A data complexity and rewritability tetrachotomy of ontology-mediated queries with a covering axiom

O. Gerasimova[1] S. Kikot[2] M. Zakharyaschev[3]
A. Kurucz[4] V. Podolskii[1,5]

[1]National Research University Higher School of Economics, Moscow, Russia

[2]London Metropolitan University, U.K.

[3]Birkbeck, University of London, U.K.

[4]King's College, University of London, U.K.

[5]Steklov Mathematical Institute, Moscow, Russia

17th International Conference on Principles of Knowledge Representation and Reasoning, Rhodes, 2020

Ontology-mediated querying by example (dates back to Prolog?)

Fact Set (a collection of *facts*, or ground atoms, a.k.a. the data)

$\mathcal{D} = \{Male(John), Parent(John, Sasha), Person(Sasha), Parent(Sasha, Ann), Female(Ann)\}$

Ontology-mediated querying by example (dates back to Prolog?)

Fact Set (a collection of *facts*, or ground atoms, a.k.a. the data)

$\mathcal{D} = \{Male(John), Parent(John, Sasha), Person(Sasha), Parent(Sasha, Ann), Female(Ann)\}$

Ontology (a set of sentences from some fragment of μFO called *description logic*)

$\mathcal{O} = \{\forall x(Person(x) \rightarrow Male(x) \vee Female(x)\}$

Ontology-mediated querying by example (dates back to Prolog?)

Fact Set (a collection of *facts*, or ground atoms, a.k.a. the data)

$\mathcal{D} = \{Male(John), Parent(John, Sasha), Person(Sasha), Parent(Sasha, Ann), Female(Ann)\}$

Ontology (a set of sentences from some fragment of μFO called *description logic*)

$\mathcal{O} = \{\forall x(Person(x) \rightarrow Male(x) \vee Female(x)\}$

Query (CQ, i.e. existentially quantified conjunction of atoms)

$\boldsymbol{q} = \exists x \exists y\, Male(x) \wedge Parent(x, y) \wedge Female(y)$ (Is there a father and a daughter ?)

Ontology-mediated querying by example (dates back to Prolog?)

Fact Set (a collection of *facts*, or ground atoms, a.k.a. the data)

$\mathcal{D} = \{Male(John), Parent(John, Sasha), Person(Sasha), Parent(Sasha, Ann), Female(Ann)\}$

Ontology (a set of sentences from some fragment of μFO called *description logic*)

$\mathcal{O} = \{\forall x(Person(x) \rightarrow Male(x) \vee Female(x)\}$

Query (CQ, i.e. existentially quantified conjunction of atoms)

$\boldsymbol{q} = \exists x \exists y\, Male(x) \wedge Parent(x, y) \wedge Female(y)$ (Is there a father and a daughter ?)

Semantics: $\mathcal{D}, \mathcal{O} \models \boldsymbol{q}$ if q holds in all intepretations for $\mathcal{D}$ and $\mathcal{O}$

$\mathcal{D}_1 = \mathcal{D} \cup \{Male(Sasha)\} \models \boldsymbol{q}$ under $x \mapsto Sasha$ and $y \mapsto Ann$
$\mathcal{D}_2 = \mathcal{D} \cup \{Female(Sasha)\} \models \boldsymbol{q}$ under $x \mapsto John$ and $y \mapsto Sasha$, thus $\mathcal{D}, \mathcal{O} \models \boldsymbol{q}$

Ontology-mediated querying by example (dates back to Prolog?)

Fact Set (a collection of *facts*, or ground atoms, a.k.a. the data)

$\mathcal{D} = \{Male(John), Parent(John, Sasha), Person(Sasha), Parent(Sasha, Ann), Female(Ann)\}$

Ontology (a set of sentences from some fragment of μFO called *description logic*)

$\mathcal{O} = \{\forall x(Person(x) \to Male(x) \vee Female(x)\}$

Query (CQ, i.e. existentially quantified conjunction of atoms)

$\boldsymbol{q} = \exists x \exists y\, Male(x) \wedge Parent(x, y) \wedge Female(y)$ (Is there a father and a daughter ?)

Semantics: $\mathcal{D}, \mathcal{O} \models \boldsymbol{q}$ if q holds in all intepretations for $\mathcal{D}$ and $\mathcal{O}$

$\mathcal{D}_1 = \mathcal{D} \cup \{Male(Sasha)\} \models \boldsymbol{q}$ under $x \mapsto Sasha$ and $y \mapsto Ann$
$\mathcal{D}_2 = \mathcal{D} \cup \{Female(Sasha)\} \models \boldsymbol{q}$ under $x \mapsto John$ and $y \mapsto Sasha$, thus $\mathcal{D}, \mathcal{O} \models \boldsymbol{q}$

Ontology-mediated query (OMQ) $\mathcal{Q} = (\mathcal{O}, \boldsymbol{q})$ checks if

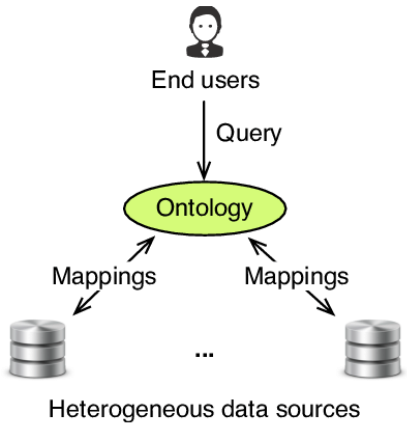
"there is a *Parent*-path over *Person* from *Male* to *Female*" (a rewriting of $\mathcal{Q}$)

- can be expressed in linear Datalog;
- NL-complete in data complexity (other examples match AC^0, L, P and coNP)

Data complexity of OMQ answering

Fundamental problem 1: how to determine the data complexity of answering a given ontology-mediated query $(\mathcal{O}, \boldsymbol{q})$ with an 'expressive' ontology $\mathcal{O}$ and a Boolean conjunctive query $\boldsymbol{q}$?

That is, the complexity of deciding whether $\mathcal{O}, \mathcal{D} \models \boldsymbol{q}$ for a given input data instance $\mathcal{D}$



Fundamental problem 1′: how to determine whether $(\mathcal{O}, \boldsymbol{q})$ is

- FO-rewritable, that is, whether there is an FO-sentence $\boldsymbol{q}'$ such that, for any data D, we have $\mathcal{O}, \mathcal{D} \models \boldsymbol{q}$ iff $\mathcal{D} \models \boldsymbol{q}'$ (AC^0)
- linear-datalog-rewritable (in NL)
- datalog-rewritable (in PTime)

Related research - 1

Boundedness of datalog programs (is recursion eliminable?)

