Query Answering with Guarded Rules and Expressive Constraints

Antoine Amarilli[1,2], Michael Benedikt[2], Pierre Bourhis[3], Michael Vanden Boom[2]

October 20, 2016

[1]Télécom ParisTech, Paris, France

[2]University of Oxford, Oxford, United Kingdom

[3]CNRS CRIStAL, Lille

Open-world query answering (QA)

- We are given:
  - Relational instance I (ground facts)
  - Logical constraints Σ
  - Boolean conjunctive query q (existentially quantified conjunction of atoms)

Open-world query answering (QA)

- We are given:
  - Relational instance I (ground facts)
  - Logical constraints Σ
  - Boolean conjunctive query q
    (existentially quantified conjunction of atoms)
- We ask:
  - Consider all possible completions $J \supseteq I$
  - Restrict to those that satisfy the constraints Σ

  $\rightarrow$ Is q certain among them?

QA example

Relational instance I

- $\mathrm{CoSup}(\mathrm{Jamie})$ – *Jamie is a co-supervised student*

QA example

Relational instance I

- $\mathrm{CoSup}(\mathrm{Jamie})$ – *Jamie is a co-supervised student*

Logical constraints Σ

- $\forall x\ \mathrm{CoSup}(x) \to \exists y\, z\ \mathrm{Adv}(x, y) \land \mathrm{Adv}(x, z) \land \mathrm{Knows}(y, z) \land y \neq z$
  A co-supervised student has two advisors that know each other

QA example

Relational instance I

- $\mathrm{CoSup}(\mathrm{Jamie})$ – *Jamie is a co-supervised student*

Logical constraints Σ

- $\forall x\ \mathrm{CoSup}(x) \to \exists y\, z\ \mathrm{Adv}(x, y) \land \mathrm{Adv}(x, z) \land \mathrm{Knows}(y, z) \land y \neq z$
  A co-supervised student has two advisors that know each other
- $\forall x\, y\ \mathrm{Adv}(x, y) \to \mathrm{Knows}(x, y)$
  Advisors know their students

QA example

Relational instance I

- $\mathrm{CoSup}(\mathrm{Jamie})$ – *Jamie is a co-supervised student*

Logical constraints Σ

- $\forall x\ \mathrm{CoSup}(x) \to \exists y\, z\ \mathrm{Adv}(x,y) \wedge \mathrm{Adv}(x,z) \wedge \mathrm{Knows}(y,z) \wedge y \neq z$
  A co-supervised student has two advisors that know each other
- $\forall x\, y\ \mathrm{Adv}(x,y) \to \mathrm{Knows}(x,y)$
  Advisors know their students
- $\forall x\, y\ \mathrm{Knows}(x,y) \to \mathrm{Knows}(y,x)$
  "Knows" is symmetric

QA example

Relational instance I

- $\mathrm{CoSup}(\mathrm{Jamie})$ – *Jamie is a co-supervised student*

Logical constraints Σ

- $\forall x\ \mathrm{CoSup}(x) \to \exists y\, z\ \mathrm{Adv}(x, y) \land \mathrm{Adv}(x, z) \land \mathrm{Knows}(y, z) \land y \neq z$
  A co-supervised student has two advisors that know each other
- $\forall x\, y\ \mathrm{Adv}(x, y) \to \mathrm{Knows}(x, y)$
  Advisors know their students
- $\forall x\, y\ \mathrm{Knows}(x, y) \to \mathrm{Knows}(y, x)$
  “Knows” is symmetric

Boolean conjunctive query q

- $\exists x\, y\, z\ \mathrm{Knows}(x, y) \land \mathrm{Knows}(y, z) \land \mathrm{Knows}(x, z)$
  Is there a clique of 3 people that know each other?

QA example

Relational instance I

- $\mathrm{CoSup}(\mathrm{Jamie})$ – *Jamie is a co-supervised student*

Logical constraints Σ

- $\forall x\ \mathrm{CoSup}(x) \rightarrow \exists y\, z\ \mathrm{Adv}(x, y) \wedge \mathrm{Adv}(x, z) \wedge \mathrm{Knows}(y, z) \wedge y \neq z$
  A co-supervised student has two advisors that know each other
- $\forall x\, y\ \mathrm{Adv}(x, y) \rightarrow \mathrm{Knows}(x, y)$
  Advisors know their students
- $\forall x\, y\ \mathrm{Knows}(x, y) \rightarrow \mathrm{Knows}(y, x)$
  "Knows" is symmetric

Boolean conjunctive query q

- $\exists x\, y\, z\ \mathrm{Knows}(x, y) \wedge \mathrm{Knows}(y, z) \wedge \mathrm{Knows}(x, z)$
  Is there a clique of 3 people that know each other?

$\rightarrow$ The query is certain under the instance and constraints

QA tasks

Is QA decidable, and what is the complexity?

QA tasks

Is QA decidable, and what is the complexity?

$\rightarrow$ Depends on the language used for logical constraints

QA tasks

Is QA decidable, and what is the complexity?

→ Depends on the language used for logical constraints

→ E.g., for arbitrary first-order constraints, QA is undecidable (because satisfiability is undecidable)

QA tasks

Is QA decidable, and what is the complexity?

$\rightarrow$ Depends on the language used for logical constraints

$\rightarrow$ E.g., for arbitrary first-order constraints, QA is undecidable (because satisfiability is undecidable)

$\rightarrow$ Find fragments of first-order logic with decidable QA

Tuple-generating dependencies (or existential rules)

$$\forall \mathbf{x}\,\mathbf{y}\ \phi(\mathbf{x},\mathbf{y}) \rightarrow \exists \mathbf{z}\ \psi(\mathbf{y},\mathbf{z})$$

where the body ϕ and head ψ are conjunctions of atoms

Tuple-generating dependencies (or existential rules)

$$\forall \mathbf{x}\, \mathbf{y}\ \phi(\mathbf{x}, \mathbf{y}) \to \exists \mathbf{z}\ \psi(\mathbf{y}, \mathbf{z})$$

where the body ϕ and head ψ are conjunctions of atoms

→ Guarded TGDs: body ϕ has an atom with all body variables $\mathbf{x}, \mathbf{y}$

$$\forall x\, y_1\, y_2\ S(x, y_1) \land S(x, y_2) \land U(x, y_1, y_2) \to \exists z\ S(y_2, z) \land T(y_1)$$

Tuple-generating dependencies (or existential rules)

$$\forall \mathbf{x}\, \mathbf{y}\ \phi(\mathbf{x}, \mathbf{y}) \rightarrow \exists \mathbf{z}\ \psi(\mathbf{y}, \mathbf{z})$$

where the body ϕ and head ψ are conjunctions of atoms

→ Guarded TGDs: body ϕ has an atom with all body variables $\mathbf{x}, \mathbf{y}$

$$\forall x\, y_1\, y_2\ S(x, y_1) \wedge S(x, y_2) \wedge U(x, y_1, y_2) \rightarrow \exists z\ S(y_2, z) \wedge T(y_1)$$

→ Frontier-one TGDs: the frontier $\mathbf{y}$ has only one variable

$$\forall x\, y\ S(x, y) \rightarrow \exists z\ S(y, z)$$

Tuple-generating dependencies (or existential rules)

$$\forall \mathbf{x}\,\mathbf{y}\ \phi(\mathbf{x}, \mathbf{y}) \rightarrow \exists \mathbf{z}\ \psi(\mathbf{y}, \mathbf{z})$$

where the body ϕ and head ψ are conjunctions of atoms

→ Guarded TGDs: body ϕ has an atom with all body variables $\mathbf{x}, \mathbf{y}$

$$\forall x\, y_1\, y_2\ S(x, y_1) \wedge S(x, y_2) \wedge U(x, y_1, y_2) \rightarrow \exists z\ S(y_2, z) \wedge T(y_1)$$

→ Frontier-one TGDs: the frontier $\mathbf{y}$ has only one variable

$$\forall x\, y\ S(x, y) \rightarrow \exists z\ S(y, z)$$

→ Frontier-guarded TGDs (FGTGDs):
body ϕ has an atom with all frontier variables $\mathbf{y}$

$$\forall x\, y_1\, y_2\ S(x, y_1) \wedge S(x, y_2) \wedge R(y_1, y_2) \rightarrow \exists z\ S(y_2, z) \wedge T(y_1)$$

Tuple-generating dependencies (or existential rules)

$$\forall \mathbf{x}\,\mathbf{y}\ \phi(\mathbf{x},\mathbf{y}) \rightarrow \exists \mathbf{z}\ \psi(\mathbf{y},\mathbf{z})$$

where the body ϕ and head ψ are conjunctions of atoms

→ Guarded TGDs: body ϕ has an atom with all body variables $\mathbf{x},\mathbf{y}$

$$\forall x\, y_1\, y_2\ S(x,y_1) \wedge S(x,y_2) \wedge U(x,y_1,y_2) \rightarrow \exists z\ S(y_2,z) \wedge T(y_1)$$

→ Frontier-one TGDs: the frontier $\mathbf{y}$ has only one variable

$$\forall x\, y\ S(x,y) \rightarrow \exists z\ S(y,z)$$

→ Frontier-guarded TGDs (FGTGDs):
body ϕ has an atom with all frontier variables $\mathbf{y}$

$$\forall x\, y_1\, y_2\ S(x,y_1) \wedge S(x,y_2) \wedge R(y_1,y_2) \rightarrow \exists z\ S(y_2,z) \wedge T(y_1)$$

→ QA for FGTGDs is decidable [Baget et al., 2011]

FGTGDs cannot express everything

- Disjunction: *Professors must have a PhD* ***or*** *an equivalent title*
  Negation: *Professors* ***without*** *an HDR must have co-supervisors*

FGTGDs cannot express everything

- Disjunction: *Professors must have a PhD **or** an equivalent title*
  Negation: *Professors **without** an HDR must have co-supervisors*
  - $\rightarrow$ Handled by guarded logics and guarded negation logics

FGTGDs cannot express everything

- Disjunction: *Professors must have a PhD* ***or*** *an equivalent title*
  Negation: *Professors* ***without*** *an HDR must have co-supervisors*
  → Handled by guarded logics and guarded negation logics

- Number constraints: *Co-supervised students have* ***two advisors***
  Functionality: *Any person is born in* ***exactly one*** *place*

FGTGDs cannot express everything

- Disjunction: *Professors must have a PhD* ***or*** *an equivalent title*
  Negation: *Professors* ***without*** *an HDR must have co-supervisors*
  - → Handled by guarded logics and guarded negation logics

- Number constraints: *Co-supervised students have* ***two advisors***
  Functionality: *Any person is born in* ***exactly one*** *place*
  - → $\forall x\, y_1\, y_2\, \mathrm{Born}(x, y_1) \wedge \mathrm{Born}(x, y_2) \rightarrow y_1 = y_2$

FGTGDs cannot express everything

- Disjunction: *Professors must have a PhD **or** an equivalent title*
  Negation: *Professors **without** an HDR must have co-supervisors*
  - → Handled by guarded logics and guarded negation logics

- Number constraints: *Co-supervised students have **two advisors***
  Functionality: *Any person is born in **exactly one** place*
  - → $\forall x\, y_1\, y_2\ \mathrm{Born}(x, y_1) \wedge \mathrm{Born}(x, y_2) \rightarrow y_1 = y_2$ ↯ unguarded frontier!

