Combining Existential Rules and Description Logics

Antoine Amarilli[1,2], Michael Benedikt[2]

[1]Télécom ParisTech, Paris, France

[2]University of Oxford, Oxford, United Kingdom

October 23, 2015

Open-world query answering (QA)

Open-world query answering:

- We are given:
  - Relational instance I (ground facts)
  - Logical constraints Σ
  - Boolean conjunctive query q

Open-world query answering (QA)

Open-world query answering:

- We are given:
  - Relational instance I (ground facts)
  - Logical constraints Σ
  - Boolean conjunctive query q
- We ask:
  - Consider all possible completions $J \supseteq I$
  - Restrict to those that satisfy the constraints Σ
  - $\rightarrow$ Is q certain among them?

Open-world query answering (QA)

Open-world query answering: – query entailment or containment

- We are given:
  - Relational instance I (ground facts) – A-Box
  - Logical constraints Σ – T-Box
  - Boolean conjunctive query q
- We ask:
  - Consider all possible completions $J \supseteq I$
  - Restrict to those that satisfy the constraints Σ
  - $\rightarrow$ Is q certain among them?

Decidable constraint languages for QA

Rich description logics (DLs)	**Frontier-guarded existential rules**

Decidable constraint languages for QA

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

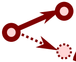
Decidable constraint languages for QA

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

Decidable constraint languages for QA

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

Decidable constraint languages for QA

Rich description logics (DLs)	**Frontier-guarded existential rules**
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.)	Poor (conjunction and implication)
Functionality asserts Funct(Mgr^{-}) 	n/a

Decidable constraint languages for QA

Rich description logics (DLs)	**Frontier-guarded existential rules**
$\mathrm{Emp} \sqsubseteq \mathrm{CEO} \sqcup (\exists \mathrm{Mgr}^{-}.\mathrm{Emp})$	$\forall pwv\ \mathrm{Acpt}(p, w, v) \rightarrow \exists f\ \mathrm{Trip}(p, f, v)$
Arity-two only	Arbitrary arity
Rich (disjunction, etc.)	Poor (conjunction and implication)
Functionality asserts $\mathrm{Funct}(\mathrm{Mgr}^{-})$	n/a

→ QA is decidable for either language

Our problem

Can we have the best of both worlds?

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

Our problem

Can we have the best of both worlds?

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

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

Our problem

Can we have the best of both worlds?

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

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

We show:

Our problem

Can we have the best of both worlds?

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

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

We show:

- QA is undecidable for rich DLs and frontier-guarded rules
- QA with rich DLs is decidable for some new rule classes
- Functional dependencies can be added under some conditions

Table of contents

Undecidability of frontier-guarded plus DLs

Theorem

QA is undecidable for rich DLs and frontier-guarded rules

Undecidability of frontier-guarded plus DLs

Theorem

QA is undecidable for rich DLs and frontier-guarded rules

Problem:

- DLs can express Funct ($\leftrightarrow$ functional dependencies, FDs)
- Frontier-guarded can express inclusion dependencies (IDs)
- 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 frontier-guarded rules

Problem:

- DLs can express Funct ($\leftrightarrow$ functional dependencies, FDs)
- Frontier-guarded can express inclusion dependencies (IDs)
- Implication of IDs and FDs is undecidable [Mitchell, 1983]
- Implication reduces to QA [Calì et al., 2003]

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

Undecidability of frontier-one plus DLs

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

Undecidability of frontier-one plus DLs

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

However:

Theorem

QA is undecidable for rich DLs and frontier-one rules

Undecidability of frontier-one plus DLs

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

However:

Theorem

QA is undecidable for rich DLs and frontier-one rules

Problem:

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

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

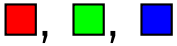
Undecidability of frontier-one plus DLs: proof

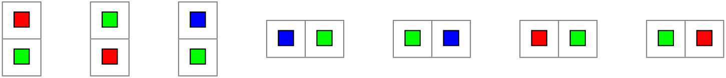
We reduce from tiling problems:

- finite set of colors: ■, ■, ■

Undecidability of frontier-one plus DLs: proof

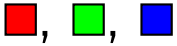
We reduce from tiling problems:

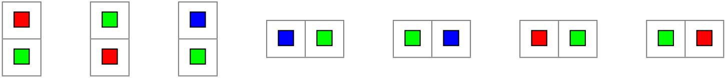
- finite set of colors: 
- horizontal and vertical allowed pairs:



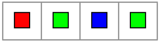
Undecidability of frontier-one plus DLs: proof

We reduce from tiling problems:

