Combining Existential Rules and Description Logics

Antoine Amarilli\textsuperscript{1,2}, Michael Benedikt\textsuperscript{2}

\textsuperscript{1} Télécom ParisTech, Paris, France
\textsuperscript{2} University of Oxford, Oxford, United Kingdom

October 29, 2015
Open-world query answering (QA)

Open-world query answering:
- We are given:
  - Relational instance $I$ (ground facts)
  - Logical constraints $\Sigma$
  - Boolean conjunctive query $q$
Open-world query answering (QA)

Open-world query answering:

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

- We ask:
  - Consider all possible completions $J \supseteq I$
  - Restrict to those that satisfy the constraints $\Sigma$
  - 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 $\Sigma$ – T-Box
  - Boolean conjunctive query $q$

- We ask:
  - Consider all possible completions $J \supseteq I$
  - Restrict to those that satisfy the constraints $\Sigma$
  - 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

\[ \text{Emp} \sqsubseteq \text{CEO} \sqcup (\exists \text{Mgr}^{-}.\text{Emp}) \]

\[ \forall pwv \ \text{Acpt}(p, w, v) \rightarrow \exists f \ \text{Trip}(p, f, v) \]
Decidable constraint languages for QA

<table>
<thead>
<tr>
<th>Rich description logics (DLs)</th>
<th>Frontier-guarded existential rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Emp ⊑ CEO ⊓ (∃Mgr⁻ .Emp)</td>
<td>∀ p,w,v Acpt(p, w, v) → ∃f Trip(p, f, v)</td>
</tr>
<tr>
<td>Arity-two only</td>
<td>Arbitrary arity</td>
</tr>
</tbody>
</table>

- Decidable constraint languages for QA
- Rich description logics (DLs)
  - Emp ⊑ CEO ⊓ (∃Mgr⁻ .Emp)
  - Arity-two only
- Frontier-guarded existential rules
  - ∀ p,w,v Acpt(p, w, v) → ∃f Trip(p, f, v)
  - Arbitrary arity
Decidable constraint languages for QA

<table>
<thead>
<tr>
<th>Rich description logics (DLs)</th>
<th>Frontier-guarded existential rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Emp $\subseteq$ CEO $\sqcup$ (∃Mgr$^-$.Emp)</td>
<td>$\forall p w v \text{ Acpt}(p, w, v) \rightarrow \exists f \text{ Trip}(p, f, v)$</td>
</tr>
<tr>
<td>Arity-two only $\triangleright$</td>
<td>Arbitrary arity $\blacktriangle$</td>
</tr>
<tr>
<td>Rich (disjunction, etc.)</td>
<td>Poor (conjunction and implication)</td>
</tr>
</tbody>
</table>
Decidable constraint languages for QA

<table>
<thead>
<tr>
<th>Rich description logics (DLs)</th>
<th>Frontier-guarded existential rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Emp ⊑ CEO ⊓ (∃Mgr⁻.Emp)</td>
<td>∀p,w,v Acpt(p, w, v) → ∃f Trip(p, f, v)</td>
</tr>
<tr>
<td>Arity-two only</td>
<td>Arbitrary arity</td>
</tr>
<tr>
<td>Rich (disjunction, etc.)</td>
<td>Poor (conjunction and implication)</td>
</tr>
<tr>
<td>Functionality asserts</td>
<td>n/a</td>
</tr>
<tr>
<td>Funct(Mgr⁻)</td>
<td></td>
</tr>
</tbody>
</table>
## Decidable constraint languages for QA

### Rich description logics (DLs)

- **Emp** ⊆ **CEO** ⊢ (∃ **Mgr**.Emp)
- **Arity-two only**
- **Rich** (disjunction, etc.)
- **Functionality asserts**
  - **Funct**(Mgr)

### Frontier-guarded existential rules

- ∀p,w,v Acpt(p, w, v) → ∃ f Trip(p, f, v)
- **Arbitrary arity**
- **Poor** (conjunction and implication)
- **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

→ 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 GC², 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:
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>
<thead>
<tr>
<th>1</th>
<th>Problem statement</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Undecidability</td>
</tr>
<tr>
<td>3</td>
<td>Decidability</td>
</tr>
<tr>
<td>4</td>
<td>Adding FDs</td>
</tr>
<tr>
<td>5</td>
<td>Conclusion</td>
</tr>
</tbody>
</table>
Undecidability of frontier-guarded plus DLs

