Evaluating the Generality of Disjunctive Model Faithful Acyclicity on OWL ontologies

Lukas Gerlach

Knowledge-Based Systems Group
Technische Universität Dresden, Germany

18.03.2021

Disjunctive Existential Rules

$$\forall \vec{x} \forall \vec{y}. Body(\vec{x}, \vec{y}) \rightarrow \bigvee_{i=1}^{n} \exists \vec{z}_i . Head^i(\vec{x}_i, \vec{z}_i)$$

Disjunctive Existential Rules

$$\forall\vec{x}\forall\vec{y}.Body(\vec{x},\vec{y}) \rightarrow \bigvee_{i=1}^{n} \exists\vec{z}_i.Head^i(\vec{x}_i,\vec{z}_i)$$

- $Body$ and $Head^i$: conjunctions of atoms
- $\vec{x},\vec{y},\vec{z}_i$: pairwise disjoint lists of variables
- $\bigcup_{i=1}^{n}\vec{x}_i = \vec{x}$

Disjunctive Existential Rules

$$\forall \vec{x} \forall \vec{y}. Body(\vec{x}, \vec{y}) \rightarrow \bigvee_{i=1}^{n} \exists \vec{z_i}. Head^i(\vec{x_i}, \vec{z_i})$$

- $Body$ and $Head^i$: conjunctions of atoms
- $\vec{x}, \vec{y}, \vec{z_i}$: pairwise disjoint lists of variables
- $\bigcup_{i=1}^{n} \vec{x_i} = \vec{x}$

$$SalamiPizza(x) \rightarrow \exists z. hasBase(x, z) \wedge CheesePizza(z)$$

Disjunctive Existential Rules

$$\forall \vec{x} \forall \vec{y}. Body(\vec{x}, \vec{y}) \to \bigvee_{i=1}^{n} \exists \vec{z}_i . Head^i(\vec{x}_i, \vec{z}_i)$$

- $Body$ and $Head^i$: conjunctions of atoms
- $\vec{x}, \vec{y}, \vec{z}_i$: pairwise disjoint lists of variables
- $\bigcup_{i=1}^{n} \vec{x}_i = \vec{x}$

$$SalamiPizza(x) \to \exists z. hasBase(x, z) \land CheesePizza(z)$$

$$Pizza(x) \to CheesePizza(x) \lor SalamiPizza(x)$$

Disjunctive Existential Rules

$$\forall \vec{x} \forall \vec{y}. Body(\vec{x}, \vec{y}) \to \bigvee_{i=1}^{n} \exists \vec{z}_i . Head^i(\vec{x}_i, \vec{z}_i)$$

- $Body$ and $Head^i$: conjunctions of atoms
- $\vec{x}, \vec{y}, \vec{z}_i$: pairwise disjoint lists of variables
- $\bigcup_{i=1}^{n} \vec{x}_i = \vec{x}$

$$SalamiPizza(x) \to \exists z. hasBase(x, z) \land CheesePizza(z)$$

$$Pizza(x) \to CheesePizza(x) \lor SalamiPizza(x)$$

$$CheesePizza(x) \to Pizza(x)$$

Disjunctive Existential Rules

$$\forall \vec{x} \forall \vec{y}.Body(\vec{x}, \vec{y}) \rightarrow \bigvee_{i=1}^{n} \exists \vec{z}_i.Head^i(\vec{x}_i, \vec{z}_i)$$

- $Body$ and $Head^i$: conjunctions of atoms
- $\vec{x}, \vec{y}, \vec{z}_i$: pairwise disjoint lists of variables
- $\bigcup_{i=1}^{n} \vec{x}_i = \vec{x}$

$$SalamiPizza(x) \rightarrow \exists z.hasBase(x, z) \wedge CheesePizza(z)$$

$$Pizza(x) \rightarrow CheesePizza(x) \vee SalamiPizza(x)$$

$$CheesePizza(x) \rightarrow Pizza(x)$$

$$SalamiPizza(x) \rightarrow Pizza(x)$$

Disjunctive Skolem Chase

$$\begin{aligned}
SalamiPizza(x) &\rightarrow \exists z.hasBase(x, z) \wedge CheesePizza(z) \\
Pizza(x) &\rightarrow CheesePizza(x) \vee SalamiPizza(x) \\
CheesePizza(x) &\rightarrow Pizza(x)
\end{aligned}$$

Disjunctive Skolem Chase

$$SalamiPizza(x) \rightarrow \exists z.hasBase(x, z) \wedge CheesePizza(z)$$
$$Pizza(x) \rightarrow CheesePizza(x) \vee SalamiPizza(x)$$
$$CheesePizza(x) \rightarrow Pizza(x)$$

- $somePizza : Pizza$

Disjunctive Skolem Chase

$$SalamiPizza(x) \rightarrow \exists z.hasBase(x, z) \wedge CheesePizza(z)$$
$$Pizza(x) \rightarrow CheesePizza(x) \vee SalamiPizza(x)$$
$$CheesePizza(x) \rightarrow Pizza(x)$$

- $somePizza : Pizza, CheesePizza$

- $somePizza : Pizza, SalamiPizza$

Disjunctive Skolem Chase

$$SalamiPizza(x) \rightarrow hasBase(x, f^z(x)) \wedge CheesePizza(f^z(x))$$
$$Pizza(x) \rightarrow CheesePizza(x) \vee SalamiPizza(x)$$
$$CheesePizza(x) \rightarrow Pizza(x)$$

- $somePizza : Pizza, CheesePizza$

- $somePizza : Pizza, SalamiPizza$

Disjunctive Skolem Chase

$$SalamiPizza(x) \to hasBase(x, f^z(x)) \wedge CheesePizza(f^z(x))$$
$$Pizza(x) \to CheesePizza(x) \vee SalamiPizza(x)$$
$$CheesePizza(x) \to Pizza(x)$$

