MANCHESTER
1824
The University of Manchester

Saturation-Based Querying Procedures for the Clique-Guarded Negation Fragment

Sen Zheng, Renate A. Schmidt

June 25, 2023

The University of Manchester
The Decision Problem in First-Order Logic (DPFO2023)

Problem Setting

- D: Ground atoms, no function symbols

Problem Setting

- D: Ground atoms, no function symbols
- Σ: Formulas in Clique-Guarded Negation Fragment (CGNF)

Problem Setting

- D: Ground atoms, no function symbols
- Σ: Formulas in Clique-Guarded Negation Fragment (CGNF)
  - $\phi ::= R(\bar{x}) \mid x \approx y \mid \exists \phi \mid \phi \wedge \phi \mid \phi \vee \phi \mid \mathcal{G}(\bar{x}, \bar{y}) \wedge \neg \phi(\bar{y})$

Problem Setting

- D: Ground atoms, no function symbols
- Σ: Formulas in Clique-Guarded Negation Fragment (CGNF)
  - $\phi ::= R(\overline{x}) \mid x \approx y \mid \exists \phi \mid \phi \wedge \phi \mid \phi \vee \phi \mid \mathcal{G}(\overline{x}, \overline{y}) \wedge \neg \phi(\overline{y})$
  - $G_1(x, y) \wedge G_2(y, z) \wedge G_3(y, z) \wedge \neg A(x, y, z)$

Problem Setting

- D: Ground atoms, no function symbols
- Σ: Formulas in Clique-Guarded Negation Fragment (CGNF)
  - $\phi ::= R(\overline{x}) \mid x \approx y \mid \exists \phi \mid \phi \wedge \phi \mid \phi \vee \phi \mid \mathcal{G}(\overline{x}, \overline{y}) \wedge \neg \phi(\overline{y})$
  - $G_1(x, y) \wedge G_2(y, z) \wedge G_3(y, z) \wedge \neg A(x, y, z)$
  - $\neg \exists x_1 ... x_6 .\ A_1(x_1, x_2) \wedge A_2(x_2, x_3) \wedge A_3(x_3, x_4, x_5) \wedge A_4(x_5, x_6) \wedge A_5(x_3, x_4)$

Problem Setting

- D: Ground atoms, no function symbols
- Σ: Formulas in Clique-Guarded Negation Fragment (CGNF)
  - $\phi ::= R(\bar{x}) \mid x \approx y \mid \exists \phi \mid \phi \wedge \phi \mid \phi \vee \phi \mid \mathcal{G}(\bar{x}, \bar{y}) \wedge \neg \phi(\bar{y})$
  - $G_1(x, y) \wedge G_2(y, z) \wedge G_3(y, z) \wedge \neg A(x, y, z)$
  - $\neg \exists x_1 ... x_6 .\ A_1(x_1, x_2) \wedge A_2(x_2, x_3) \wedge A_3(x_3, x_4, x_5) \wedge A_4(x_5, x_6) \wedge A_5(x_3, x_4)$
- q: (unions of) Boolean Conjunctive Queries (BCQs)

Problem Setting

- D: Ground atoms, no function symbols
- Σ: Formulas in Clique-Guarded Negation Fragment (CGNF)
    - $\phi ::= R(\overline{x}) \mid x \approx y \mid \exists \phi \mid \phi \wedge \phi \mid \phi \vee \phi \mid \mathcal{G}(\overline{x}, \overline{y}) \wedge \neg \phi(\overline{y})$
    - $G_1(x, y) \wedge G_2(y, z) \wedge G_3(y, z) \wedge \neg A(x, y, z)$
    - $\neg \exists x_1 ... x_6 .\ A_1(x_1, x_2) \wedge A_2(x_2, x_3) \wedge A_3(x_3, x_4, x_5) \wedge A_4(x_5, x_6) \wedge A_5(x_3, x_4)$
- q: (unions of) Boolean Conjunctive Queries (BCQs)
    - Yes or No

Problem Setting

- D: Ground atoms, no function symbols
- Σ: Formulas in Clique-Guarded Negation Fragment (CGNF)
  - $\phi ::= R(\overline{x}) \mid x \approx y \mid \exists \phi \mid \phi \wedge \phi \mid \phi \vee \phi \mid \mathcal{G}(\overline{x}, \overline{y}) \wedge \neg \phi(\overline{y})$
  - $G_1(x, y) \wedge G_2(y, z) \wedge G_3(y, z) \wedge \neg A(x, y, z)$
  - $\neg \exists x_1 ... x_6.\ A_1(x_1, x_2) \wedge A_2(x_2, x_3) \wedge A_3(x_3, x_4, x_5) \wedge A_4(x_5, x_6) \wedge A_5(x_3, x_4)$
- q: (unions of) Boolean Conjunctive Queries (BCQs)
  - Yes or No
  - $\exists \overline{x} F(\overline{x})$: A conjunction of atoms, no function symbols

Problem Setting

- D: Ground atoms, no function symbols
- Σ: Formulas in Clique-Guarded Negation Fragment (CGNF)
    - $\phi ::= R(\overline{x}) \mid x \approx y \mid \exists \phi \mid \phi \wedge \phi \mid \phi \vee \phi \mid \mathcal{G}(\overline{x}, \overline{y}) \wedge \neg \phi(\overline{y})$
    - $G_1(x, y) \wedge G_2(y, z) \wedge G_3(y, z) \wedge \neg A(x, y, z)$
    - $\neg \exists x_1 ... x_6 .\ A_1(x_1, x_2) \wedge A_2(x_2, x_3) \wedge A_3(x_3, x_4, x_5) \wedge A_4(x_5, x_6) \wedge A_5(x_3, x_4)$
- q: (unions of) Boolean Conjunctive Queries (BCQs)
    - Yes or No
    - $\exists \overline{x} F(\overline{x})$: A conjunction of atoms, no function symbols
    - $\exists x_1 ... x_6 .\ A_1(x_1, x_2) \wedge A_2(x_2, x_3) \wedge A_3(x_3, x_4, x_5) \wedge A_4(x_5, x_6) \wedge A_5(x_3, x_4)$

Problem Setting

- D: Ground atoms, no function symbols
- Σ: Formulas in CGNF
- q: BCQs

Questions

1. Can a saturation-based method check $D \cup \Sigma \models q$?
2. Can the saturation of $\{\neg q\} \cup \Sigma$ be back-translated to a FO formula with no Skolem symbols?

Problem Setting

- D: Ground atoms, no function symbols
- Σ: Formulas in CGNF
- q: BCQs

Questions

1. Can a saturation-based method check $D \cup \Sigma \models q$?
2. Can the saturation of $\{\neg q\} \cup \Sigma$ be back-translated to a FO formula with no Skolem symbols?

Saturation is one of the major techniques for automated theorem proving … computes the closure of a given set of formulas under resolution-based inferences

Motivation

Questions

1. Can a saturation-based method check $D \cup \Sigma \models q$?
2. Can the saturation of $\{\neg q\} \cup \Sigma$ be back-translated to a FO formula with no Skolem symbols?