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



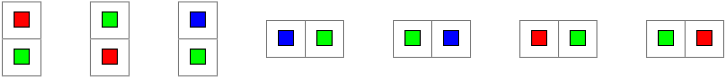
The tiling problem is:

- input: initial configuration: 

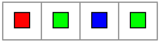
Undecidability of frontier-one plus DLs: proof

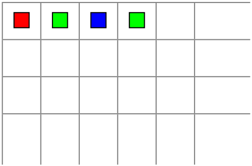
We reduce from tiling problems:

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



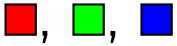
The tiling problem is:

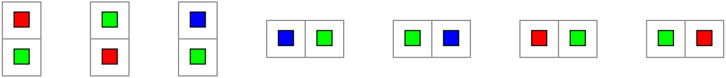
- input: initial configuration: 
- output: is there an infinite tiling?



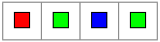
Undecidability of frontier-one plus DLs: proof

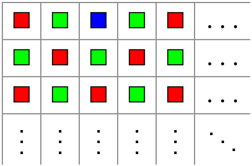
We reduce from tiling problems:

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



The tiling problem is:

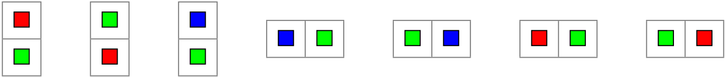
- input: initial configuration: 
- output: is there an infinite tiling?



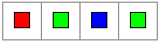
Undecidability of frontier-one plus DLs: proof

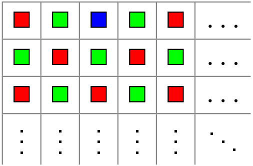
We reduce from tiling problems:

- 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 $C_{\square}$ 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 $C_{\square}$ for each color

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

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

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

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

- DL constraints for the pairs, e.g., $C_{\color{red}\blacksquare} \sqcap \exists R.C_{\color{red}\blacksquare} \sqsubseteq \bot$
- Disjunction to color tiles: $T \sqsubseteq C_{\color{red}\blacksquare} \sqcup C_{\color{lime}\blacksquare} \sqcup C_{\color{blue}\blacksquare}$

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

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

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

- DL constraints for the pairs, e.g., $C_{\color{red}\blacksquare} \sqcap \exists R.C_{\color{red}\blacksquare} \sqsubseteq \bot$
- Disjunction to color tiles: $T \sqsubseteq C_{\color{red}\blacksquare} \sqcup C_{\color{green}\blacksquare} \sqcup C_{\color{blue}\blacksquare}$

- Frontier-one rule: $\forall x \; T(x) \Rightarrow \exists yzw$

$$\begin{array}{ccc} T(x) & \xrightarrow{R} & T(y) \\ \downarrow D & & \downarrow D \\ T(z) & \xrightarrow{R} & T(w) \end{array}$$

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

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

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

- DL constraints for the pairs, e.g., $C_{\color{red}\blacksquare} \sqcap \exists R.C_{\color{red}\blacksquare} \sqsubseteq \bot$
- Disjunction to color tiles: $T \sqsubseteq C_{\color{red}\blacksquare} \sqcup C_{\color{green}\blacksquare} \sqcup C_{\color{blue}\blacksquare}$

- Frontier-one rule: $\forall x \; T(x) \Rightarrow \exists yzw$

$$\begin{array}{ccc} T(x) & \xrightarrow{R} & T(y) \\ \downarrow D & & \downarrow D \\ T(z) & \xrightarrow{R} & T(w) \end{array}$$

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

Table of contents

Decidability of non-looping frontier-one and DLs

Idea: prohibit cycles in existential rules:

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

Decidability of non-looping frontier-one and DLs

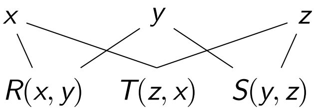
Idea: prohibit cycles in existential rules:

- $R(x, y)\ S(y, z)\ T(z, x)$ 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



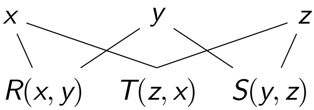
Idea: prohibit cycles in existential rules:

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

Formally:

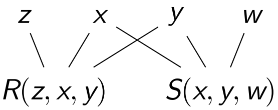
- Berge cycle: cycle in the atom–variable incidence graph

Decidability of non-looping frontier-one and DLs



Idea: prohibit cycles in existential rules:

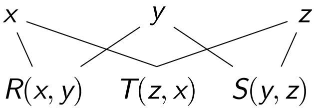
- $R(x, y)\ S(y, z)\ T(z, x)$ is a cycle
- $R(z, x, y)\ S(x, y, w)$ is also a cycle



Formally:

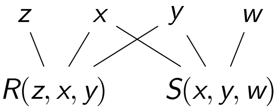
- Berge cycle: cycle in the atom–variable incidence graph

Decidability of non-looping frontier-one and DLs



Idea: prohibit cycles in existential rules:

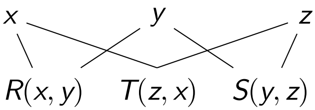
- $R(x, y)\ S(y, z)\ T(z, x)$ is a cycle
- $R(z, x, y)\ S(x, y, w)$ is also a cycle



Formally:

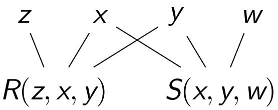
- Berge cycle: cycle in the atom–variable incidence graph
- Non-looping atoms: no Berge cycle except, e.g., $R(x, y)\ S(x, y)$
- Non-looping frontier-one: non-looping body and head

Decidability of non-looping frontier-one and DLs



Idea: prohibit cycles in existential rules:

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



Formally:

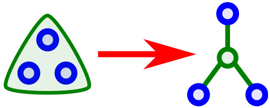
- Berge cycle: cycle in the atom–variable incidence graph
- Non-looping atoms: no Berge cycle except, e.g., $R(x, y)\ S(x, y)$
- Non-looping frontier-one: non-looping body and head

Theorem

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

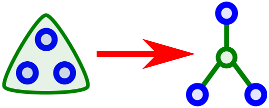
Decidability of non-looping frontier-one and DLs (proof)

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



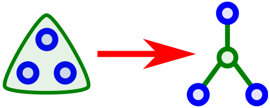
Decidability of non-looping frontier-one and DLs (proof)

- 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)

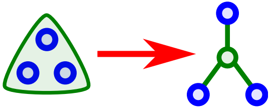
- Shred $R(a, b, c)$ to $R_1(f, a), R_2(f, b), R_3(f, c)$
- Axiomatize the R_i, e.g., $\forall f \, \exists^{=1} x \; R_1(f, x)$



→ QA for the shredded instance, rules, query, and axioms is equivalent to QA for the original instance, rules, 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., $\forall f\, \exists^{=1} x\, R_1(f, x)$

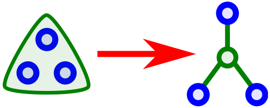


→ QA for the shredded instance, rules, query, and axioms is equivalent to QA for the original instance, rules, query

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

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., $\forall f\, \exists^{=1} x\; R_1(f, x)$

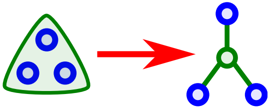


→ QA for the shredded instance, rules, query, and axioms is equivalent to QA for the original instance, rules, query

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

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., $\forall f\, \exists^{=1} x\; R_1(f, x)$

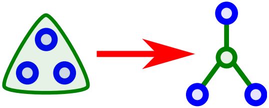


→ QA for the shredded instance, rules, query, and axioms is equivalent to QA for the original instance, rules, query

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

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., $\forall f\, \exists^{=1} x\ R_1(f, x)$

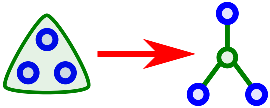


$\rightarrow$ QA for the shredded instance, rules, query, and axioms is equivalent to QA for the original instance, rules, query

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

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., $\forall f\, \exists^{=1} x\ R_1(f, x)$

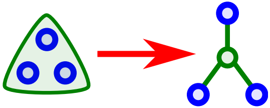


→ QA for the shredded instance, rules, query, and axioms is equivalent to QA for the original instance, rules, query

- Rewrite shredded non-looping frontier-one rules to GC^2:
  - Rewrite $\forall x\mathbf{y}\ \phi(x, \mathbf{y}) \Rightarrow \exists \mathbf{z}\ \psi(x, \mathbf{z})$ to $\forall x\ \phi'(x) \Rightarrow \psi'(x)$, with $\phi'(x)$ and $\psi'(x)$ the shredding of $\forall \mathbf{y}\, \phi(x, \mathbf{y})$ and $\exists \mathbf{z}\, \psi(x, \mathbf{y})$
  - Exemple: $\phi(x) = \exists yz\ T(x, y) \land R(x, x, z) \land A(z)$
    → $\exists yzf\ T(x, y) \land R_1(f, x) \land R_2(f, x) \land R_3(f, z) \land A(z)$
    → $\left(\exists y\ T(x, y)\right)$

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., $\forall f\, \exists^{=1} x\; R_1(f,x)$

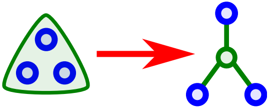


→ QA for the shredded instance, rules, query, and axioms is equivalent to QA for the original instance, rules, query