- $somePizza : Pizza, CheesePizza$

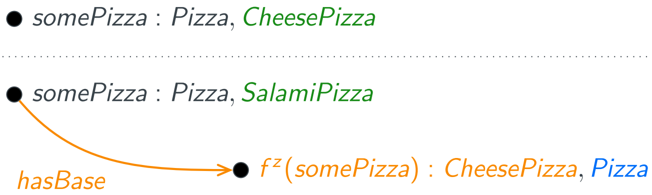
- $somePizza : Pizza, SalamiPizza$

$hasBase$

- $f^z(somePizza) : CheesePizza$

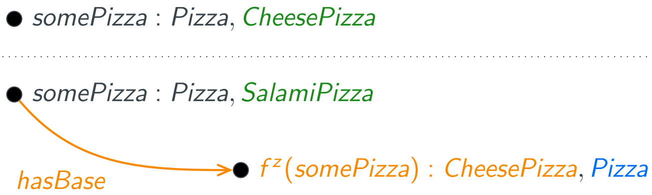
Disjunctive Skolem Chase

$$SalamiPizza(x) \rightarrow hasBase(x, f^z(x)) \land CheesePizza(f^z(x))$$
$$Pizza(x) \rightarrow CheesePizza(x) \lor SalamiPizza(x)$$
$$CheesePizza(x) \rightarrow Pizza(x)$$



Disjunctive Skolem Chase

$$SalamiPizza(x) \rightarrow hasBase(x, f^z(x)) \wedge CheesePizza(f^z(x))$$
$$Pizza(x) \rightarrow CheesePizza(x) \vee SalamiPizza(x)$$
$$CheesePizza(x) \rightarrow Pizza(x)$$



The *disjunctive skolem chase* yields a *universal model set*.

Acyclicity Notions

Problem: Termination of the chase is undecidable.
Sufficient conditions for termination have been introduced.

Acyclicity Notions

Problem: Termination of the chase is undecidable.
Sufficient conditions for termination have been introduced.

Disjunctive Existential Rule Sets

Acyclicity Notions

Problem: Termination of the chase is undecidable.
Sufficient conditions for termination have been introduced.

Disjunctive Existential Rule Sets

Terminating w.r.t.
(disjunctive) restricted chase

Acyclicity Notions

Problem: Termination of the chase is undecidable.
Sufficient conditions for termination have been introduced.

Disjunctive Existential Rule Sets

Terminating w.r.t.
(disjunctive) restricted chase

Terminating w.r.t.
disjunctive skolem chase

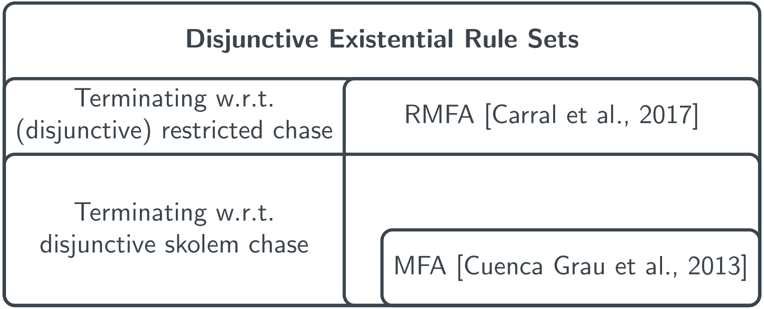
Acyclicity Notions

Problem: Termination of the chase is undecidable.
Sufficient conditions for termination have been introduced.

Disjunctive Existential Rule Sets	
Terminating w.r.t. (disjunctive) restricted chase	RMFA [Carral et al., 2017]
Terminating w.r.t. disjunctive skolem chase	

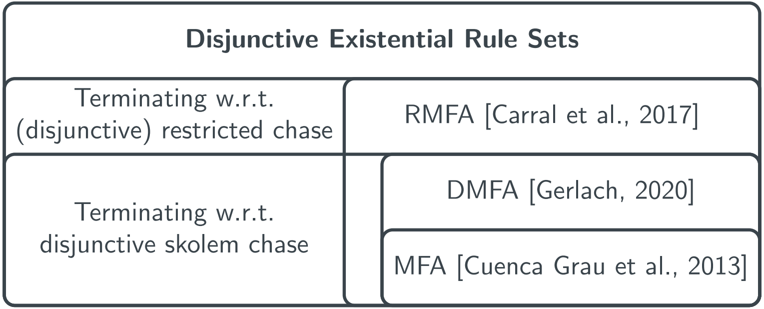
Acyclicity Notions

Problem: Termination of the chase is undecidable.
Sufficient conditions for termination have been introduced.



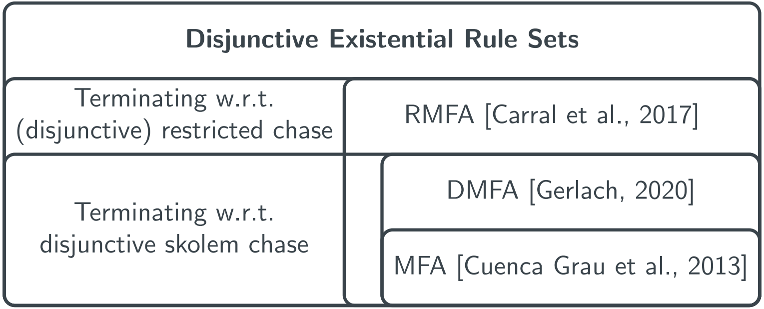
Acyclicity Notions

Problem: Termination of the chase is undecidable.
Sufficient conditions for termination have been introduced.



Acyclicity Notions

Problem: Termination of the chase is undecidable.
Sufficient conditions for termination have been introduced.



General Idea: Compute the chase on a *critical instance* using only rules that are not *blocked*.