Motivation

Questions

1. Can a saturation-based method check $D \cup \Sigma \models q$?
2. Can the saturation of $\{\neg q\} \cup \Sigma$ be back-translated to a FO formula with no Skolem symbols?

- No automated deduction method for querying CGNF

Motivation

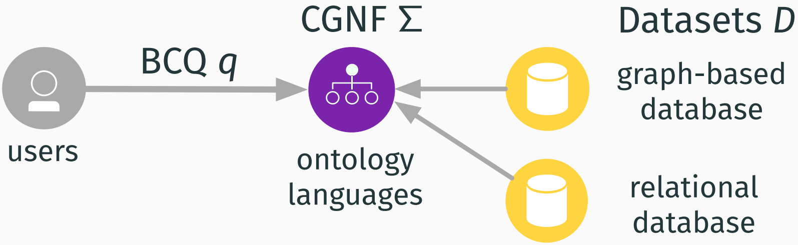
Questions

1. Can a saturation-based method check $D \cup \Sigma \models q$?
2. Can the saturation of $\{\neg q\} \cup \Sigma$ be back-translated to a FO formula with no Skolem symbols?

- No automated deduction method for querying CGNF
- Suitable for ontology-based data access

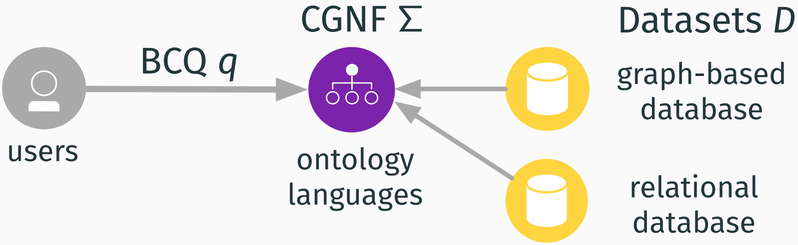
Motivation

- No automated deduction method for querying CGNF
- Suitable for ontology-based data access



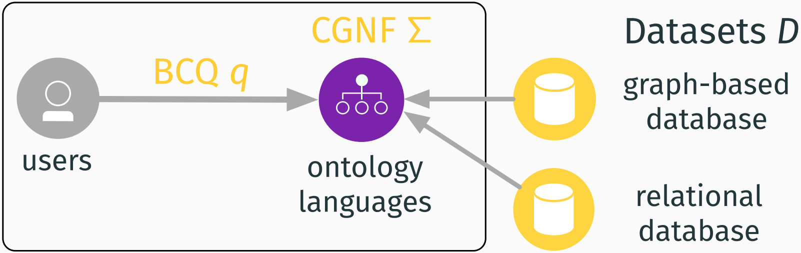
Motivation

- No automated deduction method for querying CGNF
- Suitable for ontology-based data access



Deciding $D \cup \Sigma \models q$ to deciding unsat. of $D \cup \Sigma \cup \{\neg q\}$

Motivation



Deciding $D \cup \Sigma \models q$ to deciding unsat. of $D \cup \Sigma \cup \{\neg q\}$

1. Saturating $\Sigma \cup \{\neg q\}$ first; reusable for different D_i
2. Back-translating the saturation to a FO formula for other reasoning methods

The Guarded Fragments

guarded fragment (GF)	guarded quantification fragments
loosely guarded fragment (LGF)	
clique-guarded fragment (CGF)	

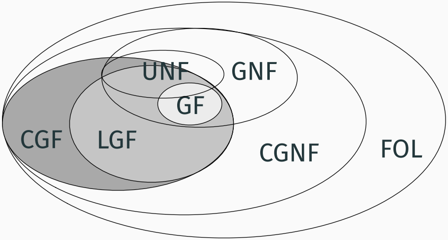
The Guarded Fragments

guarded fragment (GF) loosely guarded fragment (LGF) clique-guarded fragment (CGF)	guarded quantification fragments
unary negation fragment (UNF) guarded negation fragment (GNF) clique-guarded negation fragment (CGNF)	guarded negation fragments

The Guarded Fragments

guarded fragment (GF)
loosely guarded fragment (LGF)
clique-guarded fragment (CGF)
} **guarded quantification fragments**

unary negation fragment (UNF)
guarded negation fragment (GNF)
clique-guarded negation fragment (CGNF)
} guarded negation fragments



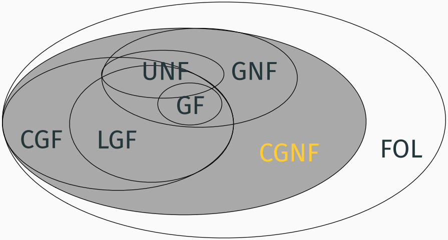
The Guarded Fragments

guarded fragment (GF)
loosely guarded fragment (LGF)
clique-guarded fragment (CGF)

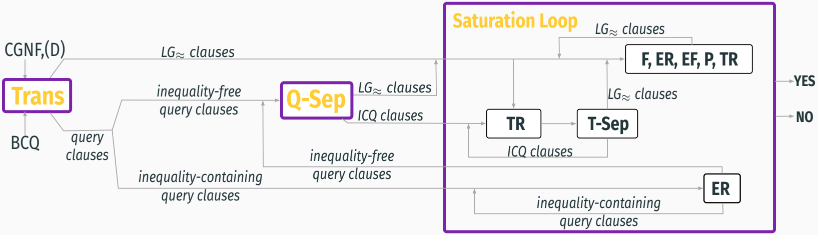
} guarded quantification fragments

unary negation fragment (UNF)
guarded negation fragment (GNF)
clique-guarded negation fragment (CGNF)