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

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., $\forall f\, \exists^{=1} x\; R_1(f,x)$

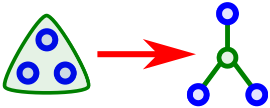


→ QA for the shredded instance, rules, query, and axioms is equivalent to QA for the original instance, rules, query

- Rewrite shredded non-looping frontier-one rules to GC^2:
  - Rewrite $\forall x\mathbf{y}\; \phi(x,\mathbf{y}) \Rightarrow \exists\mathbf{z}\; \psi(x,\mathbf{z})$ to $\forall x\; \phi'(x) \Rightarrow \psi'(x)$, with $\phi'(x)$ and $\psi'(x)$ the shredding of $\forall\mathbf{y}\,\phi(x,\mathbf{y})$ and $\exists\mathbf{z}\,\psi(x,\mathbf{y})$
  - Exemple: $\phi(x) = \exists yz\;\; T(x,y) \wedge R(x,x,z) \wedge A(z)$
    - → $\exists yzf\; T(x,y) \wedge R_1(f,x) \wedge R_2(f,x) \wedge R_3(f,z) \wedge A(z)$
    - → $\Big(\exists y\;\, T(x,y)\Big) \wedge \Big(\exists f\; R_1(f,x) \wedge R_2(f,x) \wedge \big(\exists z\; R_3(f,z) \wedge A(z)\big)\Big)$

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., $\forall f\, \exists^{=1} x\ R_1(f, x)$



→ QA for the shredded instance, rules, query, and axioms is equivalent to QA for the original instance, rules, query

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

→ Reduces to QA for GC^2: decidable [Pratt-Hartmann, 2009]

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

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

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

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

Theorem

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

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

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

Theorem

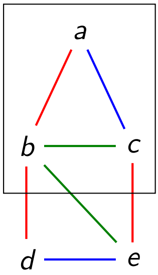
QA is decidable for head-non-looping frontier-one rules + rich DLs

Basic idea:

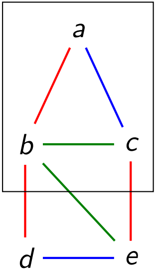
- If there is a counterexample model to QA, we can unravel it
  - → It is still a counterexample
  - → It has no cycles (except in the instance part)

→ Looping rule bodies can only match on the instance part

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

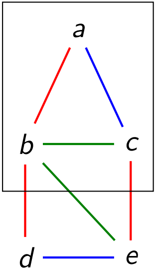


Head-non-looping frontier-one and DLs: unraveling

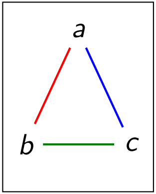


$\Rightarrow$

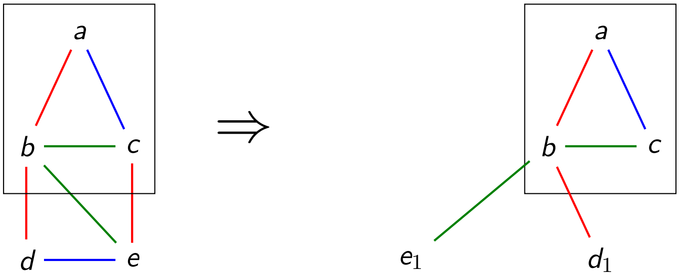
Head-non-looping frontier-one and DLs: unraveling



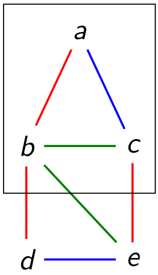
$\Rightarrow$



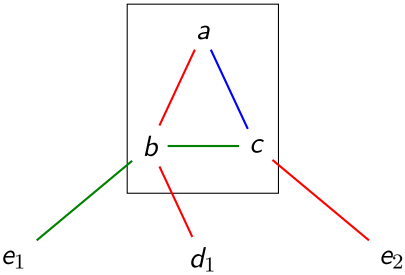
Head-non-looping frontier-one and DLs: unraveling



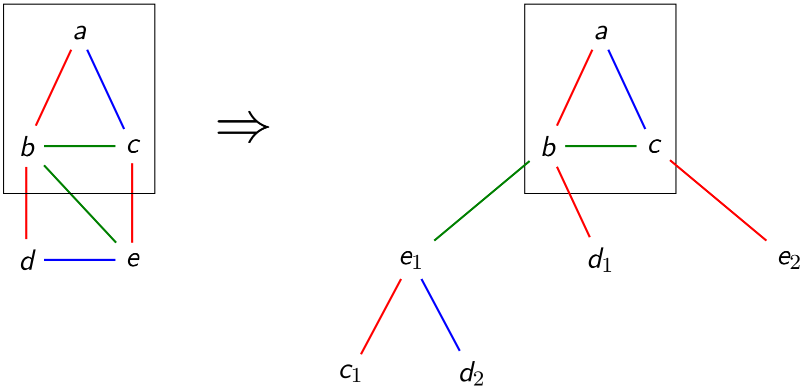
Head-non-looping frontier-one and DLs: unraveling