Undecidable (Vardi 1988), even for sirups (Marcinkowski 1996)
2ExpTime-complete for monadic programs (Cosmadakis et al. 1988)

Linearisability of datalog programs

Transform a datalog program into an equivalent linear program
i.e., ≤ 1 IDB predicate in rule bodies (Ullman, Afrati, Saraiya, etc.)

Markable disjunctive Datalog programs

A sufficient condition for plain Datalog rewritability of disjunctive Datalog programs.

(Grau, Nenov, Kaminsky 2016)

Related research - 2

Constraint satisfaction problems

Classify CSP patterns according to their complexities ...

Related research - 2

Constraint satisfaction problems

Classify CSP patterns according to their complexities ...

Reduction to CSPs

FO- and datalog-rewritability of OMQs with $\mathcal{SHIU}$ ontologies and atomic queries is decidable in NExpTime by a reduction to CSPs (Bienvenu, ten Cate, Lutz, Wolter 2014)

Related research - 2

Constraint satisfaction problems

Classify CSP patterns according to their complexities ...

Reduction to CSPs

FO- and datalog-rewritability of OMQs with $\mathcal{SHIU}$ ontologies and atomic queries is decidable in NExpTime by a reduction to CSPs

(Bienvenu, ten Cate, Lutz, Wolter 2014)

Trichotomy for $\mathcal{EL}$ OMQs

AC^0/NL/P trichotomy for the data complexity of answering OMQs with an $\mathcal{EL}$ ontology and a CQ

Can be checked in ExpTime (Lutz, Sabellek 2017-20)

Our approach

Covering axiom

$A \sqsubseteq T \sqcup F$ class A is covered by classes T and F

Example: $\mathit{Animal} \sqsubseteq \mathit{Female} \sqcup \mathit{Male}$

Global covering and disjointness

$\top \sqsubseteq T \sqcup F$ everything is covered by T and F

Disjointness: $T \sqcap F \sqsubseteq \bot$ $\quad \top \sqsubseteq \mathit{Alive} \sqcup \mathit{Dead}, \quad \mathit{Alive} \sqcap \mathit{Dead} \sqsubseteq \bot$

Notation:

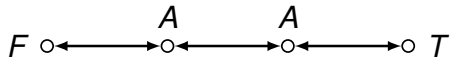
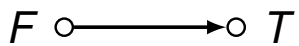
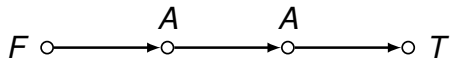
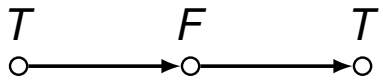
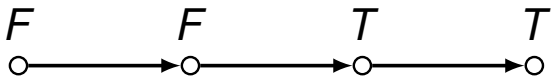
$\mathsf{Cov}_\top = \{\top \sqsubseteq F \sqcup T\}$ $\quad \mathsf{Cov}_A = \{A \sqsubseteq F \sqcup T\}$

$\mathsf{Cov}_\top^\bot = \{\top \sqsubseteq F \sqcup T,\ F \sqcap T \sqsubseteq \bot\}$ $\quad \mathsf{Cov}_A^\bot = \{A \sqsubseteq F \sqcup T,\ F \sqcap T \sqsubseteq \bot\}$

Problem

Classify syntactically CQs $\boldsymbol{q}$ w.r.t. complexity of answering $(\mathsf{Cov}, \boldsymbol{q})$

Typical examples for $\mathsf{Cov}_A = \{A \sqsubseteq F \sqcup T\}$

Complexity	CQ q	Explanation
AC^0		if $\boldsymbol{q}$ has only F atoms but no T, then Cov_A can be ignored
L		checks undirected reachability:  the answer to $\boldsymbol{Q}$ is 'yes'
NL		checks directed reachability:  the answer to $\boldsymbol{Q}$ is 'yes'
P		evaluates monotone Boolean circuits C
coNP		checks CNF satisfiability

Naïve idea: classify $\boldsymbol{q}$ according to the number of occurrences of T, F

How disjunctive predicates F and T may occur in $\boldsymbol{q}$?

$F(x) \in \boldsymbol{q}$ is **solitary** if $T(x) \notin \boldsymbol{q}$ $\quad$ $F(x), T(x) \in \boldsymbol{q}$ is a **twin**

Theorem: AC^0 vs. L-hard (emphasise the danger of twins)

- If $\boldsymbol{q}$ has no solitary F then both $(\mathrm{Cov}_A^{\perp}, \boldsymbol{q})$ and $(\mathrm{Cov}_A, \boldsymbol{q})$ are in AC^0, with $\boldsymbol{q}$ being an FO-rewriting
- Otherwise, for $\boldsymbol{q}$ without twins, $(\mathrm{Cov}_{\top}^{\perp}, \boldsymbol{q})$, $(\mathrm{Cov}_{\top}, \boldsymbol{q})$ are L-hard
- $(\mathrm{Cov}_A^{\perp}, \boldsymbol{q})$ is in AC^0 iff $\boldsymbol{q}$ has no solitary F or T (nice criterion)
- deciding if $(\mathrm{Cov}_A, \boldsymbol{q})$ is in AC^0 is PSPACE-hard if $\boldsymbol{q}$ may have twins (and highly likely 2ExpTime-hard)

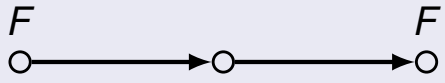
Syntactic tetrachotomy for path CQs and $\mathrm{Cov}^{\perp}_{A}$

A CQ is a *path-CQ* if its binary atoms can be arranged in a directed path.

