Rewriting with Acyclic Queries: Mind Your Head

Gaetano Geck Jens Keppeler Thomas Schwentick Christopher Spinrath

TU Dortmund University

25th ICDT March 30, 2022

Rewritings

R

1	2
2	3

S

2	1
3	1

T

1	4
2	5

relational database

Rewritings

Query $H(x, w) \leftarrow R(x, y),\ S(y, z),\ T(z, w)$

R	
1	2
2	3

S	
2	1
3	1

T	
1	4
2	5

relational database

Rewritings

Query $H(x, w) \leftarrow R(x, y),\ S(y, z),\ T(z, w)$

no direct access

R	
1	2
2	3

S	
2	1
3	1

T	
1	4
2	5

relational database

Rewritings

Query $H(x, w) \leftarrow R(x, y),\ S(y, z),\ T(z, w)$

no direct access

View
$V_1(x, z) \leftarrow R(x, y), S(y, z)$

View
$V_2(z, w) \leftarrow S(y, z), T(z, w)$

R

1	2
2	3

S

2	1
3	1

T

1	4
2	5

relational database

Rewritings

Query $H(x, w) \leftarrow R(x, y),\ S(y, z),\ T(z, w)$

no direct access

V_1	
1	1
2	1

View
$V_1(x, z) \leftarrow R(x, y), S(y, z)$

V_2	
1	4

View
$V_2(z, w) \leftarrow S(y, z), T(z, w)$

relational database

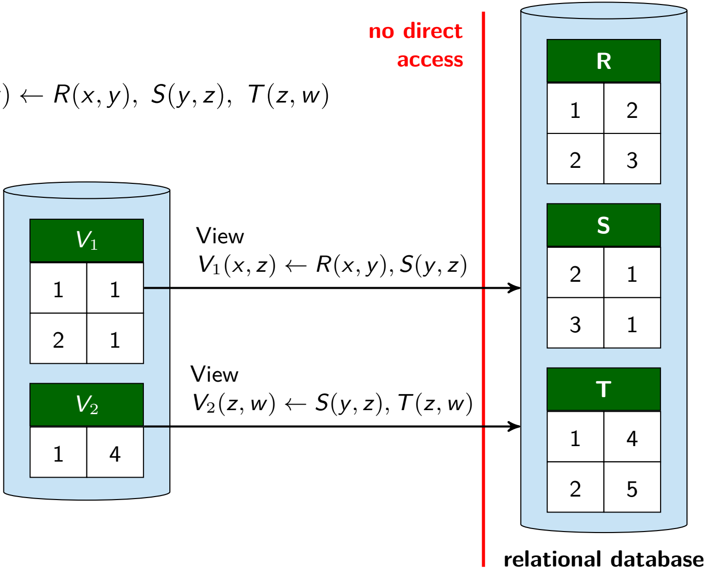
R	
1	2
2	3

S	
2	1
3	1

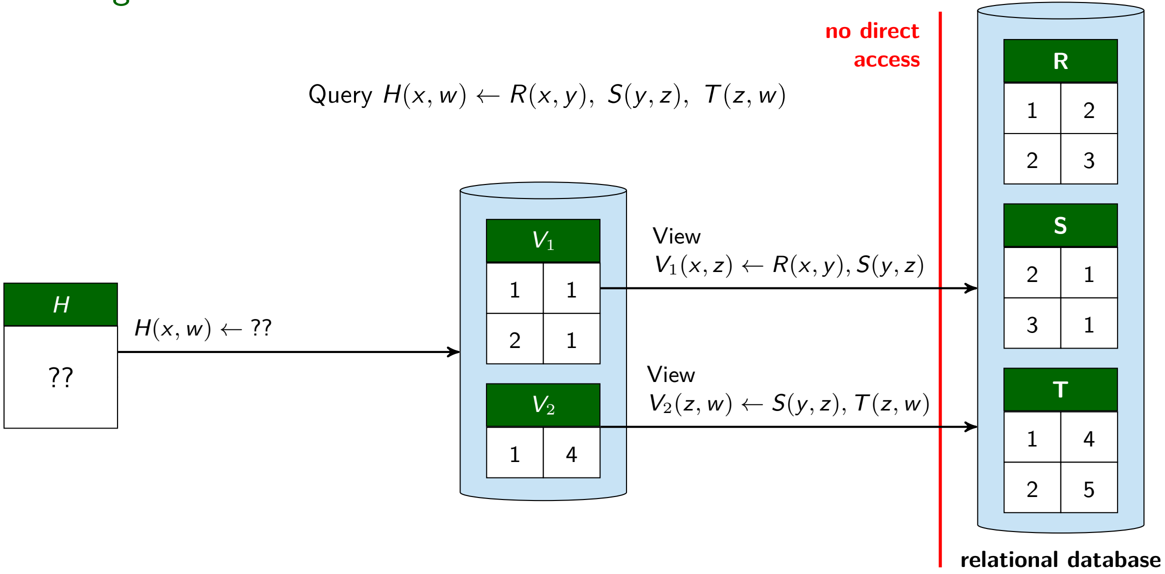
T	
1	4
2	5

Rewritings

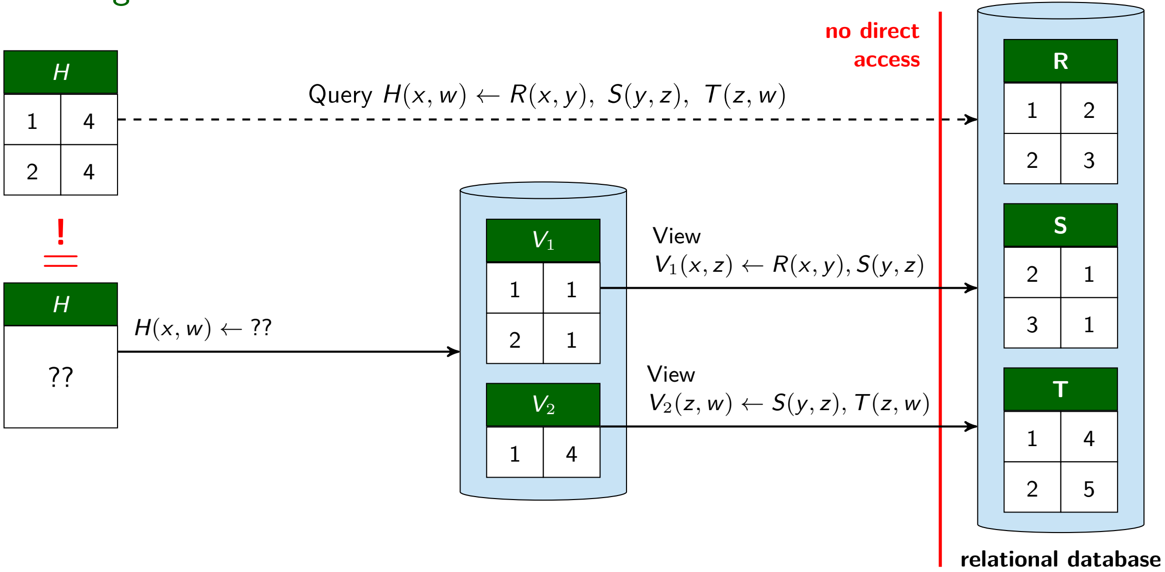
Query $H(x, w) \leftarrow R(x, y),\ S(y, z),\ T(z, w)$



