restrictions

- Safety Condition: Every variable that appears anywhere in the rule must appear in some non-negated, relational subgoal
- This is to ensure that result of a rule is finite relation and
- Rules with arithmetic subgoals or with negated subgoals makes intuitive sense

Datalog Rules and Queries

Restrictions

- `P(x, y)` $\leftarrow Q(x, z)$ `AND NOT R(w, x, z) AND` $x < y$
- Variable `y` appears in the head but not in any non-negated relational subgoal
- Variable `w` appears in negated, relational subgoal but not in the non-negated, relational subgoal
- Variable `y` appears in an arithmetic subgoal but not in a non-negated, relational subgoal

Datalog Rules and Queries

Example

- `P(x, y)` $\leftarrow$ `Q(x, z) AND R(z, y) AND NOT Q(x, y)`
- Let Q consists of (1, 2) and (1, 3)
- R consists of (2, 3) and (3, 1)
- Two non-negated relational subgoals `Q(x, z)`, `R(z, y)`

Q(x, z)	R(z, y)	Consistent?	Not Q(x, y)?	Head
(1, 2)	(2, 3)	Yes	No	-
(1, 2)	(3, 1)	No; $z = 2, 3$	Irrelevant	-
(1, 3)	(2, 3)	No; $z = 3, 2$	Irrelevant	-
(1, 3)	(3, 1)	Yes	Yes	P(1,1)

Extensional and Intensional Predicates

Definitions

Extensional Predicates (EDB) Whose relations are stored in a database

Intensional Predicates (IDB) Whose relations are Computed by applying one or more Datalog rules

Datalog Rules and Bags

Example

H(x, z) ← R(x, y) AND S(y, z)

R(A, B)		S(B,C)	
A	B	B	C
1	2	2	3
1	2	4	5
		4	5

H(x, z)	
X	Z
1	3
1	3

Example with more than one rule

Example

Consider the following two rules

- `H(x, y)` $\leftarrow$ `S(x, y) AND` $x > 1$
- `H(x, y)` $\leftarrow$ `S(x, y) AND` $y < 5$
- Rule 1 yields the H(x, y) set { (2, 3), (4, 5), (4, 5)}
- Rule 2 yields the H(x, y) set { (2, 3)}

Datalog: intersection

Definition

`I(a, b, c, d)` $\leftarrow$ `R(a, b, c, d) AND S(a, b, c, D)`

R(a, b, c, d) is EDB

S(a, b, c, d) is EDB

I(a, b, c, d) is IDB

Datalog: Union

Definition

1. `U(a, b, c, d) ← R(a, b, c, d)`
2. `U(a, b, c, d) ← S(a, b, c, d)`

1. First rule states every tuple in R is a tuple in the IDB relation U
2. Second rule states every tuple in S is a tuple in the IDB relation U
3. Thus the two rules together represent the union

Datalog: Difference

Definition

$\mathtt{I(a,\ b,\ c,\ d)} \leftarrow \mathtt{R(a,\ b,\ c,\ d)\ AND\ NOT\ S(a,\ b,\ c,\ d)}$

Datalog: Projection

Definition

`I(b, d)` $\leftarrow$ `R(a, b, c, d)`

1. A single subgoal with predicate R is sufficient.
2. Arguments of this subgoal are distinct variables one for each attribute in the relation
3. The head has an atom with arguments that are variables corresponding to the attributes in the projection list

Datalog: Selection

Selection

- Simple case is when the selection condition is the AND of one more more arithmetic comparisons
- Rules are:
    - One relational subgoal for the relation upon which we are performing the selection
    - This atom has distinct variables for each component
    - one for each attribute of the relation
    - For each comparison in the selection comparison, an arithmetic subgoal that is identical to the comparison

Datalog: Selection

Selection Example

- $\sigma_{length \geq 100 AND studioName = 'Fox'}(Movie)$
- `S(t, y, l, c, s, p)` $\leftarrow$ `Movie(y, t, l, c, s, p) AND l` $\geq$ `100 AND s = 'Fox'`

Datalog: Selection

Selection Example

1. $\sigma_{length \geq 100 OR studioName = 'Fox'}(Movie)$
2. `S(t, y, l, c, s, p)` $\leftarrow$ `Movie(y, t, l, c, s, p) AND l` $\geq$ `100`
3. `S(t, y, l, c, s, p)` $\leftarrow$ `Movie(y, t, l, c, s, p) AND s = 'Fox'`

- Rule 1 produces movies at least 100 minutes long
- Rule 2 produces movies produced by 'Fox'

Datalog: Selection

Selection Example - Negation

1. $\sigma_{NOT(length \geq 100 OR studioName = 'Fox')}(Movie)$
2. $\sigma_{NOT(length \geq 100) AND (NOT(studioName = 'Fox')}(Movie)$
3. $\sigma_{(length < 100) AND ((studioName \neq 'Fox')}(Movie)$
4. $\texttt{S(t, y, l, c, s, p)} \leftarrow \texttt{Movie(y, t, l, c, s, p) AND l} < \texttt{100 AND s} \neq \texttt{'Fox'}$

Datalog: Cross Product

Definition

`I(a, b, c, d)` $\leftarrow$ `R(a, b) AND S(c, d)`

Datalog: Natural Join

Definition

$I(a, b, c, d) \leftarrow R(a, b)$ AND $S(b, c, d)$

Datalog: Theta Join

Definition

1. $U \underset{a<d \ AND \ U.b \neq V.b}{\bowtie} V$
2. I(a, ub, uc, vb, vc, d) $\leftarrow$ U(a, ub, uc) AND V(vb, vc, d) AND a < d
3. I(a, ub, uc, vb, vc, d) $\leftarrow$ U(a, ub, uc) AND V(vb, vc, d) AND ub $\neq$ vb

Datalog: Recursive Programming

Introduction

Consider the relation

SequelOf	
movie	sequel
Rocky	Rocky II
Rocky II	Rocky III
Rocky III	Rocky IV