Theorem

QA is undecidable for rich DLs and frontier-guarded rules
Theorem

QA is undecidable for rich DLs and frontier-guarded rules

Problem:

- DLs can express **Funct** (↔ 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* (↔ 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]

→ Restrict to frontier-one rules: \( \forall x y \phi(x, y) \rightarrow \exists z \psi(x, z) \)
Restrict to frontier-one rules: \( \forall x y \phi(x, y) \rightarrow \exists z \psi(x, z) \)

QA for frontier-one IDs plus FDs is decidable (separability)
Undecidability of frontier-one plus DLs

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

![Allowed pairs](image)

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:

*Diagram showing allowed pairs with colors represented by shapes.*

The tiling problem is:

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

*Diagram showing an example of an infinite tiling with shapes representing the colors.*
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 $C_i$ for each color

**Initial instance:**

- $C_\square \xrightarrow{R} C_\square \xrightarrow{R} C_\square \xrightarrow{R} C_\square$
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:**

- **DL constraints** for the pairs, e.g., $C \cap \exists R. C \subseteq \bot$
- **Disjunction** to color tiles: $T \subseteq C \cup C \cup C$
Undecidability of frontier-one plus DLs: proof, cont’d

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

Initial instance: $C \square \xrightarrow{R} C \square \xrightarrow{R} C \square \xrightarrow{R} C \square$

- DL constraints for the pairs, e.g., $C \square \sqcap \exists R. C \square \sqsubseteq \bot$
- Disjunction to color tiles: $T \sqsubseteq C \square \sqcup C \square \sqcup C \square$

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

$T(x) \xrightarrow{R} T(y)$
$\downarrow D$
$T(z) \xrightarrow{R} T(w)$

$\downarrow D$
Undecidability of frontier-one plus DLs: proof, cont’d

- **Functional relations** $D$ for down and $R$ for right
- **Unary predicate** $T$ for tiles and $C_c$ for each color

- **Initial instance:**
  
  - $C_\blacksquare \xrightarrow{R} C_\blacksquare \xrightarrow{R} C_\blacktriangle \xrightarrow{R} C_\blackdiamond

- **DL constraints** for the pairs, e.g., $C_\blacksquare \cap \exists R. C_\blacksquare \subseteq \bot$
- **Disjunction** to color tiles: $T \sqsubseteq C_\blacksquare \cup C_\blacksquare \cup C_\blacktriangle

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

  - $T(x) \xrightarrow{R} T(y)$
  - $T(z) \xrightarrow{R} T(w)$

→ There is an extension of the instance iff there is a tiling
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:

- \( R(x, y) \quad S(y, z) \quad T(z, x) \) is a cycle
- \( R(z, x, y) \quad S(x, y, w) \) 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., $\forall f \exists^1 x \ R_1(f, x)$

$\implies$ 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 $\text{GC}^2$:
  - Rewrite $\forall xy \phi(x, y) \Rightarrow \exists z \psi(x, z)$ 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., $\forall f \exists^1 x R_1(f, x)$

$\rightarrow$ 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 $\forall xy \phi(x, y) \Rightarrow \exists z \psi(x, z)$ to $\forall x \phi'(x) \Rightarrow \psi'(x)$, with $\phi'(x)$ and $\psi'(x)$ the shredding of $\forall y \phi(x, y)$ and $\exists z \psi(x, y)$
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 $\forall xy \ \phi(x, y) \Rightarrow \exists z \ \psi(x, z)$ to $\forall x \ \phi'(x) \Rightarrow \psi'(x)$, with $\phi'(x)$ and $\psi'(x)$ the shredding of $\forall y \ \phi(x, y)$ and $\exists z \ \psi(x, y)$
  - Exemple: $\phi(x) = \exists yz \ T(x, y) \land R(x, x, z) \land A(z)$
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 $\forall xy \ \phi(x, y) \Rightarrow \exists z \ \psi(x, z)$ to $\forall x \ \phi'(x) \Rightarrow \psi'(x)$, with $\phi'(x)$ and $\psi'(x)$ the shredding of $\forall y \ \phi(x, y)$ and $\exists z \ \psi(x, y)$
  - Exemple: $\phi(x) = \exists yz \ T(x, y) \land R(x, x, z) \land A(z)$
    $\rightarrow \ \exists yzf \ T(x, y) \land R_1(f, x) \land R_2(f, x) \land R_3(f, z) \land A(z)$
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 $\forall xy \phi(x, y) \Rightarrow \exists z \psi(x, z)$ to $\forall x \phi'(x) \Rightarrow \psi'(x)$, with $\phi'(x)$ and $\psi'(x)$ the shredding of $\forall y \phi(x, y)$ and $\exists z \psi(x, y)$
  - Exemple: $\phi(x) = \exists yz T(x, y) \land R(x, x, z) \land A(z)$
    - $\Rightarrow \exists yzf T(x, y) \land R_1(f, x) \land R_2(f, x) \land R_3(f, z) \land A(z)$
    - $\Rightarrow \left( \exists y T(x, y) \right)$
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., $\forall f \exists^1 x R_1(f, x)$
  \[ \rightarrow \text{ QA for the shredded instance, rules, query, and axioms is equivalent to QA for the original instance, rules, query} \]