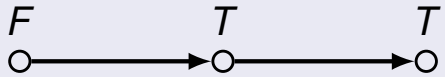
Theorem (main result)

For any $\mathcal{Q} = (\mathrm{Cov}^{\perp}_{A}, \boldsymbol{q})$ with path-CQ $\boldsymbol{q}$, the following hold:

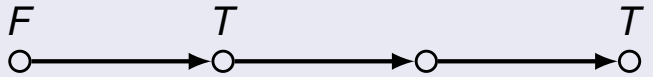
- *if $\boldsymbol{q}$ is a 0-CQ, then $\mathcal{Q}$ is in AC^0;*



- *if $\boldsymbol{q}$ is a periodic 1-CQ, then $\mathcal{Q}$ is NL-complete;*



- *if $\boldsymbol{q}$ is a non-periodic 1-CQ, then $\mathcal{Q}$ is P-complete;*



- *if $\boldsymbol{q}$ is a 2-CQ, then $\mathcal{Q}$ is coNP-complete.*

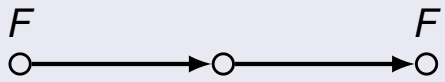
Syntactic tetrachotomy for path CQs and $\mathrm{Cov}_A^{\perp}$

A CQ is a *path-CQ* if its binary atoms can be arranged in a directed path.

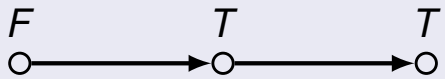
Theorem (main result)

For any $\mathcal{Q} = (\mathrm{Cov}_A^{\perp}, \boldsymbol{q})$ *with path-CQ* $\boldsymbol{q}$*, the following hold:*

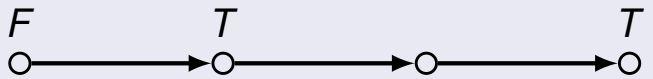
- *if* $\boldsymbol{q}$ *is a 0-CQ, then* $\mathcal{Q}$ *is in* AC^0*;*



- *if* $\boldsymbol{q}$ *is a periodic 1-CQ, then* $\mathcal{Q}$ *is NL-complete;*



- *if* $\boldsymbol{q}$ *is a non-periodic 1-CQ, then* $\mathcal{Q}$ *is P-complete;*



- *if* $\boldsymbol{q}$ *is a 2-CQ, then* $\mathcal{Q}$ *is coNP-complete.*

Questions ?

CQs without solitary F

$F(x) \in \boldsymbol{q}$ is **solitary** if $T(x) \notin \boldsymbol{q}$ $\quad$ $F(x), T(x) \in \boldsymbol{q}$ is a **twin**

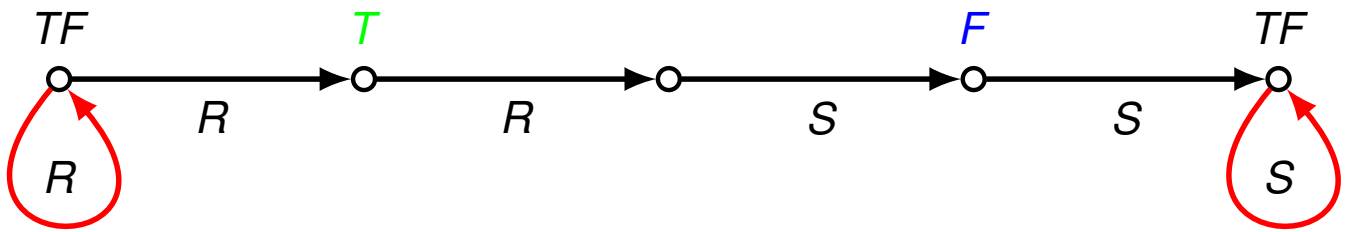
Theorem: AC^0 vs. L-hard

- If $\boldsymbol{q}$ has no solitary F then both $(\mathrm{Cov}_A^{\perp}, \boldsymbol{q})$ and $(\mathrm{Cov}_A, \boldsymbol{q})$ are in AC^0, with $\boldsymbol{q}$ being an FO-rewriting
- Otherwise, for $\boldsymbol{q}$ without twins, $(\mathrm{Cov}_{\top}^{\perp}, \boldsymbol{q})$, $(\mathrm{Cov}_{\top}, \boldsymbol{q})$ are L-hard
- $(\mathrm{Cov}_A^{\perp}, \boldsymbol{q})$ is in AC^0 iff $\boldsymbol{q}$ has no solitary F or T (nice criterion)

Path CQs

$(\mathrm{Cov}_A, \boldsymbol{q})$ with a path CQ $\boldsymbol{q}$ is in AC^0 iff $\boldsymbol{q}$ has no solitary F or T.
If path $\boldsymbol{q}$ contains both solitary F and T, then $(\mathrm{Cov}_A, \boldsymbol{q})$ is NL-hard

However, $(\mathrm{Cov}_A, \boldsymbol{q})$ with the following (minimal) non-path $\boldsymbol{q}$ is in AC^0

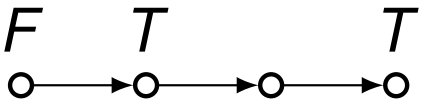


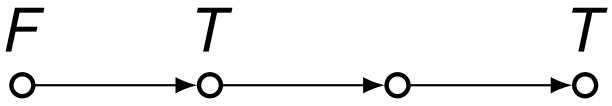
1-CQs (with one solitary F)

- Reducible to datalog, so can be done in P
- For tree-shaped $\boldsymbol{q}$ with root F, reducible to $\mathcal{EL}$ atomic OMQs, so AC^0/NL/P trichotomy decidable in ExpTime by (Lutz, Sabellek)

1-CQs (with one solitary F)

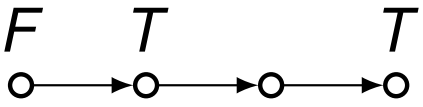
- Reducible to datalog, so can be done in P
- For tree-shaped $\boldsymbol{q}$ with root F, reducible to $\mathcal{EL}$ atomic OMQs, so AC^0/NL/P trichotomy decidable in ExpTime by (Lutz, Sabellek)

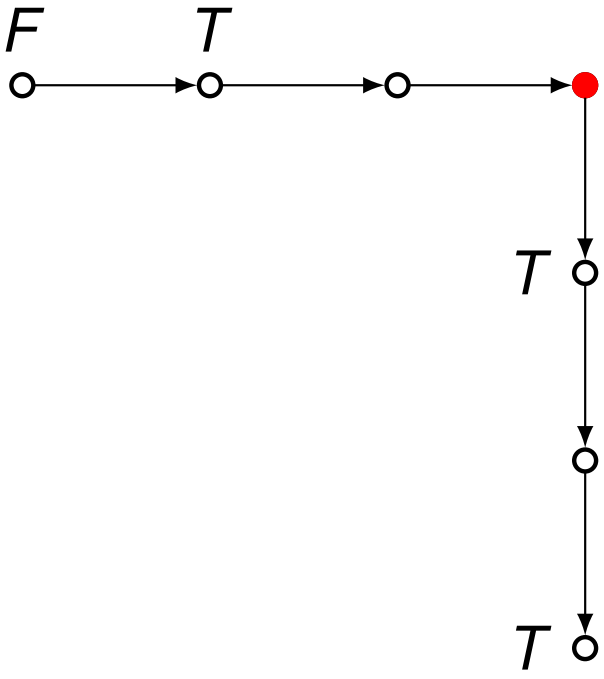
Example: Understand models of $(\mathsf{Cov}_\top, \boldsymbol{q})$ with $\boldsymbol{q} =$ 



