Answering Queries with Negation over Existential Rules

Stefan Ellmauthaler, Markus Krötzsch, Stephan Mennicke

Knowledge-Based Systems Group, TU Dresden, Dresden, Germany
stefan.ellmauthaler@tu-dresden.de, markus.kroetzsch@tu-dresden.de, stephan.mennicke@tu-dresden.de

Abstract

Ontology-based query answering with existential rules is well understood and implemented for positive queries, in particular conjunctive queries. For queries with negation, however, there is no agreed-upon semantics or standard implementation. This problem is unknown for simpler rule languages, such as Datalog, where it is intuitive and practical to evaluate negative queries over the least model. This fails for existential rules, which instead of a single least model have multiple *universal models* that may not lead to the same results for negative queries. We therefore propose *universal core models* as a basis for a meaningful (non-monotonic) semantics for queries with negation. Since cores are hard to compute, we identify syntactic conditions (on rules and queries) under which our core-based semantics can equivalently be obtained for other universal models, such as those produced by practical chase algorithms. Finally, we use our findings to propose a semantics for a broad class of existential rules with negation.

Introduction

Existential rules are a prominent approach in knowledge representation, due to theoretical and practical advances in ontology-based query answering (Baget et al. 2009; Calì, Gottlob, and Lukasiewicz 2009), but also because of applications in many other domains, such as data exchange and integration (Fagin et al. 2005). Answering conjunctive queries (CQs) over sets of existential rules is often the main goal, where a CQ is entailed if it is satisfied by all models of the given rules. This is often implemented using *universal models*, which is a class of models each representing all positive query answers, such that one such model suffices to answer CQs (Deutsch, Nash, and Remmel 2008).

However, for queries that ask for the absence of facts (i.e., queries incorporating negated atoms), universal models cannot be used because different universal models may yield different query answers. This is a major problem, since negation is an important feature in real-world queries, and a semantic prerequisite for supporting negation in rule bodies, which can be viewed as a recursive generalisation of negative queries. The severity of this limitation is further aggravated by the fact that no such problems exist for Datalog, where negative queries can safely be evaluated over the unique least model. This can be further generalised by considering *stratified negation* in rule bodies, which yields an intuitive and implementable non-monotonic semantics (Abiteboul, Hull, and Vianu 1994). But even this basic form of negation is not safe to use with existential rules.

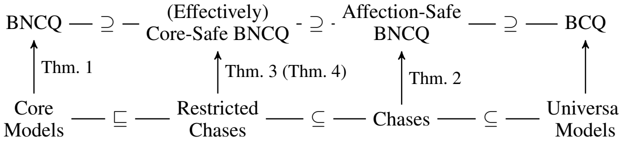

Figure 1: Summary of Results

What makes this problem challenging is that we are looking for a non-monotonic form of negation, where the absence of positive evidence is sufficient to entail negative information. Indeed, under the classical first-order semantics of negation, no negated queries are ever entailed by a set of existential rules. However, it is not immediate when a non-monotonic semantics should allow for additional, non-classical consequences, and several distinct approaches have been proposed for existential rules (Magka, Krötzsch, and Horrocks 2013; Baget et al. 2014; Gottlob et al. 2014; Alviano, Morak, and Pieris 2017; Krötzsch 2020). Our goal for this paper is to provide a semantics that agrees with the widely used and generally accepted semantics of Datalog with stratified negation for rules without existentials, but which also respects the classical reading of existential quantifiers as mere statements of existence of certain elements without any commitment to their identity (in contrast to logic programming approaches that use function terms for referring to distinguished objects).

We therefore propose to evaluate negations with respect to universal models that are *cores*, an algebraic property that, intuitively speaking, ensures that they contain no redundant structures. Many good practical and theoretical results have been obtained for core models (Fagin, Kolaitis, and Popa 2005; Deutsch, Nash, and Remmel 2008; Carral et al. 2018; Krötzsch 2020), but computing cores from arbitrary structures is expensive (Hell and Nešetřil 1992).

We therefore ask if our core-based semantics for negation can also be computed using other kinds of universal models, which are easier to compute in practice. It turns out that this is possible if we restrict the shape of the queries that we want to answer, and we define several classes of "safe" queries that can be evaluated over broader classes of models. Our results are summarised in Figure 1, which illustrates the inverse relationship between the generality of the query language and the specificity of the models that one can use to compute them. The (restricted) *chase* refers to a widely implemented type of reasoning algorithm, such that the respective query classes could be evaluated in practice (Nenov et al. 2015; Benedikt et al. 2017; Bellomarini, Sallinger, and Gottlob 2018; Carral et al. 2019). The symbol $\sqsubseteq$ expresses that core models can be embedded into every restricted chase, although no such chase is necessarily equal to the core (hence no $\subseteq$). For any of the given query fragments, we provide concrete syntactic definitions that can be decided in practice.

Our final contribution is the extension of our approach to existential rules with negation in their bodies. This is technically more challenging than mere query answering, and even the eager computation of cores during reasoning cannot guarantee a unique semantics (Krötzsch 2020). We focus on cases where rules can be stratified in a certain sense, but our conditions significantly generalise classical stratification (Abiteboul, Hull, and Vianu 1994) and the recent *full stratification* for existential rules (Krötzsch 2020). Nevertheless, we can still find a unique "perfect" model that can be used in query answering, making our approach a valid generalisation of the *perfect core model semantics* (Krötzsch 2020).

All proofs are included in the technical report of the paper (Ellmauthaler, Krötzsch, and Mennicke 2021).