Disjunctive Model Faithful Acyclicity

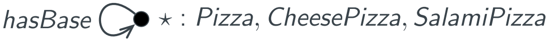
General Idea: Compute the chase on a *critical instance* using only rules that are not *blocked*.

$$SalamiPizza(x) \rightarrow hasBase(x, f^z(x)) \wedge CheesePizza(f^z(x))$$
$$Pizza(x) \rightarrow CheesePizza(x) \vee SalamiPizza(x)$$
$$CheesePizza(x) \rightarrow Pizza(x)$$

Disjunctive Model Faithful Acyclicity

General Idea: Compute the chase on a *critical instance* using only rules that are not *blocked*.

$$SalamiPizza(x) \rightarrow hasBase(x, f^z(x)) \wedge CheesePizza(f^z(x))$$
$$Pizza(x) \rightarrow CheesePizza(x) \vee SalamiPizza(x)$$
$$CheesePizza(x) \rightarrow Pizza(x)$$



Disjunctive Model Faithful Acyclicity

General Idea: Compute the chase on a *critical instance* using only rules that are not *blocked*.

$$SalamiPizza(x) \rightarrow hasBase(x, f^z(x)) \wedge CheesePizza(f^z(x))$$
$$Pizza(x) \rightarrow CheesePizza(x) \vee SalamiPizza(x)$$
$$CheesePizza(x) \rightarrow Pizza(x)$$

Disjunctive Model Faithful Acyclicity

General Idea: Compute the chase on a *critical instance* using only rules that are not *blocked*.

$$SalamiPizza(x) \rightarrow hasBase(x, f^z(x)) \wedge CheesePizza(f^z(x))$$
$$Pizza(x) \rightarrow CheesePizza(x) \vee SalamiPizza(x)$$
$$CheesePizza(x) \rightarrow Pizza(x)$$

Disjunctive Model Faithful Acyclicity

General Idea: Compute the chase on a *critical instance* using only rules that are not *blocked*.

$$SalamiPizza(x) \rightarrow hasBase(x, f^z(x)) \land CheesePizza(f^z(x))$$
$$Pizza(x) \rightarrow CheesePizza(x) \lor SalamiPizza(x)$$
$$CheesePizza(x) \rightarrow Pizza(x)$$

The second rule is blocked now.
The rule set is DMFA since the computed fact set has no *cyclic terms*.

Evaluation Outline

Datasets:

- Oxford ontology repository (OXFD)
  797 ontologies[1]
- OWL Reasoner Evaluation 2015 (ORE15)
  1920 ontologies[2]

Steps:

[1] OXFD - https://www.cs.ox.ac.uk/isg/ontologies/

[2] ORE15 - https://zenodo.org/record/18578

[3] Ontology Normalizer - https://github.com/dcarralma/OntologyNormalizer

[4] Translation - https://gitlab.com/m0nstR/owl-to-disjunctive-existential-rules-converter

[5] DMFA-Implementation https://gitlab.com/m0nstR/dmfa-checker

Evaluation Outline

Datasets:

- Oxford ontology repository (OXFD)
  797 ontologies[1]
- OWL Reasoner Evaluation 2015 (ORE15)
  1920 ontologies[2]

$$SalamiPizza \sqsubseteq \geq 1 hasBase.CheesePizza$$
$$Pizza \sqsubseteq CheesePizza \sqcup SalamiPizza$$
$$CheesePizza \sqsubseteq Pizza$$

Steps:

1. Normalize Ontologies using an existing tool[3]

[1] OXFD - https://www.cs.ox.ac.uk/isg/ontologies/
[2] ORE15 - https://zenodo.org/record/18578
[3] Ontology Normalizer - https://github.com/dcarralma/OntologyNormalizer
[4] Translation - https://gitlab.com/m0nstR/owl-to-disjunctive-existential-rules-converter
[5] DMFA-Implementation https://gitlab.com/m0nstR/dmfa-checker

Evaluation Outline

Datasets:

- Oxford ontology repository (OXFD)
  797 ontologies[1]
- OWL Reasoner Evaluation 2015 (ORE15)
  1920 ontologies[2]

Steps:

1. Normalize Ontologies using an existing tool[3]
2. Translate into Disjunctive Existential Rules[4]

$$SalamiPizza \sqsubseteq \geq 1 hasBase.CheesePizza$$
$$Pizza \sqsubseteq CheesePizza \sqcup SalamiPizza$$
$$CheesePizza \sqsubseteq Pizza$$

$$\downarrow$$

$$SalamiPizza(x) \rightarrow \exists z.hasBase(x, z) \wedge CheesePizza(z)$$
$$Pizza(x) \rightarrow CheesePizza(x) \vee SalamiPizza(x)$$
$$CheesePizza(x) \rightarrow Pizza(x)$$

[1] OXFD - https://www.cs.ox.ac.uk/isg/ontologies/
[2] ORE15 - https://zenodo.org/record/18578
[3] Ontology Normalizer - https://github.com/dcarralma/OntologyNormalizer
[4] Translation - https://gitlab.com/m0nstR/owl-to-disjunctive-existential-rules-converter
[5] DMFA-Implementation https://gitlab.com/m0nstR/dmfa-checker

Evaluation Outline

Datasets:

- Oxford ontology repository (OXFD)
  797 ontologies[1]
- OWL Reasoner Evaluation 2015 (ORE15)
  1920 ontologies[2]

Steps:

1. Normalize Ontologies using an existing tool[3]
2. Translate into Disjunctive Existential Rules[4]
3. Check MFA, DMFA, and RMFA[5]

$$SalamiPizza \sqsubseteq \geq 1 hasBase.CheesePizza$$
$$Pizza \sqsubseteq CheesePizza \sqcup SalamiPizza$$
$$CheesePizza \sqsubseteq Pizza$$