1-CQs (with one solitary F)

- Reducible to datalog, so can be done in P
- For tree-shaped $\boldsymbol{q}$ with root F, reducible to $\mathcal{EL}$ atomic OMQs, so AC^0/NL/P trichotomy decidable in ExpTime by (Lutz, Sabellek)

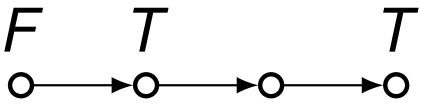
Example: Understand models of $(\mathrm{Cov}_\top, \boldsymbol{q})$ with $\boldsymbol{q} =$ 

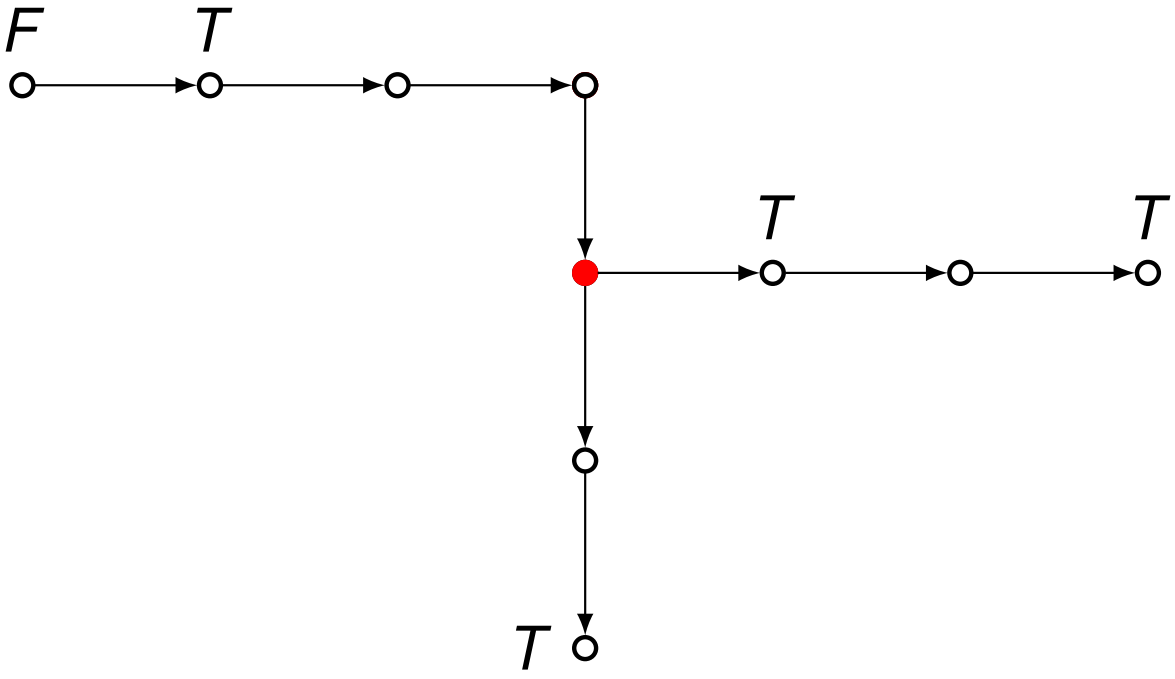


1. budding

1-CQs (with one solitary F)

- Reducible to datalog, so can be done in P
- For tree-shaped $\boldsymbol{q}$ with root F, reducible to $\mathcal{EL}$ atomic OMQs, so AC^0/NL/P trichotomy decidable in ExpTime by (Lutz, Sabellek)

Example: Understand models of $(\mathrm{Cov}_\top, \boldsymbol{q})$ with $\boldsymbol{q} =$ 

