Chase Termination

Analytical Hierarchy, Disjunctions, Sufficient Conditions

Lukas Gerlach

Knowledge-Based Systems Group, TU Dresden, Germany

11.04.2025

Brief Overview of my Work

- Theory of Existential Rules and the Chase 📋
-
-

Brief Overview of my Work

- Theory of Existential Rules and the Chase 📋
  - Chase Termination ⏱️
    - Decidability 🤔
    - Sufficient Conditions ✅ / ❌

(Carral et al. 2025)
(Gerlach and Carral 2023a)
(Gerlach and Carral 2023b)

Brief Overview of my Work

- Theory of Existential Rules and the Chase 📋
  - Chase Termination ⏱️
    - Decidability 🤔
    - Sufficient Conditions ✅ / ❌
  - Computation of Core Models ⚛️
-
-

(Carral et al. 2025)
(Gerlach and Carral 2023a)
(Gerlach and Carral 2023b)

Brief Overview of my Work

- Theory of Existential Rules and the Chase 📋
  - Chase Termination ⏱️
    - Decidability 🤔
    - Sufficient Conditions ✅ / ❌
  - Computation of Core Models ⚛️
- Porting some Termination Ideas to ASP 🚢
-

(Carral et al. 2025)
(Gerlach and Carral 2023a)
(Gerlach and Carral 2023b)

(Gerlach et al. 2024)

Brief Overview of my Work

- Theory of Existential Rules and the Chase 📋
  - Chase Termination ⏱️
    - Decidability 🤔
    - Sufficient Conditions ✅ / ❌
  - Computation of Core Models ⚛️
- Porting some Termination Ideas to ASP 🚢
- Implementation Work ⌨️
  - Working on Nemo
  - Writing Proofs in Lean 🤓

(Carral et al. 2025)
(Gerlach and Carral 2023a)
(Gerlach and Carral 2023b)

(Gerlach et al. 2024)

(Ivliev et al. 2024)

Brief Overview of my Work

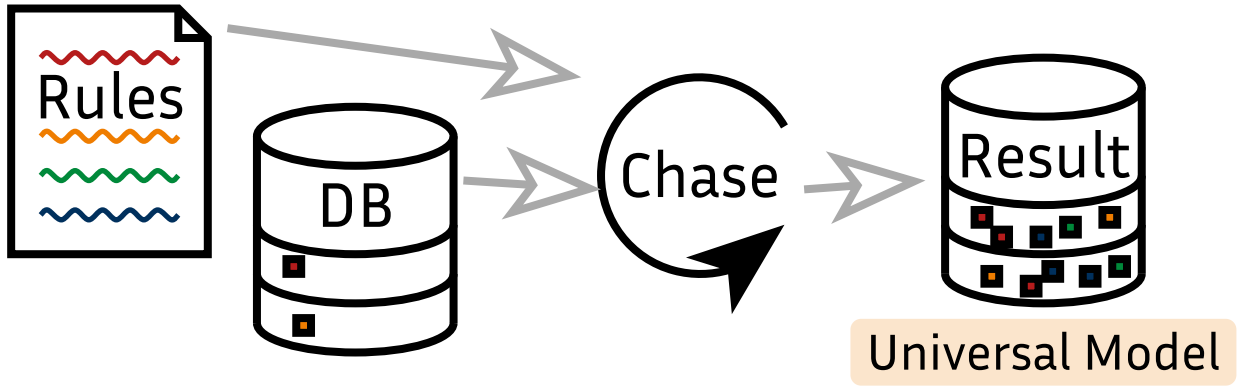
- Theory of Existential Rules and the Chase 📋
  - Chase Termination ⏱️
    - Decidability 🤔
    - Sufficient Conditions ✅ / ❌
  - Computation of Core Models ⚛️
- Porting some Termination Ideas to ASP 🚢
- Implementation Work ⌨️
  - Working on Nemo
  - Writing Proofs in Lean 🤓

(Carral et al. 2025)
(Gerlach and Carral 2023a)
(Gerlach and Carral 2023b)

(Gerlach et al. 2024)

(Ivliev et al. 2024)

Chase Crash Course for DL Folks

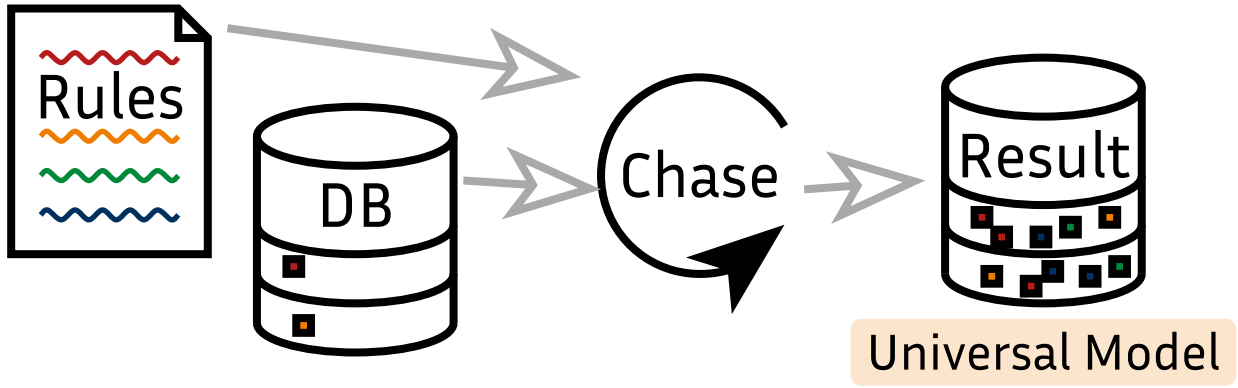


Chase Crash Course for DL Folks

$A \sqsubseteq \exists R.B$

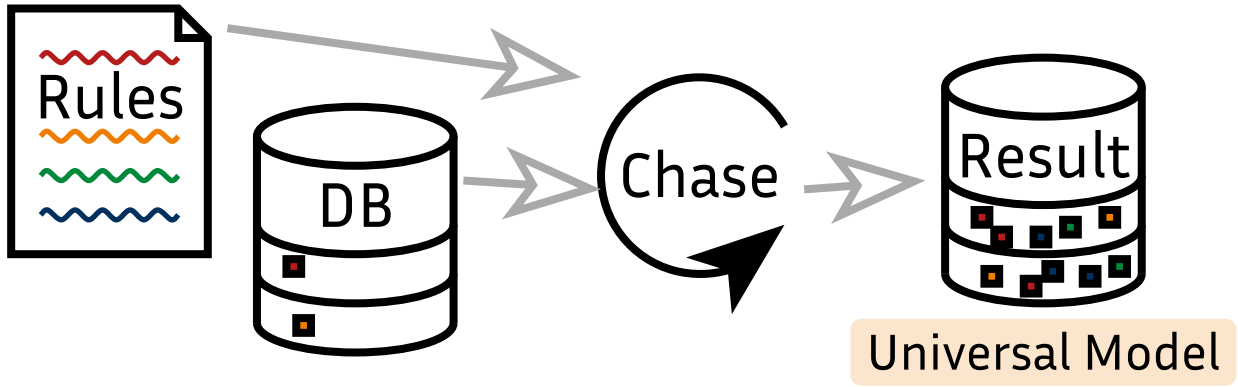
$B \sqsubseteq A$

R is symmetric



Chase Crash Course for DL Folks

$$\mathrm{A} \sqsubseteq \exists\, \mathrm{R}.\mathrm{B}$$
$$\mathrm{B} \sqsubseteq \mathrm{A}$$
R is symmetric



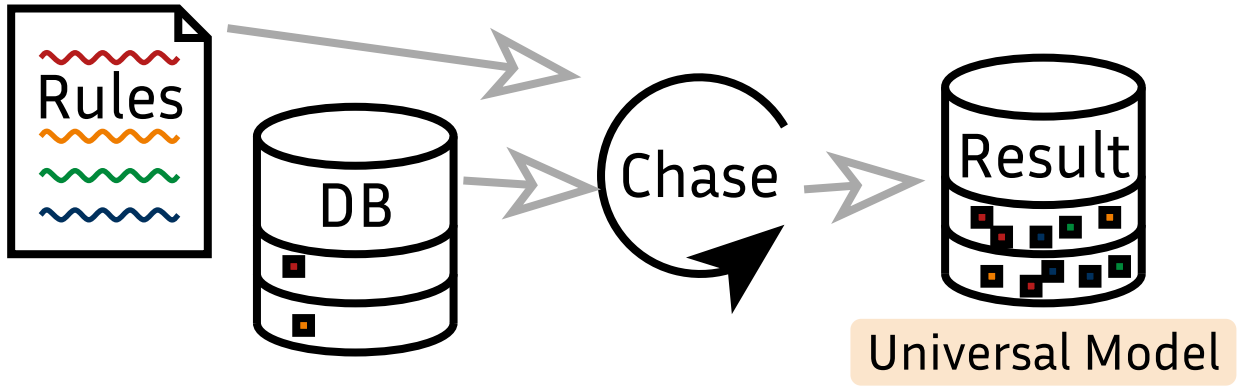
$$\mathrm{A}(x) \rightarrow \exists z.\ \mathrm{R}(x,\ z) \wedge \mathrm{B}(z)$$
$$\mathrm{B}(x) \rightarrow \mathrm{A}(x)$$
$$\mathrm{R}(x,\ y) \rightarrow \mathrm{R}(y,\ x)$$