Preliminaries

We consider a first-order signature with disjoint sets of *constants* $\mathbf{C}$, *(labelled) nulls* $\mathbf{N}$, *variables* $\mathbf{V}$, and *predicates* $\mathbf{P}$. A *term* t is an element of $\mathbf{C} \cup \mathbf{N} \cup \mathbf{V}$. Lists of terms are denoted $\boldsymbol{t} = t_1, t_2, \ldots, t_n$ with $n = |\boldsymbol{t}|$, and treated as sets when order is irrelevant. Each predicate $p \in \mathbf{P}$ has an *arity* $\mathsf{ar}(p) \in \mathbb{N}$. An *atom* is an expression $p(\boldsymbol{t})$ with $p \in \mathbf{P}$ and $\mathsf{ar}(p) = |\boldsymbol{t}|$. An atom $p(\boldsymbol{t})$ is *ground* if $\boldsymbol{t} \subseteq \mathbf{C}$. An *interpretation* $\mathcal{I}$ is a set of atoms without variables. A database $\mathcal{D}$ is a finite set of ground atoms.

Rules and Queries An *(existential) rule* r is a formula

$$r = \forall \boldsymbol{x}, \boldsymbol{y}.\ \varphi[\boldsymbol{x}, \boldsymbol{y}] \to \exists \boldsymbol{z}.\ \psi[\boldsymbol{y}, \boldsymbol{z}], \tag{1}$$

where φ and ψ are conjunctions of atoms using only terms from $\mathbf{C}$ or from the mutually disjoint lists of variables $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \subseteq \mathbf{V}$. We call φ the *body* (denoted $\mathsf{body}(r)$), ψ the *head* (denoted $\mathsf{head}(r)$), and $\boldsymbol{y}$ the *frontier* of r. For ease of notation we may treat conjunctions of atoms as sets, and we omit universal quantifiers in rules. We require that all variables in $\boldsymbol{y}$ actually occur in φ (*safety*).[1] A rule is *Datalog* if it has no existential quantifiers.

A *normal Boolean conjunctive query* (BNCQ) is a formula $q = \exists \boldsymbol{x}.\varphi \wedge \psi$, where φ is a conjunction of atoms with variables from $\boldsymbol{x}$, and ψ is a conjunction of negated atoms $\mathbf{not}\, p(\boldsymbol{t})$ using only variables that occur in φ (*safety*). We write q^+ (q^-) for the set of all non-negated atoms in φ (ψ). If $q^- = \emptyset$, then q is a *boolean conjunctive query* (BCQ).

Models and Entailment Given a set of atoms $\mathcal{A}$ and an interpretation $\mathcal{I}$, a *homomorphism* $h : \mathcal{A} \to \mathcal{I}$ is a function that maps the terms occurring in $\mathcal{A}$ to (the variable-free) terms occurring in $\mathcal{I}$, such that: (i) for all $c \in \mathbf{C}$, $h(c) = c$; (ii) for all $p \in \mathbf{P}$, $p(\boldsymbol{t}) \in \mathcal{A}$ only if $p(h(\boldsymbol{t})) \in \mathcal{I}$, where $h(\boldsymbol{t})$ is the list of h-images of the terms $\boldsymbol{t}$. If h satisfies (ii) with "only if" strengthened to "iff", h is a *strong homomorphism*. An *embedding* is an injective strong homomorphism, and an *isomorphism* is a bijective strong homomorphism (i.e., surjective embedding). We apply homomorphisms to a formula by applying them individually to all of its atoms.

A *match* of a rule r in an interpretation $\mathcal{I}$ is a homomorphism $h : \mathsf{body}(r) \to \mathcal{I}$. A match h of r in $\mathcal{I}$ is *satisfied* if there is a homomorphism $h' : \mathsf{head}(r) \to \mathcal{I}$, such that $h \subseteq h'$. We call h' an *extension of* h. Rule r is *satisfied* by $\mathcal{I}$, written $\mathcal{I} \models r$, if every match of r in $\mathcal{I}$ is satisfied. A set of rules Σ is satisfied by $\mathcal{I}$, written $\mathcal{I} \models \Sigma$, if $\mathcal{I} \models r$ for all $r \in \Sigma$. We call $\mathcal{I}$ a *model* of Σ and database $\mathcal{D}$ if $\mathcal{I} \models \Sigma$ and $\mathcal{D} \subseteq \mathcal{I}$. A BNCQ q is *satisfied* by $\mathcal{I}$, written $\mathcal{I} \models q$, if there is a homomorphism $h : q^+ \to \mathcal{I}$ with $h(q^-) \cap \mathcal{I} = \emptyset$. Note, a BNCQ q with $q^+ \cap q^- \neq \emptyset$ can never be satisfied. Herein we consider only BNCQs q that are *non-trivial* in this sense.

Universal Models and the Chase A model $\mathcal{U}$ of $\mathcal{D}$ and Σ is *universal* if there is a homomorphism $\mathcal{U} \to \mathcal{M}$ for every model $\mathcal{M}$ of $\mathcal{D}$ and Σ. Universal models can be computed with *the chase* (Deutsch, Nash, and Remmel 2008). In this paper, we will mainly deal with the *restricted chase* (also known as *standard chase*).