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

$\Rightarrow$ 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 $\forall xy \ \phi(x, y) \Rightarrow \exists z \ \psi(x, z)$ to $\forall x \ \phi'(x) \Rightarrow \psi'(x)$, with $\phi'(x)$ and $\psi'(x)$ the shredding of $\forall y \ \phi(x, y)$ and $\exists z \ \psi(x, y)$
  - Exemple: $\phi(x) = \exists y z \ T(x, y) \land R(x, x, z) \land A(z)$
    $\Rightarrow \exists y z f \ T(x, y) \land R_1(f, x) \land R_2(f, x) \land R_3(f, z) \land A(z)$
    $\Rightarrow \left( \exists y \ T(x, y) \right) \land \left( \exists f \ R_1(f, x) \land R_2(f, x) \land (\exists z \ R_3(f, z) \land A(z)) \right)$
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 $\forall xy \ \phi(x, y) \Rightarrow \exists z \ \psi(x, z)$ to $\forall x \ \phi'(x) \Rightarrow \psi'(x)$, with $\phi'(x)$ and $\psi'(x)$ the shredding of $\forall y \ \phi(x, y)$ and $\exists z \ \psi(x, y)$
  - Exemple: $\phi(x) = \exists yz \ T(x, y) \land R(x, x, z) \land A(z)$
    → $\exists yzf \ T(x, y) \land R_1(f, x) \land R_2(f, x) \land R_3(f, z) \land A(z)$
    → $\left( \exists y \ T(x, y) \right) \land \left( \exists f \ R_1(f, x) \land R_2(f, x) \land (\exists z \ R_3(f, z) \land A(z)) \right)$

→ 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

\[ \begin{array}{c}
  a \\
  \downarrow \\
  b \quad c \\
  \downarrow \quad \downarrow \\
  d \quad e
\end{array} \quad \Rightarrow \]
Head-non-looping frontier-one and DLs: unraveling

\begin{align*}
\begin{array}{ccc}
\text{Problem statement} & \text{Undecidability} & \text{Decidability} & \text{Adding FDs} & \text{Conclusion} \\
\end{array}
\end{align*}

\begin{align*}
\text{Head-non-looping frontier-one and DLs: unraveling}
\end{align*}

\begin{align*}
\begin{array}{c}
\begin{tikzpicture}
\node (a) at (0,0) {$a$};
\node (b) at (-1,-1) {$b$};
\node (c) at (0,-1) {$c$};
\node (d) at (-1,-2) {$d$};
\node (e) at (0,-2) {$e$};
\draw[red] (a) -- (b);
\draw[blue] (a) -- (c);
\draw[green] (b) -- (d);
\draw[green] (c) -- (e);
\end{tikzpicture}
\end{array}
\Rightarrow
\begin{array}{c}
\begin{tikzpicture}
\node (a) at (0,0) {$a$};
\node (b) at (-1,-1) {$b$};
\node (c) at (0,-1) {$c$};
\draw[red] (a) -- (b);
\draw[blue] (a) -- (c);
\end{tikzpicture}
\end{array}
\end{align*}
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
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**:
For every frontier-one rule with a looping body:

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

- Consider all possible mappings to the instance
  
  → 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 rule with a looping body:

- Consider all possible self-homomorphisms of the body
  \[ R(x, y) \land S(y, z) \land T(z, x) \] gives \[ R(x, y) \land S(y, x) \land T(x, x) \]

- Consider all possible mappings to the instance
  \[ R(x, y) \land S(y, z) \land T(z, x) \] gives \[ R(x, y) \land S(y', z) \land T(z', x') \]
  \[ x = a \land x' = a \land y = b \land y' = b \land z = c \land z' = c \]