Chase Crash Course for DL Folks

$\mathrm{A} \sqsubseteq \exists\, \mathrm{R}.\mathrm{B}$

$\mathrm{B} \sqsubseteq \mathrm{A}$

R is symmetric



$$\mathrm{A}(x) \rightarrow \exists z.\ \mathrm{R}(x,\ z) \wedge \mathrm{B}(z)$$

$$\mathrm{B}(x) \rightarrow \mathrm{A}(x)$$

$$\mathrm{R}(x,\ y) \rightarrow \mathrm{R}(y,\ x)$$

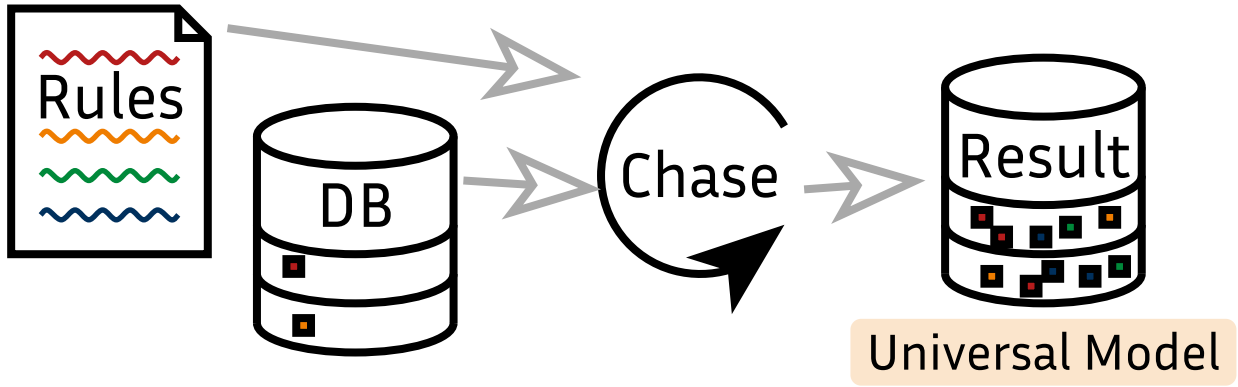
DB: $\mathrm{A}(c)$, $\mathrm{B}(c)$

Chase Crash Course for DL Folks

$\mathrm{A} \sqsubseteq \exists\, \mathrm{R}.\mathrm{B}$

$\mathrm{B} \sqsubseteq \mathrm{A}$

R is symmetric



$$\mathrm{A}(x) \rightarrow \exists z.\ \mathrm{R}(x,\ z) \wedge \mathrm{B}(z)$$

$$\mathrm{B}(x) \rightarrow \mathrm{A}(x)$$

$$\mathrm{R}(x,\ y) \rightarrow \mathrm{R}(y,\ x)$$

DB: $\mathrm{A}(c)$, $\mathrm{B}(c)$

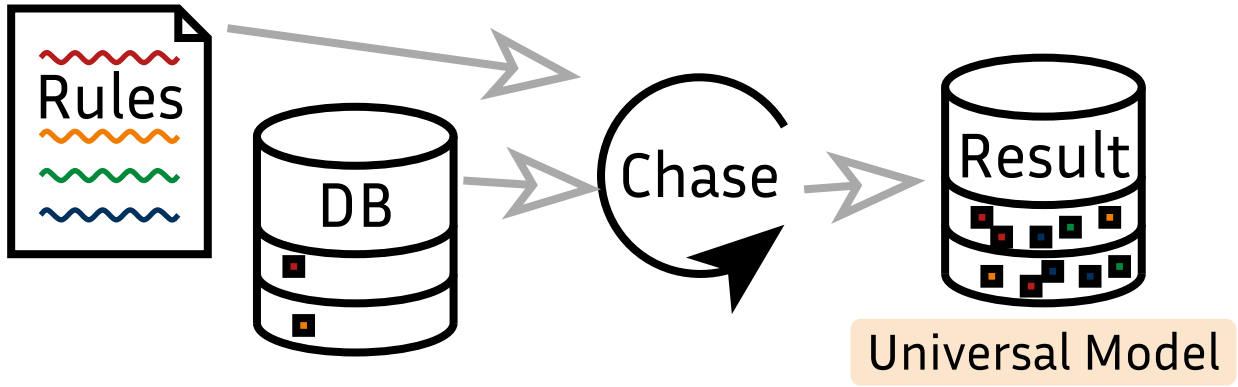
Step 1: $\mathrm{R}(c,\ f.c)$, $\mathrm{B}(f.c)$

Chase Crash Course for DL Folks

$\mathrm{A} \sqsubseteq \exists\, \mathrm{R}.\mathrm{B}$

$\mathrm{B} \sqsubseteq \mathrm{A}$

R is symmetric



$$\mathrm{A}(x) \rightarrow \exists z.\ \mathrm{R}(x,\ z) \wedge \mathrm{B}(z)$$

$$\mathrm{B}(x) \rightarrow \mathrm{A}(x)$$

$$\mathrm{R}(x,\ y) \rightarrow \mathrm{R}(y,\ x)$$

DB: $\mathrm{A}(c)$, $\mathrm{B}(c)$

Step 1: $\mathrm{R}(c,\ f.c)$, $\mathrm{B}(f.c)$

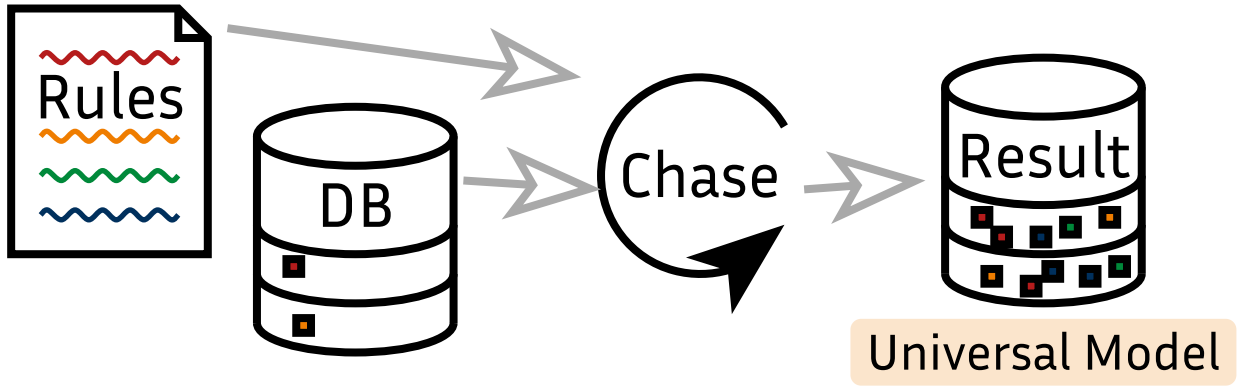
Step 2: $\mathrm{A}(f.c)$; Step 3: $\mathrm{R}(f.c,\ c)$

Chase Crash Course for DL Folks

$\mathrm{A} \sqsubseteq \exists\, \mathrm{R}.\mathrm{B}$

$\mathrm{B} \sqsubseteq \mathrm{A}$

R is symmetric



$$\mathrm{A}(x) \rightarrow \exists z.\ \mathrm{R}(x,\ z) \wedge \mathrm{B}(z)$$

$$\mathrm{B}(x) \rightarrow \mathrm{A}(x)$$

$$\mathrm{R}(x,\ y) \rightarrow \mathrm{R}(y,\ x)$$

DB: $\mathrm{A}(c)$, $\mathrm{B}(c)$

Step 1: $\mathrm{R}(c,\ f.c)$, $\mathrm{B}(f.c)$

Step 2: $\mathrm{A}(f.c)$; Step 3: $\mathrm{R}(f.c,\ c)$

Universal Models are "most general" and can answer conjunctive queries.

Chase Termination is Undecidable

$$\mathrm{A}(x) \rightarrow \exists z.\ \mathrm{R}(x,\ z) \wedge \mathrm{B}(z)$$
$$\mathrm{B}(x) \rightarrow \mathrm{A}(x)$$
$$\mathrm{R}(x,\ y) \rightarrow \mathrm{R}(y,\ x)$$

DB: $\mathrm{A}(c), \mathrm{B}(c)$

Step 1: $\mathrm{R}(c,\ f.c), \mathrm{B}(f.c)$

Step 2: $\mathrm{A}(f.c)$

Chase Termination is Undecidable

$$\mathrm{A}(x) \rightarrow \exists z.\ \mathrm{R}(x,\ z) \wedge \mathrm{B}(z)$$
$$\mathrm{B}(x) \rightarrow \mathrm{A}(x)$$
$$\mathrm{R}(x,\ y) \rightarrow \mathrm{R}(y,\ x)$$

DB: $\mathrm{A}(c), \mathrm{B}(c)$

Step 1: $\mathrm{R}(c,\ f.c), \mathrm{B}(f.c)$

Step 2: $\mathrm{A}(f.c)$

Step 3: $\mathrm{R}(f.c,\ f.f.c), \mathrm{B}(f.f.c)$

Chase Termination is Undecidable

$$\mathrm{A}(x) \rightarrow \exists z.\ \mathrm{R}(x,\ z) \wedge \mathrm{B}(z)$$
$$\mathrm{B}(x) \rightarrow \mathrm{A}(x)$$
$$\mathrm{R}(x,\ y) \rightarrow \mathrm{R}(y,\ x)$$

DB: $\mathrm{A}(c)$, $\mathrm{B}(c)$
Step 1: $\mathrm{R}(c,\ f.c)$, $\mathrm{B}(f.c)$
Step 2: $\mathrm{A}(f.c)$
Step 3: $\mathrm{R}(f.c,\ f.f.c)$, $\mathrm{B}(f.f.c)$

Termination of the example depends on rule application order (and more)!

