Updated 1/27/2023

Topic 1: Data models and query languages
Unit 2: Logic & relational calculus
Lecture 6

Wolfgang Gatterbauer

CS7240 Principles of scalable data management (sp23)

https://northeastern-datalab.github.io/cs7240/sp23/

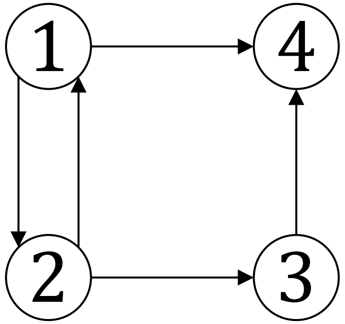
1/27/2023

Pre-class conversations

- Last class recapitulation
  - with more details and intuition

- today:
  - a bit more on logic (I maybe skimming)
  - the relational algebra (RA)

Example: Querying a Graph



A:

1	2
2	1
2	3
1	4
3	4

A encodes the directed edges of a graph ("arcs")

What do these queries return ?

$$\{ x \mid \exists y.\, A(x,y) \}$$

?

$$\{ x \mid \exists y,z,u.[A(x,y) \land A(y,z) \land A(z,u)] \}$$

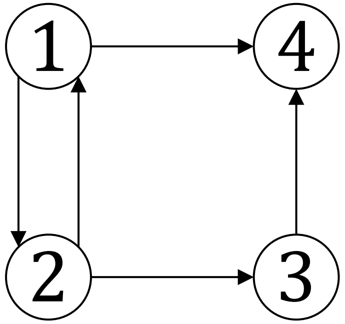
?

$$\{ (x,y) \mid \forall z.[A(x,z) \rightarrow A(y,z)] \}$$

?

Based on an example by Dan Suciu from CSE 554, 2011.

Example: Querying a Graph



A:

1	2
2	1
2	3
1	4
3	4

A encodes the directed edges of a graph ("arcs")

What do these queries return ?

$$\{ x \mid \exists y.\ A(x,y) \}$$

Nodes that have at least one child: ?

$$\{ x \mid \exists y,z,u.[A(x,y) \wedge A(y,z) \wedge A(z,u)] \}$$

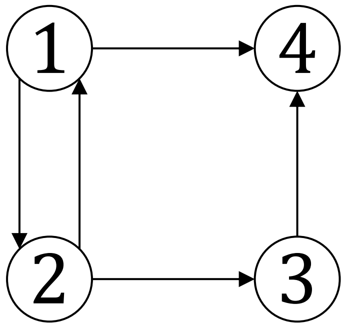
?

$$\{ (x,y) \mid \forall z.[A(x,z) \rightarrow A(y,z)] \}$$

?

Based on an example by Dan Suciu from CSE 554, 2011.

Example: Querying a Graph



A:

1	2
2	1
2	3
1	4
3	4

A encodes the directed edges of a graph ("arcs")

What do these queries return ?

$$\{\ x \mid \exists y.\ A(x,y)\ \}$$

Nodes that have at least one child: $\{1,2,3\}$

$$\{\ x \mid \exists y,z,u.[A(x,y) \land A(y,z) \land A(z,u)]\}$$

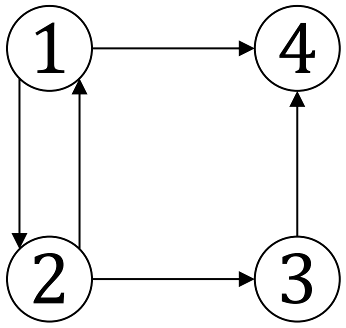
?

$$\{\ (x,y) \mid \forall z.[A(x,z) \rightarrow A(y,z)]\}$$

?

Based on an example by Dan Suciu from CSE 554, 2011.

Example: Querying a Graph



What do these queries return ?

A:

1	2
2	1
2	3
1	4
3	4

A encodes the directed edges of a graph ("arcs")

$$\{\, x \mid \exists y.\, A(x,y) \,\}$$

Nodes that have at least one child: {1,2,3}

$$\{\, x \mid \exists y,z,u.[A(x,y) \land A(y,z) \land A(z,u)]\}$$

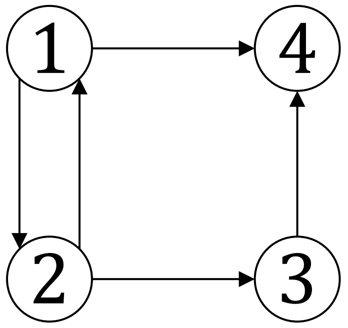
Nodes that have a great-grand-child: ?

$$\{\, (x,y) \mid \forall z.[A(x,z) \rightarrow A(y,z)]\}$$

?

Based on an example by Dan Suciu from CSE 554, 2011.