FGTGDs cannot express everything

- Disjunction: *Professors must have a PhD* ***or*** *an equivalent title*
  Negation: *Professors* ***without*** *an HDR must have co-supervisors*
  → Handled by guarded logics and guarded negation logics

- Number constraints: *Co-supervised students have* ***two advisors***
  Functionality: *Any person is born in* ***exactly one*** *place*
  → $\forall x\, y_1\, y_2\ \mathrm{Born}(x, y_1) \wedge \mathrm{Born}(x, y_2) \rightarrow y_1 = y_2$ ⚡ unguarded frontier!

- Transitivity: isLocatedIn, isPartOf, subclass...

FGTGDs cannot express everything

- Disjunction: *Professors must have a PhD* ***or*** *an equivalent title*
  Negation: *Professors* ***without*** *an HDR must have co-supervisors*
  - $\rightarrow$ Handled by guarded logics and guarded negation logics

- Number constraints: *Co-supervised students have* ***two advisors***
  Functionality: *Any person is born in* ***exactly one*** *place*
  - $\rightarrow$ $\forall x\, y_1\, y_2\ \mathrm{Born}(x, y_1) \wedge \mathrm{Born}(x, y_2) \rightarrow y_1 = y_2$ ⚡ unguarded frontier!

- Transitivity: isLocatedIn, isPartOf, subclass...
  - $\rightarrow$ $\forall x\, y_1\, y_2\ \mathrm{Part}(y_1, x) \wedge \mathrm{Part}(x, y_2) \rightarrow \mathrm{Part}(y_1, y_2)$

FGTGDs cannot express everything

- Disjunction: *Professors must have a PhD **or** an equivalent title*
  Negation: *Professors **without** an HDR must have co-supervisors*
  - → Handled by guarded logics and guarded negation logics
- Number constraints: *Co-supervised students have **two advisors***
  Functionality: *Any person is born in **exactly one** place*
  - → $\forall x\, y_1\, y_2\ \mathrm{Born}(x, y_1) \wedge \mathrm{Born}(x, y_2) \rightarrow y_1 = y_2$ ⚡ unguarded frontier!
- Transitivity: isLocatedIn, isPartOf, subclass...
  - → $\forall x\, y_1\, y_2\ \mathrm{Part}(y_1, x) \wedge \mathrm{Part}(x, y_2) \rightarrow \mathrm{Part}(y_1, y_2)$ ⚡unguarded frontier!

FGTGDs cannot express everything

- Disjunction: *Professors must have a PhD* ***or*** *an equivalent title*
  Negation: *Professors* ***without*** *an HDR must have co-supervisors*
  - $\rightarrow$ Handled by guarded logics and guarded negation logics
- Number constraints: *Co-supervised students have* ***two advisors***
  Functionality: *Any person is born in* ***exactly one*** *place*
  - $\rightarrow$ $\forall x\, y_1\, y_2\ \mathrm{Born}(x, y_1) \land \mathrm{Born}(x, y_2) \rightarrow y_1 = y_2$ ⚡ unguarded frontier!
- Transitivity: isLocatedIn, isPartOf, subclass...
  - $\rightarrow$ $\forall x\, y_1\, y_2\ \mathrm{Part}(y_1, x) \land \mathrm{Part}(x, y_2) \rightarrow \mathrm{Part}(y_1, y_2)$ ⚡unguarded frontier!
- Totality: *All elephants are bigger than all mice*

FGTGDs cannot express everything

- Disjunction: *Professors must have a PhD **or** an equivalent title*
  Negation: *Professors **without** an HDR must have co-supervisors*
  - $\rightarrow$ Handled by guarded logics and guarded negation logics

- Number constraints: *Co-supervised students have **two advisors***
  Functionality: *Any person is born in **exactly one** place*
  - $\rightarrow$ $\forall x\, y_1\, y_2\ \mathrm{Born}(x, y_1) \wedge \mathrm{Born}(x, y_2) \rightarrow y_1 = y_2$ ⚡ unguarded frontier!

- Transitivity: isLocatedIn, isPartOf, subclass...
  - $\rightarrow$ $\forall x\, y_1\, y_2\ \mathrm{Part}(y_1, x) \wedge \mathrm{Part}(x, y_2) \rightarrow \mathrm{Part}(y_1, y_2)$ ⚡unguarded frontier!

- Totality: *All elephants are bigger than all mice*
  - $\rightarrow$ $\forall y_1\, y_2\ \mathrm{Elephant}(y_1) \wedge \mathrm{Mouse}(y_2) \rightarrow y_1 < y_2$

FGTGDs cannot express everything

- Disjunction: *Professors must have a PhD **or** an equivalent title*
  Negation: *Professors **without** an HDR must have co-supervisors*
  - $\rightarrow$ Handled by guarded logics and guarded negation logics

- Number constraints: *Co-supervised students have **two advisors***
  Functionality: *Any person is born in **exactly one** place*
  - $\rightarrow$ $\forall x\, y_1\, y_2\ \mathrm{Born}(x, y_1) \wedge \mathrm{Born}(x, y_2) \rightarrow y_1 = y_2$ ⚡ unguarded frontier!

- Transitivity: isLocatedIn, isPartOf, subclass...
  - $\rightarrow$ $\forall x\, y_1\, y_2\ \mathrm{Part}(y_1, x) \wedge \mathrm{Part}(x, y_2) \rightarrow \mathrm{Part}(y_1, y_2)$ ⚡unguarded frontier!

- Totality: *All elephants are bigger than all mice*
  - $\rightarrow$ $\forall y_1\, y_2\ \mathrm{Elephant}(y_1) \wedge \mathrm{Mouse}(y_2) \rightarrow y_1 < y_2$ ⚡ unguarded frontier!

This talk

Can we extend frontier-guarded TGDs to capture some unguarded constraints while preserving the decidability of QA?

This talk

**Can we extend frontier-guarded TGDs
to capture some unguarded constraints
while preserving the decidability of QA?**

1. *Combining Existential Rules and Description Logics*
   A. A., Michael Benedikt, IJCAI'15.
   - $\rightarrow$ Extend FGTGDs with description logic constraints on arity-two data (includes disjunction, negation, counting)

This talk

Can we extend frontier-guarded TGDs to capture some unguarded constraints while preserving the decidability of QA?

1. *Combining Existential Rules and Description Logics*
   A. A., Michael Benedikt, IJCAI'15.
   $\rightarrow$ Extend FGTGDs with description logic constraints on arity-two data (includes disjunction, negation, counting)

2. *Query Answering with Transitive and Linear-Ordered Data*
   A. A., Michael Benedikt, P. Bourhis, M. Vanden Boom, IJCAI'16
   $\rightarrow$ Extend FGTGDs with transitive relations and order relations that cannot be used as guards

Table of contents

Rich Description Logics

Rich description logics **(DLs)**	FGTGDs

Rich Description Logics

Rich description logics **(DLs)**	FGTGDs
$\mathrm{Emp} \sqsubseteq \mathrm{CEO} \sqcup (\exists \mathrm{Mgr}^{-}.\mathrm{Emp})$	$\forall pwv\ \mathrm{Acpt}(p,w,v) \rightarrow \exists f\ \mathrm{Trip}(p,f,v)$

Rich Description Logics

Rich description logics **(DLs)**	FGTGDs
$\mathrm{Emp} \sqsubseteq \mathrm{CEO} \sqcup (\exists \mathrm{Mgr}^{-}.\mathrm{Emp})$	$\forall pwv\ \mathrm{Acpt}(p, w, v) \rightarrow \exists f\ \mathrm{Trip}(p, f, v)$
Arity-two only	Arbitrary arity

Rich Description Logics

Rich description logics **(DLs)**	FGTGDs
$\mathrm{Emp} \sqsubseteq \mathrm{CEO} \sqcup (\exists \mathrm{Mgr}^{-}.\mathrm{Emp})$	$\forall pwv\ \mathrm{Acpt}(p, w, v) \rightarrow \exists f\ \mathrm{Trip}(p, f, v)$
Arity-two only	Arbitrary arity
Rich: disjunction, etc.	Conjunction and implication only

Rich Description Logics

Rich description logics **(DLs)**	FGTGDs
Emp $\sqsubseteq$ CEO $\sqcup$ ($\exists$Mgr$^-$.Emp)	$\forall pwv$ Acpt$(p, w, v) \rightarrow \exists f$ Trip(p, f, v)
Arity-two only	Arbitrary arity
Rich: disjunction, etc.	Conjunction and implication only
Functionality asserts Funct(Mgr$^-$)	n/a

Our problem

Can we have the best of both worlds?