Chase Termination is Undecidable

$$\mathrm{A}(x) \rightarrow \exists z.\ \mathrm{R}(x,\ z) \wedge \mathrm{B}(z)$$
$$\mathrm{B}(x) \rightarrow \mathrm{A}(x)$$
$$\mathrm{R}(x,\ y) \rightarrow \mathrm{R}(y,\ x)$$

DB: $\mathrm{A}(c)$, $\mathrm{B}(c)$
Step 1: $\mathrm{R}(c,\ f.c)$, $\mathrm{B}(f.c)$
Step 2: $\mathrm{A}(f.c)$
Step 3: $\mathrm{R}(f.c,\ f.f.c)$, $\mathrm{B}(f.f.c)$

Termination of the example depends on rule application order (and more)!

$\mathrm{CTK}^{r}_{\forall}$ is the set of all ontologies, i.e. pairs of TBoxes, and ABoxes, on which every ($\forall$) restricted (r) chase sequence terminates.

Chase Termination is Undecidable

$$\mathrm{A}(x) \rightarrow \exists z.\ \mathrm{R}(x,\ z) \wedge \mathrm{B}(z)$$
$$\mathrm{B}(x) \rightarrow \mathrm{A}(x)$$
$$\mathrm{R}(x,\ y) \rightarrow \mathrm{R}(y,\ x)$$

DB: $\mathrm{A}(c)$, $\mathrm{B}(c)$
Step 1: $\mathrm{R}(c,\ f.c)$, $\mathrm{B}(f.c)$
Step 2: $\mathrm{A}(f.c)$
Step 3: $\mathrm{R}(f.c,\ f.f.c)$, $\mathrm{B}(f.f.c)$

Termination of the example depends on rule application order (and more)!

$\mathrm{CTK}^{r}_{\forall}$ is the set of all knowledge bases, i.e. pairs of rule sets, and databases, on which every ($\forall$) restricted (r) chase sequence terminates.

Chase Termination is Undecidable

$$\mathrm{A}(x) \rightarrow \exists z.\ \mathrm{R}(x,\ z) \wedge \mathrm{B}(z)$$
$$\mathrm{B}(x) \rightarrow \mathrm{A}(x)$$
$$\mathrm{R}(x,\ y) \rightarrow \mathrm{R}(y,\ x)$$

DB: $\mathrm{A}(c)$, $\mathrm{B}(c)$
Step 1: $\mathrm{R}(c,\ f.c)$, $\mathrm{B}(f.c)$
Step 2: $\mathrm{A}(f.c)$
Step 3: $\mathrm{R}(f.c,\ f.f.c)$, $\mathrm{B}(f.f.c)$

Termination of the example depends on rule application order (and more)!

$\mathrm{CTK}^r_\forall$ is the set of all knowledge bases, i.e. pairs of rule sets, and databases, on which every ($\forall$) restricted (r) chase sequence terminates.
$\mathrm{CTR}^r_\forall$ is analogous for rule sets by $\forall$-quantifying over all databases.

Chase Termination is Undecidable (2)

$\mathrm{CTK}^r_\forall$ is the set of all knowledge bases, i.e. pairs of rule sets, and databases, on which every ($\forall$) restricted (r) chase sequence terminates.
$\mathrm{CTR}^r_\forall$ is analogous for rule sets by $\forall$-quantifying over all databases.

Chase Termination is Undecidable (2)

$\mathrm{CTK}^r_\forall$ is the set of all knowledge bases, i.e. pairs of rule sets, and databases, on which every ($\forall$) restricted (r) chase sequence terminates.
$\mathrm{CTR}^r_\forall$ is analogous for rule sets by $\forall$-quantifying over all databases.

Overview from (Grahne and Onet 2018)

$\mathrm{CTX}^r_{\mathrm{Q}}$	$\exists$	$\forall$
K	Σ^0_1-complete	Σ^0_1-complete
R	Π^0_2-complete	Π^0_2-complete

Chase Termination is Undecidable (2)

$\mathrm{CTK}^r_\forall$ is the set of all knowledge bases, i.e. pairs of rule sets, and databases, on which every ($\forall$) restricted (r) chase sequence terminates.
$\mathrm{CTR}^r_\forall$ is analogous for rule sets by $\forall$-quantifying over all databases.

Overview from (Grahne and Onet 2018)

$\mathrm{CTX}^r_{\mathrm{Q}}$	$\exists$	$\forall$
K	Σ^0_1-complete	Σ^0_1-complete
R	Π^0_2-complete	Π^0_2-complete

Σ^0_1 - Semi-Decidable Languages
(e.g. Halting Problem)

Π^0_2 - Co-Semi-Decidable with Semi-Decision Oracle
(e.g. Universal Halting Problem)

Chase Termination is Undecidable (2)

$\mathrm{CTK}^r_\forall$ is the set of all knowledge bases, i.e. pairs of rule sets, and databases, on which every ($\forall$) restricted (r) chase sequence terminates.
$\mathrm{CTR}^r_\forall$ is analogous for rule sets by $\forall$-quantifying over all databases.

Overview from (Grahne and Onet 2018)

$\mathrm{CTX}^r_{\mathrm{Q}}$	$\exists$	$\forall$
K	Σ^0_1-complete	Σ^0_1-complete
R	Π^0_2-complete	Π^0_2-complete

Membership: Run all Chase Sequences in Parallel

Hardness: TM can be simulated with Existential Rules

Chase Termination is Undecidable (2)

$\mathrm{CTK}^r_\forall$ is the set of all knowledge bases, i.e. pairs of rule sets, and databases, on which every ($\forall$) restricted (r) chase sequence terminates.
$\mathrm{CTR}^r_\forall$ is analogous for rule sets by $\forall$-quantifying over all databases.

Overview from (Grahne and Onet 2018)

$\mathrm{CTX}^r_{\mathrm{Q}}$	$\exists$	$\forall$
K	Σ^0_1-complete	Σ^0_1-complete
R	Π^0_2-complete	Π^0_2-complete

Membership: Use $\mathrm{CTK}^r_\exists$ oracle.

Hardness: (more involved)
see (Grahne and Onet 2018)

Chase Termination is Undecidable (2)

$\mathrm{CTK}^r_\forall$ is the set of all knowledge bases, i.e. pairs of rule sets, and databases, on which every ($\forall$) restricted (r) chase sequence terminates.
$\mathrm{CTR}^r_\forall$ is analogous for rule sets by $\forall$-quantifying over all databases.

Overview from (Grahne and Onet 2018)

$\mathrm{CTX}^r_{\mathrm{Q}}$	$\exists$	$\forall$
K	Σ^0_1-complete	Σ^0_1-complete
R	Π^0_2-complete	Π^0_2-complete

Membership: Run all Chase Sequences in Parallel **?**

Chase Termination is Undecidable (2)

$\mathrm{CTK}^r_\forall$ is the set of all knowledge bases, i.e. pairs of rule sets, and databases, on which every ($\forall$) restricted (r) chase sequence terminates.
$\mathrm{CTR}^r_\forall$ is analogous for rule sets by $\forall$-quantifying over all databases.

Overview from (Grahne and Onet 2018)

$\mathrm{CTX}^r_{\mathrm{Q}}$	$\exists$	$\forall$
K	Σ^0_1-complete	Σ^0_1-complete
R	Π^0_2-complete	Π^0_2-complete

Membership: Run all Chase Sequences in Parallel **?**

This does not work because of *fairness*!

Chase Termination is Undecidable (2)

$\mathrm{CTK}^r_\forall$ is the set of all knowledge bases, i.e. pairs of rule sets, and databases, on which every ($\forall$) restricted (r) chase sequence terminates.
$\mathrm{CTR}^r_\forall$ is analogous for rule sets by $\forall$-quantifying over all databases.

Overview from (Grahne and Onet 2018)

$\mathrm{CTX}^r_{\mathrm{Q}}$	$\exists$	$\forall$
K	Σ^0_1-complete	Π^0_2-hard
R	Π^0_2-complete	Π^0_2-hard

(Carral et al. 2022) have shown $\mathrm{CTK}^r_\forall$ to be at least Π^0_2-hard.

Chase Termination is Undecidable (2)

$\mathrm{CTK}^r_\forall$ is the set of all knowledge bases, i.e. pairs of rule sets, and databases, on which every ($\forall$) restricted (r) chase sequence terminates.
$\mathrm{CTR}^r_\forall$ is analogous for rule sets by $\forall$-quantifying over all databases.

Overview from (Grahne and Onet 2018)

$\mathrm{CTX}^r_{\mathrm{Q}}$	$\exists$	$\forall$
K	Σ^0_1-complete	Π^0_2-hard
R	Π^0_2-complete	Π^0_2-hard

(Carral et al. 2022) have shown $\mathrm{CTK}^r_\forall$ to be at least Π^0_2-hard.