Definition 1. *Let $\mathcal{D}$ be a database and Σ a set of rules. A sequence $\mathcal{D}^0, \mathcal{D}^1, \ldots$ is called a* (restricted) chase sequence of $\mathcal{D}$ and Σ *iff*

1. *$\mathcal{D}^0 = \mathcal{D}$,*
2. *for every $\mathcal{D}^{i+1}$ ($i \geq 0$) there is a rule $r \in \Sigma$ of the form* (1) *and a match h for r in $\mathcal{D}^i$ with*
   (a) *h is an unsatisfied match for r in $\mathcal{D}^i$ and*
   (b) *there is an extension $h^\star : \mathsf{head}(r) \to \mathcal{D}^{i+1}$ of h, such that $h^\star(z)$ is a fresh null for each $z \in \boldsymbol{z}$, and*
3. *if h is a match for some rule $r \in \Sigma$ in $\mathcal{D}^i$ ($i \geq 0$), then h is satisfied in some $\mathcal{D}^j$ with $j \geq i$ (fairness).*

r is applicable in $\mathcal{D}^i$ *if it has an unsatisfied match in $\mathcal{D}^i$. Chase sequence $\mathcal{D}^0, \mathcal{D}^1, \ldots, \mathcal{D}^k$ is called* terminating *if $\mathcal{D}^k$ is a model of Σ and $\mathcal{D}$. We call $\mathcal{D}^\infty := \bigcup_{i \geq 0} \mathcal{D}^i$ a* chase of Σ and $\mathcal{D}$.

Other chase variants, like the Skolem/semi-oblivious or the oblivious chase, mainly differ in the respective definitions of rule applicability (2. (a) in Definition 1).

Influence of Existential Quantifiers For some of the upcoming notions, it is useful to have rule sets Σ where each rule makes use of a distinct set of variables from all the other rules. We say that Σ is *renamed-apart*. Every rule set can be rewritten to an equivalent renamed-apart rule set.

The *positions* in a predicate $p \in \mathbf{P}$ are the pairs $\langle p, 1\rangle, \ldots, \langle p, \mathsf{ar}(p)\rangle$, and we will refer to terms *at* a certain position in an atom, set of atoms, or other formula.

[1] This requirement can be relaxed, but it simplifies presentation.

$\mathsf{Var}(X)$ ($\mathsf{Var}_\exists(X)$/$\mathsf{Var}_\forall(X)$) denotes the set of all (existentially quantified/universally quantified) variables in a rule or rule set X. We reproduce the following definition from Krötzsch and Rudolph (2011) to obtain a structure allowing for syntactic reasoning of the influence of existential quantifiers within the chase.

Definition 2. *Let Σ be a renamed-apart rule set. For $x \in \mathsf{Var}(\Sigma)$, let Π_x^B (Π_x^H) be the sets of all positions at which x occurs in the body (head) of a (necessarily unique) rule in Σ. If $x \in \mathsf{Var}_\exists(\Sigma)$, then Ω_x is the smallest set of positions such that (1) $\Pi_x^H \subseteq \Omega_x$ and (2) for all $y \in \mathsf{Var}_\forall(\Sigma)$, $\Pi_y^B \subseteq \Omega_x$ implies $\Pi_y^H \subseteq \Omega_x$. The set of* jointly affected *positions is $\bigcup_{x\in\mathsf{Var}_\exists(\Sigma)} \Omega_x$. For $x, y \in \mathsf{Var}_\exists(\Sigma)$, we write $x \rightsquigarrow y$ if the (unique) rule of y has a frontier variable z with $\Pi_z^B \subseteq \Omega_x$.*

The relation $x \rightsquigarrow y$ states that nulls created for x potentially enable rule applications that create nulls for y. Of particular interest is the transitive and reflexive closure $\rightsquigarrow^*$. Starting from a set of existentially quantified variables, we seek all positions at which a null might occur that is directly or indirectly influenced by a variable in this set.

Definition 3. *Let Σ be a renamed-apart rule set and let $V \subseteq \mathsf{Var}_\exists(\Sigma)$. A position π is* V-influenced *if there are $x \in V$ and y with $x \rightsquigarrow^* y$ such that $\pi \in \Omega_y$.*

Note that *jointly affected* is the same as $\mathsf{Var}_\exists(\Sigma)$-influenced. Virtually any chase variant (restricted, Skolem, and oblivious) is *compatible* with jointly affected positions, meaning that nulls in chases $\mathcal{D}^\infty$ only occur in $\mathsf{Var}_\exists(\Sigma)$-influenced positions.

Proposition 1. *For renamed-apart rule set Σ and database $\mathcal{D}$ with (Skolem/oblivious/restricted) chase $\mathcal{D}^\infty$, if $p(t_1, \ldots, t_{\mathsf{ar}(p)}) \in \mathcal{D}^\infty$ and $t_i \in \mathbf{N}$ ($1 \leq i \leq \mathsf{ar}(p)$), then $\langle p, i\rangle$ is $\mathsf{Var}_\exists(\Sigma)$-influenced.*

The proof uses an over-approximation of the chase procedure (Ellmauthaler, Krötzsch, and Mennicke 2021).

Answering BNCQs on Core Models

A BCQ is entailed by rule set Σ and database $\mathcal{D}$ if it is satisfied by all models of Σ and $\mathcal{D}$. Moreover, every universal model of Σ and $\mathcal{D}$ satisfies exactly the BCQs that are entailed in this sense. Neither condition is appropriate to define entailment of BNCQs that may use negation. On the one hand, no such query is satisfied in all models since there is always a model where every BCQ is true. On the other hand, different universal models do not satisfy the same BNCQs.