- QA is decidable for frontier-guarded TGDs
- QA is decidable for rich DLs (i.e., expressible in GC^2, guarded two-variable first-order logic with counting)

Our problem

Can we have the best of both worlds?

- QA is decidable for frontier-guarded TGDs
- QA is decidable for rich DLs (i.e., expressible in GC^2, guarded two-variable first-order logic with counting)

$\rightarrow$ Is QA decidable for rich DLs + some classes of FGTGDs?

Our problem

Can we have the best of both worlds?

- QA is decidable for frontier-guarded TGDs
- QA is decidable for rich DLs (i.e., expressible in GC^2, guarded two-variable first-order logic with counting)

→ Is QA decidable for rich DLs + some classes of FGTGDs?

We show:

Our problem

Can we have the best of both worlds?

- QA is decidable for frontier-guarded TGDs
- QA is decidable for rich DLs (i.e., expressible in GC^2, guarded two-variable first-order logic with counting)

$\rightarrow$ Is QA decidable for rich DLs + some classes of FGTGDs?

We show:

- QA is undecidable for rich DLs and FGTGDs
- QA with rich DLs is decidable for some new FGTGD classes
- Functional dependencies can be added under some conditions even to higher-arity relations

Undecidability of frontier-guarded plus DLs

Theorem

QA is undecidable for rich DLs and FGTGDs

Undecidability of frontier-guarded plus DLs

Theorem

QA is undecidable for rich DLs and FGTGDs

Problem:

- DLs can express Funct ($\leftrightarrow$ functional dependencies, FDs)
- FGTGDs can express inclusion dependencies (IDs)
  $\forall \mathbf{x}\,\mathbf{y}\ A(\mathbf{x}, \mathbf{y}) \to \exists \mathbf{z}\ B(\mathbf{y}, \mathbf{z})$ with no variable repetitions
- Implication of IDs and FDs is undecidable [Mitchell, 1983]
- Implication reduces to QA [Calì et al., 2003]

Undecidability of frontier-guarded plus DLs

Theorem

QA is undecidable for rich DLs and FGTGDs

Problem:

- DLs can express Funct ($\leftrightarrow$ functional dependencies, FDs)
- FGTGDs can express inclusion dependencies (IDs)
  $\forall \mathbf{x}\,\mathbf{y}\; A(\mathbf{x}, \mathbf{y}) \rightarrow \exists \mathbf{z}\; B(\mathbf{y}, \mathbf{z})$ with no variable repetitions
- Implication of IDs and FDs is undecidable [Mitchell, 1983]
- Implication reduces to QA [Calì et al., 2003]

$\rightarrow$ Restrict to frontier-one TGDs: $\forall \mathbf{x}\, y\; \phi(\mathbf{x}, y) \rightarrow \exists \mathbf{z}\; \psi(y, \mathbf{z})$

Undecidability of frontier-one plus DLs

- Restrict to frontier-one TGDs: $\forall \mathbf{x}\, y\ \phi(\mathbf{x}, y) \rightarrow \exists \mathbf{z}\ \psi(y, \mathbf{z})$
- QA for frontier-one IDs plus FDs is decidable (separability)

Undecidability of frontier-one plus DLs

- Restrict to frontier-one TGDs: $\forall \mathbf{x}\, y\ \phi(\mathbf{x}, y) \rightarrow \exists \mathbf{z}\ \psi(y, \mathbf{z})$
- QA for frontier-one IDs plus FDs is decidable (separability)

However:

Theorem

QA is undecidable for rich DLs and frontier-one TGDs

Undecidability of frontier-one plus DLs

- Restrict to frontier-one TGDs: $\forall \mathbf{x}\, y\ \phi(\mathbf{x}, y) \rightarrow \exists \mathbf{z}\ \psi(y, \mathbf{z})$
- QA for frontier-one IDs plus FDs is decidable (separability)

However:

Theorem

QA is undecidable for rich DLs and frontier-one TGDs

Problem:

- Rule heads and bodies may contain cycles
- We have Funct assertions

$\rightarrow$ We can build a grid and encode tiling problems

Undecidability of frontier-one plus DLs: proof

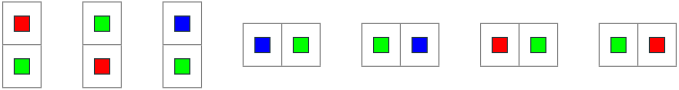
We reduce from tiling problems:

- finite set of colors: ■, ■, ■

Undecidability of frontier-one plus DLs: proof

We reduce from tiling problems:

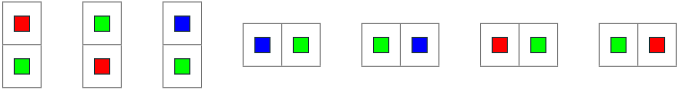
- finite set of colors: 
- horizontal and vertical allowed pairs:



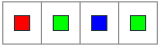
Undecidability of frontier-one plus DLs: proof

We reduce from tiling problems:

- finite set of colors: 
- horizontal and vertical allowed pairs:



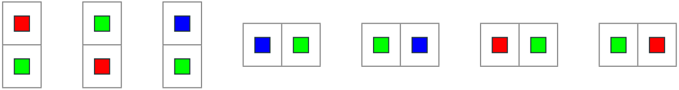
The tiling problem is:

- input: initial configuration: 

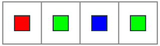
Undecidability of frontier-one plus DLs: proof

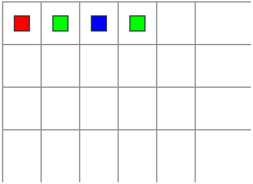
We reduce from tiling problems:

- finite set of colors: 
- horizontal and vertical allowed pairs:



The tiling problem is:

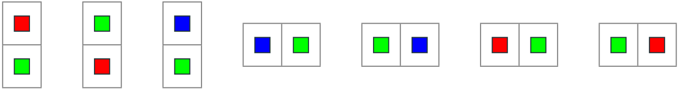
- input: initial configuration: 
- output: is there an infinite tiling?



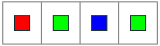
Undecidability of frontier-one plus DLs: proof

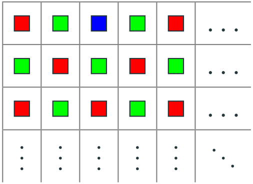
We reduce from tiling problems:

- finite set of colors: 
- horizontal and vertical allowed pairs:



The tiling problem is:

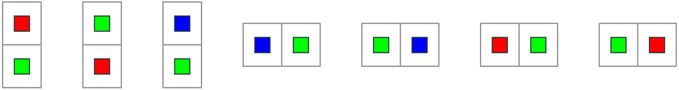
- input: initial configuration: 
- output: is there an infinite tiling?



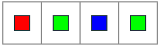
Undecidability of frontier-one plus DLs: proof

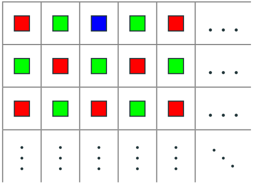
We reduce from tiling problems:

- finite set of colors: 
- horizontal and vertical allowed pairs:



The tiling problem is:

- input: initial configuration: 
- output: is there an infinite tiling?



→ Undecidable for some sets of colors and configurations

Undecidability of frontier-one plus DLs: proof, cont'd

- Functional relations D for down and R for right
- Unary predicate T for tiles and $C_{\blacksquare}$ for each color

Undecidability of frontier-one plus DLs: proof, cont'd

- Functional relations D for down and R for right
- Unary predicate T for tiles and $C_{■}$ for each color

Initial instance: $C_{■} \xrightarrow{R} C_{■} \xrightarrow{R} C_{■} \xrightarrow{R} C_{■}$

Undecidability of frontier-one plus DLs: proof, cont'd

- Functional relations D for down and R for right
- Unary predicate T for tiles and $C_{\color{gray}\blacksquare}$ for each color

Initial instance: $C_{\color{red}\blacksquare} \xrightarrow{R} C_{\color{green}\blacksquare} \xrightarrow{R} C_{\color{blue}\blacksquare} \xrightarrow{R} C_{\color{green}\blacksquare}$

Constraints:

- DL Disjunction to color tiles: $T \sqsubseteq C_{\color{red}\blacksquare} \sqcup C_{\color{green}\blacksquare} \sqcup C_{\color{blue}\blacksquare}$

Undecidability of frontier-one plus DLs: proof, cont'd

- Functional relations D for down and R for right
- Unary predicate T for tiles and $C_{\color{gray}\blacksquare}$ for each color

Initial instance: $C_{\color{red}\blacksquare} \xrightarrow{R} C_{\color{green}\blacksquare} \xrightarrow{R} C_{\color{blue}\blacksquare} \xrightarrow{R} C_{\color{green}\blacksquare}$

Constraints:

- DL Disjunction to color tiles: $T \sqsubseteq C_{\color{red}\blacksquare} \sqcup C_{\color{green}\blacksquare} \sqcup C_{\color{blue}\blacksquare}$
- Frontier-one TGD: $\forall y\ T(y) \Rightarrow \exists tzw$
$$\begin{tikzcd} T(y) \arrow[r, "R"] \arrow[d, "D"] & T(t) \arrow[d, "D"] \\ T(z) \arrow[r, "R"] & T(w) \end{tikzcd}$$

Undecidability of frontier-one plus DLs: proof, cont'd

- Functional relations D for down and R for right
- Unary predicate T for tiles and $C_{\blacksquare}$ for each color

Initial instance: $C_{\blacksquare} \xrightarrow{R} C_{\blacksquare} \xrightarrow{R} C_{\blacksquare} \xrightarrow{R} C_{\blacksquare}$

Constraints:

- DL Disjunction to color tiles: $T \sqsubseteq C_{\blacksquare} \sqcup C_{\blacksquare} \sqcup C_{\blacksquare}$
- Frontier-one TGD: $\forall y\ T(y) \Rightarrow \exists tzw$ $\begin{tikzcd} T(y) \arrow[r, "R"] \arrow[d, "D"] & T(t) \arrow[d, "D"] \\ T(z) \arrow[r, "R"] & T(w) \end{tikzcd}$

