Restricted Chase Termination: You Want More than Fairness (Extended Abstract)

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

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

Abstract

The chase is a fundamental algorithm with ubiquitous uses in database theory. Given a database and a set of existential rules (aka tuple-generating dependencies), it iteratively extends the database to ensure that the rules are satisfied in a most general way. This process may not terminate, and a major problem is to decide whether it does. This problem has been studied for many chase variants, which differ by the conditions under which a rule is applied to extend the database. Surprisingly, the complexity of the universal termination of the restricted (aka standard) chase is not fully understood. We close this gap by placing universal restricted chase termination in the analytical hierarchy. This higher hardness is due to the fairness condition, and we propose an alternative to reduce the hardness of universal termination.

Keywords

Existential Rules, Tuple-Generating Dependencies, Restricted Chase

1. The Problem of Restricted Chase Termination and Fairness

In this extended abstract of [1], we summarise our findings and outline the problem of fairness for restricted chase termination. We give an overview of the levels of undecidability for various chase termination problems including our novel results that we obtain for $\mathsf{CTK}_{\forall}^{rest}$ and $\mathsf{CTR}_{\forall}^{rest}$.

The chase is a bottom-up materialisation procedure that computes a universal model (a model that can be homomorphically embedded into all other models) for a knowledge base (KB), consisting of an (existential) rule set and a database.

Example 1. *Consider the KB $\mathcal{K}_1 = \langle \Sigma, D \rangle$ where D is the database $\{\mathtt{Bicycle}(b)\}$ and Σ contains:*

$$\forall x.\mathtt{Bicycle}(x) \rightarrow \exists y.\mathtt{HasPart}(x,y) \wedge \mathtt{Wheel}(y) \qquad \forall x,y.\mathtt{HasPart}(x,y) \rightarrow \mathtt{IsPartOf}(y,x)$$
$$\forall x.\mathtt{Wheel}(x) \rightarrow \exists y.\mathtt{IsPartOf}(x,y) \wedge \mathtt{Bicycle}(y) \qquad \forall x,y.\mathtt{IsPartOf}(x,y) \rightarrow \mathtt{HasPart}(y,x)$$

Then, $\{\mathtt{Bicycle}(b), \mathtt{HasPart}(b,t), \mathtt{IsPartOf}(t,b), \mathtt{Wheel}(t)\}$ *is a universal model of* $\mathcal{K}$.

DL 2025: 38th International Workshop on Description Logics, September 3–6, 2025, Opole, Poland
david.carral@inria.fr (D. Carral); lukas.gerlach@tu-dresden.de (L. Gerlach); lucas.larroque@inria.fr (L. Larroque); michael.thomazo@inria.fr (M. Thomazo)
0000-0001-7287-4709 (D. Carral); 0000-0003-4566-0224 (L. Gerlach); 0009-0007-2351-2681 (L. Larroque); 0000-0002-1437-6389 (M. Thomazo)
© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
CEUR Workshop Proceedings (CEUR-WS.org)

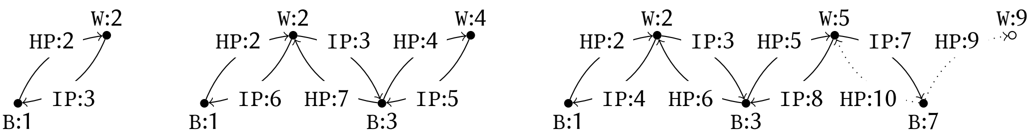

Figure 1: Three Different Restricted Chase Sequences for the KB $\mathscr{K}_1$ from Example 1

Although there are many variants of the chase, they all implement a similar strategy. Namely, they start with the database and then, in a step-by-step manner, extend this structure with new atoms to satisfy the rules in the input rule set in a most general way. Since none of these variants are guaranteed to terminate (some KBs do not even admit finite universal models), it is only natural to wonder about their respective halting problems [2, 3, 4, 5, 6, 7]. Despite intensive efforts, some results have remained open. Specifically, prior research has established tight bounds for the major classes of chase terminating KBs and rule sets, except for the following:

- The class $\mathsf{CTK}_\forall^{rest}$ of all KBs that only admit finite restricted chase sequences.
- The class $\mathsf{CTR}_\forall^{rest}$ containing a rule set Σ if $\langle \Sigma, D\rangle \in \mathsf{CTK}_\forall^{rest}$ for every database D.