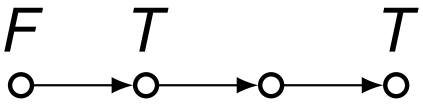


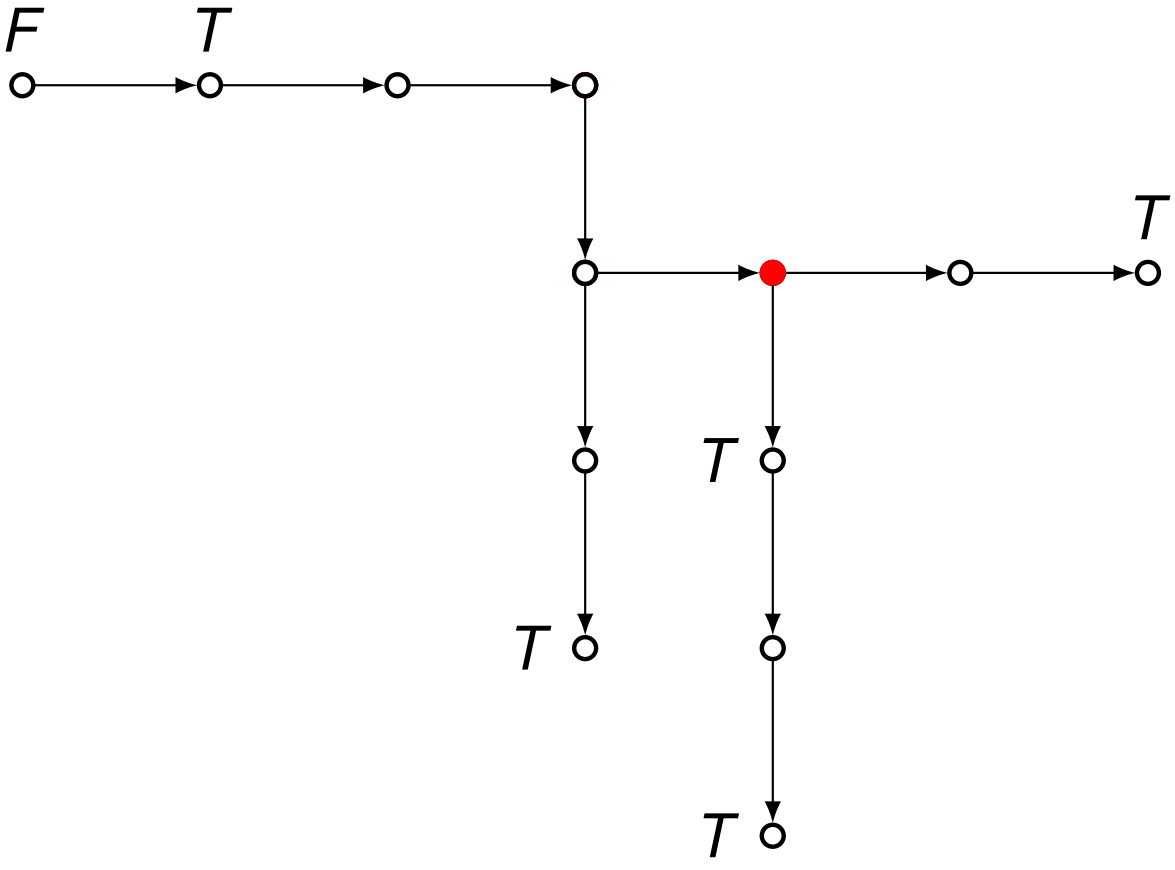
1. budding

2. budding

1-CQs (with one solitary F)

- Reducible to datalog, so can be done in P
- For tree-shaped $\boldsymbol{q}$ with root F, reducible to $\mathcal{EL}$ atomic OMQs, so AC^0/NL/P trichotomy decidable in ExpTime by (Lutz, Sabellek)

Example: Understand models of $(\mathrm{Cov}_\top, \boldsymbol{q})$ with $\boldsymbol{q} =$ 



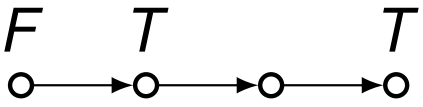
1. budding

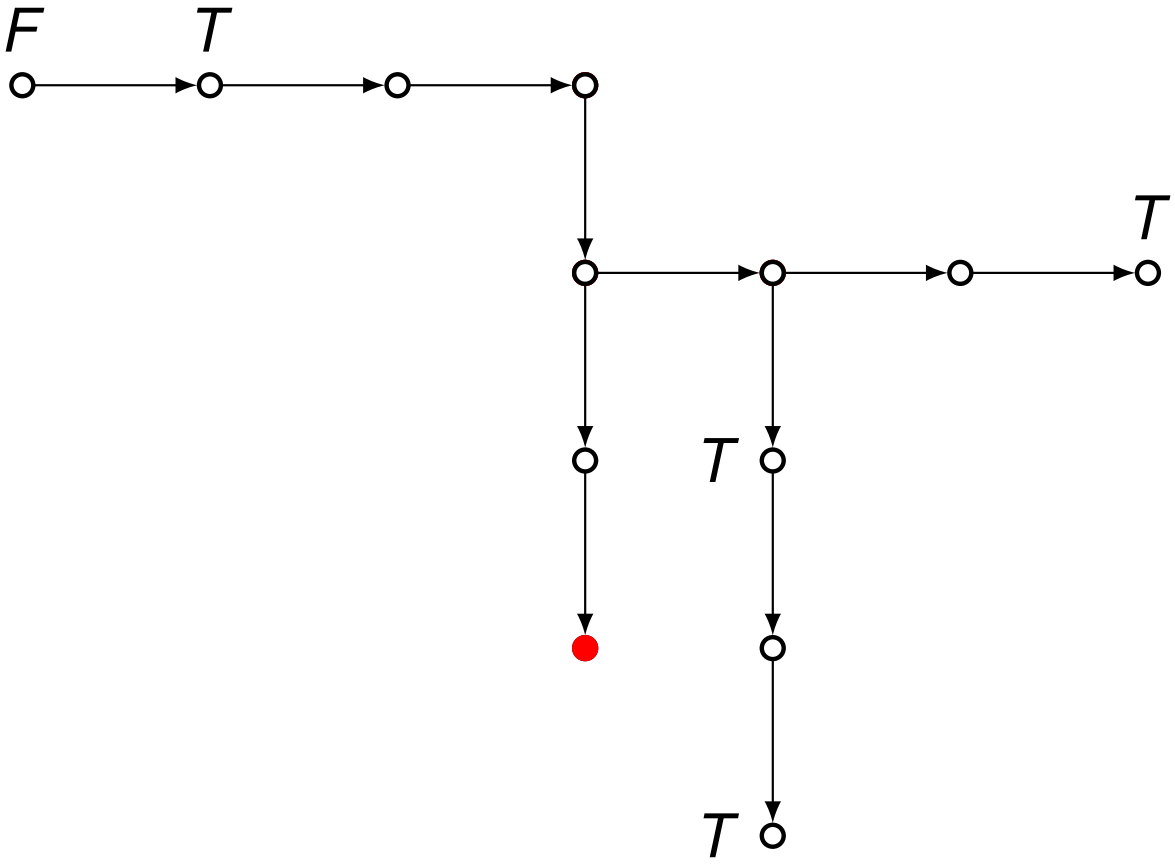
2. budding

3. budding

1-CQs (with one solitary F)

- Reducible to datalog, so can be done in P
- For tree-shaped $\boldsymbol{q}$ with root F, reducible to $\mathcal{EL}$ atomic OMQs, so AC^0/NL/P trichotomy decidable in ExpTime by (Lutz, Sabellek)

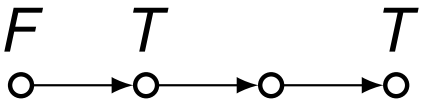
Example: Understand models of $(\mathrm{Cov}_\top, \boldsymbol{q})$ with $\boldsymbol{q} =$ 