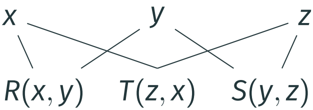
Query: $\exists x\, y\ C_{\blacksquare}(x) \xrightarrow{R} C_{\blacksquare}(y)$ for all forbidden pairs

$\rightarrow$ There is an extension of the instance iff there is a tiling

Decidability of non-looping frontier-one and DLs

Idea: prohibit cycles in FGTGDs

Decidability of non-looping frontier-one and DLs



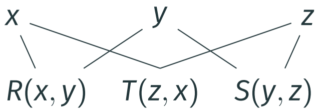
Idea: prohibit cycles in FGTGDs

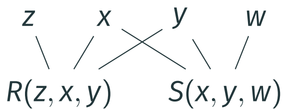
- $R(x, y)$ $S(y, z)$ $T(z, x)$ is a cycle

Decidability of non-looping frontier-one and DLs

Idea: prohibit cycles in FGTGDs

- $R(x,y)\ S(y,z)\ T(z,x)$ is a cycle
- $R(z,x,y)\ S(x,y,w)$ is also a cycle

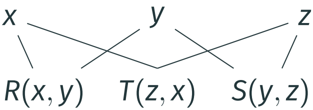


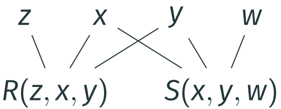


Decidability of non-looping frontier-one and DLs

Idea: prohibit cycles in FGTGDs

- $R(x, y)\ S(y, z)\ T(z, x)$ is a cycle
- $R(z, x, y)\ S(x, y, w)$ is also a cycle

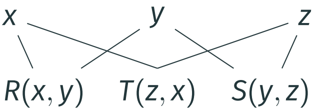




Formally:

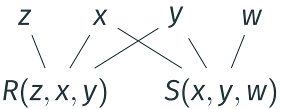
- Berge cycle: cycle in the atom–variable incidence graph

Decidability of non-looping frontier-one and DLs



Idea: prohibit cycles in FGTGDs

- $R(x, y)\ S(y, z)\ T(z, x)$ is a cycle
- $R(z, x, y)\ S(x, y, w)$ is also a cycle



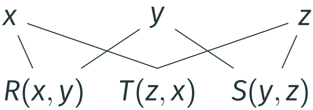
Formally:

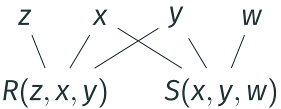
- Berge cycle: cycle in the atom–variable incidence graph
- Non-looping conjunction: no cycle except, e.g., $R(x, y)\ S(x, y)$
- Non-looping frontier-one: non-looping body and head

Decidability of non-looping frontier-one and DLs

Idea: prohibit cycles in FGTGDs

- $R(x,y)\ S(y,z)\ T(z,x)$ is a cycle
- $R(z,x,y)\ S(x,y,w)$ is also a cycle





Formally:

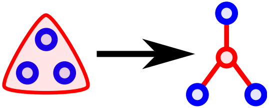
- Berge cycle: cycle in the atom–variable incidence graph
- Non-looping conjunction: no cycle except, e.g., $R(x,y)\ S(x,y)$
- Non-looping frontier-one: non-looping body and head

Theorem

QA is decidable for non-looping frontier-one TGDs + rich DLs

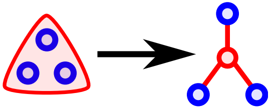
Decidability of non-looping frontier-one and DLs (proof)

- Shred $R(a, b, c)$ to $R_1(f, a), R_2(f, b), R_3(f, c)$



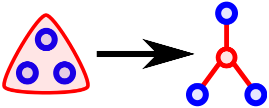
Decidability of non-looping frontier-one and DLs (proof)

- Shred $R(a, b, c)$ to $R_1(f, a), R_2(f, b), R_3(f, c)$
- Axiomatize the R_i, e.g., $\forall f\ \exists^{=1} x\ R_1(f, x)$



Decidability of non-looping frontier-one and DLs (proof)

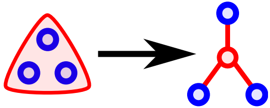
- Shred $R(a, b, c)$ to $R_1(f, a), R_2(f, b), R_3(f, c)$
- Axiomatize the R_i, e.g., $\forall f\ \exists^{=1} x\ R_1(f, x)$



$\rightarrow$ QA for the shredded instance, TGDs, query, and axioms is equivalent to QA for the original instance, TGDs, query

Decidability of non-looping frontier-one and DLs (proof)

- Shred $R(a, b, c)$ to $R_1(f, a), R_2(f, b), R_3(f, c)$
- Axiomatize the R_i, e.g., $\forall f\ \exists^{=1} x\ R_1(f, x)$

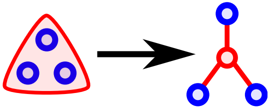


$\rightarrow$ QA for the shredded instance, TGDs, query, and axioms is equivalent to QA for the original instance, TGDs, query

- Rewrite shredded non-looping frontier-one TGDs to GC^2:
  - Rewrite $\forall \mathbf{x} y\ \phi(\mathbf{x}, y) \Rightarrow \exists \mathbf{z}\ \psi(y, \mathbf{z})$ to $\forall y\ \phi'(y) \Rightarrow \psi'(y)$,

Decidability of non-looping frontier-one and DLs (proof)

- Shred $R(a, b, c)$ to $R_1(f, a), R_2(f, b), R_3(f, c)$
- Axiomatize the R_i, e.g., $\forall f\ \exists^{=1} x\ R_1(f, x)$

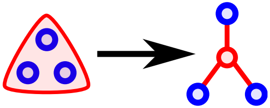


$\rightarrow$ QA for the shredded instance, TGDs, query, and axioms is equivalent to QA for the original instance, TGDs, query

- Rewrite shredded non-looping frontier-one TGDs to GC^2:
  - Rewrite $\forall \mathbf{x} y\ \phi(\mathbf{x}, y) \Rightarrow \exists \mathbf{z}\ \psi(y, \mathbf{z})$ to $\forall y\ \phi'(y) \Rightarrow \psi'(y)$, with $\phi'(y)$ and $\psi'(y)$ the shredding of $\exists \mathbf{x}\, \phi(\mathbf{x}, y)$ and $\exists \mathbf{z}\, \psi(y, \mathbf{z})$

Decidability of non-looping frontier-one and DLs (proof)

- Shred $R(a, b, c)$ to $R_1(f, a), R_2(f, b), R_3(f, c)$
- Axiomatize the R_i, e.g., $\forall f \ \exists^{=1} x \ R_1(f, x)$

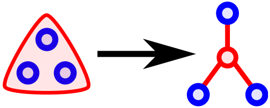


$\rightarrow$ QA for the shredded instance, TGDs, query, and axioms is equivalent to QA for the original instance, TGDs, query

- Rewrite shredded non-looping frontier-one TGDs to GC^2:
  - Rewrite $\forall \mathbf{x} y \ \phi(\mathbf{x}, y) \Rightarrow \exists \mathbf{z} \ \psi(y, \mathbf{z})$ to $\forall y \ \phi'(y) \Rightarrow \psi'(y)$, with $\phi'(y)$ and $\psi'(y)$ the shredding of $\exists \mathbf{x} \, \phi(\mathbf{x}, y)$ and $\exists \mathbf{z} \, \psi(y, \mathbf{z})$
  - Example: $\phi(y) := \exists x_1 x_2 \ T(y, x_1) \wedge R(y, y, x_2) \wedge A(x_2)$

Decidability of non-looping frontier-one and DLs (proof)

- Shred $R(a, b, c)$ to $R_1(f, a), R_2(f, b), R_3(f, c)$
- Axiomatize the R_i, e.g., $\forall f \; \exists^{=1} x \; R_1(f, x)$

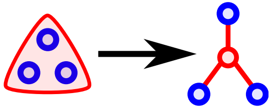


$\rightarrow$ QA for the shredded instance, TGDs, query, and axioms is equivalent to QA for the original instance, TGDs, query

- Rewrite shredded non-looping frontier-one TGDs to GC²:
  - Rewrite $\forall \mathbf{x} y \; \phi(\mathbf{x}, y) \Rightarrow \exists \mathbf{z} \; \psi(y, \mathbf{z})$ to $\forall y \; \phi'(y) \Rightarrow \psi'(y)$, with $\phi'(y)$ and $\psi'(y)$ the shredding of $\exists \mathbf{x} \, \phi(\mathbf{x}, y)$ and $\exists \mathbf{z} \, \psi(y, \mathbf{z})$
  - Example: $\phi(y) := \exists x_1 x_2 \; T(y, x_1) \wedge R(y, y, x_2) \wedge A(x_2)$
    $\rightarrow \; \exists x_1 \, x_2 \, f \; T(y, x_1) \wedge R_1(f, y) \wedge R_2(f, y) \wedge R_3(f, x_2) \wedge A(x_2)$

Decidability of non-looping frontier-one and DLs (proof)

- Shred $R(a,b,c)$ to $R_1(f,a), R_2(f,b), R_3(f,c)$
- Axiomatize the R_i, e.g., $\forall f \; \exists^{=1} x \; R_1(f,x)$

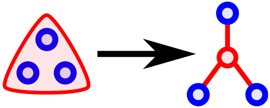


$\rightarrow$ QA for the shredded instance, TGDs, query, and axioms is equivalent to QA for the original instance, TGDs, query