$\downarrow$

$$SalamiPizza(x) \rightarrow \exists z.hasBase(x,z) \wedge CheesePizza(z)$$
$$Pizza(x) \rightarrow CheesePizza(x) \vee SalamiPizza(x)$$
$$CheesePizza(x) \rightarrow Pizza(x)$$

$\downarrow$

not MFA but DMFA and RMFA

[1] OXFD - https://www.cs.ox.ac.uk/isg/ontologies/
[2] ORE15 - https://zenodo.org/record/18578
[3] Ontology Normalizer - https://github.com/dcarralma/OntologyNormalizer
[4] Translation - https://gitlab.com/m0nstR/owl-to-disjunctive-existential-rules-converter
[5] DMFA-Implementation https://gitlab.com/m0nstR/dmfa-checker

Translation of normalized Axioms

$$A_1 \sqcap \cdots \sqcap A_n \sqsubseteq B \qquad A_1(x) \land \cdots \land A_n(x) \to B(x) \tag{1}$$

Here, $A, A_1, \ldots, A_n, B, B_1, \ldots, B_{n'}$ are classes, $R, R_1, \ldots, R_k, S$ are (inverse) properties, $a_1, \ldots, a_l$ are individuals, and m, m' are integers ≥ 0.

Translation of normalized Axioms

$$A_1 \sqcap \cdots \sqcap A_n \sqsubseteq B \qquad A_1(x) \wedge \cdots \wedge A_n(x) \rightarrow B(x) \tag{1}$$

$$A \sqsubseteq B_1 \sqcup \cdots \sqcup B_{n'} \qquad A(x) \rightarrow B_1(x) \vee \cdots \vee B_{n'}(x) \tag{2}$$

Here, $A, A_1, \ldots, A_n, B, B_1, \ldots, B_{n'}$ are classes, $R, R_1, \ldots, R_k, S$ are (inverse) properties, $a_1, \ldots, a_l$ are individuals, and m, m' are integers ≥ 0.

Translation of normalized Axioms

$$
\begin{array}{lll}
A_1 \sqcap \cdots \sqcap A_n \sqsubseteq B & A_1(x) \wedge \cdots \wedge A_n(x) \rightarrow B(x) & (1) \\
A \sqsubseteq B_1 \sqcup \cdots \sqcup B_{n'} & A(x) \rightarrow B_1(x) \vee \cdots \vee B_{n'}(x) & (2) \\
A \sqsubseteq \forall R.B & A(x) \wedge R(x,y) \rightarrow B(y) & (3) \\
A \sqsubseteq \geq mR.B & \text{(more involved)} & (4) \\
A \sqsubseteq \leq m'R.B & \text{(more involved)} & (5) \\
A \sqsubseteq \{ a_1, \ldots, a_l \} & \text{(more involved)} & (6) \\
R_1 \circ \cdots \circ R_k \sqsubseteq S & R_1(x_1,x_2) \wedge \cdots \wedge R_k(x_k,x_{k+1}) \rightarrow S(x_1,x_{k+1}) & (7) \\
R \sqcap S \sqsubseteq \bot & R(x,y) \wedge S(x,y) \rightarrow \bot & (8) \\
A \sqsubseteq \exists R.\mathrm{Self} & A(x) \rightarrow R(x,x) & (9) \\
\exists R.\mathrm{Self} \sqsubseteq A & R(x,x) \rightarrow A(x) & (10)
\end{array}
$$

Here, $A, A_1, \ldots, A_n, B, B_1, \ldots, B_{n'}$ are classes, $R, R_1, \ldots, R_k, S$ are (inverse) properties, $a_1, \ldots, a_l$ are individuals, and m, m' are integers ≥ 0.

Complicated Axioms

The translation for $A \sqsubseteq {\geq}2R.B$ is more involved:

$$
\begin{array}{lll}
A(x) \to \exists z_1.R_1(x, z_1) \wedge B(z_1) & R_1(x, y) \to R(x, y) & R(x, y) \to R_1(x, y) \vee R_2(x, y) \\
A(x) \to \exists z_2.R_2(x, z_2) \wedge B(z_2) & R_2(x, y) \to R(x, y) & R_1(x, y) \wedge R_2(x, y) \to \bot
\end{array}
$$

Complicated Axioms

The translation for $A \sqsubseteq {\geq}2R.B$ is more involved:

$$A(x) \to \exists z_1.R_1(x, z_1) \land B(z_1) \quad R_1(x, y) \to R(x, y) \quad R(x, y) \to R_1(x, y) \lor R_2(x, y)$$
$$A(x) \to \exists z_2.R_2(x, z_2) \land B(z_2) \quad R_2(x, y) \to R(x, y) \quad R_1(x, y) \land R_2(x, y) \to \bot$$

For $A \sqsubseteq {\leq}1R.B$ and $A \sqsubseteq \{a, b\}$, we require equality:

$$A(x) \land R(x, u) \land B(u) \land R(x, v) \land B(v) \to u = v \qquad A(x) \to x = a \lor x = b$$

Complicated Axioms

The translation for $A \sqsubseteq {\geq}2R.B$ is more involved:

$$
\begin{array}{lll}
A(x) \rightarrow \exists z_1.R_1(x, z_1) \wedge B(z_1) & R_1(x, y) \rightarrow R(x, y) & R(x, y) \rightarrow R_1(x, y) \vee R_2(x, y) \\
A(x) \rightarrow \exists z_2.R_2(x, z_2) \wedge B(z_2) & R_2(x, y) \rightarrow R(x, y) & R_1(x, y) \wedge R_2(x, y) \rightarrow \bot
\end{array}
$$

For $A \sqsubseteq {\leq}1R.B$ and $A \sqsubseteq \{ a, b \}$, we require equality:

$$A(x) \wedge R(x, u) \wedge B(u) \wedge R(x, v) \wedge B(v) \rightarrow u = v \qquad A(x) \rightarrow x = a \vee x = b$$

Problem: DMFA does not support equality.
For now: Omit axioms that require equality.
Later on: Axiomatize equality as suggested in [Carral and Urbani, 2020].