Example 1. *Take for instance the database $\mathcal{D} = \{a(1,2), b(2,2)\}$ together with an empty rule set. Of course, $\mathcal{D}$ is a universal model, but so is $\mathcal{U} := \mathcal{D} \cup \{a(1,n)\}$ (where $n \in \mathbf{N}$). As mentioned above, all BCQs evaluate with the same result on $\mathcal{D}$ or $\mathcal{U}$. However, for BNCQ $q = \exists x, y.\ a(x,y) \wedge \neg b(y,y)$ we get $\mathcal{U} \models q$ while $\mathcal{D} \not\models q$.*

Models that are *cores* have been suggested as the appropriate model for defining the semantics of queries that may depend on negative information (Deutsch, Nash, and Remmel 2008; Baget et al. 2014). This suggestion is substantiated by the fact that cores are unique up to isomorphism and that isomorphic structures cannot be distinguished by first-order (FO) queries (see, e.g., the *Isomorphism Lemma* by Ebbinghaus, Flum, and Thomas (1994)). Indeed, if a universal model is a core, then it satisfies fewer BNCQs than any other model (see also Theorem 1 below), resulting in the most "cautious" notion of non-monotonic entailment.

Definition 4. *A finite interpretation $\mathcal{I}$ is a* core *if every homomorphism $h : \mathcal{I} \to \mathcal{I}$ is an isomorphism. A* core model *of a rule set Σ and a database $\mathcal{D}$ is a finite universal model of Σ and $\mathcal{D}$ that is a core.*

If Σ and $\mathcal{D}$ have any finite universal model, then they admit a finite core, which is unique up to isomorphism (Deutsch, Nash, and Remmel 2008). The situation is more complicated on infinite structures, which admit several non-equivalent definitions of core (Bauslaugh 1995) and where core models might fail to exist altogether (Carral et al. 2018). We therefore focus on cases with finite models $\mathcal{M}$ and their unique cores, which we denote $\mathsf{core}(\mathcal{M})$. Many conditions have been studied to recognise cases where finite universal models are guaranteed to exist (Cuenca Grau et al. 2013). In the rest of the paper we study the notion of *core entailments* for BNCQs, being those entailments we obtain when evaluating the query on *the* core model.

Definition 5. *A rule set Σ and database $\mathcal{D}$* core-entail *a BNCQ q, written $\Sigma, \mathcal{D} \models_c q$, if Σ and $\mathcal{D}$ have a core model $\mathcal{C}$ satisfying q.*

Important Assumption Throughout this paper, we only consider pairs of rule sets Σ and databases $\mathcal{D}$ that have a (finite) core model. Definition 5 does not apply to other cases.

Reconsidering Example 1, $\mathcal{D}$ is not just any universal model, but the core model. Hence, $\Sigma, \mathcal{D} \not\models_c q$ for the query q in the example, as one might intuitively expect. As shown in Example 1, such non-entailments are not preserved in arbitrary (universal) models, but it turns out that entailments are. This is a consequence of the fact that core models are embedded in all universal models in the following sense.

Lemma 1. *Let Σ be a rule set and $\mathcal{D}$ a database with core model $\mathcal{C}$ and arbitrary universal model $\mathcal{U}$. (1) If $h : \mathcal{C} \to \mathcal{U}$ is a homomorphism, it is an embedding. We call $h(\mathcal{C})$ the* core instance *of $\mathcal{U}$ and denote it by $\mathcal{U}_c$. (2) Every homomorphism $h : \mathcal{U}_c \to \mathcal{U}_c$ is an isomorphism.*

Proof. On (1), since $\mathcal{U}$ is a universal model and $\mathcal{C}$ a (core) model of Σ and $\mathcal{D}$, there is a homomorphism $h' : \mathcal{U} \to \mathcal{C}$. Note, $h' \circ h : \mathcal{C} \to \mathcal{C}$ is an isomorphism as $\mathcal{C}$ is a core. We show that h is an embedding, i.e., h is injective and strong. For injectivity, let u and t be distinct terms occurring in $\mathcal{C}$. If $h(u) = h(t)$, it follows that $h' \circ h(u) = h' \circ h(t)$, which contradicts the assumption that $h' \circ h$ is an isomorphism. For showing that h is strong, let $\boldsymbol{u}$ be a list of terms in $\mathcal{C}$ and $p(h(\boldsymbol{u})) \in \mathcal{U}$. We need to show that $p(\boldsymbol{u}) \in \mathcal{C}$. As $h' \circ h$ is an isomorphism on $\mathcal{C}$, its inverse is also an isomorphism. Hence, $p(h' \circ h(\boldsymbol{u})) \in \mathcal{C}$ and $p((h' \circ h)^{-1} \circ (h' \circ h)(\boldsymbol{u})) \in \mathcal{C}$ allow for the conclusion $p(\boldsymbol{u}) \in \mathcal{C}$.

Towards (2), there is an isomorphism $i : \mathcal{C} \to \mathcal{U}_c$ by the argumentation above. Let $h : \mathcal{U}_c \to \mathcal{U}_c$ be a homomorphism.

Define $j := i^{-1} \circ (h \circ i)$ (i.e., $j : \mathcal{C} \to \mathcal{C}$). Since $\mathcal{C}$ is a core, j is an isomorphism. Hence, h must be an isomorphism. □

Based on the core instance, we can now show that core entailments of BNCQs carry over to all universal models.