The upper bound was still unknown! (Until slide 6)

Fairness Complicates Things

Why does the $\mathrm{CTK}^{r}_{\forall}$ membership idea (parallel chase) not work?

Fairness Complicates Things

Why does the $\mathrm{CTK}^{r}_{\forall}$ membership idea (parallel chase) not work?

There could be an infinite sequence which is not fair.

Fairness Complicates Things

Why does the $\mathrm{CTK}^{r}_{\forall}$ membership idea (parallel chase) not work?

There could be an infinite sequence which is not fair.

Example from (Gogacz et al. 2023)

DB: $\mathrm{R}(a, b, b)$

$\mathrm{R}(x, y, y) \rightarrow \exists z.\ \mathrm{R}(x, z, y) \wedge \mathrm{R}(z, y, y)$

$\mathrm{R}(x, y, z) \rightarrow \mathrm{R}(z, z, z)$

Fairness Complicates Things

Why does the $\mathrm{CTK}^r_\forall$ membership idea (parallel chase) not work?

There could be an infinite sequence which is not fair.

Example from (Gogacz et al. 2023)

DB: $\mathrm{R}(a, b, b)$

$\mathrm{R}(x, y, y) \rightarrow \exists z.\ \mathrm{R}(x, z, y) \wedge \mathrm{R}(z, y, y)$

$\mathrm{R}(x, y, z) \rightarrow \mathrm{R}(z, z, z)$

We can derive a chain of $\mathrm{R}(n_1, b, b), \mathrm{R}(n_2, b, b), \ldots$ but $\mathrm{R}(b, b, b)$ stops everything eventually.

Fairness Complicates Things

Why does the $\mathrm{CTK}^{r}_{\forall}$ membership idea (parallel chase) not work?

There could be an infinite sequence which is not fair.

Example from (Gogacz et al. 2023)

DB: $\mathrm{R}(a, b, b)$

$\mathrm{R}(x, y, y) \rightarrow \exists z.\ \mathrm{R}(x, z, y) \wedge \mathrm{R}(z, y, y)$

$\mathrm{R}(x, y, z) \rightarrow \mathrm{R}(z, z, z)$

We can derive a chain of $\mathrm{R}(n_1, b, b), \mathrm{R}(n_2, b, b), \ldots$ but $\mathrm{R}(b, b, b)$ stops everything eventually.

Fairness demands that rules are applied after finitely many steps.

Fairness Complicates Things

Why does the $\mathrm{CTK}^r_\forall$ membership idea (parallel chase) not work?

There could be an infinite sequence which is not fair.

Example from (Gogacz et al. 2023)

DB: $\mathrm{R}(a, b, b)$

$\mathrm{R}(x, y, y) \rightarrow \exists z.\ \mathrm{R}(x, z, y) \wedge \mathrm{R}(z, y, y)$

$\mathrm{R}(x, y, z) \rightarrow \mathrm{R}(z, z, z)$

We can derive a chain of $\mathrm{R}(n_1, b, b), \mathrm{R}(n_2, b, b), \ldots$ but $\mathrm{R}(b, b, b)$ stops everything eventually.

Fairness demands that rules are applied after finitely many steps.

Possible Fix: We could demand something stronger, e.g. “breadth-first”.

Back to $\mathrm{CTK}^r_\forall$: Π^1_1 Membership

Π^1_1 - *first analytical hierarchy level - beyond infinitely many Turing jumps.*

Back to $\mathrm{CTK}^{r}_{\forall}$: Π^1_1 Membership

Π^1_1 - *first analytical hierarchy level - beyond infinitely many Turing jumps.*

Complete Problem: Decide if a given non-deterministic Turing machine visits a designated state only finitely many times in each run. (Harel 1986)

Back to $\mathrm{CTK}^r_\forall$: Π^1_1 Membership

Π^1_1 - *first analytical hierarchy level - beyond infinitely many Turing jumps.*

Complete Problem: Decide if a given non-deterministic Turing machine visits a designated state only finitely many times in each run. (Harel 1986)

Proof Idea: Describe NTM that computes a chase sequence.

.

.

Back to $\mathrm{CTK}^r_\forall$: Π^1_1 Membership

Π^1_1 - *first analytical hierarchy level - beyond infinitely many Turing jumps.*

Complete Problem: Decide if a given non-deterministic Turing machine visits a designated state only finitely many times in each run. (Harel 1986)

Proof Idea: Describe NTM that computes a chase sequence.

- Keep a backlog of applicable rules R_i for each step i and a counter j.
-

Back to $\mathrm{CTK}^r_\forall$: Π^1_1 Membership

Π^1_1 - *first analytical hierarchy level - beyond infinitely many Turing jumps.*

Complete Problem: Decide if a given non-deterministic Turing machine visits a designated state only finitely many times in each run. (Harel 1986)

Proof Idea: Describe NTM that computes a chase sequence.

- Keep a backlog of applicable rules R_i for each step i and a counter j.
- If during the chase computation, all applications in R_j are obsolete, increment j and visit the designated state.

Back to $\mathrm{CTK}^r_\forall$: Π^1_1 Membership

Π^1_1 - *first analytical hierarchy level - beyond infinitely many Turing jumps.*

Complete Problem: Decide if a given non-deterministic Turing machine visits a designated state only finitely many times in each run. (Harel 1986)

Proof Idea: Describe NTM that computes a chase sequence.

- Keep a backlog of applicable rules R_i for each step i and a counter j.
- If during the chase computation, all applications in R_j are obsolete, increment j and visit the designated state.

Run visiting the desig. state infinitely often **iff** infinite fair chase sequence.

A Better Lower Bound for $\mathrm{CTK}^r_\forall$?

We know that we can simulate (N)TMs with existential rules.

Can we simulate the complete problem from the last slide?

A Better Lower Bound for $\mathrm{CTK}^{r}_{\forall}$?

Complete Problem: Decide if a given non-deterministic Turing machine visits a designated state only finitely many times in each run. (Harel 1986)

A Better Lower Bound for $\mathrm{CTK}^r_\forall$?

Complete Problem: Decide if a given non-deterministic Turing machine visits a designated state only finitely many times in each run. (Harel 1986)

Goal: Construct knowledge base that is in $\mathrm{CTK}^r_\forall$ iff NTM has this property.

A Better Lower Bound for $\mathrm{CTK}^r_\forall$?

Complete Problem: Decide if a given non-deterministic Turing machine visits a designated state only finitely many times in each run. (Harel 1986)

Goal: Construct knowledge base that is in $\mathrm{CTK}^r_\forall$ iff NTM has this property.
Issue: NTM has infinite runs that visit the desig. state only finitely often.

A Better Lower Bound for $\mathrm{CTK}^r_\forall$?

Complete Problem: Decide if a given non-deterministic Turing machine visits a designated state only finitely many times in each run. (Harel 1986)

Goal: Construct knowledge base that is in $\mathrm{CTK}^r_\forall$ iff NTM has this property.
Issue: NTM has infinite runs that visit the desig. state only finitely often.

Complement: Decide if a given non-deterministic Turing machine has a run that visits a designated state infinitely often.

A Better Lower Bound for $\mathrm{CTK}^r_\forall$?

Complete Problem: Decide if a given non-deterministic Turing machine visits a designated state only finitely many times in each run. (Harel 1986)

Goal: Construct knowledge base that is in $\mathrm{CTK}^r_\forall$ iff NTM has this property.
Issue: NTM has infinite runs that visit the desig. state only finitely often.

Complement: Decide if a given non-deterministic Turing machine has a run that visits a designated state recurringly after finitely many steps.

A Better Lower Bound for $\mathrm{CTK}^r_\forall$?

Complete Problem: Decide if a given non-deterministic Turing machine visits a designated state only finitely many times in each run. (Harel 1986)

Goal: Construct knowledge base that is in $\mathrm{CTK}^r_\forall$ iff NTM has this property.
Issue: NTM has infinite runs that visit the desig. state only finitely often.

Complement: Decide if a given non-deterministic Turing machine has a run that visits a designated state recurringly after finitely many steps.

With emergency brakes, we can force the chase to terminate after finitely many steps. If the designated state is visited, we create a new brake.

Differences in the $\mathrm{CTR}^r_\forall$ Case

Membership: Similar to $\mathrm{CTK}^r_\forall$ case by adjusting NTM problem accordingly.

Differences in the $\mathrm{CTR}^r_\forall$ Case

Membership: Similar to $\mathrm{CTK}^r_\forall$ case by adjusting NTM problem accordingly.

Hardness: Tricky since our chase needs to behave well also on ill-shaped databases that do not correspond to TM configurations...

Differences in the $\mathrm{CTR}^r_\forall$ Case

Membership: Similar to $\mathrm{CTK}^r_\forall$ case by adjusting NTM problem accordingly.

Hardness: Tricky since our chase needs to behave well also on ill-shaped databases that do not correspond to TM configurations...

Key Observation: The simulation "heals" malformed configurations, giving us a proper configuration after finitely many steps, which is good enough!

Differences in the $\mathrm{CTR}^r_\forall$ Case

Membership: Similar to $\mathrm{CTK}^r_\forall$ case by adjusting NTM problem accordingly.

Hardness: Tricky since our chase needs to behave well also on ill-shaped databases that do not correspond to TM configurations...

Key Observation: The simulation "heals" malformed configurations, giving us a proper configuration after finitely many steps, which is good enough!