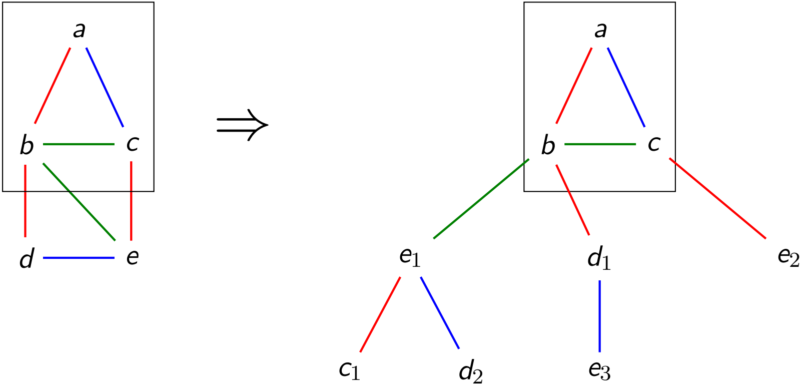
$\Rightarrow$



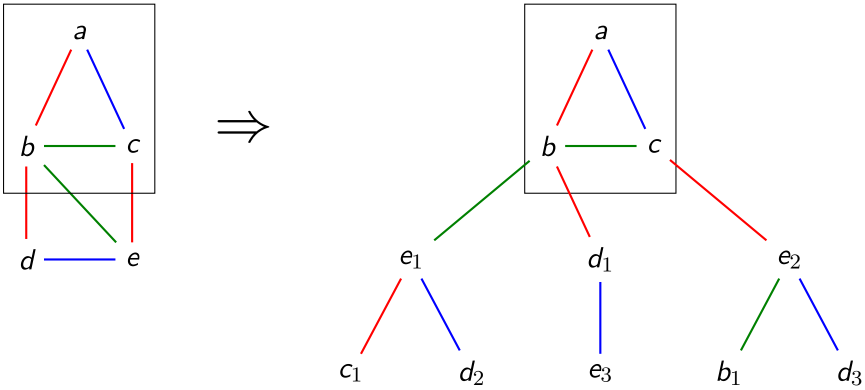
Head-non-looping frontier-one and DLs: unraveling



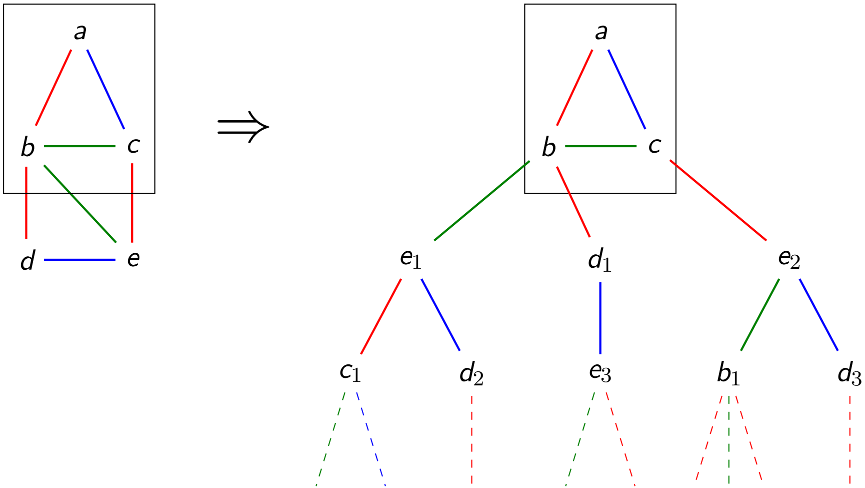
Head-non-looping frontier-one and DLs: unraveling



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



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



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

For every frontier-one rule with a looping body:

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

For every frontier-one rule with a looping body:

- Consider all possible self-homomorphisms of the body
  - → 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 rule with a looping body:

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

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

For every frontier-one rule with a looping body:

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

→ Keep the resulting fully non-looping rules

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

For every frontier-one rule with a looping body:

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

→ Keep the resulting fully non-looping rules

→ QA for the shredded instance, treefied rules, query, and axioms is equivalent to QA for the original instance, rules, query

Table of contents

Adding functional dependencies

We have shown:

Theorem

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

Adding functional dependencies

We have shown:

Theorem

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

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

Undecidability of linear frontier-one and FDs

Linear: single-atom head and body: implies non-looping.

