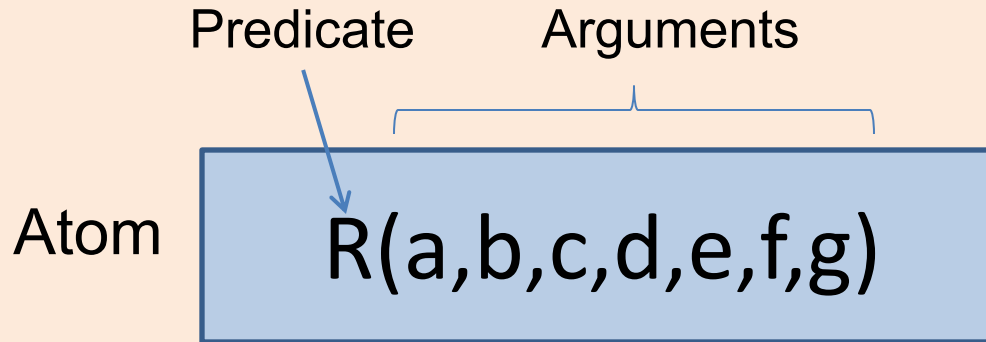


ICS 321 Spring 2011  
Algebraic and Logical Query Languages (ii)

Asst. Prof. Lipyeow Lim  
Information & Computer Science Department  
University of Hawaii at Manoa

# Datalog : Database Logic



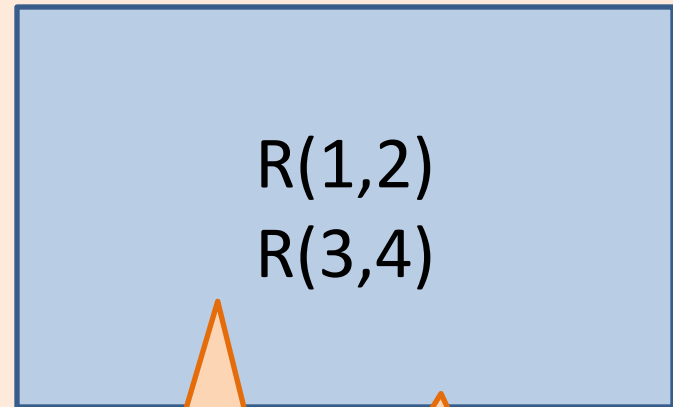
- A (relational) **atom**
  - Consists of a **predicate** and a list of **arguments**
  - Arguments can be **constants** or **variables**
  - Takes on Boolean value (true or false)
- A relation  $R$  can be represented as a predicate  $R$ 
  - A tuple  $\langle a,b,c,d,e,f,g \rangle$  is in  $R$  iff the atom  $R(a,b,c,d,e,f,g)$  is true.

# Example: tables in datalog

R

A	B
1	2
3	4

Datalog



True by default.

R(1,4) would  
be false

# Arithmetic Atoms

$$x < y$$
$$x+1 \geq y+4*z$$

Can contain  
both constants  
and variables.

# Datalog Rules

head

“if” or ←

body

Shorthand  
for AND

$\text{LongMovie}(t,y) :- \text{Movies}(t,y,l,g,s,p) , l \geq 100$

(t,y) is a tuple of LongMovie  
**IF** (t,y,l,g,s,p) is a tuple of  
Movies and length of movie is  
at least 100

These two  
“t,y” have to  
match

These two  
“l” have to  
match

Aka “subgoal”  
Can be preceded  
by negation  
operator “NOT”  
or “~”

Anonymous variables

$\text{LongMovie}(t,y) :- \text{Movies}(t,y,l,_,_,_) , l \geq 100$

# Safety Condition for Datalog Rules

Every **variable** that appears anywhere in the rule **must** appear in some **nonnegated, relational subgoal** of the body

- Without the safety condition, rules may be underspecified, resulting in an infinite relation (not allowed).
- Examples
  - $\text{LongMovie}(t,y) \text{ :- Movies}(t,y,l,\_,\_,\_) , l \geq 100$
  - $P(x,y) \text{ :- } Q(x,z), \text{ NOT } R(w,x,z), x < y$

# Alternative Interpretation: Consistency

## Datalog

```
Q(1,2)
Q(1,3)
R(2,3)
R(3,1)
P(x,y) :- Q(x,z), R(z,y), NOT Q(x,y)
```

- For each consistent assignment of nonnegated, relational subgoal,
- Check the negated, relational subgoals and the arithmetic subgoals for consistency

Q(x,z)	R(z,y)	Consistent?	NOT Q(x,y)	Head
(1,2)	(2,3)	Yes	false	
(1,2)	(3,1)	No, z=2,3		
(1,3)	(2,3)	No, z=2,3		
(1,3)	(3,1)	Yes	true	P(1,1)

# Intensional vs Extensional

## Datalog

Q(1,2)

Q(1,3)

R(2,3)

R(3,1)

P(x,y) :- Q(x,z), R(z,y), NOT Q(x,y)

extensional

intensional

- **Extensional** predicates – relations stored in a database
- **Intensional** predicates – computed by applying one or more datalog rules



# What about bag semantics ?

- Datalog still works if there are no negated, relational subgoals.
- Treat duplicates like non-duplicates

## Datalog

```
R(1,2)
R(1,2)
S(2,3)
S(4,5)
S(4,5)
H(x,z) :- R(x,y), S(y,z)
```

R(x,y)	S(y,z)	Consistent?	Head
(1,2)	(2,3)	Yes	H(1,3)
(1,2)	(4,5)	No, y=2,4	
(1,2)	(4,5)	No, y=2,4	
...	...	...	...

# Example 1

## Datalog

Answer(x,y) :- A(x,y)

Answer(x,y) :- B(x,y)

# Example 2

## Datalog

```
Answer(x,y) :- A(x,y), B(x,y)
```

# Example 3

## Datalog

```
Answer(x,y) :- A(x,y), NOT B(x,y)
```

# Example 4

## Datalog

```
Answer(x,y) :- A(x,y), x > 10, y = 200
```

# Example 5

## Datalog

Answer(x) :- A(x,y)

# Example 6

## Datalog

Answer(w,x,y,z) :- A(w,x), B(y,z)

# Example 7

## Datalog

Answer(w,x,y) :- A(w,x), B(x,y)



# Example 8

## Datalog

Answer(w,x,z) :- A(w,x), B(y,z), x>y

# Example 9

## Datalog

```
Path(x,y) :- Edge(x,y)
```

```
Path(x,z) :- Edge(x,y), Edge(y,z)
```