Restricted Chase Termination

You Want More than Fairness

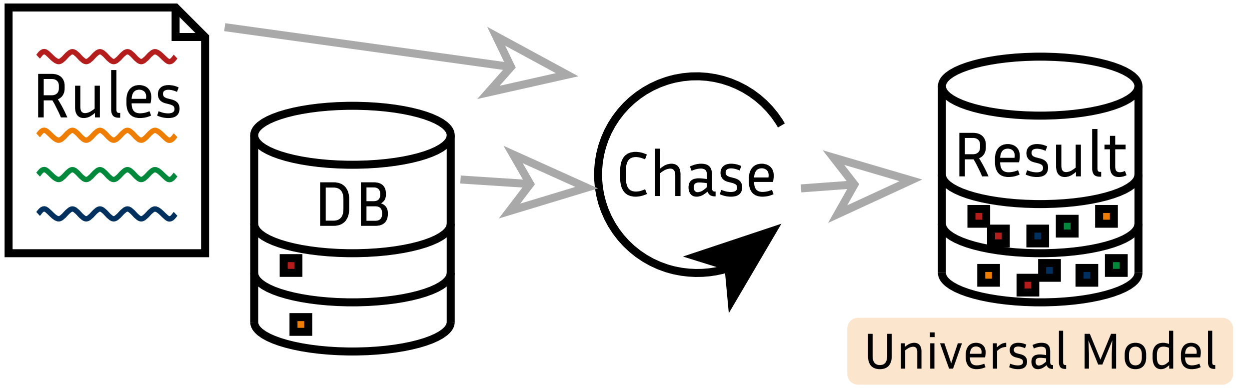
David Carral[1] Lukas Gerlach[2] Lucas Larroque[3] Michaël Thomazo[3]

[1]LIRMM, Inria, University of Montpellier, CNRS, France
[2]Knowledge-Based Systems Group, TU Dresden, Germany
[3]Inria, DI ENS, ENS, CNRS, PSL University, France

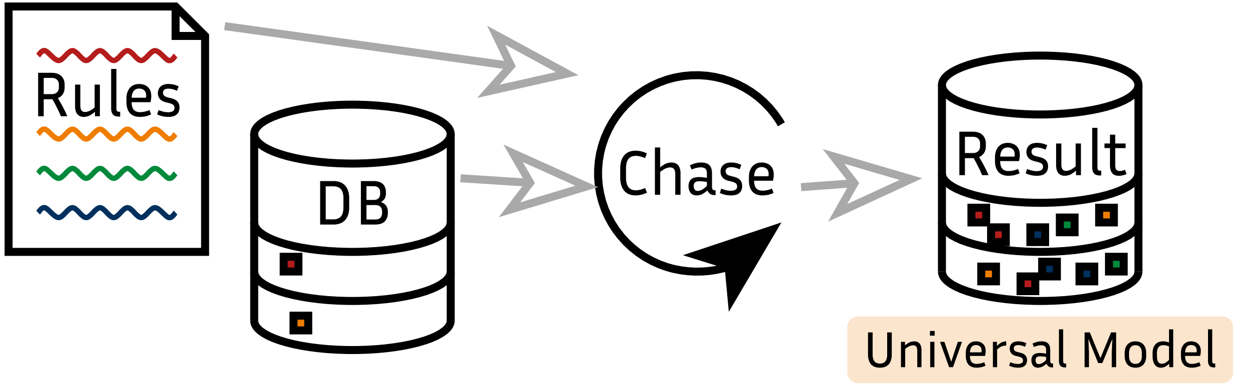
25.06.2025

TECHNISCHE UNIVERSITÄT DRESDEN International Center for Computational Logic Inria