Our main contribution is to show that both classes are Π_1^1-complete, a surprising result given that these are significantly harder than the corresponding classes for other chase variants [7].

The restricted chase differs from other variants in that it introduces new terms to satisfy existential quantifiers in rules only if these are not already satisfied by existing terms. Because of this, the order of rule applications impacts termination. The KB $\mathscr{K}_1$ from Example 1 admits both finite and infinite restricted chase sequences; some of these are represented in Fig. 1, where atoms are numbered to denote the step at which they were introduced.

$\mathsf{CTK}_\forall^{rest}$ has been claimed to be recursively enumerable (RE) in [7], probably with the following procedure in mind: given an input KB, compute all of its restricted chase sequences in parallel, and halt and accept if all of them are finite. Alas, this strategy does not work as there are terminating input KBs that admit infinitely many finite sequences that are of increasing length.

Example 2. *Consider the KB $\mathscr{K}_2 = \langle \Sigma, D\rangle$ where D is the database $\{\mathtt{Real}(a), \mathtt{E}(a,c), \mathtt{E}(c,b), \mathtt{Real}(c), \mathtt{E}(b,b), \mathtt{Brake}(b)\}$ and Σ is the rule set that contains all of the following:*

$$\forall x,y,z.\mathtt{Real}(x) \wedge \mathtt{E}(x,y) \wedge \mathtt{Real}(y) \wedge \mathtt{Brake}(z) \to \exists v.\mathtt{E}(y,v) \wedge \mathtt{E}(v,z) \wedge \mathtt{Real}(v)$$
$$\forall x.\mathtt{Brake}(x) \to \mathtt{Real}(x)$$

For any $k \geq 1$, there is a restricted chase sequence of $\mathscr{K}_2$ that yields the (finite) universal model $D \cup \{\mathtt{E}(c,t_1)\} \cup \{\mathtt{E}(t_i,t_{i+1}) \mid i < k\} \cup \{\mathtt{E}(t_i,b), \mathtt{Real}(t_i) \mid i \leq k\} \cup \{\mathtt{Real}(b)\}$ of $\mathscr{K}_2$. Such a sequence is obtained by applying the first rule k consecutive times and then applying the second one once to derive $\mathtt{Real}(b)$. After this application, the first rule is satisfied and the restricted chase halts.

The KB $\mathscr{K}_2$ in the previous example is in $\mathsf{CTK}_\forall^{rest}$ because of *fairness*. This is a built-in condition in the definition of all chase variants that guarantees that the chase yields a model of the KB by requiring that, if a rule is applicable at some point during the computation of a sequence, then this rule must be eventually satisfied. Hence, the second rule in $\mathscr{K}_2$ must sooner or later be applied in all restricted chase sequences and thus, all such sequences are finite.

	KB		Rule Set	
	Sometimes	**Always**	**Sometimes**	**Always**
Oblivious	RE-complete [6]		RE-complete [3, 5]	
Restricted	RE-complete [6]	Π^1_1-complete	Π^0_2-complete [7]	Π^1_1-complete
Core	RE-complete [6]		Π^0_2-complete [7]	

Table 1
Undecidability status of the main decision problems related to chase termination; the results presented without citations refer to our main contributions [1].

An issue with fairness is that it is not *finitely verifiable*, i.e. we cannot see if fairness is violated after a finite number of steps, intuitively because a necessary rule application might still occur later. With a stronger condition, e.g., demanding that possible rule applications are performed in a breadth-first manner, we could detect violations after a finite number of steps. In fact, in [1, Section 6], we show that such a condition lands $\mathsf{CTK}^{rest}_{\forall}$ in RE by computing chase sequences in parallel as sketched above.

The KB in Example 2 uses a technique called the *emergency brake*, initially proposed by Krötzsch et al. in [8]. The idea is to connect every term in the chase to a special term (the constant b in this example) that is not "`Real`" and acts as a "`Brake`". Eventually, this term becomes "`Real`" because of fairness, all existential restrictions are satisfied, and the restricted chase halts. The emergency brake allows to grow the chase for an arbitrary number of steps whilst guaranteeing its termination. By activating infinite sequences of emergency brakes, we emulate the eternal recurrence often displayed by Π^1_1-complete problems and thus define the reductions that lead to our main results.

2. Summary of Levels of Undecidability for Chase Termination

All decision problems related to chase termination are undecidable. However, these are complete for different classes within the arithmetical and analytical hierarchies, as summarised in Table 1.