$\mathrm{CTX}^r_{\mathrm{Q}}$	$\exists$	$\forall$
K	Σ^0_1-complete	Π^1_1-complete
R	Π^0_2-complete	Π^1_1-complete

Skolem and Restricted Chase for Disjunctive Rules

The Skolem chase only checks if what would be produced is not present yet.

$$\mathrm{A}(x) \rightarrow \exists z.\ \mathrm{R}(x,\ z) \wedge \mathrm{B}(z)$$

$$\mathrm{R}(x,\ y) \rightarrow \mathrm{A}(y) \wedge \mathrm{R}(y,\ x) \wedge \mathrm{B}(x)$$

Skolem and Restricted Chase for Disjunctive Rules

The Skolem chase only checks if what would be produced is not present yet.

$$\mathrm{A}(x) \rightarrow \exists z.\ \mathrm{R}(x,\ z) \wedge \mathrm{B}(z)$$

$$\mathrm{R}(x,\ y) \rightarrow \mathrm{A}(y) \wedge \mathrm{R}(y,\ x) \wedge \mathrm{B}(x)$$

DB: $\mathrm{A}(c)$

Skolem and Restricted Chase for Disjunctive Rules

The Skolem chase only checks if what would be produced is not present yet.

$$\mathrm{A}(x) \rightarrow \exists z.\ \mathrm{R}(x,\ z) \wedge \mathrm{B}(z)$$

$$\mathrm{R}(x,\ y) \rightarrow \mathrm{A}(y) \wedge \mathrm{R}(y,\ x) \wedge \mathrm{B}(x)$$

DB: $\mathrm{A}(c)$; Step 1: $\mathrm{R}(c,\ f.c)$, $\mathrm{B}(f.c)$

Skolem and Restricted Chase for Disjunctive Rules

The Skolem chase only checks if what would be produced is not present yet.

$$\mathrm{A}(x) \rightarrow \exists z.\ \mathrm{R}(x,\ z) \wedge \mathrm{B}(z)$$
$$\mathrm{R}(x,\ y) \rightarrow \mathrm{A}(y) \wedge \mathrm{R}(y,\ x) \wedge \mathrm{B}(x)$$

DB: $\mathrm{A}(c)$; Step 1: $\mathrm{R}(c,\ f.c)$, $\mathrm{B}(f.c)$
Step 2: $\mathrm{A}(f.c)$, $\mathrm{R}(f.c,\ c)$, $\mathrm{B}(c)$

Skolem and Restricted Chase for Disjunctive Rules

The Skolem chase only checks if what would be produced is not present yet.

$$\mathrm{A}(x) \rightarrow \exists z.\ \mathrm{R}(x,\ z) \wedge \mathrm{B}(z)$$
$$\mathrm{R}(x,\ y) \rightarrow \mathrm{A}(y) \wedge \mathrm{R}(y,\ x) \wedge \mathrm{B}(x)$$

DB: $\mathrm{A}(c)$; Step 1: $\mathrm{R}(c,\ f.c)$, $\mathrm{B}(f.c)$
Step 2: $\mathrm{A}(f.c)$, $\mathrm{R}(f.c,\ c)$, $\mathrm{B}(c)$ 

Skolem and Restricted Chase for Disjunctive Rules

The Skolem chase only checks if what would be produced is not present yet.

$$\mathrm{A}(x) \rightarrow \exists z.\ \mathrm{R}(x,\ z) \wedge \mathrm{B}(z)$$
$$\mathrm{R}(x,\ y) \rightarrow \mathrm{A}(y) \wedge \mathrm{R}(y,\ x) \wedge \mathrm{B}(x)$$

DB: $\mathrm{A}(c)$; Step 1: $\mathrm{R}(c,\ f.c)$, $\mathrm{B}(f.c)$
Step 2: $\mathrm{A}(f.c)$, $\mathrm{R}(f.c,\ c)$, $\mathrm{B}(c)$ 
Step 3: $\mathrm{R}(f.c,\ f.f.c)$, $\mathrm{B}(f.f.c)$; ...

Skolem and Restricted Chase for Disjunctive Rules

The Skolem chase only checks if what would be produced is not present yet.

$$\mathrm{A}(x) \to \exists z.\ \mathrm{R}(x,\ z) \land \mathrm{B}(z)$$
$$\mathrm{R}(x,\ y) \to \mathrm{A}(y) \land \mathrm{R}(y,\ x) \land \mathrm{B}(x)$$

DB: $\mathrm{A}(c)$; Step 1: $\mathrm{R}(c,\ f.c)$, $\mathrm{B}(f.c)$
Step 2: $\mathrm{A}(f.c)$, $\mathrm{R}(f.c,\ c)$, $\mathrm{B}(c)$ 🛑
Step 3: $\mathrm{R}(f.c,\ f.f.c)$, $\mathrm{B}(f.f.c)$; ...

With disjunctions, the Skolem chase behaves more like the restricted chase.

Skolem and Restricted Chase for Disjunctive Rules

The Skolem chase only checks if what would be produced is not present yet.

$$\mathrm{A}(x) \to \exists z.\ \mathrm{R}(x,\ z) \wedge \mathrm{B}(z)$$
$$\mathrm{R}(x,\ y) \to \mathrm{A}(y) \wedge \mathrm{R}(y,\ x) \wedge \mathrm{B}(x)$$

DB: $\mathrm{A}(c)$; Step 1: $\mathrm{R}(c,\ f.c)$, $\mathrm{B}(f.c)$
Step 2: $\mathrm{A}(f.c)$, $\mathrm{R}(f.c,\ c)$, $\mathrm{B}(c)$ 
Step 3: $\mathrm{R}(f.c,\ f.f.c)$, $\mathrm{B}(f.f.c)$; ...

With disjunctions, the Skolem chase behaves more like the restricted chase.

$$\mathrm{A}(x) \to \mathrm{B}(x)$$
$$\vee\ \exists z.\ \mathrm{R}(x,\ z) \wedge \mathrm{B}(z)$$
$$\mathrm{R}(x,\ y) \to \mathrm{A}(y) \wedge \mathrm{R}(y,\ x) \wedge \mathrm{B}(x)$$

Note: The disjunctive chase can be represented as a possibly infinite tree.

Skolem and Restricted Chase for Disjunctive Rules

The Skolem chase only checks if what would be produced is not present yet.

$$\mathrm{A}(x) \to \exists z.\ \mathrm{R}(x, z) \wedge \mathrm{B}(z)$$
$$\mathrm{R}(x, y) \to \mathrm{A}(y) \wedge \mathrm{R}(y, x) \wedge \mathrm{B}(x)$$

DB: $\mathrm{A}(c)$; Step 1: $\mathrm{R}(c, f.c), \mathrm{B}(f.c)$
Step 2: $\mathrm{A}(f.c), \mathrm{R}(f.c, c), \mathrm{B}(c)$ 
Step 3: $\mathrm{R}(f.c, f.f.c), \mathrm{B}(f.f.c)$; ...

With disjunctions, the Skolem chase behaves more like the restricted chase.

$$\mathrm{A}(x) \to \mathrm{B}(x)$$
$$\vee\ \exists z.\ \mathrm{R}(x, z) \wedge \mathrm{B}(z)$$
$$\mathrm{R}(x, y) \to \mathrm{A}(y) \wedge \mathrm{R}(y, x) \wedge \mathrm{B}(x)$$

DB: $\mathrm{A}(c)$

Note: The disjunctive chase can be represented as a possibly infinite tree.

Skolem and Restricted Chase for Disjunctive Rules

The Skolem chase only checks if what would be produced is not present yet.

$$\mathrm{A}(x) \to \exists z.\ \mathrm{R}(x,\ z) \wedge \mathrm{B}(z)$$
$$\mathrm{R}(x,\ y) \to \mathrm{A}(y) \wedge \mathrm{R}(y,\ x) \wedge \mathrm{B}(x)$$

DB: $\mathrm{A}(c)$; Step 1: $\mathrm{R}(c,\ f.c)$, $\mathrm{B}(f.c)$
Step 2: $\mathrm{A}(f.c)$, $\mathrm{R}(f.c,\ c)$, $\mathrm{B}(c)$ 
Step 3: $\mathrm{R}(f.c,\ f.f.c)$, $\mathrm{B}(f.f.c)$; ...

With disjunctions, the Skolem chase behaves more like the restricted chase.

$$\mathrm{A}(x) \to \mathrm{B}(x)$$
$$\vee\ \exists z.\ \mathrm{R}(x,\ z) \wedge \mathrm{B}(z)$$
$$\mathrm{R}(x,\ y) \to \mathrm{A}(y) \wedge \mathrm{R}(y,\ x) \wedge \mathrm{B}(x)$$

DB: $\mathrm{A}(c)$; Step 1a: $\mathrm{B}(c)$

Note: The disjunctive chase can be represented as a possibly infinite tree.

Skolem and Restricted Chase for Disjunctive Rules

The Skolem chase only checks if what would be produced is not present yet.

$$\mathrm{A}(x) \rightarrow \exists z.\ \mathrm{R}(x,\ z) \wedge \mathrm{B}(z)$$
$$\mathrm{R}(x,\ y) \rightarrow \mathrm{A}(y) \wedge \mathrm{R}(y,\ x) \wedge \mathrm{B}(x)$$

DB: $\mathrm{A}(c)$; Step 1: $\mathrm{R}(c,\ f.c)$, $\mathrm{B}(f.c)$
Step 2: $\mathrm{A}(f.c)$, $\mathrm{R}(f.c,\ c)$, $\mathrm{B}(c)$ 
Step 3: $\mathrm{R}(f.c,\ f.f.c)$, $\mathrm{B}(f.f.c)$; ...