DMFA Implementation

Input: Ruleset R
Output: Is R DMFA?

$\Delta F := \mathit{criticalInstance}(R)$

DMFA Implementation

Input: Ruleset R

Output: Is R DMFA?

$\Delta F := \mathit{criticalInstance}(R)$

$\mathit{vlogReasoner} := \mathit{VlogReasoner}(R)$ `// uses VLog bindings of Rulewerk`

$\mathit{exisReasoner} := \mathit{ExistentialReasoner}(R)$

DMFA Implementation

Input: Ruleset R

Output: Is R DMFA?

$\Delta F := \mathit{criticalInstance}(R)$

$\mathit{vlogReasoner} := \mathit{VlogReasoner}(R)$ `// uses VLog bindings of Rulewerk`

$\mathit{exisReasoner} := \mathit{ExistentialReasoner}(R)$

while $\Delta F \neq \emptyset$ **do**

DMFA Implementation

Input: Ruleset R

Output: Is R DMFA?

$\Delta F := \mathit{criticalInstance}(R)$

$\mathit{vlogReasoner} := \mathit{VlogReasoner}(R)$ `// uses VLog bindings of Rulewerk`

$\mathit{exisReasoner} := \mathit{ExistentialReasoner}(R)$

while $\Delta F \neq \emptyset$ **do**

  `// keeps track of already computed facts internally`

  $\Delta F := \Delta F \cup \mathit{vlogReasoner.newFactsFromVlogDatalogClosure}(\Delta F)$

DMFA Implementation

Input: Ruleset R

Output: Is R DMFA?

```
ΔF := criticalInstance(R)
vlogReasoner := VlogReasoner(R) // uses VLog bindings of Rulewerk
exisReasoner := ExistentialReasoner(R)
while ΔF ≠ ∅ do
    // keeps track of already computed facts internally
    ΔF := ΔF ∪ vlogReasoner.newFactsFromVlogDatalogClosure(ΔF)
    // non-exhaustive application
    // keeps track of already computed facts internally
    ΔF := exisReasoner.newFactsFromUnblockedNonDatalogRules(ΔF)
```

DMFA Implementation

Input: Ruleset R
Output: Is R DMFA?

```
ΔF := criticalInstance(R)
vlogReasoner := VlogReasoner(R) // uses VLog bindings of Rulewerk
exisReasoner := ExistentialReasoner(R)
while ΔF ≠ ∅ do
    // keeps track of already computed facts internally
    ΔF := ΔF ∪ vlogReasoner.newFactsFromVlogDatalogClosure(ΔF)
    // non-exhaustive application
    // keeps track of already computed facts internally
    ΔF := exisReasoner.newFactsFromUnblockedNonDatalogRules(ΔF)
    if hasCyclicTerm(ΔF) then
        return False
    end
end
```

DMFA Implementation

Input: Ruleset R

Output: Is R DMFA?

```
ΔF := criticalInstance(R)
vlogReasoner := VlogReasoner(R) // uses VLog bindings of Rulewerk
exisReasoner := ExistentialReasoner(R)
while ΔF ≠ ∅ do
    // keeps track of already computed facts internally
    ΔF := ΔF ∪ vlogReasoner.newFactsFromVlogDatalogClosure(ΔF)
    // non-exhaustive application
    // keeps track of already computed facts internally
    ΔF := exisReasoner.newFactsFromUnblockedNonDatalogRules(ΔF)
    if hasCyclicTerm(ΔF) then
        return False
    end
end
return True
```

Evaluation Results

OXFD: 797
ORE15: 1920

Evaluation Results

$$\begin{array}{rl} \text{OXFD: } 797 & \xrightarrow{\text{Normalization}} 791 \\ \text{ORE15: } 1920 & \quad\quad\quad\quad\quad\quad 1910 \end{array}$$

Evaluation Results

OXFD: 797 —Normalization→ 791 —Translation→ 789

ORE15: 1920 → 1910 → 1888

Evaluation Results

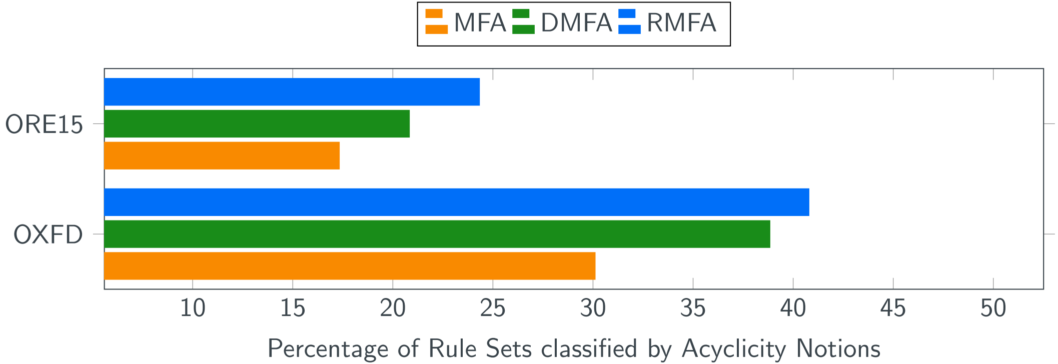
$$\begin{array}{r} \text{OXFD: } 797 \\ \text{ORE15: } 1920 \end{array} \xrightarrow{\text{Normalization}} \begin{array}{c} 791 \\ 1910 \end{array} \xrightarrow{\text{Translation}} \begin{array}{c} 789 \\ 1888 \end{array} \xrightarrow[\text{}]{\substack{\text{at least one} \\ \exists\text{- and } \vee\text{-rule}}} \begin{array}{c} 137 \\ 613 \end{array}$$

Evaluation Results

OXFD: 797	Normalization →	791	Translation →	789	at least one ∃- and ∨-rule →	137	all checks finished →	103
ORE15: 1920		1910		1888		613		514

Evaluation Results

OXFD: 797	Normalization →	791	Translation →	789	at least one ∃- and ∨-rule →	137	all checks finished →	103
ORE15: 1920		1910		1888		613		514



Where are the differences?