For the oblivious chase, application order is irrelevant, therefore $\mathsf{CTK}^{obl}_{\exists} = \mathsf{CTK}^{obl}_{\forall}$ and $\mathsf{CTR}^{obl}_{\exists} = \mathsf{CTR}^{obl}_{\forall}$. For the core chase, by Deutsch et al., $\mathsf{CTK}^{core}_{\exists} = \mathsf{CTK}^{core}_{\forall}$ and $\mathsf{CTR}^{core}_{\exists} = \mathsf{CTR}^{core}_{\forall}$. To understand why $\mathsf{CTK}^{obl}_{\exists}$ (resp. $\mathsf{CTK}^{rest}_{\exists}$ or $\mathsf{CTK}^{core}_{\exists}$) is recursively enumerable (RE), consider the following procedure: given some input KB, compute all of its oblivious (resp. restricted or core) chase sequences in parallel and accept as soon as you find a finite one. Deutsch et al. proved that $\mathsf{CTK}^{rest}_{\exists}$ is RE-hard. More precisely, they defined a reduction that takes a machine M as input and produces a KB $\mathcal{K}$ as output such that M halts on the empty word if and only $\mathcal{K}$ is in $\mathsf{CTK}^{rest}_{\exists}$; see Theorem 1 in [6].

Deutsch et al. also proved that $\mathsf{CTK}^{core}_{\exists}$ is RE-hard. More precisely, they showed that checking if a KB admits a universal model is undecidable; see Theorem 6 in [6]. Moreover, they proved that the core chase is a procedure that halts and yields a finite universal model for an input KB if this theory admits one; see Theorem 7 of the same paper. Therefore, the core chase can be applied as a semi-decision procedure for checking if a KB admits a finite universal model.

Marnette proved that $\mathsf{CTR}^{obl}_{\exists}$ is in RE. More precisely, he showed that a rule set Σ is in $\mathsf{CTR}^{obl}_{\exists}$

if and only if the KB $\langle \Sigma, D^\star_\Sigma \rangle$ is in $\mathsf{CTK}^{obl}_\exists$ where $D^\star_\Sigma = \{\mathrm{P}(\star, \dots, \star) \mid \mathrm{P} \in \mathsf{Preds}(\Sigma)\}$ is the *critical instance* and $\star$ is a special fresh constant; see Theorem 2 in [5].

Gogacz and Marcinkowski proved that $\mathsf{CTR}^{obl}_\exists$ is RE-hard. More precisely, they presented a reduction that takes a 3-counter machine M as input and produces a rule set Σ such that M halts on ε if and only if $\langle \Sigma, D^\star_\Sigma \rangle$ is in $\mathsf{CTK}^{obl}_\exists$; see Lemma 6 in [3]. Hence, M halts on the ε and only if Σ is in $\mathsf{CTR}^{obl}_\exists$ by Theorem 2 in [5]. Furthermore, Bednarczyk et al. showed that this hardness result holds even over single-head binary rule sets; see Theorem 1.1 in [2].

$\mathsf{CTR}^{rest}_\exists$ is in Π^0_2, since we can give semi-decision procedure when equipped with an oracle for $\mathsf{CTK}^{rest}_\exists$ by iterating over all databases. The argument for $\mathsf{CTR}^{core}_\exists$ being in Π^0_2 is analogous. Grahne and Onet proved that $\mathsf{CTR}^{rest}_\exists$ and $\mathsf{CTR}^{core}_\exists$ are Π^0_2-hard by reducing from the universal termination problem of word rewriting systems.

Our Contribution In [1, Section 4], we argue that $\mathsf{CTK}^{rest}_\forall$ is Π^1_1-complete. This contradicts Theorem 5.1 in [7], which states that $\mathsf{CTK}^{rest}_\forall$ is RE-complete. Specifically, it is claimed that this theorem follows from results in [6], but the authors of that paper only demonstrate that $\mathsf{CTK}^{rest}_\forall$ is undecidable without proving that it is in RE. Before our completeness result, the tightest lower bound was proven by Carral et al., who proved that this class is Π^0_2-hard; see Proposition 42 in [4]. We obtain Π^1_1-completeness by reduction to and from the following complete problem based on [9]: Decide if a given non-deterministic Turing machine (NTM) M is *non-recurring through* q_r on some word $\vec{w}$, i.e., if every non-deterministic run of M on $\vec{w}$ features q_r only finitely often. The high level intuition is that recurrence of q_r resembles the fairness condition from the chase. For membership, we compute the chase with a NTM keeping track of possible rule applications for each step i in R_i. We keep a counter j and visit q_r whenever R_j is satisfied and increase j in this case. For hardness, we construct a rule set based on a given NTM and enforce termination with the emergency brake technique except in cases where q_r is visited recurringly by always creating a new brake when q_r is visited.