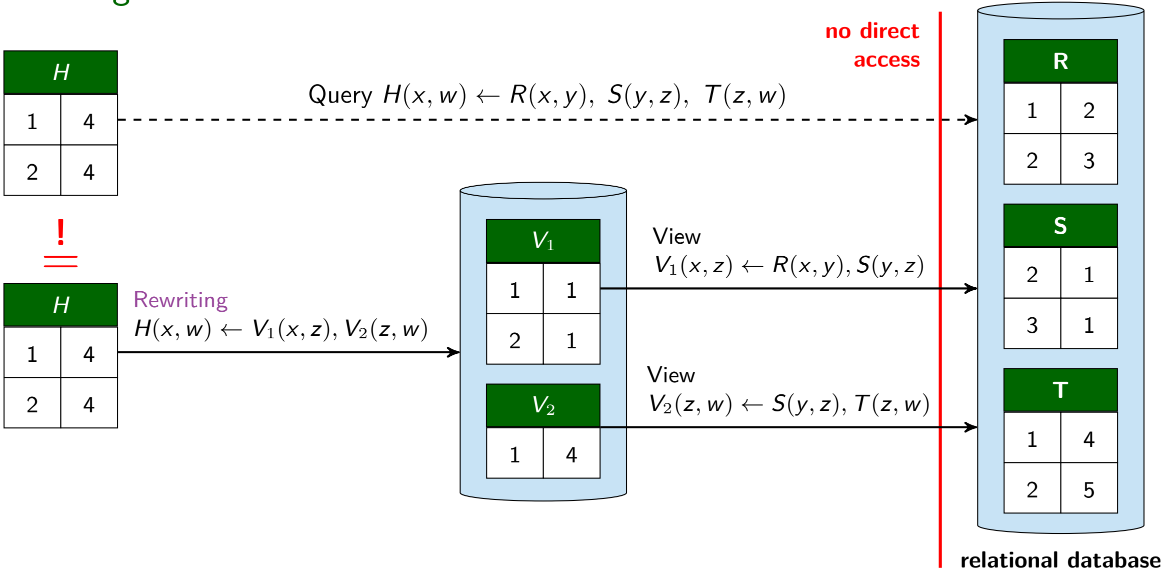
Rewritings


no direct access
Query $H(x,w) \leftarrow R(x,y),\ S(y,z),\ T(z,w)$
R
1 2
2 3
S
2 1
3 1
T
1 4
2 5
relational database
View
$V_1(x,z) \leftarrow R(x,y), S(y,z)$
View
$V_2(z,w) \leftarrow S(y,z), T(z,w)$
V_1
1 1
2 1
V_2
1 4
H
??
$H(x,w) \leftarrow$??

Rewritings


H
1 4
2 4
!
=
H
??
H(x, w) ← ??
Query H(x, w) ← R(x, y), S(y, z), T(z, w)
no direct access
V1
1 1
2 1
V2
1 4
View
V1(x, z) ← R(x, y), S(y, z)
View
V2(z, w) ← S(y, z), T(z, w)
R
1 2
2 3
S
2 1
3 1
T
1 4
2 5
relational database

Rewritings


no direct access
H
1 4
2 4
Query $H(x, w) \leftarrow R(x, y),\ S(y, z),\ T(z, w)$
R
1 2
2 3
!
=
H
1 4
2 4
Rewriting
$H(x, w) \leftarrow V_1(x, z), V_2(z, w)$
V_1
1 1
2 1
V_2
1 4
View
$V_1(x, z) \leftarrow R(x, y), S(y, z)$
S
2 1
3 1
View
$V_2(z, w) \leftarrow S(y, z), T(z, w)$
T
1 4
2 5
relational database

Rewritings for Conjunctive Queries

Conjunctive Queries are of the form

$$\underbrace{H(x,w)}_{\text{head}} \leftarrow \underbrace{R(x,y), \overbrace{S(y,z)}^{\text{atom}}, T(z,w)}_{\text{body}}$$

Rewritings for Conjunctive Queries

Conjunctive Queries are of the form

$$\underbrace{H(x,w)}_{\text{head}} \leftarrow \underbrace{R(x,y), \overbrace{S(y,z)}^{\text{atom}}, T(z,w)}_{\text{body}}$$

Views are just conjunctive queries with a special role

global-as-view,
exact, positive,
materialised or not

Rewritings for Conjunctive Queries

Conjunctive Queries are of the form

$$\underbrace{H(x,w)}_{\text{head}} \leftarrow \underbrace{R(x,y), \overbrace{S(y,z)}^{\text{atom}}, T(z,w)}_{\text{body}}$$

Views are just conjunctive queries with a special role

global-as-view,
exact, positive,
materialised or not

A Rewriting

- for a query Q
- with respect to a set $\mathcal{V}$ of views

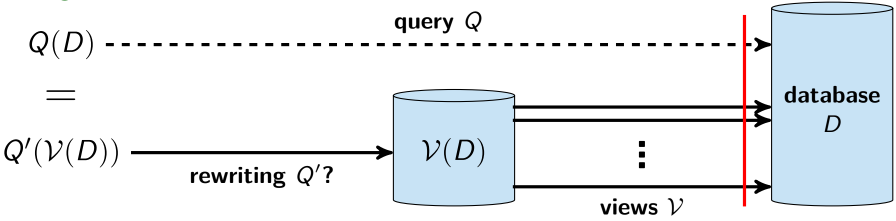
is a query Q' such that

$$Q'(\mathcal{V}(D)) = Q(D)$$

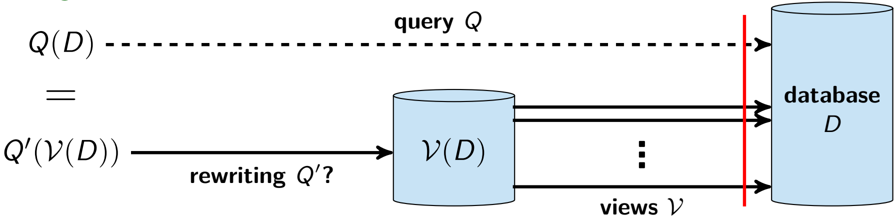
holds for all databases D

exact/equivalent
rewriting

The Rewriting Problem



The Rewriting Problem



The Rewriting Problem

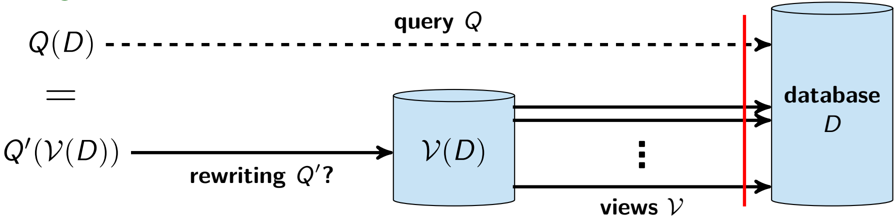
Input:

- conjunctive query Q
- set $\mathcal{V}$ of views

Question:

Is there a rewriting for Q with respect to $\mathcal{V}$?

The Rewriting Problem



The Rewriting Problem

Input:

- conjunctive query Q
- set $\mathcal{V}$ of views

Question:

Is there a rewriting for Q with respect to $\mathcal{V}$?