Theorem 1. *For every rule set Σ, database $\mathcal{D}$, and BNCQ q, if $\Sigma, \mathcal{D} \models_c q$, then $\mathcal{U} \models q$ for all universal models $\mathcal{U}$ of Σ and $\mathcal{D}$.*

Proof. Let $\mathcal{C}$ be a core model of Σ and $\mathcal{D}$. The core instance $\mathcal{U}_c \subseteq \mathcal{U}$ preserves all FO-queries, including BNCQs. We conclude that for all BNCQ q, $\mathcal{C} \models q$ implies $\mathcal{U} \models q$. □

In fact, up to minor variations, the core model is the only model with this property: the only other models for which BNCQ satisfaction is preserved in all universal models are those that can be *covered by embeddings* from the core model (i.e., all facts occur in the image of some embedding from the core). However, it turns out that *some* BNCQs allow us to consider broader classes of models for checking core entailment, as we will see in the following sections.

Affection-Safe BNCQs

In spite of its appealing semantics, core entailment has the practical disadvantage that core models are difficult to compute. This difficulty seems unavoidable if we want minimality in the sense of Theorem 1. However, this does not mean that core entailment of BNCQs does always require us to compute the core model. For example, BNCQs without negation can equivalently be answered in any universal model. We show that similar results can also be obtained for more interesting classes of BNCQs, and how these classes can be effectively recognised.

This requires us to look at more specific classes of models. Indeed, we can readily see that entailment over *all* universal models does often not coincide with core entailment since core-non-entailments are not preserved across universal models: every BNCQ q whose positive part is entailed is also true in some universal model, unless its negative part is a logical consequence (in this case, the non-entailment of q is a first-order logical consequence). This can be shown by adding redundant structures where q is satisfied, as in Example 1, but taking arbitrary rule sets into account.

Proposition 2. *Consider a rule set Σ and database $\mathcal{D}$. For every BNCQ q such that (a) $\Sigma, \mathcal{D} \models q^+$ and (b) $\Sigma, \mathcal{D} \not\models q^+ \wedge a(\boldsymbol{t})$ for all $a(\boldsymbol{t}) \in q^-$, there is a universal model $\mathcal{U}$ of Σ and $\mathcal{D}$ with $\mathcal{U} \models q$.*

Proof. Define $\mathcal{D}^+ := \mathcal{D} \cup \{p(\nu(\boldsymbol{u})) \mid p(\boldsymbol{u}) \in q^+\}$ with $\nu(t) = t$ for all $t \in \mathbf{C}$ and $\nu(t)$ a fresh null for all $t \in \mathbf{V}$. Now let $\mathcal{U}$ be a chase over Σ and $\mathcal{D}^+$ (e.g., some restricted chase). By construction, $\mathcal{U}$ is a model of Σ and $\mathcal{D}$, and ν is a match $\nu : q^+ \to \mathcal{U}$. By safety, ν is defined for $\mathsf{Var}(q^-)$.

Now for any $\alpha = a(\boldsymbol{t}) \in q^-$, let $\mathcal{U}_\alpha$ be a universal model of Σ and $\mathcal{D}$ such that $\mathcal{U}_\alpha \not\models q^+ \wedge \alpha$, which exists due to (b). Due to (a), there is a match $\nu' : q^+ \to \mathcal{U}_\alpha$ such that $\nu'(\alpha) \notin \mathcal{U}_\alpha$ (using (b)). Since, moreover, $\mathcal{D} \subseteq \mathcal{U}_\alpha$, we find that there is a homomorphism $h : \mathcal{D}^+ \to \nu'(q^+) \cup \mathcal{D}$. By soundness of the chase, h extends to a homomorphism $h' : \mathcal{U} \to \mathcal{U}_\alpha$. Since $\mathcal{U}_\alpha$ is universal, this shows that $\mathcal{U}$ is universal. By construction, $h'(\nu(\alpha)) = \nu'(\alpha) \notin \mathcal{U}_\alpha$, and hence $\nu(\alpha) \notin \mathcal{U}$. Applying the same reasoning to all $\alpha \in q^-$, we find that ν shows $\mathcal{U} \models q$. □

Hence, only limited classes of BNCQs can be evaluated over arbitrary universal models. This, however, is due to some universal models containing unmotivated clutter that users might intuitively not expect. In particular, the universal models that are computed by chase procedures are not of this form, and can only produce nulls in jointly affected positions (Proposition 1). Therefore, if all variables in q^- are bound to some position in q^+ that is not jointly affected, then these variables can only match constants.

Definition 6. *Let Σ be a rule set. A BNCQ q is* affection-safe *w.r.t. Σ if every variable in q^- occurs at a position in q^+ that is not jointly affected.*

Since all common (e.g., restricted, Skolem, oblivious) chases are compatible with joint affection, we can use their outputs to answer affection-safe BNCQs.

Theorem 2. *For every rule set Σ, database $\mathcal{D}$, and affection-safe BNCQ q, we have $\Sigma, \mathcal{D} \models_c q$ iff $\mathcal{M} \models q$ for some[2] chase $\mathcal{M}$ computed for Σ and $\mathcal{D}$ by a chase procedure compatible with jointly affected positions (Proposition 1).*