We also show in [1, Section 5] that $\mathsf{CTR}^{rest}_\forall$ is Π^1_1-complete using similar reductions as for the $\mathsf{CTK}^{rest}_\forall$ case. This contradicts Theorem 5.16 in [7], where it is stated that this class is Π^0_2-complete. The error in the upper-bound of this theorem arose from the assumption that $\mathsf{CTK}^{rest}_\forall$ is in RE, which, as previously discussed, is not the case. Regarding the lower bound, they consider an extended version of this class of rule sets where they allow the inclusion of a single "denial constraint"; i.e. an implication with an empty head that halts the chase if the body is satisfied during the computation of a chase sequence. They prove that the always restricted halting problem for rule sets is Π^0_2-hard if one such constraint is allowed. Our results imply that we do not need to consider such an extension to obtain a higher lower bound.

Acknowledgments

On TU Dresden side, this work is partly supported by DFG in project 389792660 (TRR 248, Center for Perspicuous Systems), by the BMBF in the Center for Scalable Data Analytics and Artificial Intelligence (project SCADS25B, ScaDS.AI), and by BMBF and DAAD in project 57616814 (SECAI, School of Embedded and Composite AI).

References

[1] D. Carral, L. Gerlach, L. Larroque, M. Thomazo, Restricted chase termination: You want more than fairness, Proc. ACM Manag. Data 3 (2025). doi:10.1145/3725246.

[2] B. Bednarczyk, R. Ferens, P. Ostropolski-Nalewaja, All-instances oblivious chase termination is undecidable for single-head binary tgds, in: C. Bessiere (Ed.), Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020, ijcai.org, 2020, pp. 1719–1725. URL: https://doi.org/10.24963/ijcai.2020/238. doi:10.24963/IJCAI.2020/238.

[3] T. Gogacz, J. Marcinkowski, All-instances termination of chase is undecidable, in: J. Esparza, P. Fraigniaud, T. Husfeldt, E. Koutsoupias (Eds.), Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part II, volume 8573 of *Lecture Notes in Computer Science*, Springer, 2014, pp. 293–304. URL: https://doi.org/10.1007/978-3-662-43951-7_25. doi:10.1007/978-3-662-43951-7_25.

[4] D. Carral, L. Larroque, M. Mugnier, M. Thomazo, Normalisations of existential rules: Not so innocuous!, in: G. Kern-Isberner, G. Lakemeyer, T. Meyer (Eds.), Proceedings of the 19th International Conference on Principles of Knowledge Representation and Reasoning, KR 2022, Haifa, Israel, July 31 - August 5, 2022, 2022. URL: https://proceedings.kr.org/2022/11/.

[5] B. Marnette, Generalized schema-mappings: from termination to tractability, in: J. Paredaens, J. Su (Eds.), Proceedings of the Twenty-Eigth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2009, June 19 - July 1, 2009, Providence, Rhode Island, USA, ACM, 2009, pp. 13–22. URL: https://doi.org/10.1145/1559795.1559799. doi:10.1145/1559795.1559799.

[6] A. Deutsch, A. Nash, J. B. Remmel, The chase revisited, in: M. Lenzerini, D. Lembo (Eds.), Proceedings of the Twenty-Seventh ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2008, June 9-11, 2008, Vancouver, BC, Canada, ACM, 2008, pp. 149–158. URL: https://doi.org/10.1145/1376916.1376938. doi:10.1145/1376916.1376938.

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

[8] M. Krötzsch, M. Marx, S. Rudolph, The power of the terminating chase (invited talk), in: P. Barceló, M. Calautti (Eds.), 22nd International Conference on Database Theory, ICDT 2019, March 26-28, 2019, Lisbon, Portugal, volume 127 of *LIPIcs*, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019, pp. 3:1–3:17. URL: https://doi.org/10.4230/LIPIcs.ICDT.2019.3. doi:10.4230/LIPICS.ICDT.2019.3.

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