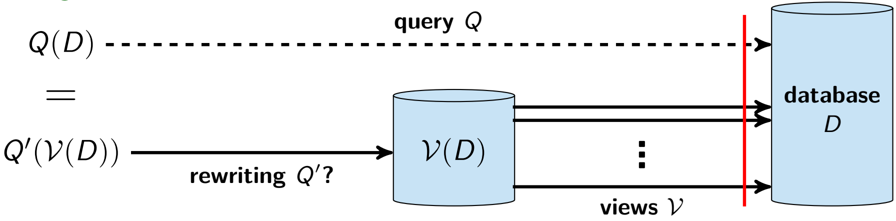
Theorem (Levy, Mendelzon, Sagiv, and Srivastava 1995)

The rewriting problem for

- *conjunctive queries and*
- *views defined by conjunctive queries*

is NP-complete.

The Rewriting Problem



The Rewriting Problem

Input:

- conjunctive query Q
- set $\mathcal{V}$ of views

Question:

Is there a rewriting for Q with respect to $\mathcal{V}$?

Theorem (Levy, Mendelzon, Sagiv, and Srivastava 1995)

The rewriting problem for

- *conjunctive queries and*
- *views defined by conjunctive queries*

is NP-complete.

→ Restrict everything to structurally simple queries

Acyclic Conjunctive Queries

For acyclic queries many problems are in polynomial time: containment, evaluation, ...

Acyclic Conjunctive Queries

For acyclic queries many problems are in polynomial time: containment, evaluation, ...

Definition

A conjunctive query is acyclic if it has a join tree

Acyclic Conjunctive Queries

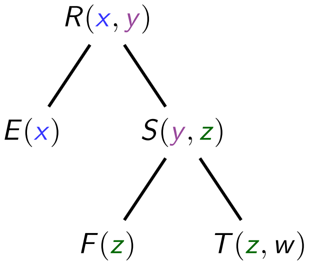
For acyclic queries many problems are in polynomial time: containment, evaluation, ...

Definition

A conjunctive query is acyclic if it has a join tree

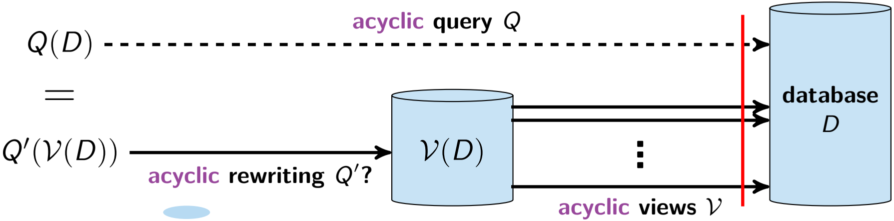
Example

$H(x, y) \leftarrow R(x, y), S(y, z), F(z), E(x), T(z, w)$ is acyclic



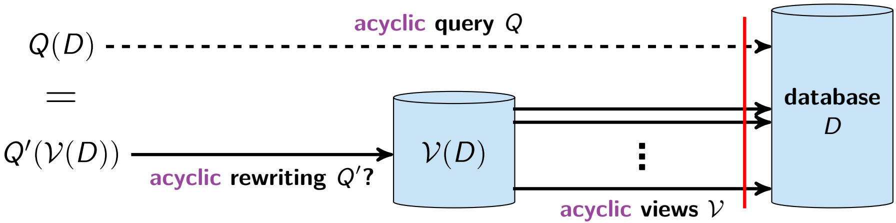
For every variable:
the induced subgraph is connected

Complexity of the Acyclic Rewriting Problem



If the query is acyclic, we would like the rewriting to be acyclic as well

Complexity of the Acyclic Rewriting Problem



Theorem

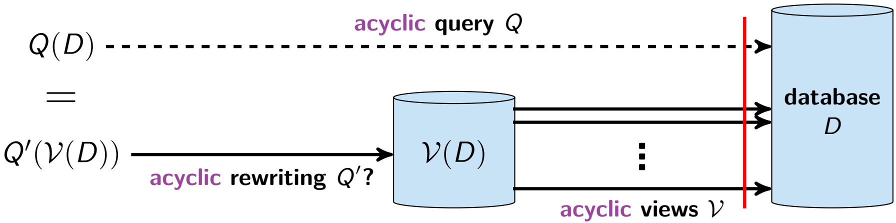
The acyclic rewriting problem for

- *acyclic queries and*
- *views defined by acyclic queries*

is NP-complete.

- Even if only a single view is given

Complexity of the Acyclic Rewriting Problem



Theorem

The acyclic rewriting problem for

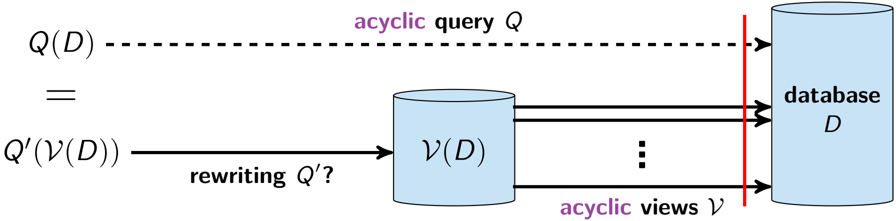
- *acyclic queries and*
- *views defined by acyclic queries*

is NP-complete.

Proof Approach

- in NP: guess and verify
- NP-hardness: reduction from 3SAT

Complexity of the Acyclic Rewriting Problem



Theorem

The rewriting problem for

- *acyclic queries and*
- *views defined by acyclic queries*

is NP-complete.

Acyclic Rewritings

Theorem

For every

- *acyclic query and*
- *views defined by conjunctive queries*

the following holds: If there is a rewriting, then there is an acyclic rewriting.

Acyclic Rewritings – Example

Acyclic Query

$$H(x, y) \leftarrow R(x, y), S(y, z), F(z), E(x), T(z, w)$$

Views

$$V_1(x, y) \leftarrow R(x, y), S(y, z)$$

$$V_2(y, z) \leftarrow S(y, z), F(z)$$

$$V_3(z, x) \leftarrow E(x), T(z, w)$$

Acyclic Rewritings – Example

Acyclic Query

$H(x, y) \leftarrow R(x, y), S(y, z), F(z), E(x), T(z, w)$

Views

$V_1(x, y) \leftarrow R(x, y), S(y, z)$

$V_2(y, z) \leftarrow S(y, z), F(z)$

$V_3(z, x) \leftarrow E(x), T(z, w)$

Rewriting

$H(x, y) \leftarrow V_1(x, y), V_2(y, z), V_3(z, x)$

Acyclic Rewritings – Example

Acyclic Query

$H(x, y) \leftarrow R(x, y), S(y, z), F(z), E(x), T(z, w)$

Views

$V_1(x, y) \leftarrow R(x, y), S(y, z)$

$V_2(y, z) \leftarrow S(y, z), F(z)$

$V_3(z, x) \leftarrow E(x), T(z, w)$

Rewriting

$H(x, y) \leftarrow V_1(x, y), V_2(y, z), V_3(z, x)$

This rewriting

- is cyclic
- minimal
  (no atom can be removed)

Acyclic Rewritings – Example

Acyclic Query

$H(x, y) \leftarrow R(x, y), S(y, z), F(z), E(x), T(z, w)$

Views

$V_1(x, y) \leftarrow R(x, y), S(y, z)$

$V_2(y, z) \leftarrow S(y, z), F(z)$

$V_3(z, x) \leftarrow E(x), T(z, w)$

Rewriting

$H(x, y) \leftarrow V_1(x, y), V_2(y, z), V_3(z, x)$

This rewriting

- is cyclic
- minimal
  (no atom can be removed)

Acyclic Rewriting

$H(x, y) \leftarrow V_1(x, y), V_2(y, z), V_3(z', x), V_3(z, x')$

Acyclic Rewritings

Theorem

For every

- *acyclic query and*
- *views defined by conjunctive queries*

the following holds: If there is a rewriting, then there is an acyclic rewriting.