Example: Querying a Graph



What do these queries return ?

$$\{\, x \mid \exists y.\, A(x,y) \,\}$$

Nodes that have at least one child: {1,2,3}

A:

1	2
2	1
2	3
1	4
3	4

$$\{\, x \mid \exists y,z,u.[A(x,y) \wedge A(y,z) \wedge A(z,u)] \}$$

Nodes that have a great-grand-child: {1,2}

y≠u not necessary!
Contrast homomorphism
vs. isomorphism
("Hamiltonian Path")

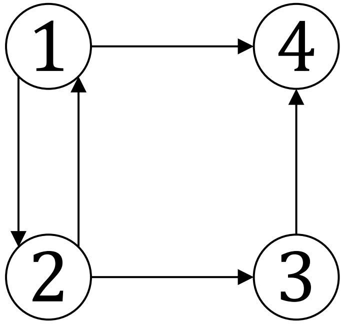
$$\{\, (x,y) \mid \forall z.[A(x,z) \rightarrow A(y,z)] \}$$

?

A encodes the directed
edges of a graph ("arcs")

Based on an example by Dan Suciu from CSE 554, 2011.

Example: Querying a Graph



A:

1	2
2	1
2	3
1	4
3	4

A encodes the directed edges of a graph ("arcs")

What do these queries return ?

$$\{ x \mid \exists y. A(x,y) \}$$

Nodes that have at least one child: {1,2,3}

$$\{ x \mid \exists y,z,u.[A(x,y) \wedge A(y,z) \wedge A(z,u)] \}$$

Nodes that have a great-grand-child: {1,2}

$\nexists z.[A(x,z) \wedge \neg A(y,z)]$

$$\{ (x,y) \mid \forall z.[A(x,z) \rightarrow A(y,z)] \}$$

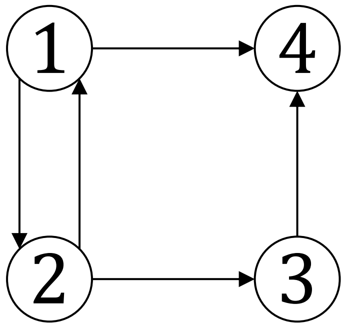
Every child of x is a child of y.

Which of the following tuples fulfill the condition? ?

(1,3) (3,1)

Based on an example by Dan Suciu from CSE 554, 2011.

Example: Querying a Graph



A:

1	2
2	1
2	3
1	4
3	4

A encodes the directed edges of a graph ("arcs")

What do these queries return ?

$$\{\, x \mid \exists y.\ A(x,y) \,\}$$

Nodes that have at least one child: {1,2,3}

$$\{\, x \mid \exists y,z,u.[A(x,y) \land A(y,z) \land A(z,u)] \}$$

Nodes that have a great-grand-child: {1,2}

$\nexists z.[A(x,z) \land \neg A(y,z)]$

$$\{\, (x,y) \mid \forall z.[A(x,z) \rightarrow A(y,z)] \}$$

Every child of x is a child of y.

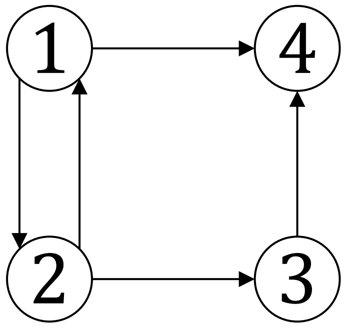
Which of the following tuples fulfill the condition?

~~(1,3)~~ (3,1)

{(1,1),(2,2),(3,1),(3,3),(4,1), (4,2), (4,3), (4,4)} if domain is set of nodes!

Based on an example by Dan Suciu from CSE 554, 2011.

Example: Querying a Graph



A:

1	2
2	1
2	3
1	4
3	4

A encodes the directed edges of a graph ("arcs")

What do these queries return ?

$$\{\, x \mid \exists y.\, A(x,y) \,\}$$

Nodes that have at least one child: {1,2,3}

$$\{\, x \mid \exists y,z,u.[A(x,y) \wedge A(y,z) \wedge A(z,u)] \}$$

Nodes that have a great-grand-child: {1,2}

$\nexists z.[A(x,z) \wedge \neg A(y,z)]$

$$\{\, (x,y) \mid \forall z.[A(x,z) \rightarrow A(y,z)] \}$$

Every child of x is a child of y.

Which of the following tuples fulfill the condition?

$\{\, (x,y) \mid N(x) \wedge N(y) \wedge \forall z.[A(x,z) \rightarrow A(y,z)] \}$

{(1,1),(2,2),(3,1),(3,3),(4,1), (4,2), (4,3), (4,4)}

if domain is set of nodes!

Based on an example by Dan Suciu from CSE 554, 2011.