Proof. The "only if" direction follows from Theorem 1. For the "if" direction, let $\mathcal{M}$ be as in the claim. By Proposition 1, terms t at positions $\langle a, i \rangle$ in $\mathcal{M}$ that are not jointly affected are constants. Let $\mathcal{C}$ be the core model of Σ and $\mathcal{D}$. By universality, there are homomorphisms $h_1 : \mathcal{M} \to \mathcal{C}$ and $h_2 : \mathcal{C} \to \mathcal{M}$. Suppose $\mathcal{M} \models q$ by a match $h : q^+ \to \mathcal{M}$ with $h(q^-) \cap \mathcal{M} = \emptyset$, but $\Sigma, \mathcal{D} \not\models_c q$. Since $h_1 \circ h(q^+) \subseteq \mathcal{C}$, this means $h_1 \circ h(q^-) \neq \emptyset$. Hence there is $p(\boldsymbol{t}) \in q^-$ with $p(h_1 \circ h(\boldsymbol{t})) \in \mathcal{C} \setminus \mathcal{M}$, and therefore $p(h_2 \circ h_1 \circ h(\boldsymbol{t})) \in \mathcal{M}$. As $h(\boldsymbol{t})$ contains only constants, this implies $p(h(\boldsymbol{t})) \in \mathcal{M}$, contradicting our assumption that $\mathcal{M} \models q$. □

Core-Safe BNCQs

Affection safety ensures that chase-based models only entail negative BNCQ atoms if they match null-free facts, which always agree across all universal models. We now relax this requirement based on the recent notion of *restraints* (Krötzsch 2020), which allow us to identify a larger set of "safe" positions to bind to. The next example, adopted from Alviano, Morak, and Pieris (2017), illustrates the problem:

Example 2. *Take as database $\mathcal{D} = \{p(A), f(B,A)\}$ (mnemonics: **p**arent, **f**ather-of, **e**qual) and rules*

$$f(x,y) \to e(x,x) \tag{2}$$

$$p(x) \to \exists y.\, f(y,x) \wedge e(y,y) \tag{3}$$

We can obtain two restricted chases on $\Sigma = \{(2),(3)\}$ and $\mathcal{D}$: $\mathcal{U}_1 = \mathcal{D} \cup \{e(B,B)\}$ and $\mathcal{U}_2 = \mathcal{D} \cup \{f(n,A), e(n,n), e(B,B)\}$. For $\mathcal{U}_1$, we only apply (2) *for match $h_1 = \{x \mapsto B, y \mapsto A\}$; then* (3) *is satisfied. For $\mathcal{U}_2$, we apply* (3) *for match $h_2 = \{x \mapsto A\}$, and then rule* (2) *for*

[2] The restricted chase, e.g., can yield distinct chase results depending on rule application order. Any of them can be used here.

match h_1. The BNCQ $q = \exists x_1, x_2, y.\ f(x_1, y) \wedge f(x_2, y) \wedge \neg e(x_1, x_2)$ is such that $\mathcal{U}_1 \not\models q$ and $\mathcal{U}_2 \models q$. Since $\mathcal{U}_1$ is the core model of Σ and $\mathcal{D}$, we obtain $\Sigma, \mathcal{D} \not\models_c q$.

Arguably, the construction of $\mathcal{U}_2$ should not have applied (3) with the extended match $h_2^\star = \{x \mapsto \mathrm{A}, y \mapsto n\}$, because $\mathcal{U}_2$ eventually does not need n to satisfy (3). Indeed, $h_2^+ = \{x \mapsto \mathrm{A}, y \mapsto \mathrm{B}\}$ would be another way to extend h_2 to satisfy (3). Functions that remap heads of prior rule applications to alternative structures in a chase are called *alternative matches*, and they can be shown to occur whenever a restricted chase fails to produce a core (Krötzsch 2020).

Definition 7. *Let $\mathcal{I}_a \subseteq \mathcal{I}_b$ be interpretations such that $\mathcal{I}_a$ is obtained from applying rule r for extended match $h^\star$. A homomorphism $h' : h^\star(\mathsf{head}(r)) \to \mathcal{I}_b$ is an* alternative match *of h if*

- *$h'(t) = t$ for all terms t in $h^\star(\mathsf{body}(r))$, and*
- *there is a null n in $h^\star(\mathsf{head}(r))$ that does not occur in $h'(h^\star(\mathsf{head}(r)))$.*

The occurrence of alternative matches in a restricted chase is associated with structures that do not occur in the core (Krötzsch 2020). Such structures could lead to additional BNCQ matches. Since alternative matches involve nulls, affection safety can mitigate this by forcing variables in negative atoms to match constants. However, it is more general and still safe if we merely restrict to elements that are not directly or indirectly related to potential alternative matches. The existence of alternative matches in a real chase is undecidable, but we can (over)approximate such matches by considering chase-like interactions of pairs of rules. Based on this approach of defining *restraints* between rules (Krötzsch 2020), we can identify existential variables that are at risk of producing redundant structures:

Definition 8. *A rule r_1* restrains *rule r_2, written $r_1 \prec^\square r_2$, if there are interpretations $\mathcal{I}_a \subseteq \mathcal{I}_b$ and a function h_2 where*

1. *$\mathcal{I}_b$ is obtained by applying r_1 for match h_1,*
2. *$\mathcal{I}_a$ is obtained by applying r_2 for match h_2,*
3. *h_2 has an alternative match h': $h_2^\star(\mathsf{head}(r_2)) \to \mathcal{I}_b$, and*
4. *h_2 has no alternative match $h_2^\star(\mathsf{head}(r_2)) \to \mathcal{I}_b \setminus h_1^\star(\mathsf{head}(r_1))$.*

In this situation, a variable $x \in \mathsf{Var}_\exists(r_2)$ is restrained *if $h_2^\star(x)$ does not occur in $h'(h_2^\star(\mathsf{head}(r_2)))$. We write R_Σ for the set of all restrained variables of a rule set Σ.*

Variables in R_Σ may produce nulls that are not represented in the core model. Moreover, such nulls may be involved in further rule applications that derive additional structures that deviate from the core. To find positions of elements that are certain to agree with the core, we therefore consider all positions that are influenced by restrained variables in the sense of Definition 3.