With disjunctions, the Skolem chase behaves more like the restricted chase.

$$\mathrm{A}(x) \rightarrow \mathrm{B}(x)$$
$$\vee\ \exists z.\ \mathrm{R}(x,\ z) \wedge \mathrm{B}(z)$$
$$\mathrm{R}(x,\ y) \rightarrow \mathrm{A}(y) \wedge \mathrm{R}(y,\ x) \wedge \mathrm{B}(x)$$

DB: $\mathrm{A}(c)$; Step 1a: $\mathrm{B}(c)$
Step 1b: $\mathrm{R}(c,\ f.c)$, $\mathrm{B}(f.c)$

Note: The disjunctive chase can be represented as a possibly infinite tree.

Skolem and Restricted Chase for Disjunctive Rules

The Skolem chase only checks if what would be produced is not present yet.

$$\mathrm{A}(x) \rightarrow \exists z.\ \mathrm{R}(x,\ z) \wedge \mathrm{B}(z)$$
$$\mathrm{R}(x,\ y) \rightarrow \mathrm{A}(y) \wedge \mathrm{R}(y,\ x) \wedge \mathrm{B}(x)$$

DB: $\mathrm{A}(c)$; Step 1: $\mathrm{R}(c,\ f.c)$, $\mathrm{B}(f.c)$
Step 2: $\mathrm{A}(f.c)$, $\mathrm{R}(f.c,\ c)$, $\mathrm{B}(c)$ 
Step 3: $\mathrm{R}(f.c,\ f.f.c)$, $\mathrm{B}(f.f.c)$; ...

With disjunctions, the Skolem chase behaves more like the restricted chase.

$$\mathrm{A}(x) \rightarrow \mathrm{B}(x)$$
$$\vee\ \exists z.\ \mathrm{R}(x,\ z) \wedge \mathrm{B}(z)$$
$$\mathrm{R}(x,\ y) \rightarrow \mathrm{A}(y) \wedge \mathrm{R}(y,\ x) \wedge \mathrm{B}(x)$$

DB: $\mathrm{A}(c)$; Step 1a: $\mathrm{B}(c)$
Step 1b: $\mathrm{R}(c,\ f.c)$, $\mathrm{B}(f.c)$
Step 2b: $\mathrm{A}(f.c)$, $\mathrm{R}(f.c,\ c)$, $\mathrm{B}(c)$

Note: The disjunctive chase can be represented as a possibly infinite tree.

Skolem and Restricted Chase for Disjunctive Rules

The Skolem chase only checks if what would be produced is not present yet.

$$\mathrm{A}(x) \rightarrow \exists z.\ \mathrm{R}(x,\ z) \wedge \mathrm{B}(z)$$
$$\mathrm{R}(x,\ y) \rightarrow \mathrm{A}(y) \wedge \mathrm{R}(y,\ x) \wedge \mathrm{B}(x)$$

DB: $\mathrm{A}(c)$; Step 1: $\mathrm{R}(c,\ f.c)$, $\mathrm{B}(f.c)$
Step 2: $\mathrm{A}(f.c)$, $\mathrm{R}(f.c,\ c)$, $\mathrm{B}(c)$ 🛑
Step 3: $\mathrm{R}(f.c,\ f.f.c)$, $\mathrm{B}(f.f.c)$; ...

With disjunctions, the Skolem chase behaves more like the restricted chase.

$$\mathrm{A}(x) \rightarrow \mathrm{B}(x)$$
$$\vee\ \exists z.\ \mathrm{R}(x,\ z) \wedge \mathrm{B}(z)$$
$$\mathrm{R}(x,\ y) \rightarrow \mathrm{A}(y) \wedge \mathrm{R}(y,\ x) \wedge \mathrm{B}(x)$$

DB: $\mathrm{A}(c)$; Step 1a: $\mathrm{B}(c)$
Step 1b: $\mathrm{R}(c,\ f.c)$, $\mathrm{B}(f.c)$
Step 2b: $\mathrm{A}(f.c)$, $\mathrm{R}(f.c,\ c)$, $\mathrm{B}(c)$ ✅

Note: The disjunctive chase can be represented as a possibly infinite tree.

Ensuring Termination - DMFA

Model-Faithful Acyclicity (MFA) (Grau et al. 2013) guarantees Skolem chase termination ($\mathrm{CTR}^{s}_{\forall}$ Membership) and is extendable for disjunctions.

Ensuring Termination - DMFA

Model-Faithful Acyclicity (MFA) (Grau et al. 2013) guarantees Skolem chase termination ($\mathrm{CTR}^{s}_{\forall}$ Membership) and is extendable for disjunctions.

- Start on critical instance of rule set R: $\{P(\star, ..., \star) \mid P \in \mathrm{Predicates}(R)\}$
- Compute Skolem chase and abort if cyclic term (e.g. $f.f.\star$) is reached.

Ensuring Termination - DMFA

Model-Faithful Acyclicity (MFA) (Grau et al. 2013) guarantees Skolem chase termination ($\mathrm{CTR}^{s}_{\forall}$ Membership) and is extendable for disjunctions.

- Start on critical instance of rule set R: $\{P(\star, ..., \star) \mid P \in \mathrm{Predicates}(R)\}$
- Compute Skolem chase and abort if cyclic term (e.g. $f.f.\star$) is reached.

The chase on the critical instance necessarily terminates if no cyclic term occurs. Also, every chase on every database can be embedded ($g : \_ \mapsto \star$).

Ensuring Termination - DMFA

Model-Faithful Acyclicity (MFA) (Grau et al. 2013) guarantees Skolem chase termination ($\mathrm{CTR}^{s}_{\forall}$ Membership) and is extendable for disjunctions.

- Start on critical instance of rule set R: $\{P(\star, ..., \star) \mid P \in \mathrm{Predicates}(R)\}$
- Compute Skolem chase and abort if cyclic term (e.g. $f.f.\star$) is reached.

The chase on the critical instance necessarily terminates if no cyclic term occurs. Also, every chase on every database can be embedded ($g : \_ \mapsto \star$).

To detect $\mathrm{CTR}^{s}_{\forall}$ Membership for more rule sets, we can ignore blocked rule applications (Gerlach and Carral 2023a). This has been done for the restricted chase similarly in RMFA (Carral et al. 2017).

High-Level Idea for Blocking

We can have unique Skolem function symbols for each existential variable.

High-Level Idea for Blocking

We can have unique Skolem function symbols for each existential variable. From the Skolem term t, we can reconstruct a set of facts F_t necessary to derive t.

High-Level Idea for Blocking

We can have unique Skolem function symbols for each existential variable. From the Skolem term t, we can reconstruct a set of facts F_t necessary to derive t. If the rule is not applicable for F_t, then it is blocked.

High-Level Idea for Blocking

We can have unique Skolem function symbols for each existential variable. From the Skolem term t, we can reconstruct a set of facts F_t necessary to derive t. If the rule is not applicable for F_t, then it is blocked. Skolem and restricted chase differ in the definition of applicable.

High-Level Idea for Blocking

We can have unique Skolem function symbols for each existential variable. From the Skolem term t, we can reconstruct a set of facts F_t necessary to derive t. If the rule is not applicable for F_t, then it is blocked. Skolem and restricted chase differ in the definition of applicable.

$$\begin{aligned} \mathrm{A}(x) &\rightarrow \mathrm{B}(x) \\ &\vee \exists z.\ \mathrm{R}(x,\ z) \wedge \mathrm{B}(z) \\ \mathrm{R}(x,\ y) &\rightarrow \mathrm{A}(y) \wedge \mathrm{R}(y,\ x) \wedge \mathrm{B}(x) \end{aligned}$$

High-Level Idea for Blocking

We can have unique Skolem function symbols for each existential variable. From the Skolem term t, we can reconstruct a set of facts F_t necessary to derive t. If the rule is not applicable for F_t, then it is blocked. Skolem and restricted chase differ in the definition of applicable.

$$\begin{aligned} \mathrm{A}(x) &\rightarrow \mathrm{B}(x) \\ &\vee \exists z.\ \mathrm{R}(x,\ z) \wedge \mathrm{B}(z) \\ \mathrm{R}(x,\ y) &\rightarrow \mathrm{A}(y) \wedge \mathrm{R}(y,\ x) \wedge \mathrm{B}(x) \end{aligned}$$

Consider $t = f.c$ (f is for z)

F_t: $\mathrm{R}(c,\ f.c), \mathrm{B}(f.c),$

High-Level Idea for Blocking

We can have unique Skolem function symbols for each existential variable. From the Skolem term t, we can reconstruct a set of facts F_t necessary to derive t. If the rule is not applicable for F_t, then it is blocked. Skolem and restricted chase differ in the definition of applicable.

$$\mathrm{A}(x) \to \mathrm{B}(x)$$
$$\vee\ \exists z.\ \mathrm{R}(x,\ z) \wedge \mathrm{B}(z)$$
$$\mathrm{R}(x,\ y) \to \mathrm{A}(y) \wedge \mathrm{R}(y,\ x) \wedge \mathrm{B}(x)$$

Consider $t = f.c$ (f is for z)