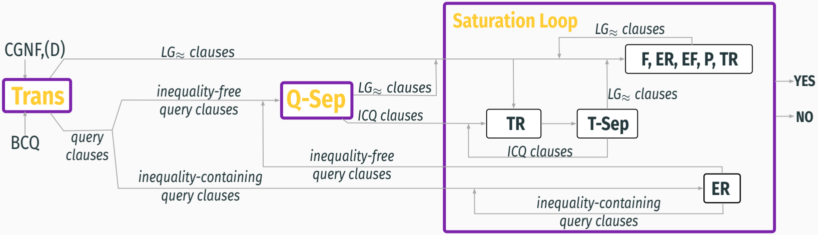
} guarded negation fragments



Saturation-based method for deciding $D \cup \Sigma \models q$

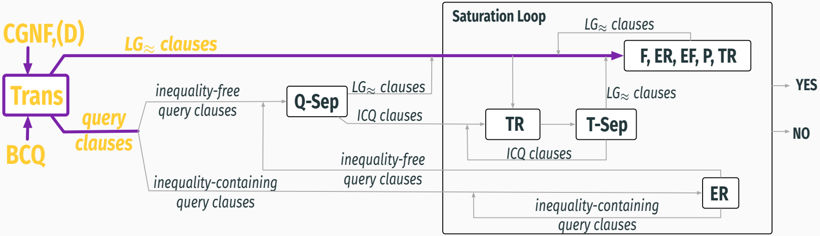


Saturation-based method for deciding $D \cup \Sigma \models q$

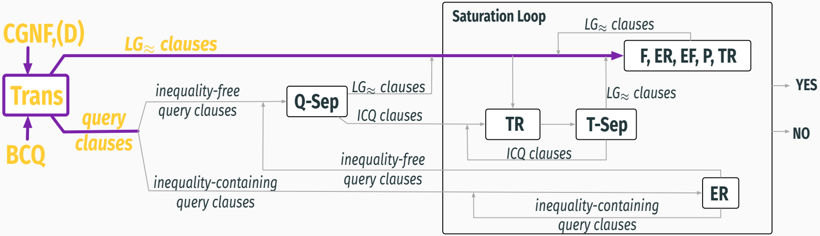


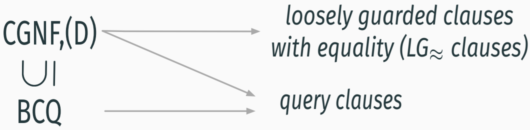
- **Trans**: Customised clausification
- **Q-Sep**: Separating (Simplifying) queries
- **Saturation Loop**: Inferences

Clausification Trans

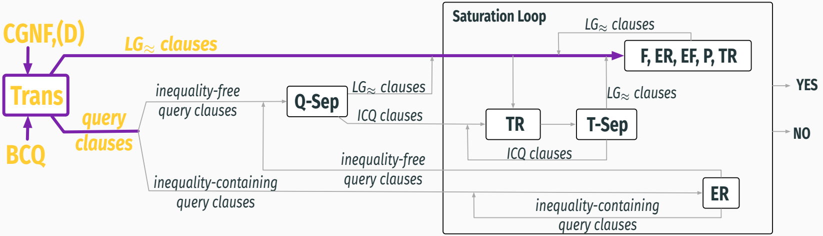


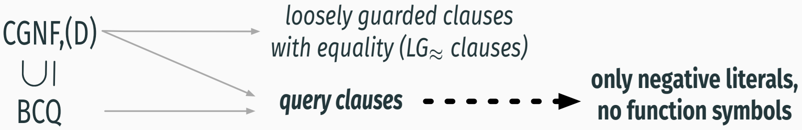
Clausification Trans



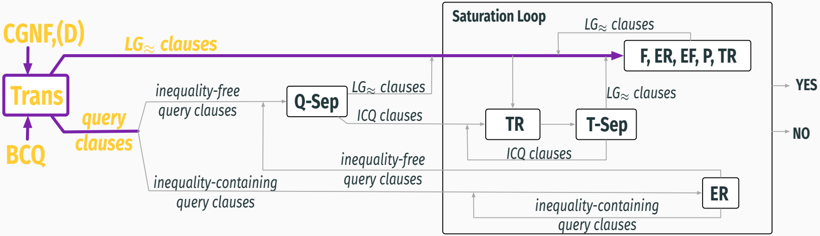


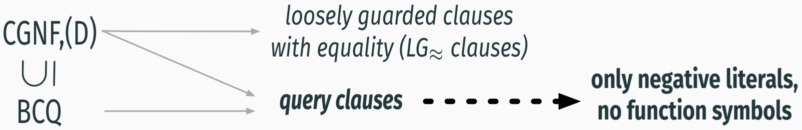
Clausification Trans





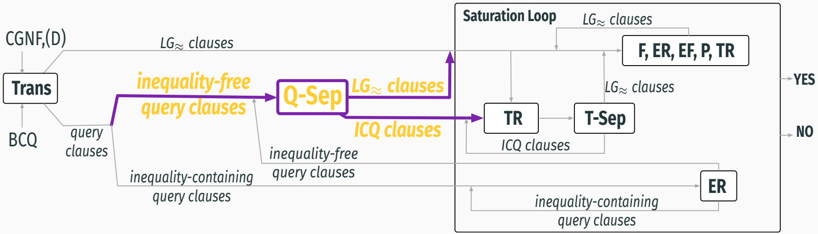
Clausification Trans



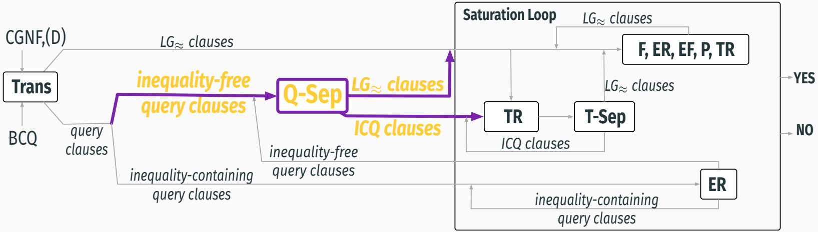


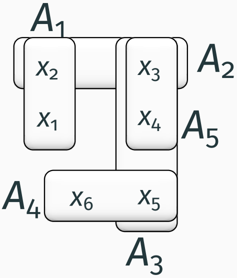
Trans reduces BCQ answering for CGNF to deciding $LG_{\approx}$ and query clauses

Separating Query Clauses Q-Sep



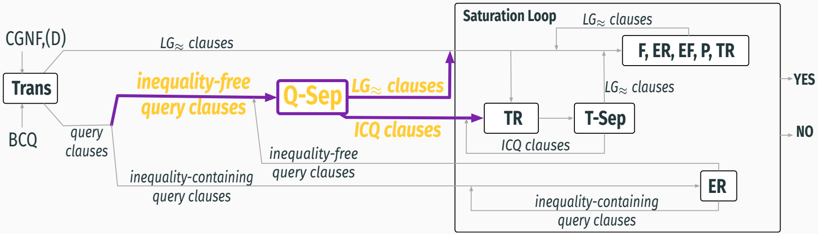
Separating Query Clauses Q-Sep

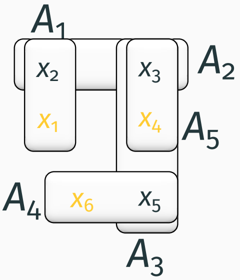




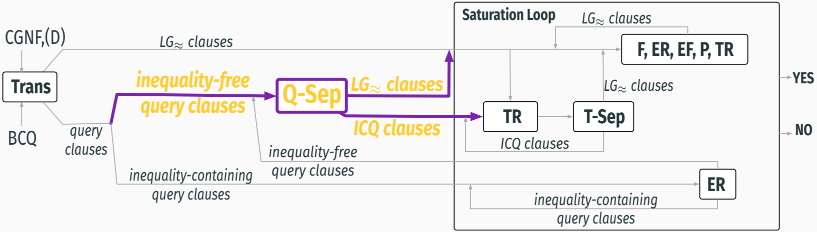
$$\neg A_1(x_1, x_2) \vee \neg A_2(x_2, x_3) \vee$$
$$\neg A_3(x_3, x_4, x_5) \vee \neg A_4(x_5, x_6) \vee$$
$$\neg A_5(x_3, x_4)$$