Observation: 6 out of 9 rule sets for OXFD and 11 out of 18 rule sets for ORE15 that are DMFA but not MFA come from BioPAX.[6]

[6] BioPAX - http://www.biopax.org/

Where are the differences?

Observation: 6 out of 9 rule sets for OXFD and 11 out of 18 rule sets for ORE15 that are DMFA but not MFA come from BioPAX.[6]
They are similar or even identical.

[6] BioPAX - http://www.biopax.org/

Where are the differences?

Observation: 6 out of 9 rule sets for OXFD and 11 out of 18 rule sets for ORE15 that are DMFA but not MFA come from BioPAX.[6]
They are similar or even identical.

We have to careful in generalizing our findings here.

[6] BioPAX - `http://www.biopax.org/`

Where are the differences?

Observation: 6 out of 9 rule sets for OXFD and 11 out of 18 rule sets for ORE15 that are DMFA but not MFA come from BioPAX.[6]
They are similar or even identical.

We have to careful in generalizing our findings here.

OXFD 00007: DMFA but not MFA

$$
\begin{aligned}
\mathit{Evidence}(x) &\rightarrow \exists z.\mathit{ConfidenceProp}(x, z) \\
\mathit{ConfidenceProp}(x, y) &\rightarrow \mathit{Confidence}(y) \\
\mathit{Confidence}(x) &\rightarrow \mathit{Xref}(x) \\
\mathit{Xref}(x) &\rightarrow \mathit{Evidence}(x) \lor \mathit{Confidence}(x) \\
\mathit{Evidence}(x) \land \mathit{Confidence}(x) &\rightarrow \bot
\end{aligned}
$$

[6] BioPAX - `http://www.biopax.org/`

Where are the differences?

Observation: 6 out of 9 rule sets for OXFD and 11 out of 18 rule sets for ORE15 that are DMFA but not MFA come from BioPAX.[6]
They are similar or even identical.

We have to careful in generalizing our findings here.

OXFD 00007: DMFA but not MFA

$$\begin{aligned}
\mathit{Evidence}(x) &\to \exists z.\mathit{ConfidenceProp}(x,z) \\
\mathit{ConfidenceProp}(x,y) &\to \mathit{Confidence}(y) \\
\mathit{Confidence}(x) &\to \mathit{Xref}(x) \\
\mathit{Xref}(x) &\to \mathit{Evidence}(x) \vee \mathit{Confidence}(x) \\
\mathit{Evidence}(x) \wedge \mathit{Confidence}(x) &\to \bot
\end{aligned}$$

OXFD 00114: RMFA but not DMFA

$$\begin{aligned}
\mathit{Sibling}(x) &\to \exists z.\mathit{hasSibling}(z,x) \wedge \mathit{Person}(z) \\
\mathit{Person}(x) \wedge \mathit{hasSibling}(x,y) &\to \mathit{Sibling}(y) \\
\mathit{Sibling}(x) &\to \mathit{Person}(x) \\
\mathit{hasSibling}(x,y) &\to \mathit{hasSibling}(y,x)
\end{aligned}$$

Not even terminating w.r.t. the disjunctive skolem chase.

[6] BioPAX - http://www.biopax.org/

Second Example in Detail

$$Sibling(x) \rightarrow \exists z.hasSibling(z, x) \land Person(z)$$
$$Person(x) \land hasSibling(x, y) \rightarrow Sibling(y)$$
$$Sibling(x) \rightarrow Person(x)$$
$$hasSibling(x, y) \rightarrow hasSibling(y, x)$$

Second Example in Detail

$$Sibling(x) \rightarrow \exists z.hasSibling(z, x) \wedge Person(z)$$
$$Person(x) \wedge hasSibling(x, y) \rightarrow Sibling(y)$$
$$Sibling(x) \rightarrow Person(x)$$
$$hasSibling(x, y) \rightarrow hasSibling(y, x)$$

$me : Sibling$ ●

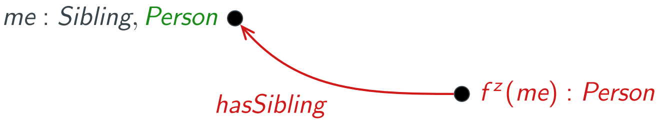
Second Example in Detail

$$Sibling(x) \rightarrow \exists z.hasSibling(z, x) \wedge Person(z)$$
$$Person(x) \wedge hasSibling(x, y) \rightarrow Sibling(y)$$
$$Sibling(x) \rightarrow Person(x)$$
$$hasSibling(x, y) \rightarrow hasSibling(y, x)$$

$me : Sibling, Person$ ●

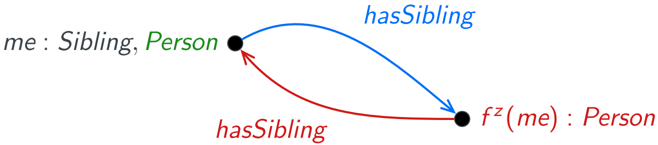
Second Example in Detail

$$Sibling(x) \rightarrow \exists z. hasSibling(z, x) \wedge Person(z)$$
$$Person(x) \wedge hasSibling(x, y) \rightarrow Sibling(y)$$
$$Sibling(x) \rightarrow Person(x)$$
$$hasSibling(x, y) \rightarrow hasSibling(y, x)$$