Chase Crash Course



Chase Crash Course

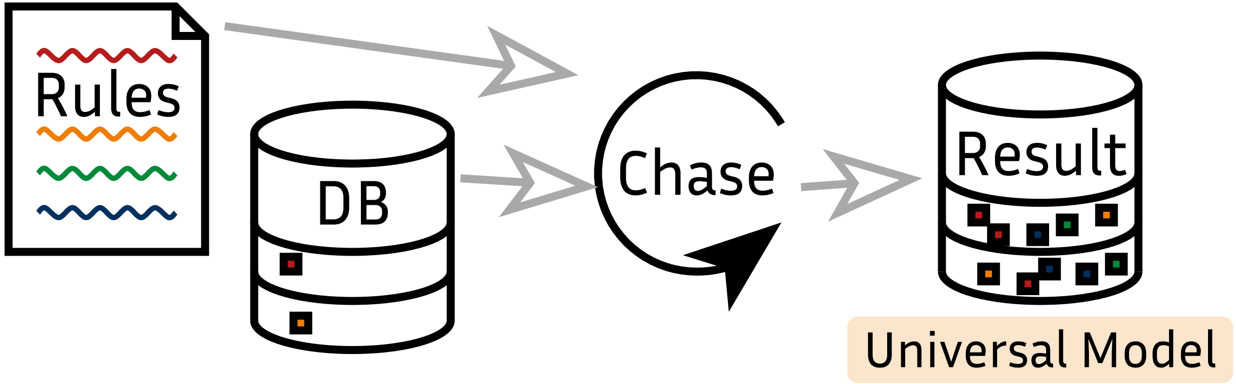


$$A(x) \rightarrow \exists z.\ R(x, z) \land B(z)$$

$$B(x) \rightarrow A(x)$$

$$R(x, y) \rightarrow R(y, x)$$

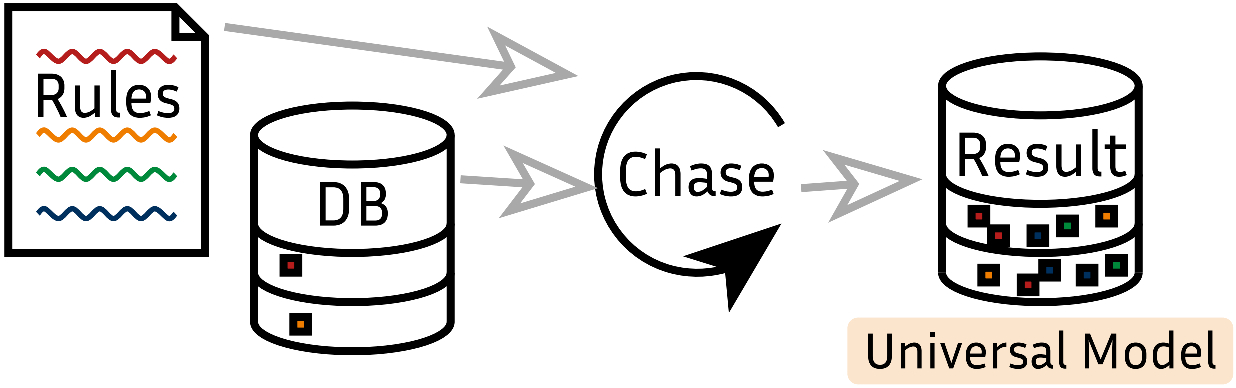
Chase Crash Course



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

DB: $A(c)$, $B(c)$

Chase Crash Course

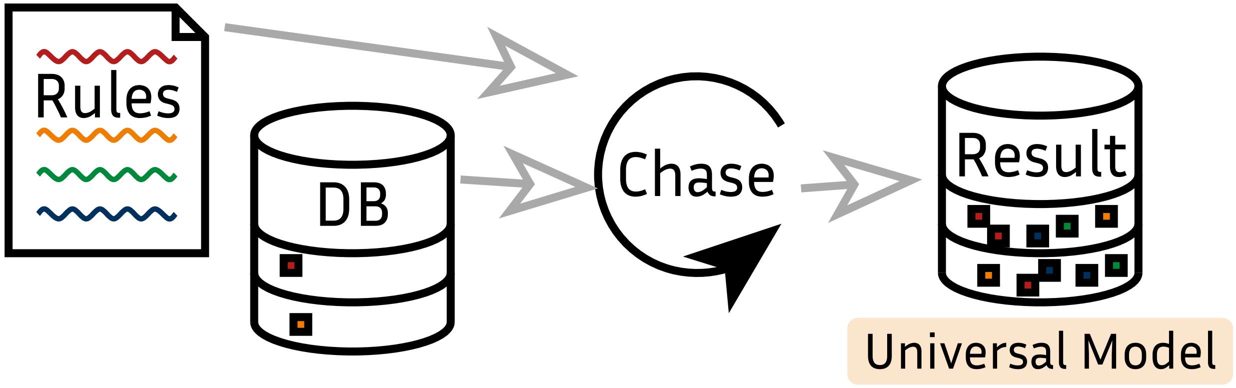


$$A(x) \rightarrow \exists z.\ R(x, z) \land B(z)$$
$$B(x) \rightarrow A(x)$$
$$R(x, y) \rightarrow R(y, x)$$