Undecidability of linear frontier-one and FDs

Linear: single-atom head and body: implies non-looping.

Theorem

QA for FDs and linear frontier-one rules is undecidable.

Undecidability of linear frontier-one and FDs

Linear: single-atom head and body: implies non-looping.

Theorem

QA for FDs and linear frontier-one rules is undecidable.

Proof ideas:

- Reduce from implication of unary FDs and frontier-2 IDs
- Leverage variable reuse and FDs to export two variables: to encode the ID $R[1,2] \subseteq R[3,4]$ with the FD $R[1] \rightarrow R[2]$, write $R(x, y, z, w) \Rightarrow R(x, y', x, y')$: we must have $y = y'$

$\rightarrow$ We need an additional restriction for decidability

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

Consider QA under single-head rules Σ and FDs Φ

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

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

Consider QA under single-head rules Σ and FDs Φ

- Σ and Φ are separable if $\mathsf{QA}(\Sigma, \Phi) \Leftrightarrow \mathsf{QA}(\Sigma)$ when $I \models \Phi$
- Separability guaranteed under the non-conflicting condition:

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

Consider QA under single-head rules Σ and FDs Φ

- Σ and Φ are separable if $\mathrm{QA}(\Sigma, \Phi) \Leftrightarrow \mathrm{QA}(\Sigma)$ when $I \models \Phi$
- Separability guaranteed under the non-conflicting condition:
  - For every rule 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 rules and FDs [Calì et al., 2012]

Consider QA under single-head rules Σ and FDs Φ

- Σ and Φ are separable if $\mathrm{QA}(\Sigma, \Phi) \Leftrightarrow \mathrm{QA}(\Sigma)$ when $I \models \Phi$
- Separability guaranteed under the non-conflicting condition:
  - For every rule 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:

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

Consider QA under single-head rules Σ and FDs Φ

- Σ and Φ are separable if $\mathrm{QA}(\Sigma, \Phi) \Leftrightarrow \mathrm{QA}(\Sigma)$ when $I \models \Phi$
- Separability guaranteed under the non-conflicting condition:
  - For every rule 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

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

Consider QA under single-head rules Σ and FDs Φ

- Σ and Φ are separable if $\mathrm{QA}(\Sigma, \Phi) \Leftrightarrow \mathrm{QA}(\Sigma)$ when $I \models \Phi$
- Separability guaranteed under the non-conflicting condition:
  - For every rule 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

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

Consider QA under single-head rules Σ and FDs Φ

- Σ and Φ are separable if $QA(\Sigma, \Phi) \Leftrightarrow QA(\Sigma)$ when $I \models \Phi$
- Separability guaranteed under the non-conflicting condition:
  - For every rule 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 rules and FDs [Calì et al., 2012]

Consider QA under single-head rules Σ and FDs Φ

- Σ and Φ are separable if $\mathrm{QA}(\Sigma, \Phi) \Leftrightarrow \mathrm{QA}(\Sigma)$ when $I \models \Phi$
- Separability guaranteed under the non-conflicting condition:
  - For every rule 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 rules and FDs [Calì et al., 2012]

Consider QA under single-head rules Σ and FDs Φ

- Σ and Φ are separable if $\mathrm{QA}(\Sigma, \Phi) \Leftrightarrow \mathrm{QA}(\Sigma)$ when $I \models \Phi$
- Separability guaranteed under the non-conflicting condition:
  - For every rule 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

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

Consider QA under single-head rules Σ and FDs Φ

- Σ and Φ are separable if $\mathrm{QA}(\Sigma, \Phi) \Leftrightarrow \mathrm{QA}(\Sigma)$ when $I \models \Phi$
- Separability guaranteed under the non-conflicting condition:
  - For every rule 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 rules and FDs [Calì et al., 2012]

Consider QA under single-head rules Σ and FDs Φ

- Σ and Φ are separable if $\mathrm{QA}(\Sigma, \Phi) \Leftrightarrow \mathrm{QA}(\Sigma)$ when $I \models \Phi$
- Separability guaranteed under the non-conflicting condition:
  - For every rule 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 rules and FDs [Calì et al., 2012]

Consider QA under single-head rules Σ and FDs Φ

- Σ and Φ are separable if $\mathrm{QA}(\Sigma, \Phi) \Leftrightarrow \mathrm{QA}(\Sigma)$ when $I \models \Phi$
- Separability guaranteed under the non-conflicting condition:
  - For every rule 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 rules and FDs [Calì et al., 2012]

Consider QA under single-head rules Σ and FDs Φ

- Σ and Φ are separable if $\mathrm{QA}(\Sigma, \Phi) \Leftrightarrow \mathrm{QA}(\Sigma)$ when $I \models \Phi$
- Separability guaranteed under the non-conflicting condition:
  - For every rule 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

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

Consider QA under single-head rules Σ and FDs Φ

- Σ and Φ are separable if $\mathrm{QA}(\Sigma, \Phi) \Leftrightarrow \mathrm{QA}(\Sigma)$ when $I \models \Phi$
- Separability guaranteed under the non-conflicting condition:
  - For every rule 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...