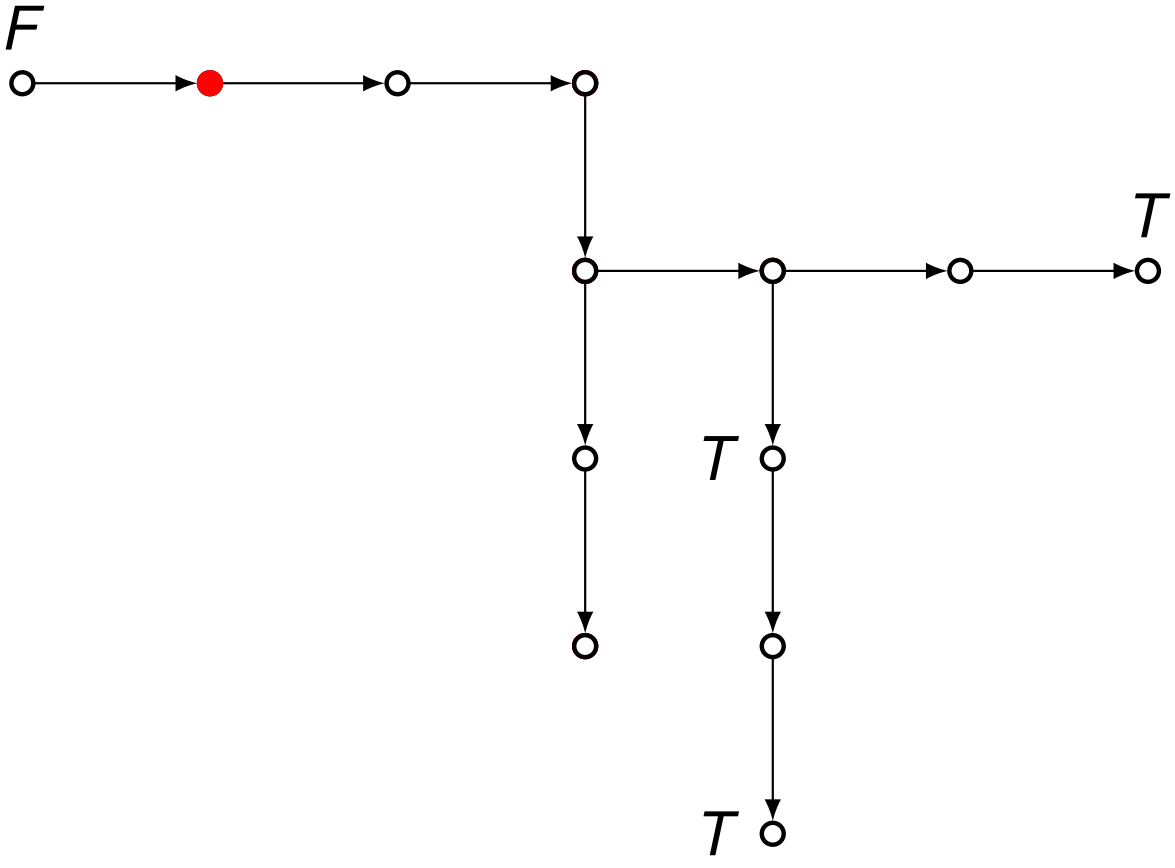


1. budding
2. budding
3. budding
4. pruning

1-CQs (with one solitary F)

- Reducible to datalog, so can be done in P
- For tree-shaped $\boldsymbol{q}$ with root F, reducible to $\mathcal{EL}$ atomic OMQs, so AC^0/NL/P trichotomy decidable in ExpTime by (Lutz, Sabellek)

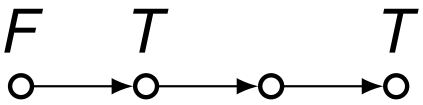
Example: Understand models of $(\mathrm{Cov}_\top, \boldsymbol{q})$ with $\boldsymbol{q} =$ 

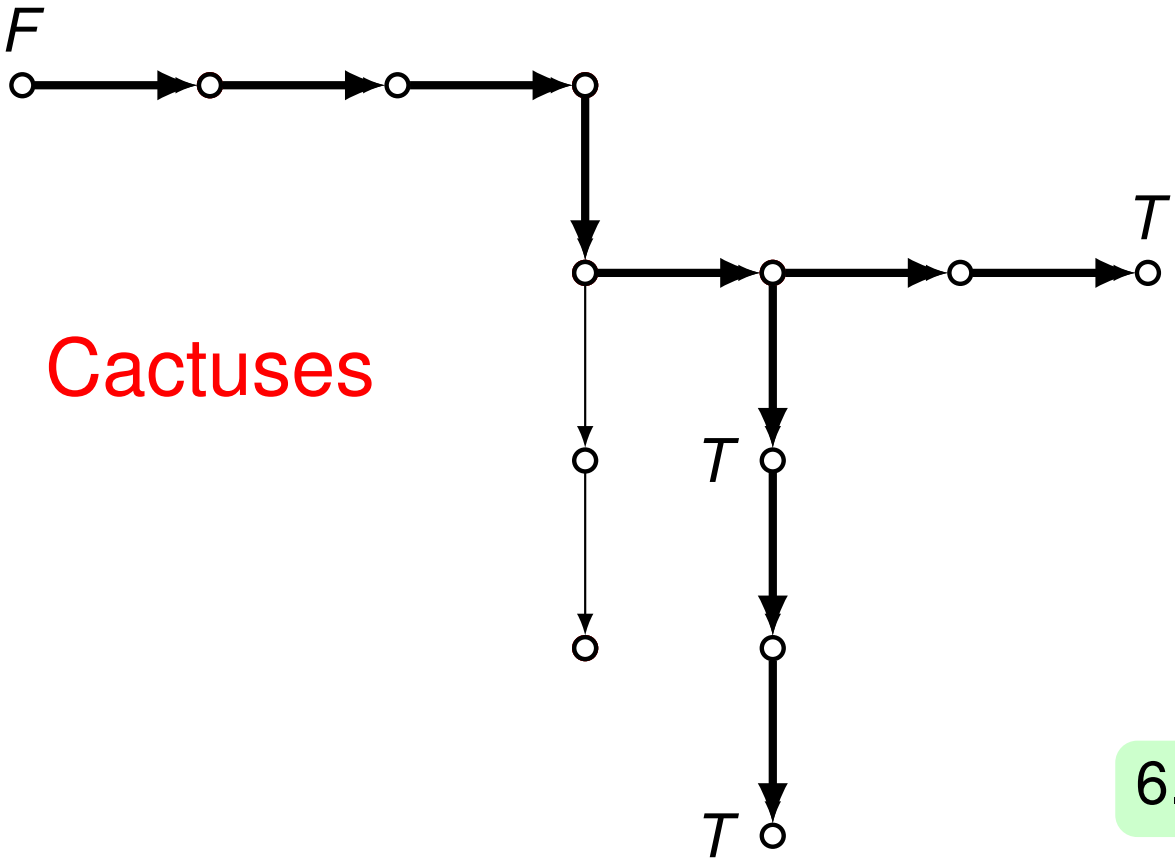


1. budding
2. budding
3. budding
4. pruning
5. pruning

1-CQs (with one solitary F)