DB: $A(c)$, $B(c)$

Step 1: $R(c, n_1)$, $B(n_1)$

Chase Crash Course



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

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

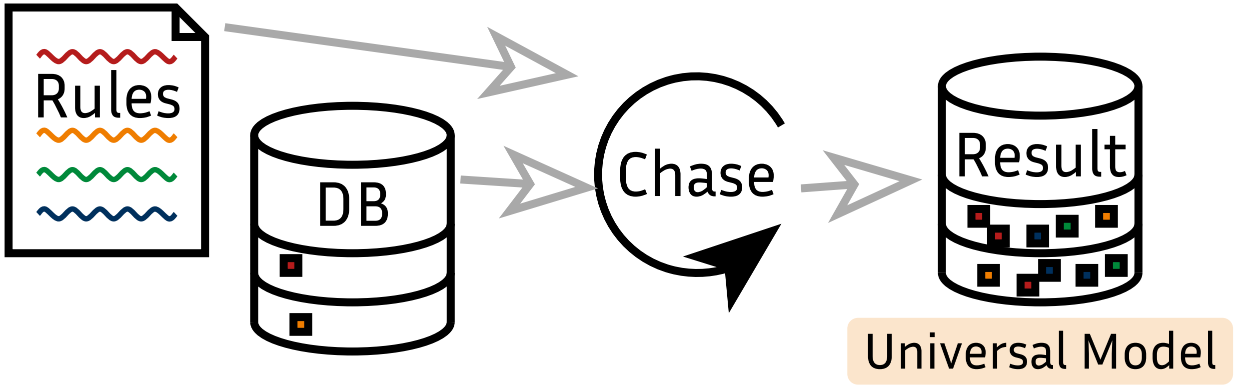
$$\mathrm{R}(x, y) \rightarrow \mathrm{R}(y, x)$$

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

Step 1: $\mathrm{R}(c, n_1)$, $\mathrm{B}(n_1)$

Step 2: $\mathrm{A}(n_1)$; Step 3: $\mathrm{R}(n_1, c)$

Chase Crash Course



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

DB: $A(c)$, $B(c)$
Step 1: $R(c, n_1)$, $B(n_1)$
Step 2: $A(n_1)$; Step 3: $R(n_1, c)$

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

Chase Termination is Undecidable

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

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

Step 1-2: $\mathrm{R}(c, n_1)$, $\mathrm{B}(n_1)$, $\mathrm{A}(n_1)$

Chase Termination is Undecidable

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

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

Step 1-2: $\mathrm{R}(c, n_1)$, $\mathrm{B}(n_1)$, $\mathrm{A}(n_1)$

Step 3: $\mathrm{R}(n_1, n_2)$, $\mathrm{B}(n_2)$

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 1-2: $\mathrm{R}(c, n_1)$, $\mathrm{B}(n_1)$, $\mathrm{A}(n_1)$

Step 3: $\mathrm{R}(n_1, n_2)$, $\mathrm{B}(n_2)$

Step N: $\mathrm{R}(n_1, c)$ (due to Fairness)

Chase Termination is Undecidable

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

DB: $A(c)$, $B(c)$
Step 1-2: $R(c, n_1)$, $B(n_1)$, $A(n_1)$
Step 3: $R(n_1, n_2)$, $B(n_2)$
Step N: $R(n_1, c)$ (due to Fairness)

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

Chase Termination is Undecidable

$A(x) \to \exists z.\, R(x, z) \wedge B(z)$

$B(x) \to A(x)$

$R(x, y) \to R(y, x)$

DB: $A(c)$, $B(c)$

Step 1-2: $R(c, n_1)$, $B(n_1)$, $A(n_1)$

Step 3: $R(n_1, n_2)$, $B(n_2)$

Step N: $R(n_1, c)$ (due to Fairness)