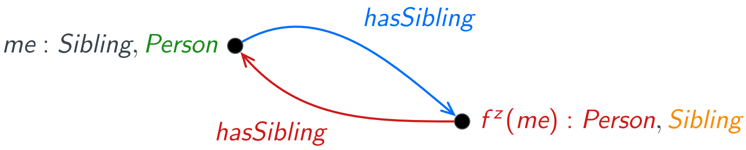
Second Example in Detail

$$Sibling(x) \rightarrow \exists z.hasSibling(z, x) \wedge Person(z)$$
$$Person(x) \wedge hasSibling(x, y) \rightarrow Sibling(y)$$
$$Sibling(x) \rightarrow Person(x)$$
$$hasSibling(x, y) \rightarrow hasSibling(y, x)$$



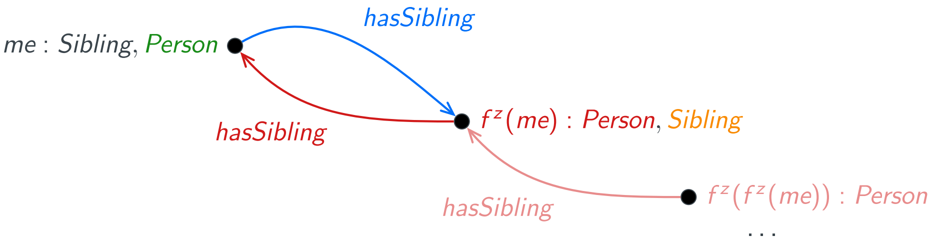
Second Example in Detail

$$Sibling(x) \rightarrow \exists z.hasSibling(z, x) \land Person(z)$$
$$Person(x) \land hasSibling(x, y) \rightarrow Sibling(y)$$
$$Sibling(x) \rightarrow Person(x)$$
$$hasSibling(x, y) \rightarrow hasSibling(y, x)$$



Second Example in Detail

$$Sibling(x) \rightarrow \exists z.hasSibling(z, x) \wedge Person(z)$$
$$Person(x) \wedge hasSibling(x, y) \rightarrow Sibling(y)$$
$$Sibling(x) \rightarrow Person(x)$$
$$hasSibling(x, y) \rightarrow hasSibling(y, x)$$



Conclusion

Results:

- DMFA improves upon MFA and comes closer to RMFA in practice.
- A significant number of rule sets is not even RMFA.
- We find rule sets that are RMFA but not DMFA that do not terminate w.r.t. the disjunctive skolem chase.

Conclusion

Results:

- DMFA improves upon MFA and comes closer to RMFA in practice.
- A significant number of rule sets is not even RMFA.
- We find rule sets that are RMFA but not DMFA that do not terminate w.r.t. the disjunctive skolem chase.

Open Questions:

- What is the impact of the translation (especially equality) on the results?
- How tight is DMFA?

Conclusion

Results:

- DMFA improves upon MFA and comes closer to RMFA in practice.
- A significant number of rule sets is not even RMFA.
- We find rule sets that are RMFA but not DMFA that do not terminate w.r.t. the disjunctive skolem chase.

Open Questions:

- What is the impact of the translation (especially equality) on the results?
- How tight is DMFA?

Future Work:

- Handling for equality in the translation.
- Investigation and evaluation of cyclicity notions for the disjunctive skolem chase.
- Performance centric evaluation of DMFA.
- Implementation of the disjunctive skolem chase, e.g. with ASP solvers with lazy grounding.

References I

(2020).
The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020. AAAI Press.

Abiteboul, S., Hull, R., and Vianu, V. (1995).
Foundations of Databases.
Addison-Wesley.

Adrian, W. T., Alviano, M., Calimeri, F., Cuteri, B., Dodaro, C., Faber, W., Fuscà, D., Leone, N., Manna, M., Perri, S., Ricca, F., Veltri, P., and Zangari, J. (2018).
The ASP system DLV: advancements and applications.
Künstliche Intell., 32(2-3):177–179.

References II

Aho, A. V., Beeri, C., and Ullman, J. D. (1979).
The theory of joins in relational databases.
ACM Trans. Database Syst., 4(3):297–314.

Alviano, M., Amendola, G., Dodaro, C., Leone, N., Maratea, M., and Ricca, F. (2019).
Evaluation of disjunctive programs in WASP.
In [Balduccini et al., 2019], pages 241–255.

Baget, J., Leclère, M., Mugnier, M., and Salvat, E. (2011).
On rules with existential variables: Walking the decidability line.
Artif. Intell., 175(9-10):1620–1654.

Balduccini, M., Lierler, Y., and Woltran, S., editors (2019).
Logic Programming and Nonmonotonic Reasoning - 15th International Conference, LPNMR 2019, volume 11481 of *Lecture Notes in Computer Science*. Springer.

References III

Beeri, C. and Vardi, M. Y. (1981).
The implication problem for data dependencies.
In [Even and Kariv, 1981], pages 73–85.

Benedikt, M., Konstantinidis, G., Mecca, G., Motik, B., Papotti, P., Santoro, D., and Tsamoura, E. (2017).
Benchmarking the chase.
In [Sallinger et al., 2017], pages 37–52.

Bessiere, C., editor (2020).
Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020. ijcai.org.
Scheduled for July 2020, Yokohama, Japan, postponed due to the Corona pandemic.

References IV

Bourhis, P., Leclère, M., Mugnier, M., Tison, S., Ulliana, F., and Gallois, L. (2019).
Oblivious and semi-oblivious boundedness for existential rules.
In [Kraus, 2019], pages 1581–1587.

Bourhis, P., Manna, M., Morak, M., and Pieris, A. (2016).
Guarded-based disjunctive tuple-generating dependencies.
ACM Trans. Database Syst., 41(4):27:1–27:45.

Bourhis, P., Morak, M., and Pieris, A. (2013).
The impact of disjunction on query answering under guarded-based existential rules.
In [Eiter et al., 2013], pages 539–551.