- 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
  \[ R(x, y) \land S(y, z) \land T(z, x) \text{ gives } R(x, y) \land S(y, x) \land T(x, x) \]

- Consider all possible mappings to the instance
  \[ R(x, y) \land S(y, z) \land T(z, x) \text{ gives } R(x, y) \land S(y', z) \land T(z', x') \]
  \[ \land x = a \land x' = a \land y = b \land y' = b \land z = c \land z' = c \]

  \[ \rightarrow \text{ Keep the resulting fully non-looping rules} \]

\[ \rightarrow \text{ QA for the shredded instance, treefied rules, query, and axioms is equivalent to QA for the original instance, rules, query} \]
# Table of contents

1. Problem statement
2. Undecidability
3. Decidability
4. Adding FDs
5. Conclusion
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** \( \text{Funct}(R) \) on binary relations
- Could we also allow FDs on **higher-arity relations**?
  **Ex.:** Talk[\text{speaker, session}] determines Talk[\text{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 $y = y'$

→ We need an additional restriction for decidability
Non-conflicting rules and FDs [Calì et al., 2012]

Consider QA under single-head rules $\Sigma$ and FDs $\Phi$

- $\Sigma$ and $\Phi$ are separable if $QA(\Sigma, \Phi) \Leftrightarrow QA(\Sigma)$ when $I \models \Phi$
Non-conflicting rules and FDs [Calì et al., 2012]

Consider QA under single-head rules \( \Sigma \) and FDs \( \Phi \)

- \( \Sigma \) and \( \Phi \) are separable if \( QA(\Sigma, \Phi) \Leftrightarrow QA(\Sigma) \) 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 $\Sigma$ and FDs $\Phi$

- $\Sigma$ and $\Phi$ 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
    - $\bar{S} :=$ positions with an existentially quantified variable
Non-conflicting rules and FDs [Calì et al., 2012]

Consider QA under single-head rules $\Sigma$ and FDs $\Phi$

- $\Sigma$ and $\Phi$ are separable if $QA(\Sigma, \Phi) \iff QA(\Sigma)$ 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 $\Phi$: 

\[ \text{(Continued...)} \]
Consider QA under **single-head rules** $\Sigma$ and **FDs** $\Phi$

- $\Sigma$ and $\Phi$ are **separable** if $QA(\Sigma, \Phi) \iff QA(\Sigma)$ when $I \models \Phi$

- Separability guaranteed under the **non-conflicting condition**:  
  - For every **rule head** $H = R(x_1, \ldots, x_n)$: 
    - $S := \text{positions of } H \text{ with a frontier variable}$ 
    - $\overline{S} := \text{positions with an existentially quantified variable}$
  - For each **FD** $R[S'] \rightarrow R[i]$ of $\Phi$: 
    - $\rightarrow$ if $S' \subsetneq S$, fail
Non-conflicting rules and FDs [Calì et al., 2012]

Consider QA under single-head rules $\Sigma$ and FDs $\Phi$

- $\Sigma$ and $\Phi$ 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
    - $\bar{S} :=$ positions with an existentially quantified variable
  - For each FD $R[S'] \rightarrow R[i]$ of $\Phi$:
    - $\rightarrow$ if $S' \subset S$, fail
    - $\rightarrow$ if $S' = S$ and some variable occurs twice in $\bar{S}$, fail
Non-conflicting rules and FDs [Calì et al., 2012]

Consider QA under single-head rules $\Sigma$ and FDs $\Phi$

- $\Sigma$ and $\Phi$ 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 $\Phi$:
    - if $S' \subsetneq S$, fail
    - 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** $\Sigma$ and **FDs** $\Phi$

- $\Sigma$ and $\Phi$ are **separable** if $\text{QA}(\Sigma, \Phi) \iff \text{QA}(\Sigma)$ 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 $\Phi$:
    - $\rightarrow$ if $S' \subset S$, 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 rules and FDs [Calì et al., 2012]

Consider QA under **single-head rules** \( \Sigma \) and **FDs** \( \Phi \)

- \( \Sigma \) and \( \Phi \) 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 \( \Phi \):
    - \( \rightarrow \) 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 rules and FDs [Calì et al., 2012]

Consider QA under single-head rules $\Sigma$ and FDs $\Phi$

- $\Sigma$ and $\Phi$ are separable if $QA(\Sigma, \Phi) \iff QA(\Sigma)$ 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
    - $\bar{S} :=$ positions with an existentially quantified variable
  - For each FD $R[S'] \rightarrow R[i]$ of $\Phi$:
    - $\rightarrow$ if $S' \subsetneq S$, fail
    - $\rightarrow$ if $S' = S$ and some variable occurs twice in $\bar{S}$, fail