Acyclic Rewritings

Theorem

For every

- *acyclic query and*
- *views defined by conjunctive queries*

the following holds: If there is a rewriting, then there is an acyclic rewriting.

Proof ingredients

1. Characterisation: There is a rewriting if and only if there is a consistent cover partition (similar to other characterisations in the literature)
2. Refinement of the partition along a join tree

Proof Ingredient: Characterisation

Query

$H(x, y) \leftarrow R(x, y), S(y, z), F(z), E(x), T(z, w)$

Views

$V_1(x, y) \leftarrow R(x, y), S(y, z)$

$V_2(y, z) \leftarrow S(y, z), F(z)$

$V_3(z, x) \leftarrow E(x), T(z, w)$

Rewriting

$H(x, y) \leftarrow V_1(x, y), \quad V_2(y, z), \qquad V_3(z, x)$

Proof Ingredient: Characterisation

Query

$$H(x,y) \leftarrow \underbrace{R(x,y)}_{A_1}, \underbrace{S(y,z), F(z)}_{A_2}, \underbrace{E(x),\ T(z,w)}_{A_3}$$

partition A_1 A_2 A_3

Views

$$V_1(x,y) \leftarrow R(x,y), S(y,z)$$
$$V_2(y,z) \leftarrow S(y,z), F(z)$$
$$V_3(z,x) \leftarrow E(x), T(z,w)$$

Rewriting

$$H(x,y) \leftarrow V_1(x,y), \quad V_2(y,z), \quad V_3(z,x)$$

Proof Ingredient: Characterisation

Query

$$H(x,y) \leftarrow \underbrace{R(x,y)}_{A_1}, \underbrace{S(y,z), F(z)}_{A_2}, \underbrace{E(x),\ T(z,w)}_{A_3}$$

partition A_1 A_2 A_3

$A_1 \leftrightarrow V_1(x,y)$, $A_2 \leftrightarrow V_2(y,z)$, $A_3 \leftrightarrow V_3(z,x)$

Rewriting

$$H(x,y) \leftarrow V_1(x,y),\quad V_2(y,z),\quad V_3(z,x)$$

Views

$$V_1(x,y) \leftarrow R(x,y), S(y,z)$$

$$V_2(y,z) \leftarrow S(y,z), F(z)$$

$$V_3(z,x) \leftarrow E(x), T(z,w)$$

Proof Ingredient: Refinement

Query

$H(x, y) \leftarrow R(x, y), S(y, z), F(z), E(x), T(z, w)$

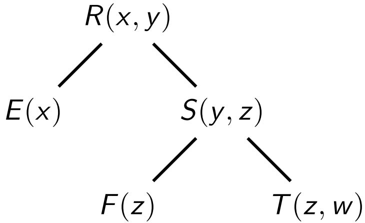
Views

$V_1(x, y) \leftarrow R(x, y), S(y, z)$

$V_2(y, z) \leftarrow S(y, z), F(z)$

$V_3(z, x) \leftarrow E(x), T(z, w)$

Join tree for the query



Rewriting

$H(x, y) \leftarrow V_1(x, y), V_2(y, z), V_3(z, x)$

Proof Ingredient: Refinement

Query

$H(x,y) \leftarrow R(x,y), S(y,z), F(z), E(x), T(z,w)$

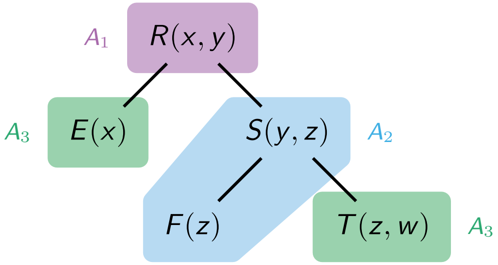
Views

$V_1(x,y) \leftarrow R(x,y), S(y,z)$

$V_2(y,z) \leftarrow S(y,z), F(z)$

$V_3(z,x) \leftarrow E(x), T(z,w)$

Join tree for the query



Rewriting

$H(x,y) \leftarrow V_1(x,y), V_2(y,z), V_3(z,x)$

Proof Ingredient: Refinement

Query

$H(x,y) \leftarrow R(x,y), S(y,z), F(z), E(x), T(z,w)$

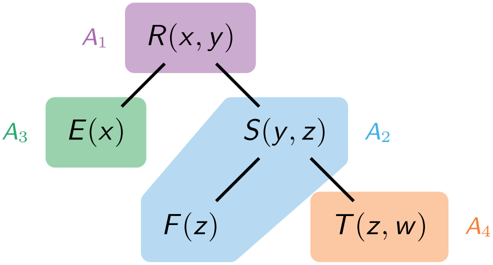
Views

$V_1(x,y) \leftarrow R(x,y), S(y,z)$

$V_2(y,z) \leftarrow S(y,z), F(z)$

$V_3(z,x) \leftarrow E(x), T(z,w)$

Join tree for the query



Rewriting

$H(x,y) \leftarrow V_1(x,y), V_2(y,z), V_3(z,x)$

Proof Ingredient: Refinement

Query

$H(x,y) \leftarrow R(x,y), S(y,z), F(z), E(x), T(z,w)$

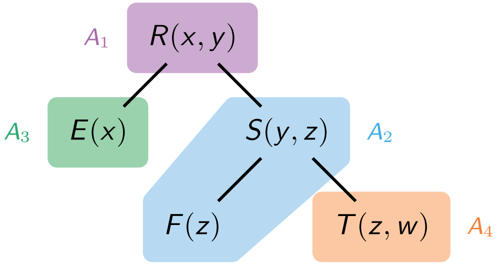
Views

$V_1(x,y) \leftarrow R(x,y), S(y,z)$

$V_2(y,z) \leftarrow S(y,z), F(z)$

$V_3(z,x) \leftarrow E(x), T(z,w)$

Join tree for the query



Rewriting

$H(x,y) \leftarrow V_1(x,y), V_2(y,z), V_3(z,x) \rightsquigarrow H(x,y) \leftarrow V_1(x,y), V_2(y,z), V_3(z,x), V_3(z,x)$

Proof Ingredient: Refinement

Query

$H(x,y) \leftarrow R(x,y), S(y,z), F(z), E(x), T(z,w)$

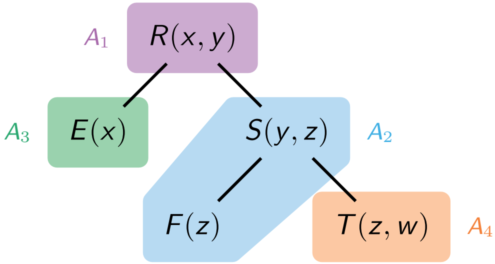
Views

$V_1(x,y) \leftarrow R(x,y), S(y,z)$

$V_2(y,z) \leftarrow S(y,z), F(z)$

$V_3(z,x) \leftarrow E(x), T(z,w)$

Join tree for the query



Rewriting

$H(x,y) \leftarrow V_1(x,y), V_2(y,z), V_3(z,x) \rightsquigarrow H(x,y) \leftarrow V_1(x,y), V_2(y,z), V_3(\mathbf{z}',x), V_3(z,\mathbf{x}')$