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

Consider QA under single-head rules Σ and FDs Φ

- Σ and Φ are separable if $\mathrm{QA}(\Sigma, \Phi) \Leftrightarrow \mathrm{QA}(\Sigma)$ when $I \models \Phi$
- Separability guaranteed under the non-conflicting condition:
  - For every rule 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)

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

Consider QA under single-head rules Σ and FDs Φ

- Σ and Φ are separable if $\mathrm{QA}(\Sigma, \Phi) \Leftrightarrow \mathrm{QA}(\Sigma)$ when $I \models \Phi$
- Separability guaranteed under the non-conflicting condition:
  - For every rule 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 rules and FDs [Calì et al., 2012]

Consider QA under single-head rules Σ and FDs Φ

- Σ and Φ are separable if $\mathrm{QA}(\Sigma, \Phi) \Leftrightarrow \mathrm{QA}(\Sigma)$ when $I \models \Phi$
- Separability guaranteed under the non-conflicting condition:
  - For every rule 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... 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:

Theorem

QA is decidable for head-non-looping frontier-one rules + 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:

Theorem

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

We show:

Theorem

QA is decidable for:

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

Decidability for non-conflicting FDs: proof ideas

- Non-conflicting: the FDs are not violated in the chase
- Unraveling is a bit like chasing

Decidability for non-conflicting FDs: proof ideas

- Non-conflicting: the FDs are not violated in the chase
- Unraveling is a bit like chasing

→ Tweak the unraveling to also respect FDs

Decidability for non-conflicting FDs: proof ideas

- Non-conflicting: the FDs are not violated in the chase
- Unraveling is a bit like chasing

→ Tweak the unraveling to also respect FDs

Intuition: When unraveling (the shredding of) a higher-arity fact, consider the positions S where the previous element occurs:

Decidability for non-conflicting FDs: proof ideas

- Non-conflicting: the FDs are not violated in the chase
- Unraveling is a bit like chasing

→ Tweak the unraveling to also respect FDs

Intuition: When unraveling (the shredding of) a higher-arity fact, consider the positions S where the previous element occurs:

- if $S' \subsetneq S$, for S' an FD determiner
  - → ignore this fact (it's not required by the constraints)

Decidability for non-conflicting FDs: proof ideas

- Non-conflicting: the FDs are not violated in the chase
- Unraveling is a bit like chasing

→ Tweak the unraveling to also respect FDs

Intuition: When unraveling (the shredding of) a higher-arity fact, consider the positions S where the previous element occurs:

- if $S' \subsetneq S$, for S' an FD determiner
  → ignore this fact (it's not required by the constraints)
- if $S' = S$ for S' an FD determiner
  → copy only one such fact, distinguish its other elements (no equality between them is required by the constraints)

Table of contents

Summary of results

Combining Existential Rules and Description Logics

Summary of results

Combining Existential Rules and Description Logics

- Open-world query answering (QA) under:
  - Rich DL constraints
  - Existential rules
- For which rule classes is QA decidable with rich DLs?

Summary of results

Combining Existential Rules and Description Logics

- Open-world query answering (QA) under:
  - Rich DL constraints
  - Existential rules
- For which rule classes is QA decidable with rich DLs?

→ Must restrict to frontier-one rules

→ Must prohibit cycles in rule heads

Summary of results

Combining Existential Rules and Description Logics

- Open-world query answering (QA) under:
  - Rich DL constraints
  - Existential rules
- For which rule classes is QA decidable with rich DLs?

→ Must restrict to frontier-one rules

→ Must prohibit cycles in rule heads

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

→ Can add non-conflicting FDs

Summary of results

Combining Existential Rules and Description Logics

- Open-world query answering (QA) under:
  - Rich DL constraints
  - Existential rules
- For which rule classes is QA decidable with rich DLs?