- $T(x) \Rightarrow R(y, y, x)$ 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 $\Sigma$ and FDs $\Phi$

- $\Sigma$ and $\Phi$ 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 $\Phi$:
    $\rightarrow$ if $S' \subset S$, fail
    $\rightarrow$ if $S' = S$ and some variable occurs twice in $\overline{S}$, fail

- $T(x) \Rightarrow R(y, y, x)$ 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** $\Sigma$ and **FDs** $\Phi$

- $\Sigma$ and $\Phi$ are **separable** if $QA(\Sigma, \Phi) \iff QA(\Sigma)$ 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 $\Phi$:
    - $\rightarrow$ if $S' \subset S$, 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 rules and FDs [Calì et al., 2012]

Consider QA under single-head rules $\Sigma$ and FDs $\Phi$

- $\Sigma$ and $\Phi$ are separable if $QA(\Sigma, \Phi) \iff QA(\Sigma)$ 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 $\Phi$:
    - $\rightarrow$ if $S' \subset S$, fail
    - $\rightarrow$ if $S' = S$ and some variable occurs twice in $\overline{S}$, fail


- $T(x) \Rightarrow R(y, y, x)$ 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 $\Sigma$ and FDs $\Phi$

- $\Sigma$ and $\Phi$ are separable if $\text{QA}(\Sigma, \Phi) \equiv \text{QA}(\Sigma)$ when $I \models \Phi$

- Separability guaranteed under the non-conflicting condition:
  - For every rule head $H = R(x_1, \ldots, x_n)$:
    - $S := \text{positions of } H \text{ with a frontier variable}$
    - $\overline{S} := \text{positions with an existentially quantified variable}$
  - For each FD $R[S'] \rightarrow R[i]$ of $\Phi$:
    $\rightarrow$ if $S' \subset S$, fail
    $\rightarrow$ if $S' = S$ and some variable occurs twice in $\overline{S}$, fail


- $T(x) \Rightarrow R(y, y, x)$ 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 \( \Sigma \) and FDs \( \Phi \)

- \( \Sigma \) and \( \Phi \) 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 \( \Phi \):
    - \( \rightarrow \) if \( S' \subsetneq S \), fail
    - \( \rightarrow \) if \( S' = S \) and some variable occurs twice in \( \overline{S} \), fail

- \( T(x) \Rightarrow R(y, y, x) \) 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 $\Sigma$ and FDs $\Phi$

- $\Sigma$ and $\Phi$ 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
    - $\bar{S} :=$ positions with an existentially quantified variable
  - For each FD $R[S'] \rightarrow R[i]$ of $\Phi$:
    - $\rightarrow$ if $S' \subsetneq S$, fail
    - $\rightarrow$ if $S' = S$ and some variable occurs twice in $\bar{S}$, fail

- $T(x) \Rightarrow R(y, y, x)$ 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 $\Sigma$ and FDs $\Phi$

- $\Sigma$ and $\Phi$ 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 $\Phi$:
    - $\rightarrow$ if $S' \subsetneq S$, fail
    - $\rightarrow$ if $S' = S$ and some variable occurs twice in $\overline{S}$, fail


- $T(x) \Rightarrow R(y, y, x)$ 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**

\[ \text{QA is decidable for single-head frontier-guarded } + \text{ non-conflicting FDs} \]

We have shown:

**Theorem**

\[ \text{QA is decidable for head-non-looping frontier-one rules } + \text{ rich DLs} \]

We show:

**Theorem**

\[ \text{QA is decidable for:} \]

- *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 $S' \subset S$, 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)
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)
Related things I work on

- 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
Related things I work on

- 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

---

\(^1\)With Michael Benedikt, ongoing work
\(^2\)With Michael Benedikt, [Amarilli and Benedikt, 2015], LICS’15
Related things I work on

- 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

- Also: probabilistic databases, partial orders, crowdsourcing...

---

\(^1\)With Michael Benedikt, ongoing work
\(^2\)With Michael Benedikt, [Amarilli and Benedikt, 2015], LICS’15
Related things I work on

- 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

- Also: probabilistic databases, partial orders, crowdsourcing...

---

\(^1\)With Michael Benedikt, ongoing work
\(^2\)With Michael Benedikt, [Amarilli and Benedikt, 2015], LICS’15

Thanks for your attention!
References