- Rewrite shredded non-looping frontier-one TGDs to GC²:
  - Rewrite $\forall \mathbf{x} y \; \phi(\mathbf{x}, y) \Rightarrow \exists \mathbf{z} \; \psi(y, \mathbf{z})$ to $\forall y \; \phi'(y) \Rightarrow \psi'(y)$, with $\phi'(y)$ and $\psi'(y)$ the shredding of $\exists \mathbf{x} \, \phi(\mathbf{x}, y)$ and $\exists \mathbf{z} \, \psi(y, \mathbf{z})$
  - Example: $\phi(y) := \exists x_1 x_2 \; T(y, x_1) \wedge R(y, y, x_2) \wedge A(x_2)$
    - $\rightarrow \exists x_1 \, x_2 \, f \; T(y, x_1) \wedge R_1(f, y) \wedge R_2(f, y) \wedge R_3(f, x_2) \wedge A(x_2)$
    - $\rightarrow \left(\exists x_1 \; T(y, x_1)\right)$

Decidability of non-looping frontier-one and DLs (proof)

- Shred $R(a, b, c)$ to $R_1(f, a), R_2(f, b), R_3(f, c)$
- Axiomatize the R_i, e.g., $\forall f\ \exists^{=1} x\ R_1(f, x)$



$\rightarrow$ QA for the shredded instance, TGDs, query, and axioms is equivalent to QA for the original instance, TGDs, query

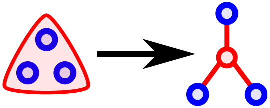
- Rewrite shredded non-looping frontier-one TGDs to GC^2:
  - Rewrite $\forall \mathbf{x} y\ \phi(\mathbf{x}, y) \Rightarrow \exists \mathbf{z}\ \psi(y, \mathbf{z})$ to $\forall y\ \phi'(y) \Rightarrow \psi'(y)$, with $\phi'(y)$ and $\psi'(y)$ the shredding of $\exists \mathbf{x}\, \phi(\mathbf{x}, y)$ and $\exists \mathbf{z}\, \psi(y, \mathbf{z})$
  - Example: $\phi(y) := \exists x_1 x_2\ T(y, x_1) \wedge R(y, y, x_2) \wedge A(x_2)$

    $\rightarrow\ \exists x_1\, x_2\, f\ T(y, x_1) \wedge R_1(f, y) \wedge R_2(f, y) \wedge R_3(f, x_2) \wedge A(x_2)$

    $\rightarrow\ \Big(\exists x_1\ T(y, x_1)\Big) \wedge \Big(\exists x_2\, f\ R_1(f, y) \wedge R_2(f, y)$

Decidability of non-looping frontier-one and DLs (proof)

- Shred $R(a, b, c)$ to $R_1(f, a), R_2(f, b), R_3(f, c)$
- Axiomatize the R_i, e.g., $\forall f\ \exists^{=1} x\ R_1(f, x)$

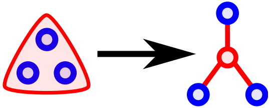


$\rightarrow$ QA for the shredded instance, TGDs, query, and axioms is equivalent to QA for the original instance, TGDs, query

- Rewrite shredded non-looping frontier-one TGDs to GC^2:
  - Rewrite $\forall \mathbf{x} y\ \phi(\mathbf{x}, y) \Rightarrow \exists \mathbf{z}\ \psi(y, \mathbf{z})$ to $\forall y\ \phi'(y) \Rightarrow \psi'(y)$, with $\phi'(y)$ and $\psi'(y)$ the shredding of $\exists \mathbf{x}\, \phi(\mathbf{x}, y)$ and $\exists \mathbf{z}\, \psi(y, \mathbf{z})$
  - Example: $\phi(y) := \exists x_1 x_2\ T(y, x_1) \land R(y, y, x_2) \land A(x_2)$
    $\rightarrow\ \exists x_1\, x_2\, f\ T(y, x_1) \land R_1(f, y) \land R_2(f, y) \land R_3(f, x_2) \land A(x_2)$
    $\rightarrow\ \Big(\exists x_1\ T(y, x_1)\Big) \land \Big(\exists x_2\, f\ R_1(f, y) \land R_2(f, y) \land \big(\exists y_2\ R_3(f, y_2) \land A(y_2)\big)\Big)$

Decidability of non-looping frontier-one and DLs (proof)

- Shred $R(a, b, c)$ to $R_1(f, a), R_2(f, b), R_3(f, c)$
- Axiomatize the R_i, e.g., $\forall f\ \exists^{=1} x\ R_1(f, x)$



$\rightarrow$ QA for the shredded instance, TGDs, query, and axioms is equivalent to QA for the original instance, TGDs, query

- Rewrite shredded non-looping frontier-one TGDs to $\mathbf{GC}^2$:
  - Rewrite $\forall \mathbf{x} y\ \phi(\mathbf{x}, y) \Rightarrow \exists \mathbf{z}\ \psi(y, \mathbf{z})$ to $\forall y\ \phi'(y) \Rightarrow \psi'(y)$, with $\phi'(y)$ and $\psi'(y)$ the shredding of $\exists \mathbf{x}\, \phi(\mathbf{x}, y)$ and $\exists \mathbf{z}\, \psi(y, \mathbf{z})$
  - Example: $\phi(y) := \exists x_1 x_2\ T(y, x_1) \wedge R(y, y, x_2) \wedge A(x_2)$
    - $\rightarrow$ $\exists x_1\, x_2\, f\ T(y, x_1) \wedge R_1(f, y) \wedge R_2(f, y) \wedge R_3(f, x_2) \wedge A(x_2)$
    - $\rightarrow$ $\left(\exists x_1\ T(y, x_1)\right) \wedge \left(\exists x_2\, f\ R_1(f, y) \wedge R_2(f, y) \wedge (\exists y_2\ R_3(f, y_2) \wedge A(y_2))\right)$

  $\rightarrow$ Reduces to QA for $\mathbf{GC}^2$: decidable [Pratt-Hartmann, 2009]

Decidability of head-non-looping frontier-one and DLs

Head-non-looping frontier-one TGDs: no cycles in head

Decidability of head-non-looping frontier-one and DLs

Head-non-looping frontier-one TGDs: no cycles in head

Theorem

QA is decidable for head-non-looping frontier-one TGDs + rich DLs

Decidability of head-non-looping frontier-one and DLs

Head-non-looping frontier-one TGDs: no cycles in head

Theorem

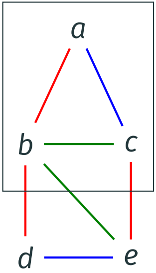
QA is decidable for head-non-looping frontier-one TGDs + rich DLs

Basic idea:

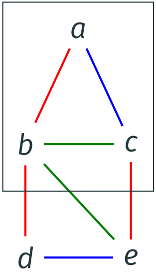
- If there is a counterexample model to QA, we can unravel it
  - $\rightarrow$ It is still a counterexample
  - $\rightarrow$ It has no cycles (except in the instance part)

$\rightarrow$ Looping TGD bodies can only match on the instance part so non-looping frontier-one TGDs can be made head-non-looping

Head-non-looping frontier-one and DLs: unraveling

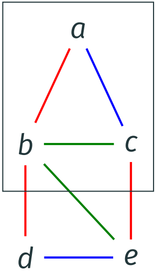


Head-non-looping frontier-one and DLs: unraveling

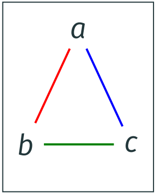


$\Rightarrow$

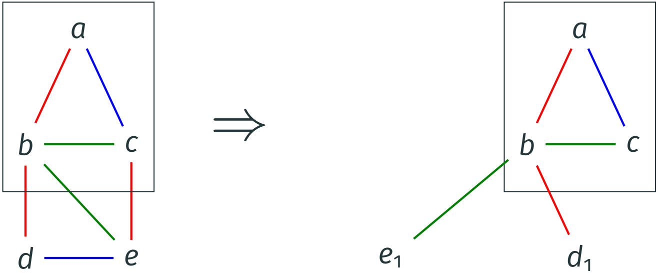
Head-non-looping frontier-one and DLs: unraveling



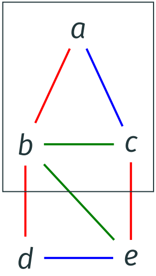
$\Longrightarrow$



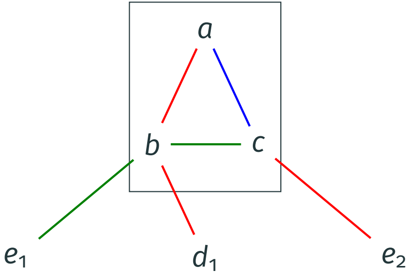
Head-non-looping frontier-one and DLs: unraveling



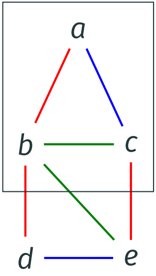
Head-non-looping frontier-one and DLs: unraveling



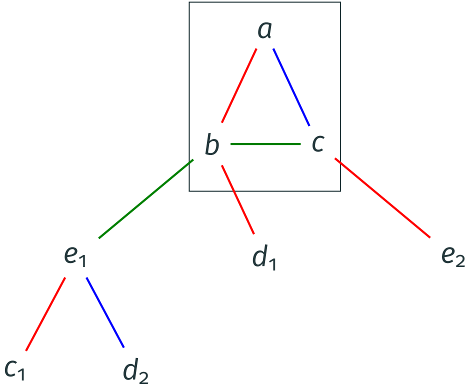
$\Rightarrow$



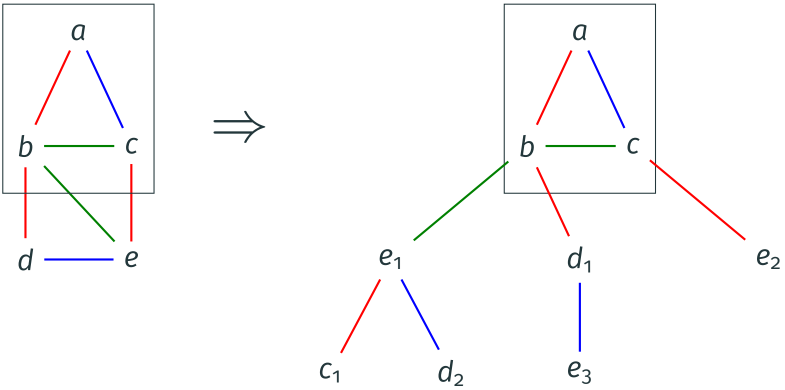
Head-non-looping frontier-one and DLs: unraveling