F_t: $\mathrm{R}(c,\ f.c)$, $\mathrm{B}(f.c)$, $\mathrm{A}(c)$,

High-Level Idea for Blocking

We can have unique Skolem function symbols for each existential variable. From the Skolem term t, we can reconstruct a set of facts F_t necessary to derive t. If the rule is not applicable for F_t, then it is blocked. Skolem and restriction chase differ in the definition of applicable.

$$\begin{aligned} \mathrm{A}(x) &\to \mathrm{B}(x) \\ &\vee \exists z.\ \mathrm{R}(x,\ z) \wedge \mathrm{B}(z) \\ \mathrm{R}(x,\ y) &\to \mathrm{A}(y) \wedge \mathrm{R}(y,\ x) \wedge \mathrm{B}(x) \end{aligned}$$

Consider $t = f.c$ (f is for z)

F_t: $\mathrm{R}(c,\ f.c)$, $\mathrm{B}(f.c)$, $\mathrm{A}(c)$, $\mathrm{A}(f.c)$, $\mathrm{R}(f.c,\ c)$, $\mathrm{B}(c)$

High-Level Idea for Blocking

We can have unique Skolem function symbols for each existential variable. From the Skolem term t, we can reconstruct a set of facts F_t necessary to derive t. If the rule is not applicable for F_t, then it is blocked. Skolem and restricted chase differ in the definition of applicable.

$$\mathrm{A}(x) \to \mathrm{B}(x)$$
$$\vee\ \exists z.\ \mathrm{R}(x,\ z) \wedge \mathrm{B}(z)$$
$$\mathrm{R}(x,\ y) \to \mathrm{A}(y) \wedge \mathrm{R}(y,\ x) \wedge \mathrm{B}(x)$$

Consider $t = f.c$ (f is for z)

F_t: $\mathrm{R}(c,\ f.c)$, $\mathrm{B}(f.c)$, $\mathrm{A}(c)$, $\mathrm{A}(f.c)$, $\mathrm{R}(f.c,\ c)$, $\mathrm{B}(c)$

Trying to apply the first rule to $\mathrm{A}(f.c)$ would be detected to be blocked since $\mathrm{B}(f.c)$ is present in F_t.

Ensuring Non-Termination - DMFC/RPC

We aim to find a DB for that every chase tree has an infinite branch - this yields a sufficient condition for $\mathrm{CTR}^{s}_{\exists}/\mathrm{CTR}^{r}_{\exists}$ *non-membership.*

Ensuring Non-Termination - DMFC/RPC

We aim to find a DB for that every chase tree has an infinite branch - this yields a sufficient condition for $\mathrm{CTR}^s_\exists / \mathrm{CTR}^r_\exists$ *non-membership.*

Inspired by MFC/RMFC (Carral et al. 2017). However, RMFC proof is incorrect.

Ensuring Non-Termination - DMFC/RPC

We aim to find a DB for that every chase tree has an infinite branch - this yields a sufficient condition for $\mathrm{CTR}^s_\exists/\mathrm{CTR}^r_\exists$ non-membership.
Inspired by MFC/RMFC (Carral et al. 2017). However, RMFC proof is incorrect.

Based on rule body $\mathrm{A}(c)$, we try to complete a cycle, i.e. reach $\mathrm{A}(f.c)$.

$$\begin{aligned} \mathrm{A}(x) &\rightarrow \mathrm{R}(x, x) \\ &\vee \exists z.\ \mathrm{R}(x, z) \wedge \mathrm{B}(z) \\ \mathrm{R}(x, y) &\rightarrow \mathrm{A}(y) \wedge \mathrm{R}(y, x) \wedge \mathrm{B}(x) \end{aligned}$$

Ensuring Non-Termination - DMFC/RPC

We aim to find a DB for that every chase tree has an infinite branch - this yields a sufficient condition for $\mathrm{CTR}^s_\exists/\mathrm{CTR}^r_\exists$ non-membership.
Inspired by MFC/RMFC (Carral et al. 2017). However, RMFC proof is incorrect.

Based on rule body $\mathrm{A}(c)$, we try to complete a cycle, i.e. reach $\mathrm{A}(f.c)$.
Only unblockable applications!

$$\begin{aligned} \mathrm{A}(x) &\rightarrow \mathrm{R}(x, x) \\ &\vee \exists z.\ \mathrm{R}(x, z) \wedge \mathrm{B}(z) \\ \mathrm{R}(x, y) &\rightarrow \mathrm{A}(y) \wedge \mathrm{R}(y, x) \wedge \mathrm{B}(x) \end{aligned}$$

Ensuring Non-Termination - DMFC/RPC

We aim to find a DB for that every chase tree has an infinite branch - this yields a sufficient condition for $\mathrm{CTR}^s_\exists/\mathrm{CTR}^r_\exists$ non-membership.
Inspired by MFC/RMFC (Carral et al. 2017). However, RMFC proof is incorrect.

Based on rule body $\mathrm{A}(c)$, we try to complete a cycle, i.e. reach $\mathrm{A}(f.c)$.
Only unblockable applications!

$$\begin{aligned} \mathrm{A}(x) &\rightarrow \mathrm{R}(x, x) \\ &\vee \exists z.\ \mathrm{R}(x, z) \wedge \mathrm{B}(z) \\ \mathrm{R}(x, y) &\rightarrow \mathrm{A}(y) \wedge \mathrm{R}(y, x) \wedge \mathrm{B}(x) \end{aligned}$$

Besides applicability, unblockable applications ensure that corresponding applications in all following cycles are also unblockable.

Ensuring Non-Termination - DMFC/RPC

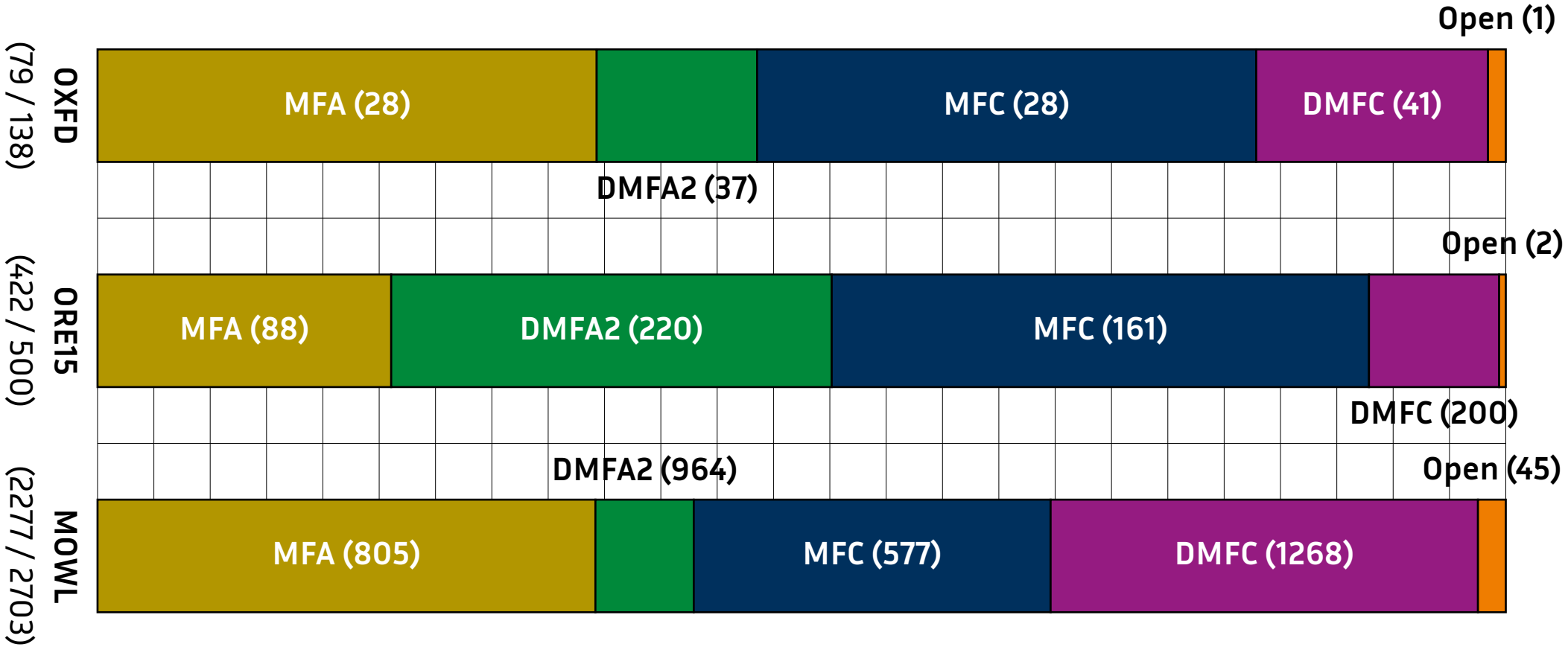
We aim to find a DB for that every chase tree has an infinite branch - this yields a sufficient condition for $\mathrm{CTR}^s_\exists/\mathrm{CTR}^r_\exists$ non-membership.
Inspired by MFC/RMFC (Carral et al. 2017). However, RMFC proof is incorrect.

Based on rule body $\mathrm{A}(c)$, we try to complete a cycle, i.e. reach $\mathrm{A}(f.c)$.
Only unblockable applications!