- Reducible to datalog, so can be done in P
- For tree-shaped $\boldsymbol{q}$ with root F, reducible to $\mathcal{EL}$ atomic OMQs, so AC^0/NL/P trichotomy decidable in ExpTime by (Lutz, Sabellek)

Example: Understand models of $(\mathrm{Cov}_\top, \boldsymbol{q})$ with $\boldsymbol{q} =$ 



Cactuses

1. budding
2. budding
3. budding
4. pruning
5. pruning

6. branching degree 1

Cactuses (inspired by Cosmodakis et al. 1988, Lutz-Sabellek 2017-18)

For $\boldsymbol{Q} = (\mathrm{Cov}_A, \boldsymbol{q})$ with 1-CQ $\boldsymbol{q}$, $\mathfrak{K}_{\boldsymbol{Q}}$ is the set of all cactuses for $\boldsymbol{Q}$
$\mathfrak{K}^{\dagger}_{\boldsymbol{Q}}$ are minimal cactuses in $\mathfrak{K}_{\boldsymbol{Q}}$ (without proper sub-cactuses in $\mathfrak{K}_{\boldsymbol{Q}}$).

Cactus theorem for $\boldsymbol{Q} = (\mathrm{Cov}_A, \boldsymbol{q})$ with 1-CQ $\boldsymbol{q}$

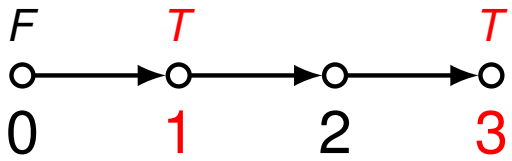
$\boldsymbol{Q}$ is in AC^0 iff for every $\boldsymbol{C} \in \mathfrak{K}^{\dagger}_{\boldsymbol{Q}}$, there is a homomorphism $h\colon \boldsymbol{q} \to \boldsymbol{C}$;
$\boldsymbol{Q}$ is linear-datalog-rewritable (so in NL) if $\mathfrak{K}^{\dagger}_{\boldsymbol{Q}}$ is boundedly branching.

Proof: represent cactuses as words in a tree alphabet and construct a tree automaton $\mathfrak{A}_{\boldsymbol{Q}}$ such that $\mathfrak{K}^{\dagger}_{\boldsymbol{Q}} \subseteq L(\mathfrak{A}_{\boldsymbol{Q}}) \subseteq \mathfrak{K}_{\boldsymbol{Q}}$.

For boundedly branching $\mathfrak{K}^{\dagger}_{\boldsymbol{Q}}$, $\mathfrak{A}_{\boldsymbol{Q}}$ is converted to a linear-datalog-rewriting of $\boldsymbol{Q}$

Arithmetic progression theorem

Let $\boldsymbol{q}$ be an F-path without twins with a single binary relation R.
Let $B_{\boldsymbol{q}}$ be the set of positions with T:



$B_{\boldsymbol{q}} = \{1, 3\}$

Theorem (NL/P dichotomy)

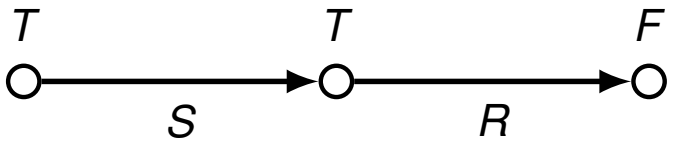
$\boldsymbol{Q} = (\mathrm{Cov}_A, \boldsymbol{q})$ *is in NL iff the set of integers* $\{0\} \cup B_{\boldsymbol{q}}$ *is an arithmetic progression (unless NL* $=$ *P). Otherwise* $\boldsymbol{Q}$ *is P-hard.*

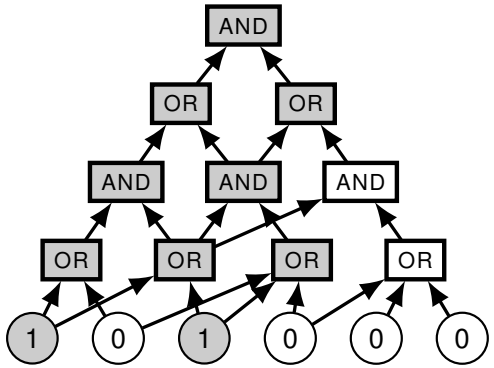
NB: For $\boldsymbol{q}$ above: $(\mathrm{Cov}_A, \boldsymbol{q})$ is P-hard as 0,1,3 is not a progression
but $(\mathrm{Cov}_\top, \boldsymbol{q})$ is in NL (A is essential!)

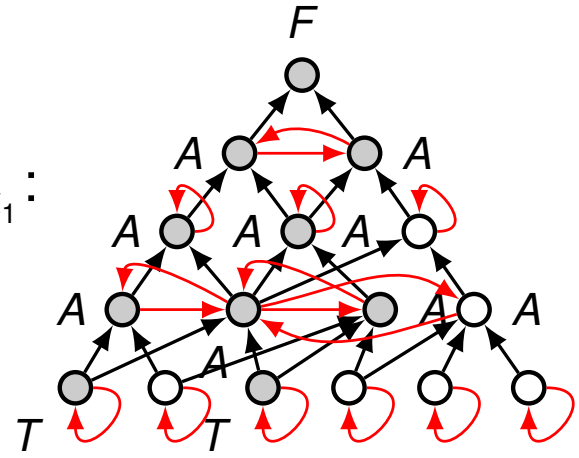
Question

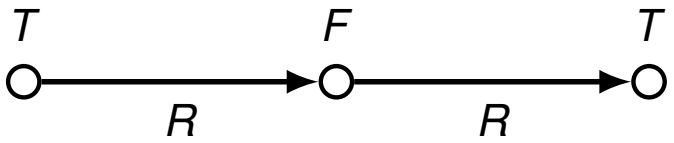
How to generalise this theorem to $\mathrm{Cov}_\top$?