The person/bar/drinks schema

Likes(person, drink)
Frequents(person, bar)
Serves(bar, drink)

331

What does the following query return?

$$\{\, x \mid \forall y.[\text{Frequents}(x,y) \rightarrow \exists z.[\text{Serves}(y,z) \wedge \text{Likes}(x,z)]] \,\}$$

?

Schema adapted from Jeff Ullman's drinkers/bars/beers example to avoid attributes with same first letters. https://dl.acm.org/doi/book/10.5555/42790

The person/bar/drinks schema

Likes(person, drink)
Frequents(person, bar)
Serves(bar, drink)

331

What does the following query return?

$$\{\, x \mid \forall y.[\text{Frequents}(x,y) \rightarrow \exists z.[\text{Serves}(y,z) \wedge \text{Likes}(x,z)]] \,\}$$

Find drinkers that frequent only bars
that serve some drink they like.

Is this query domain independent? ?

SQL example available at: https://github.com/northeastern-datalab/cs3200-activities/tree/master/sql
Schema adapted from Jeff Ullman's drinkers/bars/beers example to avoid attributes with same first letters. https://dl.acm.org/doi/book/10.5555/42790
Wolfgang Gatterbauer. Principles of scalable data management: https://northeastern-datalab.github.io/cs7240/

The person/bar/drinks schema

331

Likes(person, drink)
Frequents(person, bar)
Serves(bar, drink)

What does the following query return?

$$\{\, x \mid \forall y.[\text{Frequents}(x,y) \rightarrow \exists z.[\text{Serves}(y,z) \wedge \text{Likes}(x,z)]]\}$$

Find drinkers that frequent only bars
that serve some drink they like.

This query is not domain independent.
How to fix?

?

Its output would include all
values from the domain that do
not appear in the Frequents(x,_)

SQL example available at: https://github.com/northeastern-datalab/cs3200-activities/tree/master/sql
Schema adapted from Jeff Ullman's drinkers/bars/beers example to avoid attributes with same first letters. https://dl.acm.org/doi/book/10.5555/42790
Wolfgang Gatterbauer. Principles of scalable data management: https://northeastern-datalab.github.io/cs7240/

The person/bar/drinks schema

Likes(person, drink)
Frequents(person, bar)
Serves(bar, drink)

331

What does the following query return?

Frequents(x,_) ∧ ...
Likes(x,_) ∧ ...

Are those two options to make it safe identical ?

$$\{ x \mid \forall y.[\text{Frequents}(x,y) \rightarrow \exists z.[\text{Serves}(y,z) \wedge \text{Likes}(x,z)]]\}$$

Find drinkers that frequent <u>only</u> bars that serve <u>some</u> drink they like.

SQL example available at: https://github.com/northeastern-datalab/cs3200-activities/tree/master/sql
Schema adapted from Jeff Ullman's drinkers/bars/beers example to avoid attributes with same first letters. https://dl.acm.org/doi/book/10.5555/42790

The person/bar/drinks schema

Likes(person, drink)
Frequents(person, bar)
Serves(bar, drink)

What does the following query return?

Frequents(x,_) ∧ ...
Likes(x,_) ∧ ...

Both safe, but not identical. Tip: Should a drinker who likes a drink but does not frequent any bar be returned?

$$\{ x \mid \forall y.[\text{Frequents}(x,y) \rightarrow \exists z.[\text{Serves}(y,z) \wedge \text{Likes}(x,z)]] \}$$

Find drinkers that frequent only bars that serve some drink they like.

Challenge: write this query without the ∀ quantifier!
And then in SQL

?

SQL example available at: https://github.com/northeastern-datalab/cs3200-activities/tree/master/sql
Schema adapted from Jeff Ullman's drinkers/bars/beers example to avoid attributes with same first letters. https://dl.acm.org/doi/book/10.5555/42790

The person/bar/drinks example

Likes(person, drink)
Frequents(person, bar)
Serves(bar, drink)

331

Challenge: write these in SQL.
Solutions at: https://demo.queryvis.com

Find persons that frequent some bar that serves some drink they like.

?

Find persons that frequent only bars that serve some drink they like.

$$\{ x \mid \exists w.[\text{Likes}(x,w) \wedge \forall y.[\text{Frequents}(x,y) \rightarrow \exists z.[\text{Serves}(y,z) \wedge \text{Likes}(x,z)]]]\}$$

Find persons that frequent some bar that serves only drinks they like.

?

Find persons that frequent only bars that serve only drinks they like.
(= Find persons who like all drinks that are served in all the bars they visit.)
(= Find persons for which there does not exist a bar they frequent that serves a drink they do not like.)

?

SQL example available at: https://github.com/northeastern-datalab/cs3200-activities/tree/master/sql
Schema adapted from Jeff Ullman's drinkers/bars/beers example to avoid attributes with same first letters. https://dl.acm.org/doi/book/10.5555/42790