→ Must restrict to frontier-one rules

→ Must prohibit cycles in rule heads

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

→ Can add non-conflicting FDs

- What about QA on finite models?
- Could we have an expressive frontier-one language?
  (FDs, disjunctions... like DLs but higher-arity)

Summary of the other things I do

- Adding transitive and order relations to existential rules[1]
  - → QA for frontier-guarded is decidable with transitive relations
  - → Also for order relations (with atom-covered requirement)

[1] With Michael Benedikt, ongoing work

[2] With Michael Benedikt, [Amarilli and Benedikt, 2015], LICS'15

[3] With Pierre Bourhis and Pierre Senellart, [Amarilli et al., 2015], ICALP'15

Summary of the other things I do

- Adding transitive and order relations to existential rules[1]
  - $\rightarrow$ QA for frontier-guarded is decidable with transitive relations
  - $\rightarrow$ Also for order relations (with atom-covered requirement)
- QA on finite models[2]
  - $\rightarrow$ Frontier-one IDs and FDs are finitely controllable up to closure

---

[1] With Michael Benedikt, ongoing work

[2] With Michael Benedikt, [Amarilli and Benedikt, 2015], LICS'15

[3] With Pierre Bourhis and Pierre Senellart, [Amarilli et al., 2015], ICALP'15

Summary of the other things I do

- Adding transitive and order relations to existential rules[1]
  → QA for frontier-guarded is decidable with transitive relations
  → Also for order relations (with atom-covered requirement)
- QA on finite models[2]
  → Frontier-one IDs and FDs are finitely controllable up to closure
- Database tasks on treelike instances[3]
  - Tractable probability evaluation, semiring provenance circuits
  → Necessity of bounded treewidth, via grid minor results
  → Query-specific tree decompositions for richer safe classes?

---

[1]With Michael Benedikt, ongoing work

[2]With Michael Benedikt, [Amarilli and Benedikt, 2015], LICS'15

[3]With Pierre Bourhis and Pierre Senellart, [Amarilli et al., 2015], ICALP'15

Summary of the other things I do

- Adding transitive and order relations to existential rules[1]
  → QA for frontier-guarded is decidable with transitive relations
  → Also for order relations (with atom-covered requirement)
- QA on finite models[2]
  → Frontier-one IDs and FDs are finitely controllable up to closure
- Database tasks on treelike instances[3]
  - Tractable probability evaluation, semiring provenance circuits

  → Necessity of bounded treewidth, via grid minor results
  → Query-specific tree decompositions for richer safe classes?
- Also: partially ordered databases, crowdsourcing, and more!

[1]With Michael Benedikt, ongoing work
[2]With Michael Benedikt, [Amarilli and Benedikt, 2015], LICS'15
[3]With Pierre Bourhis and Pierre Senellart, [Amarilli et al., 2015], ICALP'15

Summary of the other things I do

- Adding transitive and order relations to existential rules[1]
  - → QA for frontier-guarded is decidable with transitive relations
  - → Also for order relations (with atom-covered requirement)
- QA on finite models[2]
  - → Frontier-one IDs and FDs are finitely controllable up to closure
- Database tasks on treelike instances[3]
  - Tractable probability evaluation, semiring provenance circuits
  - → Necessity of bounded treewidth, via grid minor results
  - → Query-specific tree decompositions for richer safe classes?
- Also: partially ordered databases, crowdsourcing, and more!

→ If you're interested by one of these, come chat with me today!

[1] With Michael Benedikt, ongoing work
[2] With Michael Benedikt, [Amarilli and Benedikt, 2015], LICS'15
[3] With Pierre Bourhis and Pierre Senellart, [Amarilli et al., 2015], ICALP'15

Summary of the other things I do

- Adding transitive and order relations to existential rules[1]
  → QA for frontier-guarded is decidable with transitive relations
  → Also for order relations (with atom-covered requirement)
- QA on finite models[2]
  → Frontier-one IDs and FDs are finitely controllable up to closure
- Database tasks on treelike instances[3]
  - Tractable probability evaluation, semiring provenance circuits
  → Necessity of bounded treewidth, via grid minor results
  → Query-specific tree decompositions for richer safe classes?
- Also: partially ordered databases, crowdsourcing, and more!

→ If you're interested by one of these, come chat with me today!

Thanks for your attention!

[1] With Michael Benedikt, ongoing work
[2] With Michael Benedikt, [Amarilli and Benedikt, 2015], LICS'15
[3] With Pierre Bourhis and Pierre Senellart, [Amarilli et al., 2015], ICALP'15

References I

Amarilli, A. and Benedikt, M. (2015).
Finite open-world query answering with number restrictions.
In *Proc. LICS*.

Amarilli, A., Bourhis, P., and Senellart, P. (2015).
Provenance circuits for trees and treelike instances.
In *Proc. ICALP*.

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

References II

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

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

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