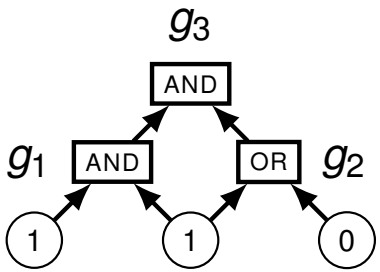
P-hardness via monotone circuits

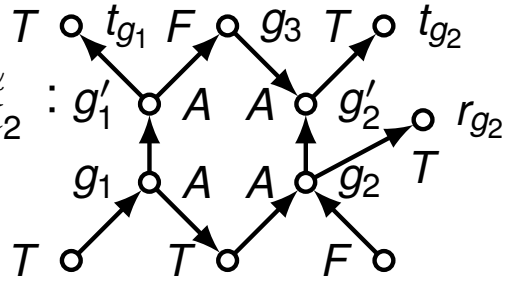
- $\boldsymbol{q}_1 =$ 

$\mathfrak{C}_1$: 

ABox $\mathcal{A}^{\alpha}_{\mathfrak{C}_1}$: 

- $\boldsymbol{q}_2 =$ 

$\mathfrak{C}_2$: 

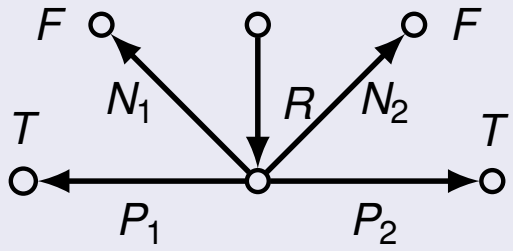
ABox $\mathcal{A}^{\alpha}_{\mathfrak{C}_2}$: 

$\mathfrak{C}(\vec{\alpha}) = 1$ iff $\mathsf{Cov}, \mathcal{A}^{\alpha}_{\mathfrak{C}} \models \boldsymbol{q}$.

coNP-hardness

Schaerf's (1993) example

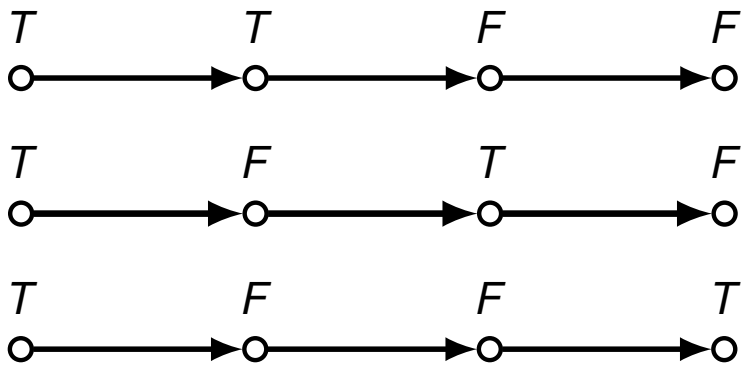
$(\mathsf{Cov}_{\top}, \boldsymbol{q})$ with $\boldsymbol{q}$ on the right is coNP-complete proof by reduction of 2+2 CNF



Conjecture

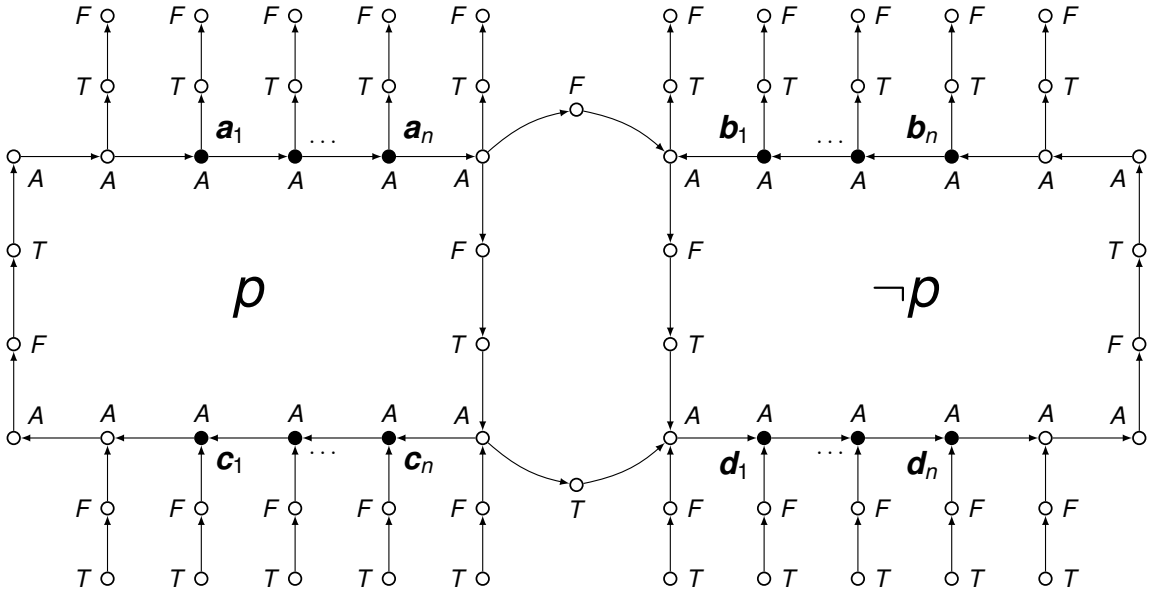
For (tree-shaped) $\boldsymbol{q}$ with at least two solitary F and two solitary T, $(\mathsf{Cov}_{A}, \boldsymbol{q})$ is coNP-hard

This is so for all 2-2 path-CQ of length 4

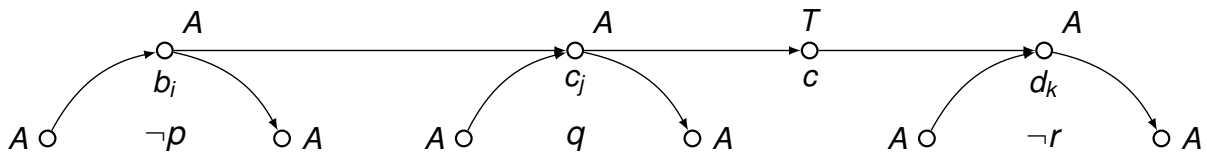


Proving coNP-hardness by reduction to 3CNF

Gadget for p and $\neg p$ in a 3CNF with n clauses:



Gadget for clause $c = (p \vee q \vee \neg r)$:



The resulting ABox $\mathcal{A}_\psi$ is s.t. 3CNF ψ is satisfiable iff $\mathsf{Cov}_A, \mathcal{A}_\psi \not\models \boldsymbol{q}$