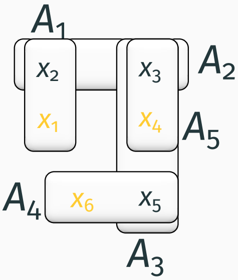
Separating Query Clauses Q-Sep





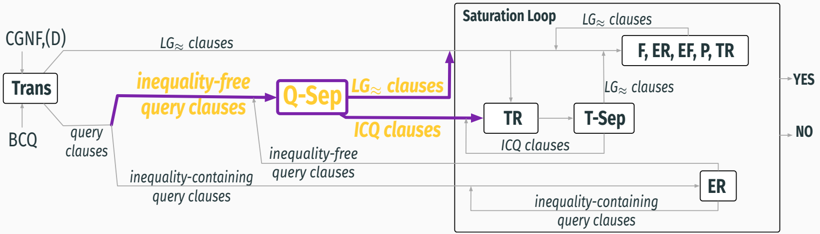
Separating Query Clauses Q-Sep

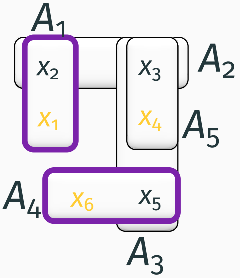




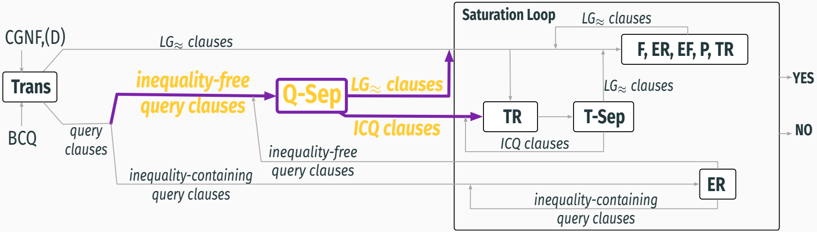
- Chained variables: x_2, x_3, x_5
- Isolated variables: x_1, x_4, x_6

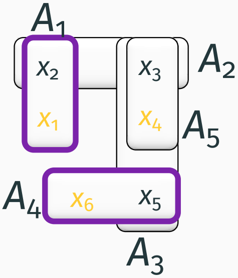
Separating Query Clauses Q-Sep





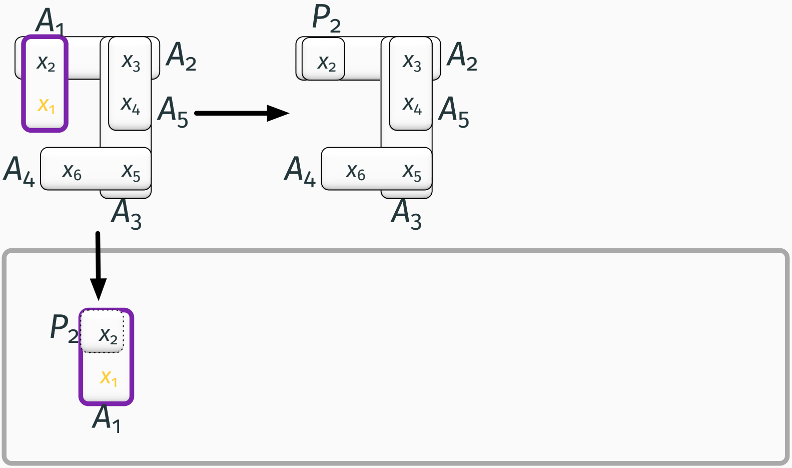
Separating Query Clauses Q-Sep



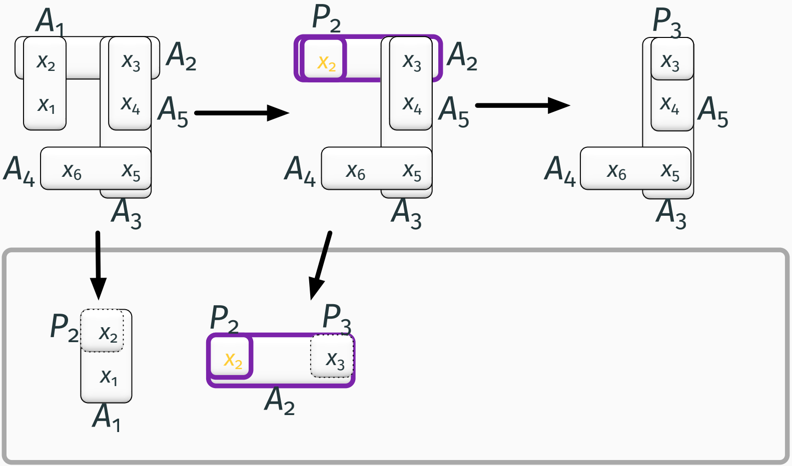


Cutting off branches!