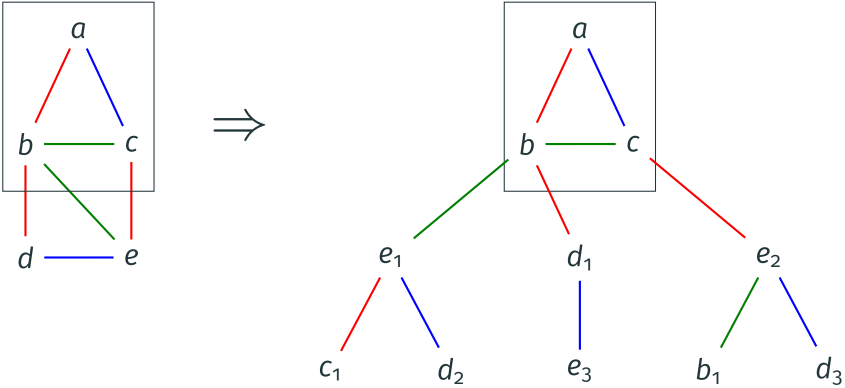
$\Longrightarrow$



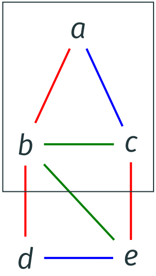
Head-non-looping frontier-one and DLs: unraveling



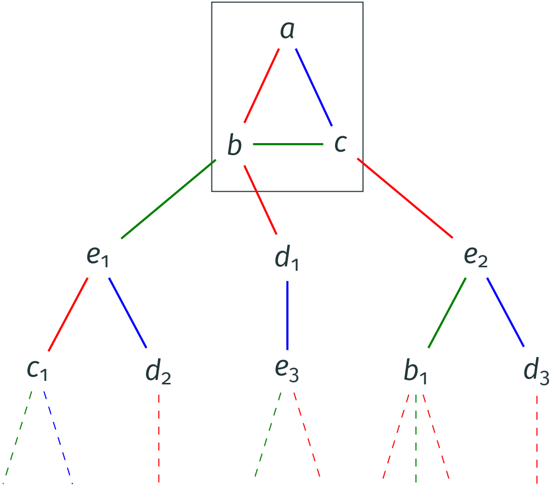
Head-non-looping frontier-one and DLs: unraveling



Head-non-looping frontier-one and DLs: unraveling



$\Longrightarrow$



Head-non-looping frontier-one and DLs: unraveling

For every frontier-one TGD with a looping body:

Head-non-looping frontier-one and DLs: unraveling

For every frontier-one TGD with a looping body:

- Consider all possible self-homomorphisms of the body
  - $\rightarrow$ Ex.: $R(x, y) \wedge S(y, z) \wedge T(z, x)$ gives $R(x, y) \wedge S(y, x) \wedge T(x, x)$

Head-non-looping frontier-one and DLs: unraveling

For every frontier-one TGD with a looping body:

- Consider all possible self-homomorphisms of the body
  - $\rightarrow$ Ex.: $R(x, y) \land S(y, z) \land T(z, x)$ gives $R(x, y) \land S(y, x) \land T(x, x)$
- Consider all possible mappings to the instance
  - $\rightarrow$ Ex.: $R(x, y) \land S(y, z) \land T(z, x)$ gives $R(x, y) \land S(y', z) \land T(z', x')$ $\land x = a \land x' = a \land y = b \land y' = b \land z = c \land z' = c$

Head-non-looping frontier-one and DLs: unraveling

For every frontier-one TGD with a looping body:

- Consider all possible self-homomorphisms of the body
  - $\rightarrow$ Ex.: $R(x,y) \wedge S(y,z) \wedge T(z,x)$ gives $R(x,y) \wedge S(y,x) \wedge T(x,x)$
- Consider all possible mappings to the instance
  - $\rightarrow$ Ex.: $R(x,y) \wedge S(y,z) \wedge T(z,x)$ gives $R(x,y) \wedge S(y',z) \wedge T(z',x')$ $\wedge\, x = a \wedge x' = a \wedge y = b \wedge y' = b \wedge z = c \wedge z' = c$

$\rightarrow$ Keep the resulting fully non-looping TGDs

Head-non-looping frontier-one and DLs: unraveling

For every frontier-one TGD with a looping body:

- Consider all possible self-homomorphisms of the body
  - $\rightarrow$ Ex.: $R(x, y) \wedge S(y, z) \wedge T(z, x)$ gives $R(x, y) \wedge S(y, x) \wedge T(x, x)$
- Consider all possible mappings to the instance
  - $\rightarrow$ Ex.: $R(x, y) \wedge S(y, z) \wedge T(z, x)$ gives $R(x, y) \wedge S(y', z) \wedge T(z', x')$ $\wedge\, x = a \wedge x' = a \wedge y = b \wedge y' = b \wedge z = c \wedge z' = c$

$\rightarrow$ Keep the resulting fully non-looping TGDs

QA for the shredded instance, treeified TGDs, query, and axioms is equivalent to QA for the original instance, TGDs, query

Adding functional dependencies

We have shown:

Theorem

QA is decidable for head-non-looping frontier-one TGDs + rich DLs

Adding functional dependencies

We have shown:

Theorem

QA is decidable for head-non-looping frontier-one TGDs + rich DLs

- We have functional dependencies $\mathrm{Funct}(R)$ on binary relations
- Could we also allow FDs on higher-arity relations?
  Ex.: $\mathrm{Talk}[\mathit{speaker}, \mathit{session}]$ determines $\mathrm{Talk}[\mathit{title}]$

Methods for TGDs + higher-arity FDs (no DLs)

- Consider QA under TGDs Σ and FDs Φ
- Σ and Φ are separable if $\mathrm{QA}(\Sigma, \Phi) \Leftrightarrow \mathrm{QA}(\Sigma)$ when $I \models \Phi$

Methods for TGDs + higher-arity FDs (no DLs)

- Consider QA under TGDs Σ and FDs Φ
- Σ and Φ are separable if $\mathrm{QA}(\Sigma, \Phi) \Leftrightarrow \mathrm{QA}(\Sigma)$ when $I \models \Phi$

$\rightarrow$ Separable higher-arity FDs can be ignored during QA

Methods for TGDs + higher-arity FDs (no DLs)

- Consider QA under TGDs Σ and FDs Φ
- Σ and Φ are separable if $\mathrm{QA}(\Sigma, \Phi) \Leftrightarrow \mathrm{QA}(\Sigma)$ when $I \models \Phi$

$\rightarrow$ Separable higher-arity FDs can be ignored during QA

- Inclusion dependencies (IDs) and FDs generally not separable
- Frontier-one IDs and FDs are always separable
- [New:] Frontier-one TGDs with single-atom body and head (i.e., IDs with variable repetitions) and FDs are not separable and QA is undecidable for them (in our paper)

Non-conflicting TGDs and FDs [Calì et al., 2012]

Non-conflicting condition: sufficient condition for separability of single-head TGDs Σ and FDs Φ,

- For every TGD head $H = R(x_1, \ldots, x_n)$:
  - $S :=$ positions of H with a frontier variable
  - $\overline{S} :=$ positions with an existentially quantified variable

Non-conflicting TGDs and FDs [Calì et al., 2012]

Non-conflicting condition: sufficient condition for separability of single-head TGDs Σ and FDs Φ,

- For every TGD head $H = R(x_1, \ldots, x_n)$:
  - $S :=$ positions of H with a frontier variable
  - $\overline{S} :=$ positions with an existentially quantified variable
- For each FD $R[S'] \to R[i]$ of Φ:

Non-conflicting TGDs and FDs [Calì et al., 2012]

Non-conflicting condition: sufficient condition for separability of single-head TGDs Σ and FDs Φ,

- For every TGD head $H = R(x_1, \ldots, x_n)$:
  - $S :=$ positions of H with a frontier variable
  - $\overline{S} :=$ positions with an existentially quantified variable
- For each FD $R[S'] \to R[i]$ of Φ:

$\to$ if $S' \subsetneq S$, fail

Non-conflicting TGDs and FDs [Calì et al., 2012]

Non-conflicting condition: sufficient condition for separability of single-head TGDs Σ and FDs Φ,

- For every TGD head $H = R(x_1, \ldots, x_n)$:
  - $S :=$ positions of H with a frontier variable
  - $\overline{S} :=$ positions with an existentially quantified variable
- For each FD $R[S'] \to R[i]$ of Φ:

$\to$ if $S' \subsetneq S$, fail

$\to$ if $S' = S$ and some variable occurs twice in $\overline{S}$, fail

Non-conflicting TGDs and FDs [Calì et al., 2012]

Non-conflicting condition: sufficient condition for separability of single-head TGDs Σ and FDs Φ,

- For every TGD head $H = R(x_1, \ldots, x_n)$:
  - $S :=$ positions of H with a frontier variable
  - $\overline{S} :=$ positions with an existentially quantified variable
- For each FD $R[S'] \rightarrow R[i]$ of Φ:

$\rightarrow$ if $S' \subsetneq S$, fail

$\rightarrow$ if $S' = S$ and some variable occurs twice in $\overline{S}$, fail

Examples: for the FD $R[1] \rightarrow R[3]$:

Non-conflicting TGDs and FDs [Calì et al., 2012]

Non-conflicting condition: sufficient condition for separability of single-head TGDs Σ and FDs Φ,

- For every TGD head $H = R(x_1, \ldots, x_n)$:
  - $S :=$ positions of H with a frontier variable
  - $\overline{S} :=$ positions with an existentially quantified variable
- For each FD $R[S'] \to R[i]$ of Φ:

$\to$ if $S' \subsetneq S$, fail

$\to$ if $S' = S$ and some variable occurs twice in $\overline{S}$, fail

Examples: for the FD $R[1] \to R[3]$:

- $T(x) \Rightarrow R(y, y, x)$ is...