Proof Ingredient: Refinement

Query

$H(x, y) \leftarrow R(x, y), S(y, z), F(z), E(x), T(z, w)$

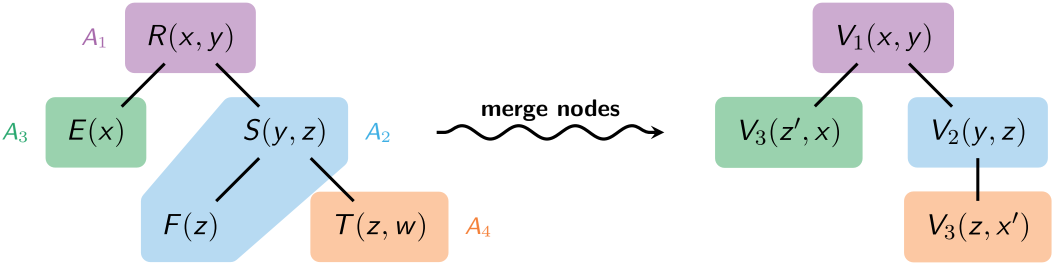
Views

$V_1(x, y) \leftarrow R(x, y), S(y, z)$

$V_2(y, z) \leftarrow S(y, z), F(z)$

$V_3(z, x) \leftarrow E(x), T(z, w)$

Join tree for the query



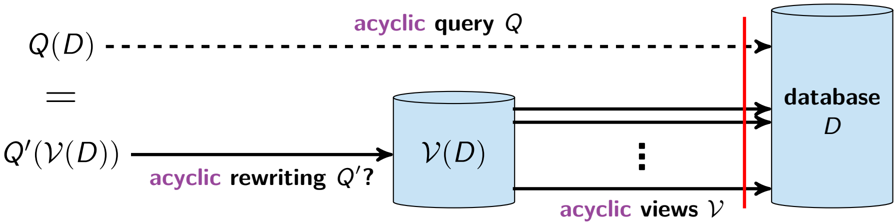
Rewriting

$H(x, y) \leftarrow V_1(x, y), V_2(y, z), V_3(z, x)$

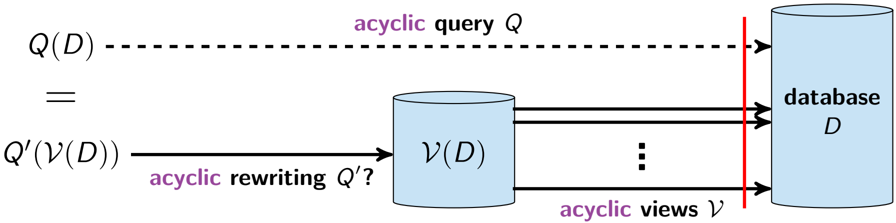
Acyclic Rewriting

$H(x, y) \leftarrow V_1(x, y), V_2(y, z), V_3(\mathbf{z}', x), V_3(z, \mathbf{x}')$

Complexity Results – The Good News

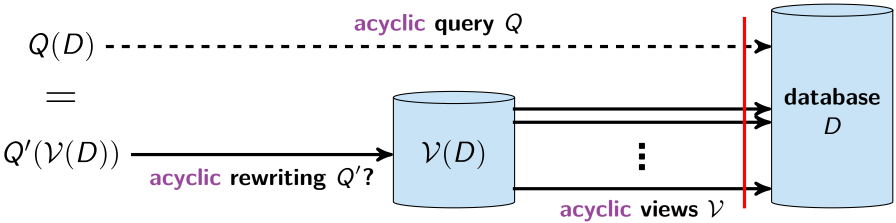


Complexity Results – The Good News



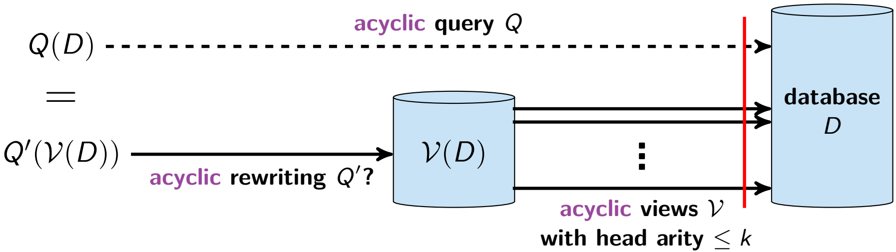
For tractability: Mind your head!

Complexity Results – The Good News



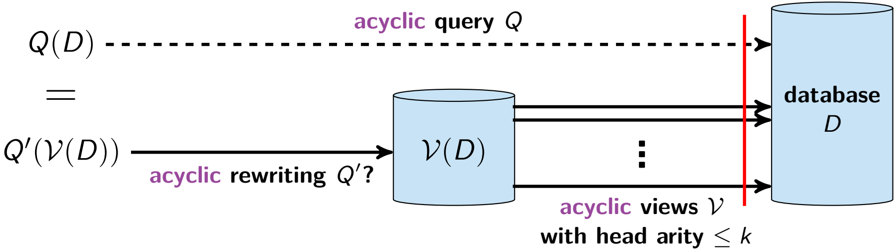
For tractability: Mind the views' heads!

Complexity Results – The Good News



For tractability: Mind the views' heads!

Complexity Results – The Good News



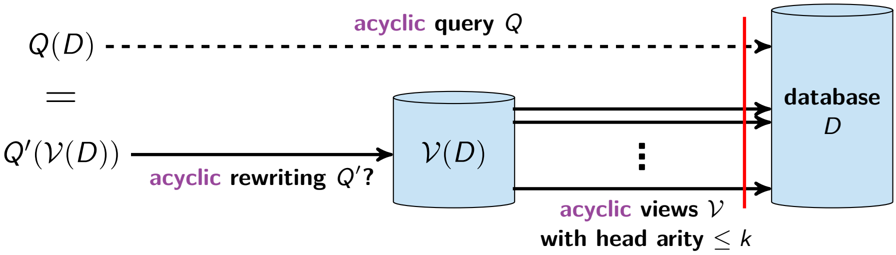
Theorem

For every $k \geq 0$ the acyclic rewriting problem for

- *acyclic queries and*
- *acyclic views with heads of arity at most k*

is in polynomial time.

Complexity Results – The Good News



Theorem

For every $k \geq 0$ the acyclic rewriting problem for

- *acyclic queries and*
- *acyclic views with heads of arity at most k*

is in polynomial time.

- If an acyclic rewriting exists, it can be computed in polynomial time

Acyclic Views with Bounded Head Arity

Proposition (Nash, Segoufin, and Vianu 2010)

For every

- *conjunctive query and*
- *views defined by conjunctive queries*

there is a rewriting, if and only if the canonical candidate is a rewriting.

Acyclic Views with Bounded Head Arity

Proposition (Nash, Segoufin, and Vianu 2010)

For every

- *conjunctive query and*
- *views defined by conjunctive queries*

there is a rewriting, if and only if the canonical candidate is a rewriting.