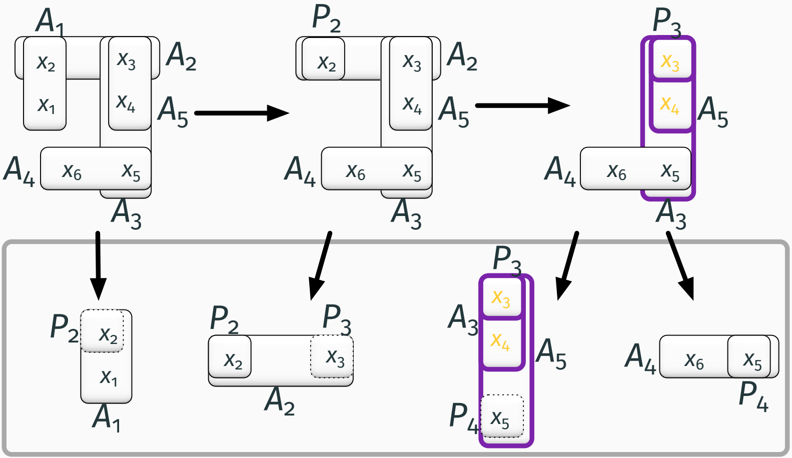
Separating Query Clauses Q-Sep



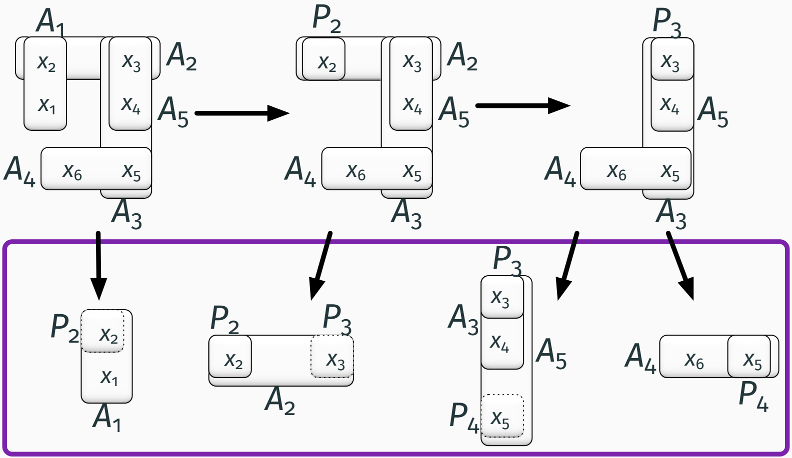
Separating Query Clauses Q-Sep



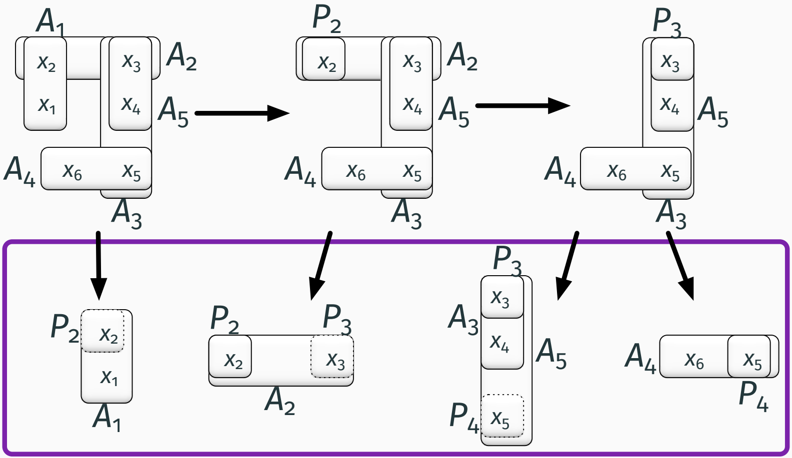
Query Simplification Q-Sep



Separating Query Clauses Q-Sep



Separating Query Clauses Q-Sep



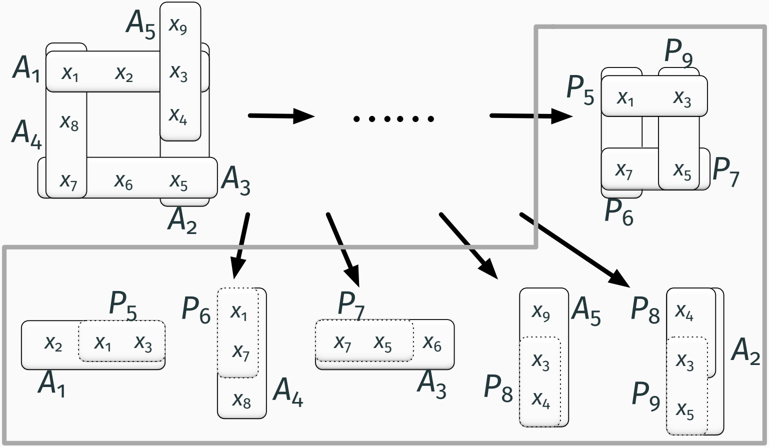
The separated clauses are $\text{LG}_{\approx}$ clauses

Separating Query Clauses Q-Sep

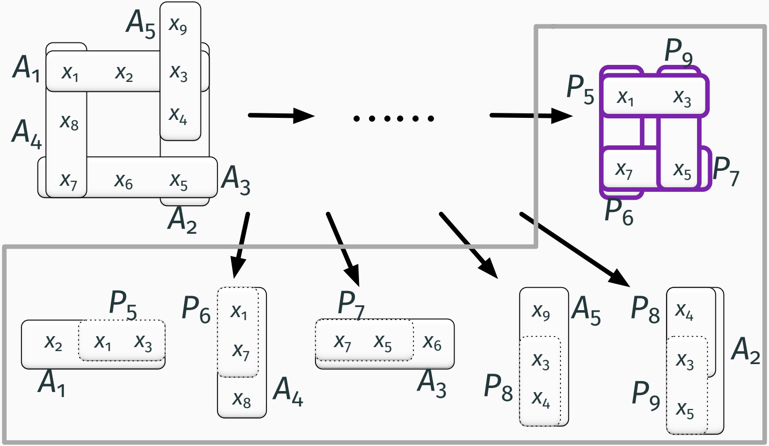
$$\neg A_1(x_1, x_2, x_3) \vee \neg A_2(x_3, x_4, x_5) \vee \neg A_3(x_5, x_6, x_7) \vee \neg A_4(x_1, x_7, x_8) \vee \neg A_5(x_3, x_4, x_9)$$

Separating Query Clauses Q-Sep

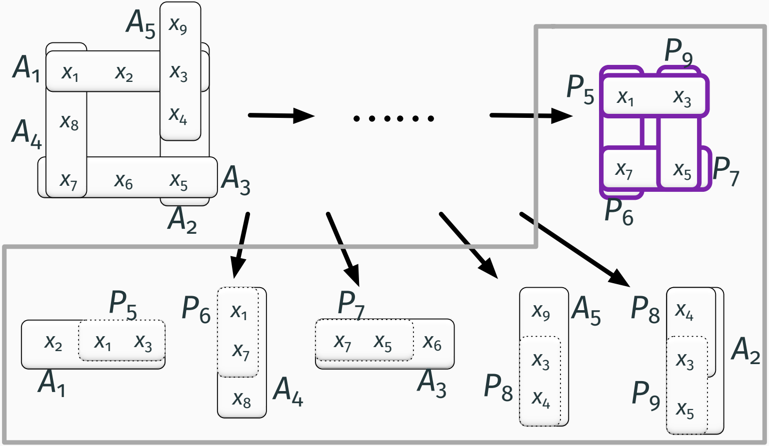
$\neg A_1(x_1, x_2, x_3) \vee \neg A_2(x_3, x_4, x_5) \vee \neg A_3(x_5, x_6, x_7) \vee$
$\neg A_4(x_1, x_7, x_8) \vee \neg A_5(x_3, x_4, x_9)$



Separating Query Clauses Q-Sep

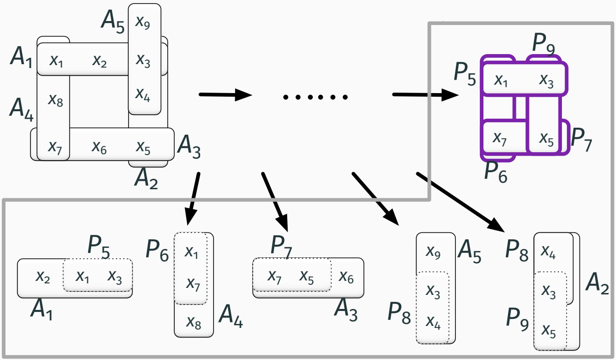


Separating Query Clauses Q-Sep



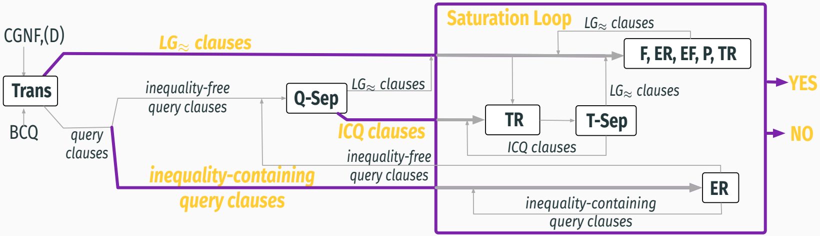
Indecomposable Chained-only Query (ICQ) clauses