Non-conflicting TGDs and FDs [Calì et al., 2012]

Non-conflicting condition: sufficient condition for separability of single-head TGDs Σ and FDs Φ,

- For every TGD head $H = R(x_1, \ldots, x_n)$:
  - $S :=$ positions of H with a frontier variable
  - $\overline{S} :=$ positions with an existentially quantified variable
- For each FD $R[S'] \rightarrow R[i]$ of Φ:

$\rightarrow$ if $S' \subsetneq S$, fail

$\rightarrow$ if $S' = S$ and some variable occurs twice in $\overline{S}$, fail

Examples: for the FD $R[1] \rightarrow R[3]$:

- $T(x) \Rightarrow R(y, y, x)$ is... non-conflicting

Non-conflicting TGDs and FDs [Calì et al., 2012]

Non-conflicting condition: sufficient condition for separability of single-head TGDs Σ and FDs Φ,

- For every TGD head $H = R(x_1, \ldots, x_n)$:
  - $S :=$ positions of H with a frontier variable
  - $\overline{S} :=$ positions with an existentially quantified variable
- For each FD $R[S'] \to R[i]$ of Φ:

$\rightarrow$ if $S' \subsetneq S$, fail

$\rightarrow$ if $S' = S$ and some variable occurs twice in $\overline{S}$, fail

Examples: for the FD $R[1] \to R[3]$:

- $T(x) \Rightarrow R(y, y, x)$ is... non-conflicting
- $T(x, y) \Rightarrow R(x, y, z)$ is...

Non-conflicting TGDs and FDs [Calì et al., 2012]

Non-conflicting condition: sufficient condition for separability of single-head TGDs Σ and FDs Φ,

- For every TGD head $H = R(x_1, \ldots, x_n)$:
  - $S :=$ positions of H with a frontier variable
  - $\overline{S} :=$ positions with an existentially quantified variable
- For each FD $R[S'] \rightarrow R[i]$ of Φ:

$\rightarrow$ if $S' \subsetneq S$, fail

$\rightarrow$ if $S' = S$ and some variable occurs twice in $\overline{S}$, fail

Examples: for the FD $R[1] \rightarrow R[3]$:

- $T(x) \Rightarrow R(y, y, x)$ is... non-conflicting
- $T(x, y) \Rightarrow R(x, y, z)$ is... conflicting (superset)

Non-conflicting TGDs and FDs [Calì et al., 2012]

Non-conflicting condition: sufficient condition for separability of single-head TGDs Σ and FDs Φ,

- For every TGD head $H = R(x_1, \ldots, x_n)$:
  - $S :=$ positions of H with a frontier variable
  - $\overline{S} :=$ positions with an existentially quantified variable
- For each FD $R[S'] \to R[i]$ of Φ:

$\rightarrow$ if $S' \subsetneq S$, fail

$\rightarrow$ if $S' = S$ and some variable occurs twice in $\overline{S}$, fail

Examples: for the FD $R[1] \to R[3]$:

- $T(x) \Rightarrow R(y, y, x)$ is... non-conflicting
- $T(x, y) \Rightarrow R(x, y, z)$ is... conflicting (superset)
- $T(x) \Rightarrow R(x, y, z)$ is...

Non-conflicting TGDs and FDs [Calì et al., 2012]

Non-conflicting condition: sufficient condition for separability of single-head TGDs Σ and FDs Φ,

- For every TGD head $H = R(x_1, \ldots, x_n)$:
  - $S :=$ positions of H with a frontier variable
  - $\overline{S} :=$ positions with an existentially quantified variable
- For each FD $R[S'] \rightarrow R[i]$ of Φ:

$\rightarrow$ if $S' \subsetneq S$, fail

$\rightarrow$ if $S' = S$ and some variable occurs twice in $\overline{S}$, fail

Examples: for the FD $R[1] \rightarrow R[3]$:

- $T(x) \Rightarrow R(y, y, x)$ is... non-conflicting
- $T(x, y) \Rightarrow R(x, y, z)$ is... conflicting (superset)
- $T(x) \Rightarrow R(x, y, z)$ is... non-conflicting

Non-conflicting TGDs and FDs [Calì et al., 2012]

Non-conflicting condition: sufficient condition for separability of single-head TGDs Σ and FDs Φ,

- For every TGD head $H = R(x_1, \ldots, x_n)$:
  - $S :=$ positions of H with a frontier variable
  - $\overline{S} :=$ positions with an existentially quantified variable
- For each FD $R[S'] \rightarrow R[i]$ of Φ:

$\rightarrow$ if $S' \subsetneq S$, fail

$\rightarrow$ if $S' = S$ and some variable occurs twice in $\overline{S}$, fail

Examples: for the FD $R[1] \rightarrow R[3]$:

- $T(x) \Rightarrow R(y, y, x)$ is... non-conflicting
- $T(x, y) \Rightarrow R(x, y, z)$ is... conflicting (superset)
- $T(x) \Rightarrow R(x, y, z)$ is... non-conflicting
- $T(x) \Rightarrow R(x, y, y)$ is...

Non-conflicting TGDs and FDs [Calì et al., 2012]

Non-conflicting condition: sufficient condition for separability of single-head TGDs Σ and FDs Φ,

- For every TGD head $H = R(x_1, \ldots, x_n)$:
  - $S :=$ positions of H with a frontier variable
  - $\overline{S} :=$ positions with an existentially quantified variable
- For each FD $R[S'] \rightarrow R[i]$ of Φ:

$\rightarrow$ if $S' \subsetneq S$, fail

$\rightarrow$ if $S' = S$ and some variable occurs twice in $\overline{S}$, fail

Examples: for the FD $R[1] \rightarrow R[3]$:

- $T(x) \Rightarrow R(y, y, x)$ is... non-conflicting
- $T(x, y) \Rightarrow R(x, y, z)$ is... conflicting (superset)
- $T(x) \Rightarrow R(x, y, z)$ is... non-conflicting
- $T(x) \Rightarrow R(x, y, y)$ is... conflicting (variable reuse)

Non-conflicting TGDs and FDs [Calì et al., 2012]

Non-conflicting condition: sufficient condition for separability of single-head TGDs Σ and FDs Φ,

- For every TGD head $H = R(x_1, \ldots, x_n)$:
  - $S :=$ positions of H with a frontier variable
  - $\overline{S} :=$ positions with an existentially quantified variable
- For each FD $R[S'] \to R[i]$ of Φ:

$\rightarrow$ if $S' \subsetneq S$, fail

$\rightarrow$ if $S' = S$ and some variable occurs twice in $\overline{S}$, fail

Examples: for the FD $R[1] \to R[3]$:

- $T(x) \Rightarrow R(y, y, x)$ is... non-conflicting
- $T(x, y) \Rightarrow R(x, y, z)$ is... conflicting (superset)
- $T(x) \Rightarrow R(x, y, z)$ is... non-conflicting
- $T(x) \Rightarrow R(x, y, y)$ is... conflicting (variable reuse)
- $T(y) \Rightarrow R(x, y, z)U(z)$ is...

Non-conflicting TGDs and FDs [Calì et al., 2012]

Non-conflicting condition: sufficient condition for separability of single-head TGDs Σ and FDs Φ,

- For every TGD head $H = R(x_1, \ldots, x_n)$:
  - $S :=$ positions of H with a frontier variable
  - $\overline{S} :=$ positions with an existentially quantified variable
- For each FD $R[S'] \to R[i]$ of Φ:

$\to$ if $S' \subsetneq S$, fail

$\to$ if $S' = S$ and some variable occurs twice in $\overline{S}$, fail

Examples: for the FD $R[1] \to R[3]$:

- $T(x) \Rightarrow R(y, y, x)$ is... non-conflicting
- $T(x, y) \Rightarrow R(x, y, z)$ is... conflicting (superset)
- $T(x) \Rightarrow R(x, y, z)$ is... non-conflicting
- $T(x) \Rightarrow R(x, y, y)$ is... conflicting (variable reuse)
- $T(y) \Rightarrow R(x, y, z)U(z)$ is... not allowed (not single-head)

Decidability for non-conflicting FDs

We know from [Calì et al., 2012]:

Theorem

QA decidable for single-head frontier-guarded + non-conflicting FDs

Decidability for non-conflicting FDs

We know from [Calì et al., 2012]:

Theorem

*QA **decidable** for single-head frontier-guarded + non-conflicting FDs*

We have shown before, without higher-arity FDs:

Theorem

*QA is **decidable** for head-non-looping frontier-one TGDs + rich DLs*

Decidability for non-conflicting FDs

We know from [Calì et al., 2012]:

Theorem

QA decidable for single-head frontier-guarded + non-conflicting FDs

We have shown before, without higher-arity FDs:

Theorem

QA is decidable for head-non-looping frontier-one TGDs + rich DLs

We show that we can combine these two results:

Theorem

QA is decidable for:

- *Rich DL constraints (with Funct)*
- *Single-head (hence, head-non-looping) frontier-one TGDs*
- *Non-conflicting FDs (on higher-arity predicates)*

Summary of results

Combining Existential Rules and Description Logics

Summary of results
Combining Existential Rules and Description Logics

- Setting: Open-world query answering (QA) under:
  - Frontier-guarded TGDs (FGTGDs)
  - Rich DL constraints
- Question: For which FGTGD classes is QA decidable with rich DLs?
- Answers:

Summary of results
Combining Existential Rules and Description Logics

- Setting: Open-world query answering (QA) under:
  - Frontier-guarded TGDs (FGTGDs)
  - Rich DL constraints
- Question: For which FGTGD classes is QA decidable with rich DLs?
- Answers:

  $\rightarrow$ We must restrict to frontier-one TGDs

  $\rightarrow$ We must prohibit cycles in TGD heads

Summary of results
Combining Existential Rules and Description Logics

- Setting: Open-world query answering (QA) under:
  - Frontier-guarded TGDs (FGTGDs)
  - Rich DL constraints
- Question: For which FGTGD classes is QA decidable with rich DLs?
- Answers:
  - $\rightarrow$ We must restrict to frontier-one TGDs
  - $\rightarrow$ We must prohibit cycles in TGD heads
  - $\rightarrow$ QA is decidable for head-non-looping frontier-one + rich DLs
  - $\rightarrow$ We can also add non-conflicting FDs on higher-arity facts

Table of contents

Query answering with distinguished relations

We separate the **signature** σ (set of allowed relations) into:

- σ_{B}: the **base** relations (e.g., $\mathrm{Advisor}$)
- σ_{D}: the **distinguished** relations
  (e.g., $\mathrm{LocatedIn}$, required to be transitive)

Query answering with distinguished relations

We separate the signature σ (set of allowed relations) into:

- σ_{B}: the base relations (e.g., $\mathrm{Advisor}$)
- σ_{D}: the distinguished relations
  (e.g., $\mathrm{LocatedIn}$, required to be transitive)

- Distinguished relations have specific built-in requirements
  (which implicitly adds unguarded logical constraints)

  QAtr: QA where each distinguished relation is transitive

  QAtc: QA where each distinguished relation is the transitive closure of another relation

Undecidability

- Problem: QAtr already known to be undecidable with FGTGDs [Gottlob et al., 2013]

Undecidability

- Problem: QAtr already known to be undecidable with FGTGDs [Gottlob et al., 2013]

→ Solution: impose that guards are base relations

Undecidability

- **Problem:** QAtr already known to be undecidable with FGTGDs [Gottlob et al., 2013]

$\rightarrow$ Solution: impose that guards are base relations

$\rightarrow$ **Base** Frontier-guarded TGDs (BaseFGTGDs):
body ϕ has an atom with all frontier variables **y**
and this atom is for a base relation $R \in \sigma_{\mathbf{B}}$

$$\forall x\, y_1\, y_2\; S(x, y_1) \wedge S(x, y_2) \wedge R(y_1, y_2) \rightarrow \exists z\; S(y_2, z) \wedge T(y_1)$$

Theorem

The QAtr and QAtc problems are decidable for BaseFGTGDs in 2EXPTIME combined complexity and PTIME data complexity.

Idea: Reduce QAtr to QA for FGTGDs by "axiomatizing" transitivity: change constraints Σ to Σ' enforcing transitivity within facts

$$\forall x\, y_1\, y_2\; F(x, y_1, y_2) \wedge T(y_1, x) \wedge T(x, y_2) \rightarrow T(y_1, y_2)$$

Lemma: Some superinstances of I_0 satisfying Σ' and violating Q can be extended by completing transitive relations to be transitive so that they still contain I_0, satisfy Σ and violate Q

Order relations

QAlin: QA where each distinguished relation is a total order (antisymmetric, transitive, total)

Order relations

QAlin: QA where each distinguished relation is a total order (antisymmetric, transitive, total)

$$\forall x x' d d' \ \mathrm{Advisor}(x, x') \land \mathrm{GradDate}(x, d) \land \mathrm{GradDate}(x', d') \rightarrow d < d'$$

Order relations

QAlin: QA where each distinguished relation is a total order (antisymmetric, transitive, total)

$$\forall x x' d d' \ \mathrm{Advisor}(x, x') \land \mathrm{GradDate}(x, d) \land \mathrm{GradDate}(x', d') \rightarrow d < d'$$

- Problem: QAlin is undecidable for BaseFGTGDs!

$\rightarrow$ Intuition:

  - $x < y \lor y < x$ codes inequality $x \neq y$
  - QA with inequalities in the query is often undecidable [Gutiérrez-Basulto et al., 2013]

Recovering decidability with order

$\rightarrow$ Solution: impose that guards are base relations
and that distinguished facts in body are covered by base facts

Recovering decidability with order

→ Solution: impose that guards are base relations and that distinguished facts in body are covered by base facts

→ **Base-covered** Frontier-guarded TGDs (BaseCovFGTGDs): BaseFGTGDs (with base guard) s.t. for every body σ_{D}-atom $x < y$ there is a σ_{B}-atom using its variables

$$\forall x\, y_1\, y_2\; x < y_1 \wedge S(x, y_1) \wedge S(x, y_2) \wedge R(y_1, y_2) \rightarrow \exists z\; S(y_2, z) \wedge T(y_1)$$

Recovering decidability with order

$\rightarrow$ Solution: impose that guards are base relations and that distinguished facts in body are covered by base facts

$\rightarrow$ **Base-covered** Frontier-guarded TGDs (BaseCovFGTGDs): BaseFGTGDs (with base guard) s.t. for every body σ_{D}-atom $x < y$ there is a σ_{B}-atom using its variables

$$\forall x\, y_1\, y_2\; x < y_1 \wedge S(x, y_1) \wedge S(x, y_2) \wedge R(y_1, y_2) \rightarrow \exists z\; S(y_2, z) \wedge T(y_1)$$

Theorem

QAlin is decidable for BaseCovFGTGDs and base-covered queries

Complexity results
Query Answering with Transitive and Linear-Ordered Data

	QAtr		QAtc		QAlin	
	data	**combined**	**data**	**combined**	**data**	**combined**
BaseFGTGD	in coNP	2EXP-c	coNP-c	2EXP-c	undecidable	
BaseCovFGTGD	P-c	2EXP-c	coNP-c	2EXP-c	coNP-c	2EXP-c

Complexity results

Query Answering with Transitive and Linear-Ordered Data

	QAtr		QAtc		QAlin	
	data	**combined**	**data**	**combined**	**data**	**combined**
BaseFGTGD	in coNP	2EXP-c	coNP-c	2EXP-c	undecidable	
BaseCovFGTGD	P-c	2EXP-c	coNP-c	2EXP-c	coNP-c	2EXP-c

- Combined complexity no worse than vanilla QA for FGTGDs

Complexity results
Query Answering with Transitive and Linear-Ordered Data

	QAtr		**QAtc**		**QAlin**	
	data	**combined**	**data**	**combined**	**data**	**combined**
BaseFGTGD	in coNP	2EXP-c	coNP-c	2EXP-c	undecidable	
BaseCovFGTGD	P-c	2EXP-c	coNP-c	2EXP-c	coNP-c	2EXP-c

- Combined complexity no worse than vanilla QA for FGTGDs
- Only gap is BaseFGTGD and QAtr

Complexity results
Query Answering with Transitive and Linear-Ordered Data

	QAtr		**QAtc**		**QAlin**	
	data	**combined**	**data**	**combined**	**data**	**combined**
BaseFGTGD	in coNP	2EXP-c	coNP-c	2EXP-c	undecidable	
BaseCovFGTGD	P-c	2EXP-c	coNP-c	2EXP-c	coNP-c	2EXP-c

- Combined complexity no worse than vanilla QA for FGTGDs
- Only gap is BaseFGTGD and QAtr
- Data complexity for QAtc and QAlin goes from PTIME to coNP-c

Complexity results
Query Answering with Transitive and Linear-Ordered Data

	QAtr		**QAtc**		**QAlin**	
	data	**combined**	**data**	**combined**	**data**	**combined**
BaseFGTGD	in coNP	2EXP-c	coNP-c	2EXP-c	undecidable	
BaseCovFGTGD	P-c	2EXP-c	coNP-c	2EXP-c	coNP-c	2EXP-c

- Combined complexity no worse than vanilla QA for FGTGDs
- Only gap is BaseFGTGD and QAtr
- Data complexity for QAtc and QAlin goes from PTIME to coNP-c

$\rightarrow$ Intuition: QAtc and QAlin can code disjunctive IDs

QAtc: In $T^+(a, b)$, is the T-path of length 1, 2, 3, ...?

QAlin: In $A(a, b)$, does $a < b$ or $b < a$ or $a = b$?

Table of contents

Summary

We have studied QA with FGTGD constraints and:

- Number restrictions:
  - OK on low-arity when restricting FGTGD shapes
  - OK when further imposing non-conflicting condition
- Transitivity:
  - OK when not used as guards
- Orders:
  - OK when not used as guards and covered

Ongoing work:

- Successor relations of linear orders:
  functional in both ways, acyclic, isomorphic to $\mathbb{Z}$
- General techniques for this type of problems?
- Handling finiteness of relations

Summary

We have studied QA with FGTGD constraints and:

- Number restrictions:
  - OK on low-arity when restricting FGTGD shapes
  - OK when further imposing non-conflicting condition
- Transitivity:
  - OK when not used as guards
- Orders:
  - OK when not used as guards and covered

Ongoing work:

- Successor relations of linear orders:
  functional in both ways, acyclic, isomorphic to $\mathbb{Z}$
- General techniques for this type of problems?
- Handling finiteness of relations

Thanks for your attention!

References I

Baget, J., Mugnier, M., Rudolph, S., and Thomazo, M. (2011).
Walking the complexity lines for generalized guarded existential rules.
In *IJCAI*.

Calì, A., Gottlob, G., and Pieris, A. (2012).
Towards more expressive ontology languages: The query answering problem.
Artif. Intel., 193.

Calì, A., Lembo, D., and Rosati, R. (2003).
Query rewriting and answering under constraints in data integration systems.
In *IJCAI*.

References II

Gottlob, G., Pieris, A., and Tendera, L. (2013).
Querying the guarded fragment with transitivity.
In *ICALP*.

Gutiérrez-Basulto, V., Ibañez Garcia, Y., Kontchakov, R., and Kostylev, E. V. (2013).
Conjunctive queries with negation over DL-Lite: A closer look.
In *Web Reasoning and Rule Systems*.

Mitchell, J. C. (1983).
The implication problem for functional and inclusion dependencies.
Information and Control, 56(3).

References III

Pratt-Hartmann, I. (2009).
Data-complexity of the two-variable fragment with counting quantifiers.
Inf. Comput., 207(8).