Definition 9. *Let Σ be a rule set. A position π is* core-safe *(w.r.t. Σ) if it is not R_Σ-influenced. A BNCQ q is* core-safe *w.r.t. Σ if every variable in q^- occurs at a core-safe position in q^+.*

Reconsidering query q of Example 2, we find that q is not core-safe w.r.t. the given rule set. Positions $\langle e, 1\rangle, \langle e, 2\rangle, \langle f, 1\rangle$ are not core-safe and variables x_1 and x_2 both occur at $\langle f, 1\rangle$ in q^+. This explains why this example admits restricted chase sequences on which the entailment of q disagrees with the core model. Indeed, for core-safe BNCQs, this problem can never occur:

Theorem 3. *For every rule set Σ, database $\mathcal{D}$, core-safe BNCQ q, and restricted chase $\mathcal{M}$ of Σ and $\mathcal{D}$, we find that $\Sigma, \mathcal{D} \models_c q$ iff $\mathcal{M} \models q$.*

The "only if" direction is again clear from Theorem 1. The proof for the "if" direction is given below. The key insight is that terms at core-safe positions in a chase $\mathcal{M}$ must belong to the core instance $\mathcal{M}_c$ of $\mathcal{M}$ (cf. Lemma 1). This is a similar situation as for affection-safety, where we showed that variables at affection-safe positions must be instantiated with constants, which therefore occur in the core.

Lemma 2. *For a restricted chase $\mathcal{M}$ of rule set Σ and database $\mathcal{D}$ (with finite core model), and core instance $\mathcal{M}_c$ of $\mathcal{M}$, if a term t occurs at a core-safe position in $\mathcal{M}$, then t occurs in $\mathcal{M}_c$.*

Proof Sketch. Let $\mathcal{C}$ be the core model of Σ and $\mathcal{D}$, $h_1 : \mathcal{M} \to \mathcal{C}$ a homomorphism (existence by $\mathcal{M}$ being universal), and $h_2 : \mathcal{C} \to \mathcal{M}_c$ the isomorphism for which h_2^{-1} agrees with h_1, i.e., $h_2^{-1}(t) = u$ if $h_1(t) = u$. The existence of an isomorphism i is ensured by Lemma 1. Furthermore, consider the restriction $h_1' : \mathcal{M}_c \to \mathcal{C}$ of h_1 to $\mathcal{M}_c$. By Lemma 1 (2), $i \circ h_1' : \mathcal{M}_c \to \mathcal{M}_c$ is an isomorphism. Set $h_2 := (h_1')^{-1}$, which is an isomorphism since $(i^{-1} \circ i \circ h_1') = h_1'$ is an isomorphism.

We subsequently analyse the homomorphism $h' := h_2 \circ h_1$. With respect to the choice of h_2, we show that if t occurs at a core-safe position in $\mathcal{M}$, then $h'(t) = t$. The rest of the proof is an induction on the chase sequence of $\mathcal{M}$ and is included in our technical report (Ellmauthaler, Krötzsch, and Mennicke 2021). □

Proof of Theorem 3. For the remaining "if" direction, suppose $\mathcal{M} \models q$. Then there is a homomorphism $h : q^+ \to \mathcal{M}$ with $h(q^-) \cap \mathcal{M} = \emptyset$. For core model $\mathcal{C}$ of Σ and $\mathcal{D}$ and core instance $\mathcal{M}_c$ of $\mathcal{M}$, let $h_1 : \mathcal{M} \to \mathcal{C}$ / $h_2 : \mathcal{C} \to \mathcal{M}_c$ be the respective homomorphism/isomorphism (by Lemma 1).

Let $r(\mathbf{t}) \in q^-$. We show that $r(h(\mathbf{t})) \notin \mathcal{M}$ implies $r(h_1(h(\mathbf{t}))) \notin \mathcal{C}$. Every variable in $\mathbf{t}$ occurs in at least one core-safe position in q^+ (by core-safety of q). Thus, every term in $\mathbf{u} = h(\mathbf{t})$ occurs in $\mathcal{M}_c$ by Lemma 2. Since $r(\mathbf{u}) \notin \mathcal{M}$ (and $r(\mathbf{u}) \notin \mathcal{M}_c$), $r(h_1(\mathbf{u})) \notin \mathcal{C}$ because, otherwise, $r(h_2(h_1(\mathbf{u}))) \in \mathcal{M}_c$, which contradicts $r(\mathbf{u}) \notin \mathcal{M}_c$ (therefore, $\notin \mathcal{M}$) because $h_2 \circ h_1$ is an isomorphism on $\mathcal{M}_c$ (by Lemma 1 (2)). □

Core-safe BNCQs therefore are a significant generalisation of affection-safe BNCQs, at the cost of requiring the use of the restricted chase – or any other correct chase procedure that ensures that its results are subsets of some restricted chase. In practice, this includes chase implementations that use specific strategies to decide the order in which rules are applied, e.g., by prioritising Datalog rules (which never introduce unnecessary nulls) (Urbani et al. 2018). Since Definition 8 identifies restrained variables under the assumption that rule r_1 might be applied before rule r_2, the fact that some rule application orders are generally impossible (or simply did not happen in a specific run) allows us to consider more variables *effectively core-safe*.