Separating Query Clauses Q-Sep

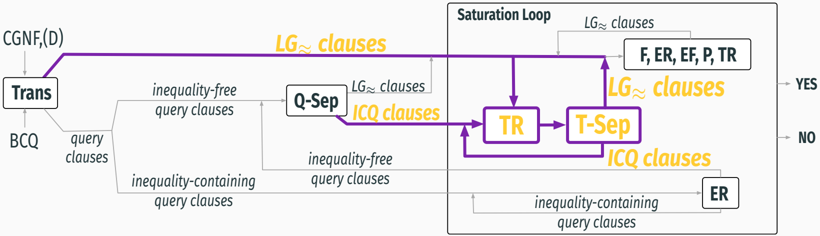


Q-Sep replaces a query clause by $LG_{\approx}$ and ICQ clauses

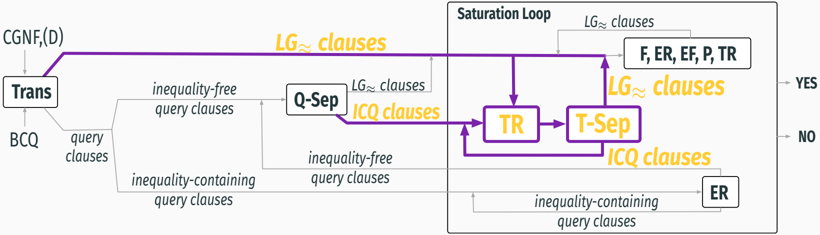
Saturation Loop

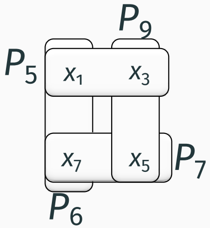


Top-Variable Resolution TR

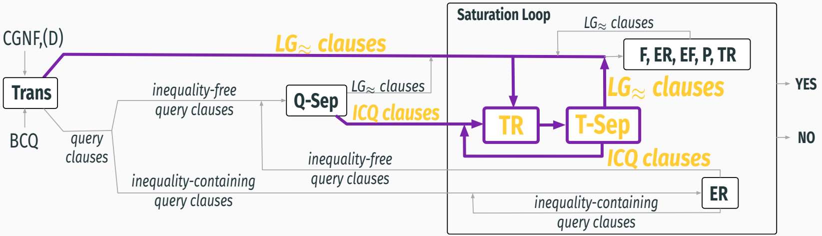


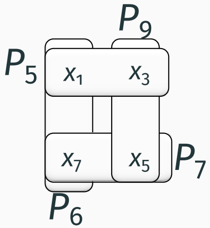
Top-Variable Resolution TR





Top-Variable Resolution TR





$$\neg P_5(x_1, x_3) \vee \neg P_9(x_3, x_5) \vee$$
$$\neg P_7(x_5, x_7) \vee \neg P_6(x_1, x_7)$$

TR Example

$$Q = \neg P_5(x_1, x_3) \vee \neg P_9(x_3, x_5) \vee \neg P_7(x_5, x_7) \vee \neg P_6(x_1, x_7),$$
$$C_1 = P_5(x, g(x)) \vee \neg G_1(x),$$
$$C_2 = P_9(g(y), y) \vee A(h(y)) \vee \neg G_2(y),$$
$$C_3 = P_7(f(x), x) \vee \neg G_3(x),$$
$$C_4 = P_6(f(x), x) \vee \neg G_4(x).$$

TR Example

$$Q = \neg P_5(x_1, x_3) \vee \neg P_9(x_3, x_5) \vee \neg P_7(x_5, x_7) \vee \neg P_6(x_1, x_7),$$
$$C_1 = P_5(x, g(x)) \vee \neg G_1(x),$$
$$C_2 = P_9(g(y), y) \vee A(h(y)) \vee \neg G_2(y),$$
$$C_3 = P_7(f(x), x) \vee \neg G_3(x),$$
$$C_4 = P_6(f(x), x) \vee \neg G_4(x).$$

Lexicographical Path Ordering:
$f \succ g \succ h \succ P_5 \succ P_6 \succ P_7 \succ P_9 \succ A \succ G_1 \succ G_2 \succ G_3 \succ G_4$

TR Example

$$Q = \neg P_5(x_1, x_3) \vee \neg P_9(x_3, x_5) \vee \neg P_7(x_5, x_7) \vee \neg P_6(x_1, x_7),$$
$$C_1 = P_5(x, g(x)) \vee \neg G_1(x),$$
$$C_2 = P_9(g(y), y) \vee A(h(y)) \vee \neg G_2(y),$$
$$C_3 = P_7(f(x), x) \vee \neg G_3(x),$$
$$C_4 = P_6(f(x), x) \vee \neg G_4(x).$$

Lexicographical Path Ordering:
$f \succ g \succ h \succ P_5 \succ P_6 \succ P_7 \succ P_9 \succ A \succ G_1 \succ G_2 \succ G_3 \succ G_4$

TR Example

$$Q = \neg P_5(x_1, x_3) \vee \neg P_9(x_3, x_5) \vee \neg P_7(x_5, x_7) \vee \neg P_6(x_1, x_7),$$
$$C_1 = P_5(x, g(x)) \vee \neg G_1(x),$$
$$C_2 = P_9(g(y), y) \vee A(h(y)) \vee \neg G_2(y),$$
$$C_3 = P_7(f(x), x) \vee \neg G_3(x),$$
$$C_4 = P_6(f(x), x) \vee \neg G_4(x).$$

- $x_5 \to f(x)$

TR Example

$$Q = \neg P_5(x_1, x_3) \vee \neg P_9(x_3, x_5) \vee \neg P_7(x_5, x_7) \vee \neg P_6(x_1, x_7),$$

$$C_1 = P_5(x, g(x)) \vee \neg G_1(x),$$

$$C_2 = P_9(g(y), y) \vee A(h(y)) \vee \neg G_2(y),$$

$$C_3 = P_7(f(x), x) \vee \neg G_3(x),$$

$$C_4 = P_6(f(x), x) \vee \neg G_4(x).$$

- $x_5 \to f(x)$

TR Example

$$Q = \neg P_5(x_1, x_3) \vee \neg P_9(x_3, x_5) \vee \neg P_7(x_5, x_7) \vee \neg P_6(x_1, x_7),$$
$$C_1 = P_5(x, g(x)) \vee \neg G_1(x),$$
$$C_2 = P_9(g(y), y) \vee A(h(y)) \vee \neg G_2(y),$$
$$C_3 = P_7(f(x), x) \vee \neg G_3(x),$$
$$C_4 = P_6(f(x), x) \vee \neg G_4(x).$$

- $x_5 \rightarrow f(x)$
- $y \rightarrow f(x)$, hence $A(h(f(x)))$ occurs in the conclusion