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

$\mathrm{CTK}_{\forall}^{r}$ 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

$A(x) \to \exists z.\, R(x, z) \wedge B(z)$

$B(x) \to A(x)$

$R(x, y) \to R(y, x)$

DB: $A(c)$, $B(c)$

Step 1-2: $R(c, n_1)$, $B(n_1)$, $A(n_1)$

Step 3: $R(n_1, n_2)$, $B(n_2)$

Step N: $R(n_1, c)$ (due to Fairness)

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)

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

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

Overview from (Grahne and Onet 2018)

$\mathrm{CTX}_{\mathrm{Q}}^{r}$	$\exists$	$\forall$
K	Σ_1^0-complete	Σ_1^0-complete
R	Π_2^0-complete	Π_2^0-complete

Σ_1^0 - Semi-Decidable Languages (e.g. Halting Problem)

Π_2^0 - Co-Semi-Decidable with Semi-Decision Oracle (e.g. Universal Halting Problem)

Chase Termination is Undecidable (2)

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

Overview from (Grahne and Onet 2018)

$\mathrm{CTX}_{\mathrm{Q}}^{r}$	$\exists$	$\forall$
K	Σ_1^0-complete	Σ_1^0-complete
R	Π_2^0-complete	Π_2^0-complete

Membership: Run all Chase Sequences in Parallel

Hardness: TM can be simulated with Existential Rules

Chase Termination is Undecidable (2)

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

Overview from (Grahne and Onet 2018)

$\mathrm{CTX}_{\mathrm{Q}}^{r}$	$\exists$	$\forall$
K	Σ_1^0-complete	Σ_1^0-complete
R	Π_2^0-complete	Π_2^0-complete

Membership: Use $\mathrm{CTK}_{\exists}^{r}$ 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)

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

Overview from (Grahne and Onet 2018)

$\mathrm{CTX}_{\mathrm{Q}}^{r}$	$\exists$	$\forall$
K	Σ_1^0-complete	Σ_1^0-complete
R	Π_2^0-complete	Π_2^0-complete

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

This does not work because of *fairness*!

Chase Termination is Undecidable (2)

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

Overview from (Grahne and Onet 2018)

$\mathrm{CTX}_{\mathrm{Q}}^{r}$	$\exists$	$\forall$
K	Σ_1^0-complete	Π_2^0-hard
R	Π_2^0-complete	Π_2^0-hard

(Carral et al. 2022) have shown $\mathrm{CTK}_{\forall}^{r}$ to be at least Π_2^0-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 inf. unfair sequence while all fair sequences are finite.

Fairness Complicates Things

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

There could be an inf. unfair sequence while all fair sequences are finite.

Example from (Gogacz et al. 2023)

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

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

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

Fairness Complicates Things

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

There could be an inf. unfair sequence while all fair sequences are finite.

Example from (Gogacz et al. 2023)

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

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

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

We can derive a chain of $\mathrm{R}(n_1, b, b)$, $\mathrm{R}(n_2, b, b)$, ... 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 inf. unfair sequence while all fair sequences are finite.

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 $\mathrm{R}(n_1, b, b)$, $\mathrm{R}(n_2, b, b)$, ... 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 inf. unfair sequence while all fair sequences are finite.

Example from (Gogacz et al. 2023)

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

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

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

We can derive a chain of $\mathrm{R}(n_1, b, b)$, $\mathrm{R}(n_2, b, b)$, ... 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”.

Fairness for NTMs - based on (Harel 1986)

Decide if a given non-deterministic Turing machine admits a run on ε that visits a designated state q_r infinitely often. (Harel 1986)

Fairness for NTMs - based on (Harel 1986)

Decide if a given non-deterministic Turing machine admits a run on w that visits a designated state q_r recurringly after finitely many steps.

If this is the case, we say that the NTM is recurring through q_r on w.

Fairness for NTMs - based on (Harel 1986)

Decide if a given non-deterministic Turing machine admits a run on w that visits a designated state q_r recurringly after finitely many steps.

If this is the case, we say that the NTM is recurring through q_r on w.

This problem is Σ_1^1-complete - first analytical hierarchy level - beyond infinitely many Turing jumps. (Harel 1986)

Back to $\mathrm{CTK}_{\forall}^{r}$: Π_1^1 Membership

Decide if a given non-deterministic Turing machine admits a run on w that visits a designated state q_r recurringly after finitely many steps.