Algorithm

1. Compute the canonical candidate

Acyclic Views with Bounded Head Arity

Proposition (Nash, Segoufin, and Vianu 2010)

For every

- *conjunctive query and*
- *views defined by conjunctive queries*

there is a rewriting, if and only if the canonical candidate is a rewriting.

Algorithm

1. Compute the canonical candidate
2. Test whether the canonical candidate is a rewriting

Acyclic Views with Bounded Head Arity

Proposition (Nash, Segoufin, and Vianu 2010)

For every

- *conjunctive query and*
- *views defined by conjunctive queries*

there is a rewriting, if and only if the canonical candidate is a rewriting.

Algorithm

1. Compute the canonical candidate
2. Test whether the canonical candidate is a rewriting
3. Transform the canonical candidate into an acyclic rewriting

Acyclic Views with Bounded Head Arity

Proposition (Nash, Segoufin, and Vianu 2010)

For every

- *conjunctive query and*
- *views defined by conjunctive queries*

there is a rewriting, if and only if the canonical candidate is a rewriting.

Algorithm

1. Compute the canonical candidate
2. Test whether the canonical candidate is a rewriting
   - in polynomial time for acyclic queries
3. Transform the canonical candidate into an acyclic rewriting

Acyclic Views with Bounded Head Arity

Proposition (Nash, Segoufin, and Vianu 2010)

For every

- *conjunctive query and*
- *views defined by conjunctive queries*

there is a rewriting, if and only if the canonical candidate is a rewriting.

Algorithm

1. Compute the canonical candidate
   - can be of exponential size for acyclic views
   - is of polynomial size for acyclic views with bounded head arity
2. Test whether the canonical candidate is a rewriting
   - in polynomial time for acyclic queries
3. Transform the canonical candidate into an acyclic rewriting