TR Example

$$Q = \neg P_5(x_1, x_3) \vee \neg P_9(x_3, x_5) \vee \neg P_7(x_5, x_7) \vee \neg P_6(x_1, x_7),$$
$$C_1 = P_5(x, g(x)) \vee \neg G_1(x),$$
$$C_2 = P_9(g(y), y) \vee A(h(y)) \vee \neg G_2(y),$$
$$C_3 = P_7(f(x), x) \vee \neg G_3(x),$$
$$C_4 = P_6(f(x), x) \vee \neg G_4(x).$$

- $x_5 \to f(x)$
- $y \to f(x)$, hence $A(h(f(x)))$ occurs in the conclusion

TR Example

$$Q = \neg P_5(x_1, x_3) \vee \neg P_9(x_3, x_5) \vee \neg P_7(x_5, x_7) \vee \neg P_6(x_1, x_7),$$
$$C_1 = P_5(x, g(x)) \vee \neg G_1(x),$$
$$C_2 = P_9(g(y), y) \vee A(h(y)) \vee \neg G_2(y),$$
$$C_3 = P_7(f(x), x) \vee \neg G_3(x),$$
$$C_4 = P_6(f(x), x) \vee \neg G_4(x).$$

- $x_5 \to f(x)$
- $y \to f(x)$, hence $A(h(f(x)))$ occurs in the conclusion
- $\{x_1 \to f(x), x_3 \to g(f(x)), x_5 \to f(x), x_7 \to x\}$

TR Example

$$Q = \neg P_5(x_1, x_3) \vee \neg P_9(x_3, x_5) \vee \neg P_7(x_5, x_7) \vee \neg P_6(x_1, x_7),$$
$$C_1 = P_5(x, g(x)) \vee \neg G_1(x),$$
$$C_2 = P_9(g(y), y) \vee A(h(y)) \vee \neg G_2(y),$$
$$C_3 = P_7(f(x), x) \vee \neg G_3(x),$$
$$C_4 = P_6(f(x), x) \vee \neg G_4(x).$$

- Resolving Q, C_1, C_2 derives
  $\neg P_7(x, x_7) \vee \neg P_6(x, x_7) \vee \neg G_1(x) \vee A(h(x)) \vee \neg G_2(x)$

TR Example

$$Q = \neg P_5(x_1, x_3) \vee \neg P_9(x_3, x_5) \vee \neg P_7(x_5, x_7) \vee \neg P_6(x_1, x_7),$$

$$C_1 = P_5(x, g(x)) \vee \neg G_1(x),$$

$$C_2 = P_9(g(y), y) \vee A(h(y)) \vee \neg G_2(y),$$

$$C_3 = P_7(f(x), x) \vee \neg G_3(x),$$

$$C_4 = P_6(f(x), x) \vee \neg G_4(x).$$

- Resolving Q, C_1, C_2 derives $\neg P_7(x, x_7) \vee \neg P_6(x, x_7) \vee \neg G_1(x) \vee A(h(x)) \vee \neg G_2(x)$
- No nested term occurs!

TR Example

$$Q = \neg P_5(x_1, x_3) \vee \neg P_9(x_3, x_5) \vee \neg P_7(x_5, x_7) \vee \neg P_6(x_1, x_7),$$
$$C_1 = P_5(x, g(x)) \vee \neg G_1(x),$$
$$C_2 = P_9(g(y), y) \vee A(h(y)) \vee \neg G_2(y),$$
$$C_3 = P_7(f(x), x) \vee \neg G_3(x),$$
$$C_4 = P_6(f(x), x) \vee \neg G_4(x).$$

- Resolving Q, C_1, C_2 derives $\neg P_7(x, x_7) \vee \neg P_6(x, x_7) \vee \neg G_1(x) \vee A(h(x)) \vee \neg G_2(x)$
- No nested term occurs!
- Make the resolution on $Q, C_1, \ldots, C_4$ redundant

Top-Variable Resolution TR

In an inference:

positive premises → $C_1, \dots, C_n$

negative premise → C

$$\frac{C_1, \dots, C_n \qquad C}{R}$$

Top-Variable Resolution TR

Two inferences I and I'

$$I : \frac{C_1, \ldots, C_n \qquad C}{R} \qquad\qquad I' : \frac{C_1, \ldots, C_m \qquad C}{R'} \ (m < n)$$

Top-Variable Resolution TR

Two inferences I and I'

$$I: \frac{C_1, \ldots, C_n \qquad C}{R} \qquad\qquad I': \frac{C_1, \ldots, C_m \qquad C}{R'} \; (m < n)$$

- I' makes I redundant since

Top-Variable Resolution TR

Two inferences I and I'

$$I : \frac{C_1, \ldots, C_n \qquad C}{R} \qquad\qquad I' : \frac{C_1, \ldots, C_m \qquad C}{R'} \quad (m < n)$$

- I' makes I redundant since
    1. $C \succ R'$, by orderings

Top-Variable Resolution TR

Two inferences I and I'

$$I : \frac{C_1, \dots, C_n \qquad C}{R} \qquad\qquad I' : \frac{C_1, \dots, C_m \qquad C}{R'} \; (m < n)$$

- I' makes I redundant since
  1. $C \succ R'$, by orderings
  2. $C_1, \dots, C_n, R' \models R$, by resolution

Top-Variable Resolution TR

Two inferences I and I'

$$I : \frac{C_1, \ldots, C_n \qquad C}{R} \qquad\qquad I' : \frac{C_1, \ldots, C_m \qquad C}{R'} \; (m < n)$$

- I' makes I redundant since
  1. $C \succ R'$, by orderings
  2. $C_1, \ldots, C_n, R' \models R$, by resolution
  3. $C_1, \ldots, C_n, C \models R$ is redundant, by 1. and 2.

Top-Variable Resolution TR

Two inferences I and I'

$$I: \ \frac{C_1, \ldots, C_n \qquad C}{R} \qquad\qquad I': \ \frac{C_1, \ldots, C_m \qquad C}{R'} \ (m < n)$$

- I' makes I redundant since
  1. $C \succ R'$, by orderings
  2. $C_1, \ldots, C_n, R' \models R$, by resolution
  3. $C_1, \ldots, C_n, C \models R$ is redundant, by 1. and 2.
- Only resolving the positive premises where unification peaks

Top-Variable Resolution TR

Two inferences I and I'

$$I: \frac{C_1, \dots, C_n \qquad C}{R} \qquad\qquad I': \frac{C_1, \dots, C_m \qquad C}{R'} \; (m < n)$$

- I' makes I redundant since
  - $C \succ R'$, by orderings
  - $C_1, \dots, C_n, R' \models R$, by resolution
  - Premises in $C_1, \dots, C_n, R' \models R$ are smaller than these in $C_1, \dots, C_n, C \models R$
- Only resolving the positive premises where unification peaks

Applying **TR** (and **T-Sep**) to ICQ clauses and $\mathrm{LG}_{\approx}$ clauses derives $\mathrm{LG}_{\approx}$ and ICQ clauses

Back-translatable Saturations

Back-translatable Saturations

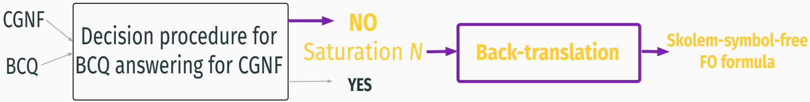
what: Eliminate Skolem symbols in N

- in general undecidable

why: Prepare N for other reasoning methods in deciding $D_i \cup N$

how: Align arguments (variables) that are under the same function symbol across clauses

Back-translatable Saturations



what: Eliminate Skolem symbols in N

- in general undecidable

why: Prepare N for other reasoning methods in deciding $D_i \cup N$

how: Align arguments (variables) that are under the same function symbol across clauses

Saturation-based BCQ rewriting!