Calì, A., Gottlob, G., and Pieris, A. (2010).
Query answering under non-guarded rules in datalog+/-.
In [Hitzler and Lukasiewicz, 2010], pages 1–17.

References V

Calvanese, D., Lenzerini, M., and Motwani, R., editors (2002).
Database Theory - ICDT 2003, 9th International Conference, volume 2572 of *Lecture Notes in Computer Science*. Springer.

Carral, D., Dragoste, I., and Krötzsch, M. (2017).
Restricted chase (non)termination for existential rules with disjunctions.
In [Sierra, 2017], pages 922–928.

Carral, D. and Krötzsch, M. (2020).
Rewriting the description logic ALCHIQ to disjunctive existential rules.
In [Bessiere, 2020], pages 1777–1783.
Scheduled for July 2020, Yokohama, Japan, postponed due to the Corona pandemic.

Carral, D. and Urbani, J. (2020).
Checking chase termination over ontologies of existential rules with equality.
In [DBL, 2020], pages 2758–2765.

References VI

Carro, M., King, A., Saeedloei, N., and Vos, M. D., editors (2016).
Technical Communications of the 32nd International Conference on Logic Programming, ICLP 2016 TCs, volume 52 of *OASICS*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

Cuenca Grau, B., Horrocks, I., Krötzsch, M., Kupke, C., Magka, D., Motik, B., and Wang, Z. (2013).
Acyclicity notions for existential rules and their application to query answering in ontologies.
J. Artif. Intell. Res., 47:741–808.

Deutsch, A., Nash, A., and Remmel, J. B. (2008).
The chase revisited.
In [Lenzerini and Lembo, 2008], pages 149–158.

References VII

Deutsch, A. and Tannen, V. (2003).
Reformulation of XML queries and constraints.
In [Calvanese et al., 2002], pages 225–241.

Eiter, T., Glimm, B., Kazakov, Y., and Krötzsch, M., editors (2013).
Informal Proceedings of the 26th International Workshop on Description Logics, volume 1014 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Esparza, J., Fraigniaud, P., Husfeldt, T., and Koutsoupias, E., editors (2014).
Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014, volume 8573 of *Lecture Notes in Computer Science*. Springer.

Even, S. and Kariv, O., editors (1981).
Automata, Languages and Programming, 8th Colloquium, volume 115 of *Lecture Notes in Computer Science*. Springer.

References VIII

Fagin, R., Kolaitis, P. G., Miller, R. J., and Popa, L. (2005).
Data exchange: semantics and query answering.
Theor. Comput. Sci., 336(1):89–124.

Gebser, M., Kaminski, R., Kaufmann, B., Ostrowski, M., Schaub, T., and Wanko, P. (2016).
Theory solving made easy with clingo 5.
In [Carro et al., 2016], pages 2:1–2:15.

Gebser, M., Kaminski, R., Kaufmann, B., and Schaub, T. (2012a).
Answer Set Solving in Practice.
Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers.

References IX

Gebser, M., Kaufmann, B., and Schaub, T. (2012b).
Multi-threaded ASP solving with clasp.
Theory Pract. Log. Program., 12(4-5):525–545.

Gerlach, L. (2020).
Don't repeat yourself: Termination of the skolem chase on disjunctive existential rules.
(student thesis available at `https://iccl.inf.tu-dresden.de/web/Thema3509`).

Gogacz, T. and Marcinkowski, J. (2014).
All-instances termination of chase is undecidable.
In [Esparza et al., 2014], pages 293–304.

Grahne, G. and Onet, A. (2018).
Anatomy of the chase.
Fundam. Inform., 157(3):221–270.

References X

Groth, P., Simperl, E., Gray, A. J. G., Sabou, M., Krötzsch, M., Lécué, F., Flöck, F., and Gil, Y., editors (2016).
The Semantic Web - ISWC 2016 - 15th International Semantic Web Conference, Kobe, Japan, October 17-21, 2016, Proceedings, Part II, volume 9982 of *Lecture Notes in Computer Science*.

Hitzler, P. and Lukasiewicz, T., editors (2010).
Web Reasoning and Rule Systems - Fourth International Conference, RR 2010, volume 6333 of *Lecture Notes in Computer Science*. Springer.

Kraus, S., editor (2019).
Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019. ijcai.org.

References XI

- Lenzerini, M. and Lembo, D., editors (2008).
  Proceedings of the Twenty-Seventh ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2008. ACM.
- Maier, D., Mendelzon, A. O., and Sagiv, Y. (1979).
  Testing implications of data dependencies.
  ACM Trans. Database Syst., 4(4):455–469.
- Marnette, B. (2009).
  Generalized schema-mappings: from termination to tractability.
  In [Paredaens and Su, 2009], pages 13–22.
- Paredaens, J. and Su, J., editors (2009).
  Proceedings of the Twenty-Eigth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2009. ACM.

References XII

- Parsia, B., Matentzoglu, N., Gonçalves, R. S., Glimm, B., and Steigmiller, A. (2016).
  The OWL reasoner evaluation (ORE) 2015 resources.
  In [Groth et al., 2016], pages 159–167.
- Parsia, B., Matentzoglu, N., Gonçalves, R. S., Glimm, B., and Steigmiller, A. (2017).
  The OWL reasoner evaluation (ORE) 2015 competition report.
  J. Autom. Reason., 59(4):455–482.
- Rudolph, S., Krötzsch, M., and Hitzler, P. (2012).
  Type-elimination-based reasoning for the description logic shiqbs using decision diagrams and disjunctive datalog.
  Logical Methods in Computer Science, 8(1).
- Sallinger, E., den Bussche, J. V., and Geerts, F., editors (2017).
  Proceedings of the 36th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS 2017. ACM.

References XIII

Sierra, C., editor (2017).
Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017. ijcai.org.