The Canonical Candidate

Query

$H(x, y) \leftarrow R(x, y), S(y, z), F(z), E(x), T(z, w)$

Views

$V_1(x, y) \leftarrow R(x, y), S(y, z)$

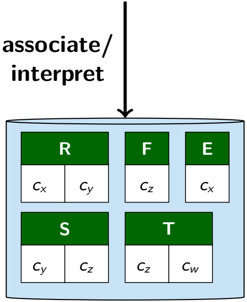
$V_2(y, z) \leftarrow S(y, z), F(z)$

$V_3(z, x) \leftarrow E(x), T(z, w)$

The Canonical Candidate

Query

$$H(x, y) \leftarrow R(x, y), S(y, z), F(z), E(x), T(z, w)$$



Views

$$V_1(x, y) \leftarrow R(x, y), S(y, z)$$

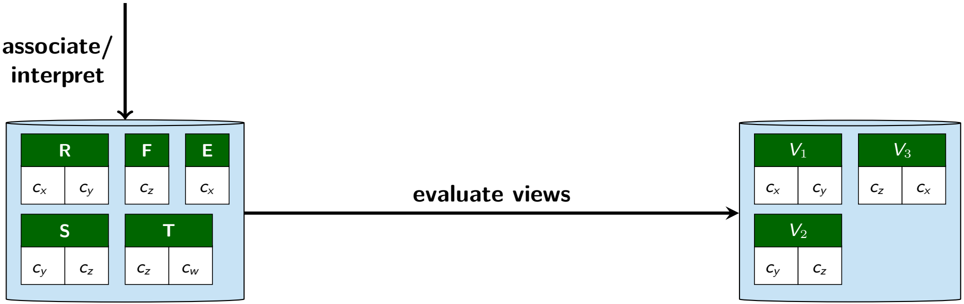
$$V_2(y, z) \leftarrow S(y, z), F(z)$$

$$V_3(z, x) \leftarrow E(x), T(z, w)$$

The Canonical Candidate

Query

$H(x, y) \leftarrow R(x, y), S(y, z), F(z), E(x), T(z, w)$



Views

$V_1(x, y) \leftarrow R(x, y), S(y, z)$

$V_2(y, z) \leftarrow S(y, z), F(z)$

$V_3(z, x) \leftarrow E(x), T(z, w)$

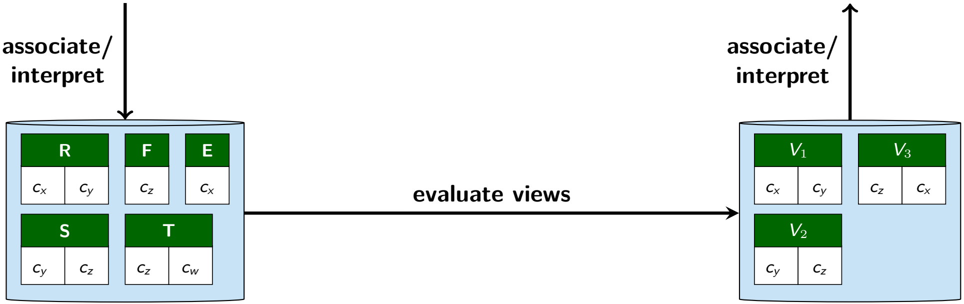
The Canonical Candidate

Query

$H(x, y) \leftarrow R(x, y), S(y, z), F(z), E(x), T(z, w)$

Canonical Candidate/Rewriting

$H(x, y) \leftarrow V_1(x, y), V_2(y, z), V_3(z, x)$



Views

$V_1(x, y) \leftarrow R(x, y), S(y, z)$

$V_2(y, z) \leftarrow S(y, z), F(z)$

$V_3(z, x) \leftarrow E(x), T(z, w)$

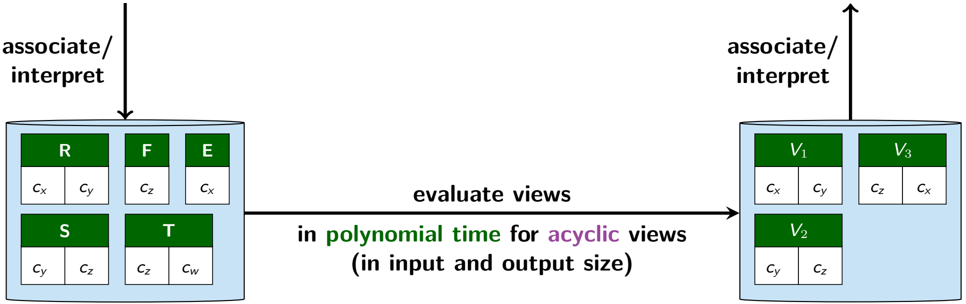
The Canonical Candidate

Query

$H(x,y) \leftarrow R(x,y), S(y,z), F(z), E(x), T(z,w)$

Canonical Candidate/Rewriting

$H(x,y) \leftarrow V_1(x,y), V_2(y,z), V_3(z,x)$



Views

$V_1(x,y) \leftarrow R(x,y), S(y,z)$

$V_2(y,z) \leftarrow S(y,z), F(z)$

$V_3(z,x) \leftarrow E(x), T(z,w)$

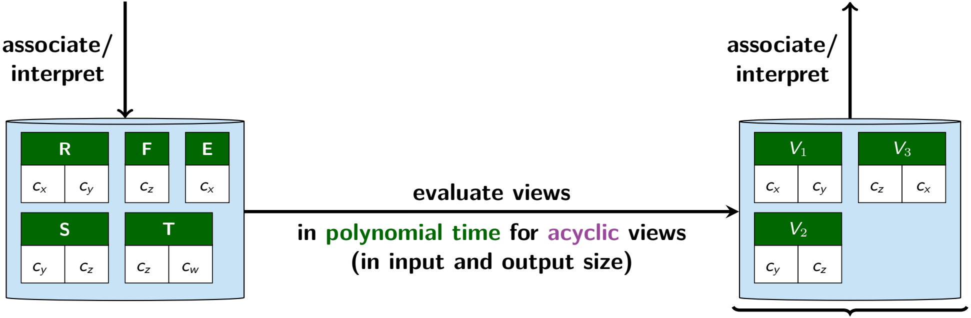
The Canonical Candidate

Query

$H(x,y) \leftarrow R(x,y), S(y,z), F(z), E(x), T(z,w)$

Canonical Candidate/Rewriting

$H(x,y) \leftarrow V_1(x,y), V_2(y,z), V_3(z,x)$



can be of exponential size
in the arity of the views

Views

$V_1(x,y) \leftarrow R(x,y), S(y,z)$

$V_2(y,z) \leftarrow S(y,z), F(z)$

$V_3(z,x) \leftarrow E(x), T(z,w)$

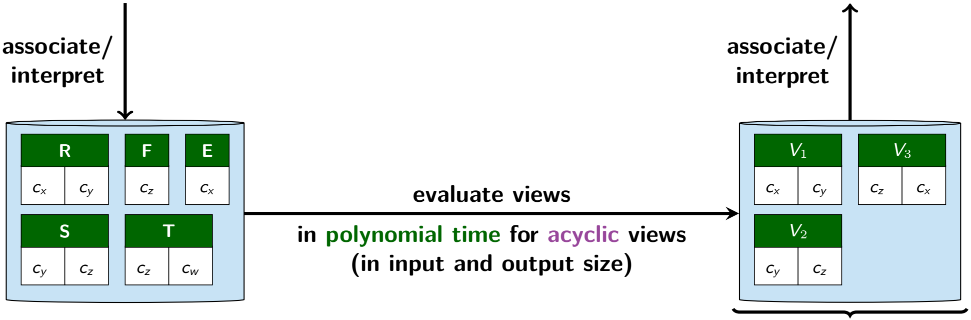
The Canonical Candidate

Query

$$H(x, y) \leftarrow R(x, y), S(y, z), F(z), E(x), T(z, w)$$

Canonical Candidate/Rewriting

$$H(x, y) \leftarrow V_1(x, y), V_2(y, z), V_3(z, x)$$



is of **polynomial size** for views with **bounded head arity**

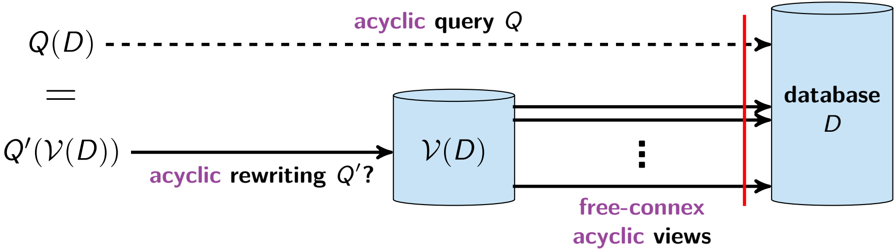
Views

$$V_1(x, y) \leftarrow R(x, y), S(y, z)$$

$$V_2(y, z) \leftarrow S(y, z), F(z)$$

$$V_3(z, x) \leftarrow E(x), T(z, w)$$

Complexity Results – The Good News



Free-Connex Acyclic Views

A conjunctive query is free-connex acyclic if

- it is acyclic and
- there is a join tree which includes the query's head atom

Free-Connex Acyclic Views

A conjunctive query is free-connex acyclic if

- it is acyclic and
- there is a join tree which includes the query's head atom

Example

$V(y, z, w) \leftarrow R(x, y), S(y, z), T(z, w), E(w)$ is free-connex acyclic

Free-Connex Acyclic Views

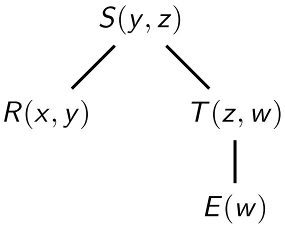
A conjunctive query is free-connex acyclic if

- it is acyclic and
- there is a join tree which includes the query's head atom

Example

$V(y,z,w) \leftarrow R(x,y), S(y,z), T(z,w), E(w)$ is free-connex acyclic

Join tree



Free-Connex Acyclic Views

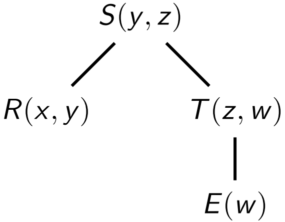
A conjunctive query is free-connex acyclic if

- it is acyclic and
- there is a join tree which includes the query's head atom

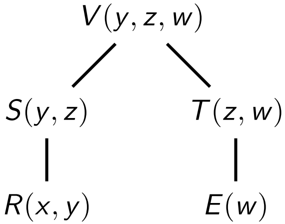
Example

$V(y,z,w) \leftarrow R(x,y), S(y,z), T(z,w), E(w)$ is free-connex acyclic

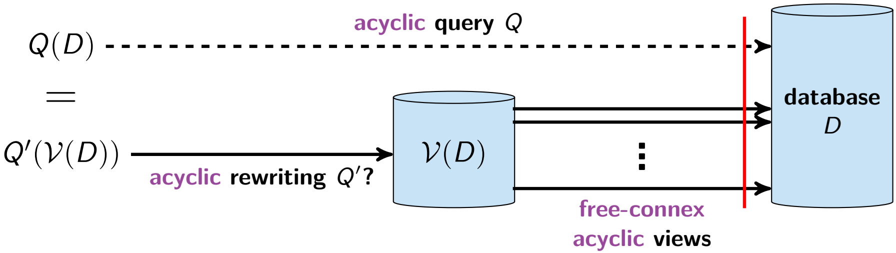
Join tree



Join tree (with head atom)



Free-Connex Acyclic Views



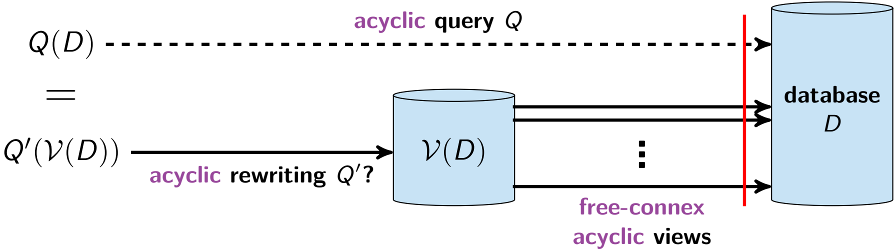
Theorem

For every $k \geq 0$ the acyclic rewriting problem for

- *acyclic queries and*
- *free-connex acyclic views*
- *over a database schema of arity at most k*

is in polynomial time.

Free-Connex Acyclic Views



Theorem

For every $k \geq 0$ the acyclic rewriting problem for

- *acyclic queries and*
- *free-connex acyclic views*
- *over a database schema of arity at most k*

is in polynomial time.