Chase Example

- Well-established query answering algorithm
- Many variations

Chase Example

- Well-established query answering algorithm
- Many variations

Simplified example from [Krötzsch et al., ICDT'19]: use standard chase to decide guarded Datalog$^{\pm}$ and data

$$D = \{\mathit{Bicycle}(c)\}$$

$$\Sigma = \{\mathit{Bicycle}(x) \to \exists v.\mathit{hasPart}(x, v) \wedge \mathit{Wheel}(v), \quad \mathit{Wheel}(x) \to \exists w.\mathit{properPartOf}(x, w) \wedge \mathit{Bicycle}(w)\}$$

Chase Example

- Well-established query answering algorithm
- Many variations

Simplified example from [Krötzsch et al., ICDT'19]: use standard chase to decide guarded Datalog$^{\pm}$ and data

$$D = \{Bicycle(c)\}$$

$$\Sigma = \{Bicycle(x) \to \exists v.hasPart(x, v) \land Wheel(v),$$
$$Wheel(x) \to \exists w.properPartOf(x, w) \land Bicycle(w)\}$$

$$D_1 = D \cup \{hasPart(c, n_1), Wheel(n_1)\}$$
$$D_2 = D_1 \cup \{properPartOf(n_1, n_2), Bicycle(n_2)\}$$

...

Chase Example

- Well-established query answering algorithm
- Many variations

Simplified example from [Krötzsch et al., ICDT'19]: use standard chase to decide guarded Datalog$^{\pm}$ and data

$$D = \{Bicycle(c)\}$$

$$\Sigma = \{Bicycle(x) \rightarrow \exists v.hasPart(x, v) \wedge Wheel(v),$$
$$Wheel(x) \rightarrow \exists w.properPartOf(x, w) \wedge Bicycle(w)\}$$

$D_1 = D \cup \{hasPart(c, n_1), Wheel(n_1)\}$

$D_2 = D_1 \cup \{properPartOf(n_1, n_2), Bicycle(n_2)\}$

...

May not terminate!

Saturation for the Chase Example

$$D = \{Bicycle(c)\}$$

$$\Sigma = \{Bicycle(x) \rightarrow \exists v.hasPart(x, v) \wedge Wheel(v),$$
$$Wheel(x) \rightarrow \exists w.properPartOf(x, w) \wedge Bicycle(w)\}$$

Saturation for the Chase Example

$$D = \{Bicycle(c)\}$$

$$\Sigma = \{Bicycle(x) \rightarrow \exists v.hasPart(x, v) \wedge Wheel(v),$$
$$Wheel(x) \rightarrow \exists w.properPartOf(x, w) \wedge Bicycle(w)\}$$

$Bicycle(c)$,

$\neg Bicycle(x) \vee hasPart(x, f(x))$,

$\neg Bicycle(x) \vee Wheel(f(x))$,

$\neg Wheel(x) \vee properPartOf(x, g(x))$,

$\neg Wheel(x) \vee Bicycle(g(x))$

Saturation for the Chase Example

$Bicycle(c)$,

$\neg Bicycle(x) \lor hasPart(x, f(x))$,

$\neg Bicycle(x) \lor Wheel(f(x))$,

$\neg Wheel(x) \lor properPartOf(x, g(x))$,

$\neg Wheel(x) \lor Bicycle(g(x))$

Lexicographical Path Ordering:
$f \succ g \succ c \succ hasPart \succ properPartOf \succ Wheel \succ Bicycle$

Saturation for the Chase Example

$$
\begin{aligned}
&Bicycle(c),\\
&\neg Bicycle(x) \lor hasPart(x, f(x)),\\
&\neg Bicycle(x) \lor Wheel(f(x)),\\
&\neg Wheel(x) \lor properPartOf(x, g(x)),\\
&\neg Wheel(x) \lor Bicycle(g(x))
\end{aligned}
$$

Lexicographical Path Ordering:
$f \succ g \succ c \succ hasPart \succ properPartOf \succ Wheel \succ Bicycle$

Saturation for the Chase Example

$Bicycle(c)$,

$\neg Bicycle(x) \vee hasPart(x, f(x))$,

$\neg Bicycle(x) \vee Wheel(f(x))$,

$\neg Wheel(x) \vee properPartOf(x, g(x))$,

$\neg Wheel(x) \vee Bicycle(g(x))$

Lexicographical Path Ordering:
$f \succ g \succ c \succ hasPart \succ properPartOf \succ Wheel \succ Bicycle$

Saturated; $D \cup \Sigma$ is **satisfiable**

Back-translate the Saturation

$\neg Bicycle(x) \vee hasPart(x, f(x))$,

$\neg Bicycle(x) \vee Wheel(f(x))$,

$\neg Wheel(x) \vee properPartOf(x, g(x))$,

$\neg Wheel(x) \vee Bicycle(g(x))$

$$\forall x \exists v ((\neg Bicycle(x) \vee hasPart(x, v)) \wedge$$
$$(\neg Bicycle(x) \vee Wheel(v))) \wedge$$
$$\forall x \exists w ((\neg Wheel(x) \vee properPartOf(x, w)) \wedge$$
$$(\neg Wheel(x) \vee Bicycle(w)))$$

Back-translate the Saturation

$$\neg Bicycle(x) \lor hasPart(x, f(x)),$$
$$\neg Bicycle(x) \lor Wheel(f(x)),$$
$$\neg Wheel(x) \lor properPartOf(x, g(x)),$$
$$\neg Wheel(x) \lor Bicycle(g(x))$$

$$\forall x(Bicycle(x) \rightarrow \exists v(hasPart(x, v) \land Wheel(v)))\land$$
$$\forall x(Wheel(x) \rightarrow \exists w(properPartOf(x, w) \land Bicycle(w)))$$

Can be complicated if inferences are performed!

Conclusions and Future Work

- First automated decision method for BCQ answering for CGNF
- First saturation-based BCQ rewriting method for implementing ontology-based data access over CGNF and its subfragment
- Back-translation of rewriting allows alternative reasoning methods to be used for ontology-based data access

Conclusions and Future Work

- First automated decision method for BCQ answering for CGNF
- First saturation-based BCQ rewriting method for implementing ontology-based data access over CGNF and its subfragment
- Back-translation of rewriting allows alternative reasoning methods to be used for ontology-based data access

- Implement and evaluate the procedures
- Complexity analysis
- Query other fragments, e.g., triguarded, guarded adjacent fragment
- Retrieve non-Boolean answers

Thank you. Questions?