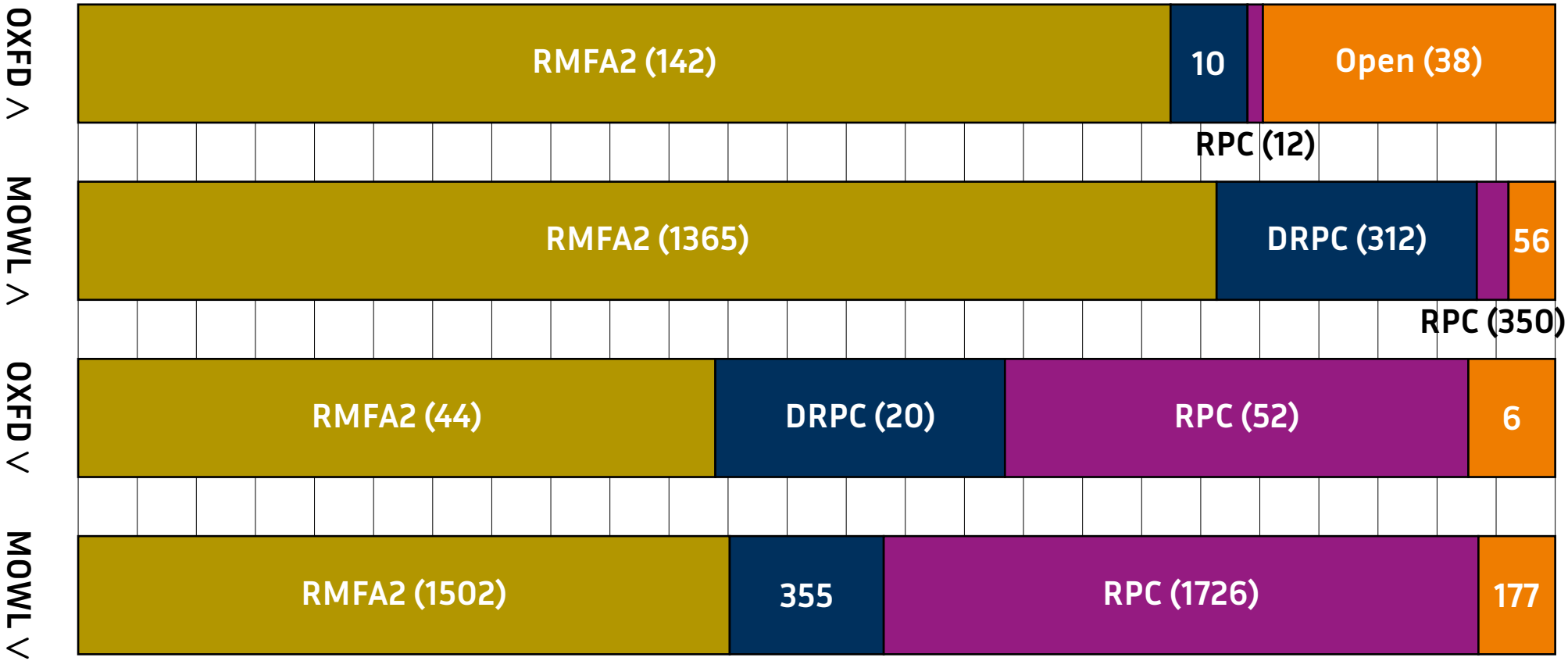
$$\mathrm{A}(x) \rightarrow \mathrm{R}(x, x)$$
$$\vee\ \exists z.\ \mathrm{R}(x, z) \wedge \mathrm{B}(z)$$
$$\mathrm{R}(x, y) \rightarrow \mathrm{A}(y) \wedge \mathrm{R}(y, x) \wedge \mathrm{B}(x)$$

Besides applicability, unblockable applications ensure that corresponding applications in all following cycles are also unblockable.
This is in a way dual to blocking and reuses something along the lines of F_t.

Evaluation of Sufficient Conditions (DMFA/DMFC)



Evaluation of Sufficient Conditions (RPC)



Summing up...

Summing up...

- Fairness makes Chase Termination highly Undecidable
-

$\mathrm{CTX}^{r}_{\mathrm{Q}}$	$\exists$	$\forall$
K	Σ^0_1-complete	Π^1_1-complete
R	Π^0_2-complete	Π^1_1-complete

Summing up...

- Fairness makes Chase Termination highly Undecidable
- Sufficient Conditions cover many rule set but especially non-termination is tricky to get right.

$\mathrm{CTX}^r_{\mathrm{Q}}$	$\exists$	$\forall$
K	Σ^0_1-complete	Π^1_1-complete
R	Π^0_2-complete	Π^1_1-complete

$$\mathrm{DMFA} \subset \mathrm{CTR}^s_\forall \subset \mathrm{CTR}^s_\exists \subset \mathrm{DMFC}$$

$$\mathrm{RMFA} \subset \mathrm{CTR}^r_\forall \subset \mathrm{CTR}^r_\exists \subset \mathrm{RPC}$$

Summing up...

- Fairness makes Chase Termination highly Undecidable
- Sufficient Conditions cover many rule set but especially non-termination is tricky to get right.

$\mathrm{CTX}^r_{\mathrm{Q}}$	$\exists$	$\forall$
K	Σ^0_1-complete	Π^1_1-complete
R	Π^0_2-complete	Π^1_1-complete

$$\mathrm{DMFA} \subset \mathrm{CTR}^s_\forall \subset \mathrm{CTR}^s_\exists \subset \mathrm{DMFC}$$

$$\mathrm{RMFA} \subset \mathrm{CTR}^r_\forall \subset \mathrm{CTR}^r_\exists \subset \mathrm{RPC}$$

Open Problems / Ongoing Endeavors (among many others):

- Disjunctive Skolem Chase Termination complete for Π^1_1 **?** (Likely true.)
-

Summing up...

- Fairness makes Chase Termination highly Undecidable
- Sufficient Conditions cover many rule set but especially non-termination is tricky to get right.

$\mathrm{CTX}^r_{\mathrm{Q}}$	$\exists$	$\forall$
K	Σ^0_1-complete	Π^1_1-complete
R	Π^0_2-complete	Π^1_1-complete

$$\mathrm{DMFA} \subset \mathrm{CTR}^s_\forall \subset \mathrm{CTR}^s_\exists \subset \mathrm{DMFC}$$

$$\mathrm{RMFA} \subset \mathrm{CTR}^r_\forall \subset \mathrm{CTR}^r_\exists \subset \mathrm{RPC}$$

Open Problems / Ongoing Endeavors (among many others):

- Disjunctive Skolem Chase Termination complete for Π^1_1 **?** (Likely true.)
- Lean Formalization of Sufficient Conditions (MFA done) - dmfa.dev/lean

References

Carral D, Dragoste I, Krötzsch M (2017) Restricted Chase (Non)Termination for Existential Rules with Disjunctions. In: Sierra C (ed) Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017. ijcai.org, pp 922–928

Carral D, Gerlach L, Larroque L, Thomazo M (2025) Restricted Chase Termination: You Want More than Fairness. Proc ACM Manag Data 3:. https://doi.org/10.1145/3725246

References

Carral D, Larroque L, Mugnier M-L, Thomazo M (2022) Normalisations of Existential Rules: Not so Innocuous!. In: Kern-Isberner G, Lakemeyer G, Meyer T (eds) Proceedings of the 19th International Conference on Principles of Knowledge Representation and Reasoning, KR 2022, Haifa, Israel, July 31 - August 5, 2022

Gerlach L, Carral D (2023a) General Acyclicity and Cyclicity Notions for the Disjunctive Skolem Chase. In: Williams B, Chen Y, Neville J (eds) Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial

References

Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023. AAAI Press, pp 6372–6379

Gerlach L, Carral D (2023b) Do Repeat Yourself: Understanding Sufficient Conditions for Restricted Chase Non-Termination. In: Marquis P, Son TC, Kern-Isberner G (eds) Proceedings of the 20th International Conference on Principles of Knowledge Representation and Reasoning, KR 2023, Rhodes, Greece, September 2-8, 2023. pp 301–310

Gerlach L, Carral D, Hecher M (2024) Finite Groundings for ASP with Functions: A Journey through Consistency. In: Proceedings of the Thirty-

References

Third International Joint Conference on Artificial Intelligence, IJCAI 2024, Jeju, South Korea, August 3-9, 2024. ijcai.org, pp 3386–3394

Gogacz T, Marcinkowski J, Pieris A (2023) Uniform Restricted Chase Termination. SIAM J Comput 52:641–683. https://doi.org/10.1137/20M1377035

Grahne G, Onet A (2018) Anatomy of the Chase. Fundam Informaticae 157:221–270. https://doi.org/10.3233/FI-2018-1627

References

Grau BC, Horrocks I, Krötzsch M, et al (2013) Acyclicity Notions for Existential Rules and Their Application to Query Answering in Ontologies. J Artif Intell Res 47:741–808. https://doi.org/10.1613/JAIR.3949

Harel D (1986) Effective transformations on infinite trees, with applications to high undecidability, dominoes, and fairness. J ACM 33:224–248. https://doi.org/10.1145/4904.4993

Ivliev A, Gerlach L, Meusel S, et al (2024) Nemo: Your Friendly and Versatile Rule Reasoning Toolkit. In: Marquis P, Ortiz M, Pagnucco M (eds) Proceedings of the 21st International Conference on Principles of

References

Knowledge Representation and Reasoning, KR 2024, Hanoi, Vietnam. November 2-8, 2024