Back to $\mathrm{CTK}_{\forall}^{r}$: Π_1^1 Membership

Decide if a given non-deterministic Turing machine admits a run on w that visits a designated state q_r recurringly after finitely many steps.

Proof Idea: Describe NTM that computes a chase sequence for a KB.

-
-

Back to $\mathrm{CTK}_{\forall}^{r}$: Π_1^1 Membership

Decide if a given non-deterministic Turing machine admits a run on w that visits a designated state q_r recurringly after finitely many steps.

Proof Idea: Describe NTM that computes a chase sequence for a KB.

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

Back to $\mathrm{CTK}_{\forall}^{r}$: Π_1^1 Membership

Decide if a given non-deterministic Turing machine admits a run on w that visits a designated state q_r recurringly after finitely many steps.

Proof Idea: Describe NTM that computes a chase sequence for a KB.

- 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 $\mathrm{CTK}_\forall^r$: Π_1^1 Membership

Decide if a given non-deterministic Turing machine admits a run on w that visits a designated state q_r recurringly after finitely many steps.

Proof Idea: Describe NTM that computes a chase sequence for a KB.

- 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.
Hence: Complement of $\mathrm{CTK}_\forall^r$ is in Σ_1^1; therefore $\mathrm{CTK}_\forall^r$ is in Π_1^1.

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

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

Can we simulate the non-recurrence problem through q_r on ε with rules?

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

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

Can we simulate the non-recurrence problem through q_r on ε with rules?

Goal: Construct KB that is in $\mathrm{CTK}_{\forall}^{r}$ iff NTM has this property.

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

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

Can we simulate the non-recurrence problem through q_r on ε with rules?

Goal: Construct KB that is in $\mathrm{CTK}_{\forall}^{r}$ iff NTM has this property.

Issue: NTM has infinite runs that visit the desig. state only finitely often but we need to ensure that the chase terminates in such cases.

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

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

Can we simulate the non-recurrence problem through q_r on ε with rules?

Goal: Construct KB that is in $\mathrm{CTK}_{\forall}^{r}$ iff NTM has this property.

Issue: NTM has infinite runs that visit the desig. state only finitely often but we need to ensure that the chase terminates in such cases.

Idea: 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 using universal non-recurrence.

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

Membership: Similar to $\mathrm{CTK}^r_\forall$ case using universal non-recurrence.

Hardness: Tricky even with robust non-recurrence since ill-shaped databases may not correspond to TM configurations...

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

Membership: Similar to $\mathrm{CTK}_{\forall}^{r}$ case using universal non-recurrence.

Hardness: Tricky even with robust non-recurrence since ill-shaped databases may 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 using universal non-recurrence.

Hardness: Tricky even with robust non-recurrence since ill-shaped databases may 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

Summing up...

Fairness makes Chase Termination highly Undecidable.

Stricter, finitely verifiable conditions could solve this (e.g. breadth-first).

$\mathrm{CTX}^r_\mathrm{Q}$	$\exists$	$\forall$
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.

Stricter, finitely verifiable conditions could solve this (e.g. breadth-first).

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

Some Open Problems:

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

Summing up...

Fairness makes Chase Termination highly Undecidable.

Stricter, finitely verifiable conditions could solve this (e.g. breadth-first).

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

Some Open Problems:

- Disjunctive Skolem Chase Termination complete for Π^1_1 **?** (Likely true.)
- What about single-head rules (in the $\mathrm{CTR}^r_\forall$ case)?

Summing up...

Fairness makes Chase Termination highly Undecidable.

Stricter, finitely verifiable conditions could solve this (e.g. breadth-first).

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

Some Open Problems:

- Disjunctive Skolem Chase Termination complete for Π^1_1 **?** (Likely true.)
- What about single-head rules (in the $\mathrm{CTR}^r_{\forall}$ case)?

📢 *Talk to me about Lean and Typst* 🤓

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

Krötzsch M, Marx M, Rudolph S (2019) The Power of the Terminating Chase (Invited Talk). In: Barceló P, Calautti M (eds) 22nd International Conference on Database Theory, ICDT 2019, March 26-28, 2019, Lisbon, Portugal. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, pp 1–17