- If an acyclic rewriting exists, it can be computed in polynomial time

Free-Connex Acyclic Views

View

$V(y, z, w) \leftarrow R(x, y), S(y, z), T(z, w), E(w)$

Database relations have arity at most $k = 2$

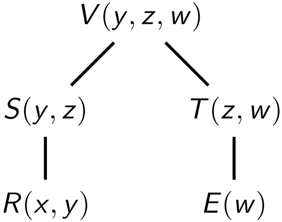
Free-Connex Acyclic Views

View

$V(y, z, w) \leftarrow R(x, y), S(y, z), T(z, w), E(w)$

Database relations have arity at most $k = 2$

Join tree (with head atom)



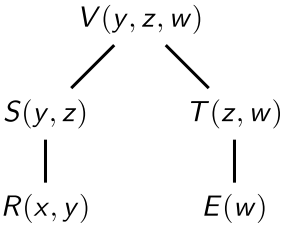
Free-Connex Acyclic Views

View

$V(y, z, w) \leftarrow R(x, y), S(y, z), T(z, w), E(w)$

Database relations have arity at most $k = 2$

Join tree (with head atom)



Algorithm

1. Root the tree at the head atom

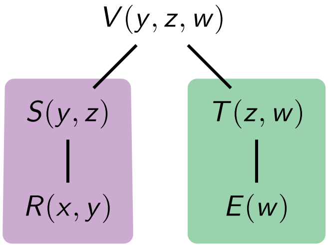
Free-Connex Acyclic Views

View

$V(y, z, w) \leftarrow R(x, y), S(y, z), T(z, w), E(w)$

Database relations have arity at most $k = 2$

Join tree (with head atom)



Algorithm

❶ Root the tree at the head atom

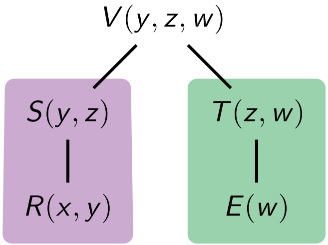
Free-Connex Acyclic Views

View

$V(y, z, w) \leftarrow R(x, y), S(y, z), T(z, w), E(w)$

Database relations have arity at most $k = 2$

Join tree (with head atom)



Algorithm

1. Root the tree at the head atom
2. For each subtree create a view with head arity $\leq k = 2$

$V_1(y, z) \leftarrow R(x, y), S(y, z), T(z, w), E(w)$

$V_2(z, w) \leftarrow R(x, y), S(y, z), T(z, w), E(w)$

$V(y, z, w)$ is "equivalent" to $V_1(y, z), V_2(z, w)$

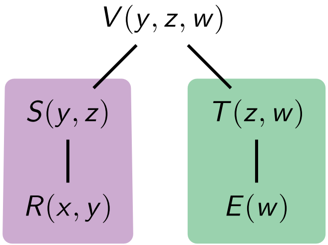
Free-Connex Acyclic Views

View

$V(y, z, w) \leftarrow R(x, y), S(y, z), T(z, w), E(w)$

Database relations have arity at most $k = 2$

Join tree (with head atom)



Algorithm

1. Root the tree at the head atom
2. For each subtree create a view with head arity $\leq k = 2$

   $V_1(y, z) \leftarrow R(x, y), S(y, z), T(z, w), E(w)$

   $V_2(z, w) \leftarrow R(x, y), S(y, z), T(z, w), E(w)$

   $V(y, z, w)$ is "equivalent" to $V_1(y, z), V_2(z, w)$
3. Compute a rewriting with views V_1, V_2 (or return "`false`" if none exists)

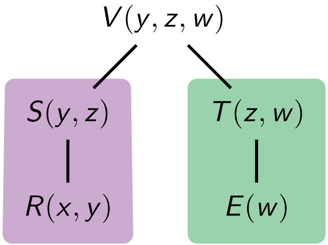
Free-Connex Acyclic Views

View

$V(y, z, w) \leftarrow R(x, y), S(y, z), T(z, w), E(w)$

Database relations have arity at most $k = 2$

Join tree (with head atom)



Algorithm

1. Root the tree at the head atom
2. For each subtree create a view with head arity $\leq k = 2$

   $V_1(y, z) \leftarrow R(x, y), S(y, z), T(z, w), E(w)$

   $V_2(z, w) \leftarrow R(x, y), S(y, z), T(z, w), E(w)$

   $V(y, z, w)$ is "equivalent" to $V_1(y, z), V_2(z, w)$
3. Compute a rewriting with views V_1, V_2 (or return "`false`" if none exists)
4. Replace every atom $V_1(u, v)$ with an atom $V(u, v, w')$ where w' is a fresh variable

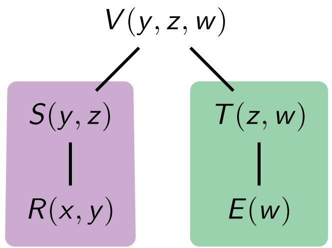
Free-Connex Acyclic Views

View

$V(y,z,w) \leftarrow R(x,y), S(y,z), T(z,w), E(w)$

Database relations have arity at most $k = 2$

Join tree (with head atom)



Algorithm

1. Root the tree at the head atom
2. For each subtree create a view with head arity $\leq k = 2$

   $V_1(y,z) \leftarrow R(x,y), S(y,z), T(z,w), E(w)$

   $V_2(z,w) \leftarrow R(x,y), S(y,z), T(z,w), E(w)$

   $V(y,z,w)$ is "equivalent" to $V_1(y,z), V_2(z,w)$
3. Compute a rewriting with views V_1, V_2 (or return "`false`" if none exists)
4. Replace every atom $V_1(u,v)$ with an atom $V(u,v,w')$ where w' is a fresh variable
5. Analogously for atoms $V_2(u,v)$

Conclusion

Summary

- If there is any rewriting for an acyclic query, then there is an acyclic rewriting
- The acyclic rewriting problem is NP-complete for acyclic queries and acyclic views
- The acyclic rewriting problem is in polynomial time for acyclic queries and
  - acyclic views with bounded head arity
  - free-connex acyclic views if the arity of the database schema is bounded

Conclusion

Summary

- If there is any rewriting for an acyclic query, then there is an acyclic rewriting
- The acyclic rewriting problem is NP-complete for acyclic queries and acyclic views
- The acyclic rewriting problem is in polynomial time for acyclic queries and
  - acyclic views with bounded head arity
  - free-connex acyclic views if the arity of the database schema is bounded

Further Prospects

- The result for free-connex acyclic views can be generalised to acyclic views with bounded weak head arity
- Analogous results for hierarchical and q-hierarchical queries

Conclusion

Summary

- If there is any rewriting for an acyclic query, then there is an acyclic rewriting
- The acyclic rewriting problem is NP-complete for acyclic queries and acyclic views
- The acyclic rewriting problem is in polynomial time for acyclic queries and
  - acyclic views with bounded head arity
  - free-connex acyclic views if the arity of the database schema is bounded

Further Prospects

- The result for free-connex acyclic views can be generalised to acyclic views with bounded weak head arity
- Analogous results for hierarchical and q-hierarchical queries

Mind your head!

References

Levy, Alon Y., Alberto O. Mendelzon, Yehoshua Sagiv, and Divesh Srivastava (1995). "Answering Queries Using Views." In: Proceedings of the Fourteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Ed. by Mihalis Yannakakis and Serge Abiteboul. ACM Press, pp. 95–104.

Nash, Alan, Luc Segoufin, and Victor Vianu (2010). "Views and queries: Determinacy and rewriting." In: ACM Trans. Database Syst. 35.3, 21:1–21:41.