Definition 10. *Let Σ be a rule set, $\mathcal{D}$ a database, and $S = \mathcal{D}^0, \mathcal{D}^1, \mathcal{D}^2, \ldots$ a restricted chase sequence with $\mathcal{D}^{i-1} \xrightarrow{r_i, h_i} \mathcal{D}^i$ for $i \geq 1$, $r_i \in \Sigma$ and (unsatisfied) match h_i. A variable $x \in R_\Sigma$ is* effectively restrained in *S if there are $j < k$, such that x occurs in rule r_j and $r_k \prec^\square r_j$. The set of all effectively restrained variables in S is denoted R_S.*

A position $\langle a, i\rangle$ is effectively core-safe w.r.t. *S if it is not R_S-influenced. A BNCQ q is* effectively core-safe w.r.t. *S if every variable x in q^- occurs at an effectively core-safe position in q^+.*

Effective core safety marks even more positions as safe for BNCQs to query for.

Theorem 4. *For every rule set Σ, database $\mathcal{D}$, restricted chase sequence $S = \mathcal{D}^0, \mathcal{D}^1, \mathcal{D}^2, \ldots$ over Σ and $\mathcal{D}$ with chase $\mathcal{M} = \bigcup_{i\geq 0} \mathcal{D}^i$, and BNCQ q that is effectively core-safe for S, it holds that $\Sigma, \mathcal{D} \models_c q$ iff $\mathcal{M} \models q$.*

Proof sketch. Similar argumentation as for the proof of Theorem 3. In particular, terms at effectively core-safe positions do occur in the core instance $\mathcal{M}_c$ of $\mathcal{M}$ (cf. Lemma 2). □

Our results put BNCQ answering under core entailment semantics into reach for practical implementations. Indeed, chase procedures, including the restricted chase, are supported by efficient implementations, and the computation of restraints is of the same complexity as the application of a single rule, namely Σ_2^{P}-complete (Krötzsch 2020). This may at first seem harder than computing the core, which is known to be DP-complete (Fagin, Kolaitis, and Popa 2005), but the crucial difference is that the worst-case complexity of core computation refers to the size of the whole chase (often millions of facts), whereas for rule applications and restraint checking, it is about the size of a single rule (typically dozens of facts). This may explain why no general implementation of the core chase is available today.

Rules with Negation

We turn our attention to *normal rules*, which admit negated atoms in their bodies and can naturally be viewed as a recursive generalisation of BNCQ answering. Using our previous insights, we first define a chase-based semantics for such rules in cases where rule bodies are core-safe. We then generalise this by *stratifying* rule sets in a way that is compatible with restraints, generalising the notion of *full stratification* for normal existential rules (Krötzsch 2020). This yields a unique and well-defined semantics that we call *perfect core semantics*.

A *normal existential rule* is an expression $r = \forall \boldsymbol{x}, \boldsymbol{y}.\ \varphi[\boldsymbol{x}, \boldsymbol{y}] \wedge \chi[\boldsymbol{x}, \boldsymbol{y}] \rightarrow \exists \boldsymbol{z}.\ \psi[\boldsymbol{y}, \boldsymbol{z}]$, such that $\exists \boldsymbol{x}, \boldsymbol{y}.\ \varphi[\boldsymbol{x}, \boldsymbol{y}] \wedge \chi[\boldsymbol{x}, \boldsymbol{y}]$ is a BNCQ, and $\psi[\boldsymbol{y}, \boldsymbol{z}]$ is a conjunction of atoms. We use $\mathsf{body}^+(r)$ and $\mathsf{body}^-(r)$ for the sets of all atoms in φ and χ. A *match* of r in an interpretation $\mathcal{I}$ is a homomorphism $h : \mathsf{body}^+(r) \rightarrow \mathcal{I}$ with $h(\mathsf{body}^-(r)) \cap \mathcal{I} = \emptyset$. Other notions are as defined for rules without negation.

The restricted chase procedure of Definition 1 can then directly be applied to normal rules. This does not in general lead to a sound reasoning algorithm, since a rule might be applicable due to the absence of a negated atom that is inferred later on in the chase. Chase sequences where this does not happen have been called *generating* and can be used to define a kind of stable model semantics for normal existential rules (Baget et al. 2014).

A well-known approach to obtain generating chase sequences is *stratification*, where rules are partitioned into a sequence of sets ("strata") such that rules in higher strata cannot derive facts that occur negatively in rules of lower strata. For (normal) existential rules, stratifications have been defined using three relations between rules: *positive reliances* $r_1 \prec^+ r_2$ express that r_1 might derive facts that allow r_2 to be applied, *negative reliances* $r_1 \prec^- r_2$ express that r_1 might derive facts that prevent a possible application of r_2 since they occur in $\mathsf{body}^-(r_2)$, and *restraints* $r_1 \prec^\square r_2$ are as in Definition 8 with the additional condition that h_1 and h_2 are also matches (for normal rules) with respect to $\mathcal{I}_b$. Full formal definitions of these notions are given in our report (Ellmauthaler, Krötzsch, and Mennicke 2021).

Example 3. *Consider the rules (mnemonics:* ***p****arent,* ***f****ather-of,* ***m****ale,* ***c****hild-of,* ***a****dult,* ***o****lder-than-3,* ***t****ired)*

$$p(x) \rightarrow \exists v.\ f(x, v) \wedge m(v) \quad (4)$$

$$f(x, y) \rightarrow m(y) \wedge c(y, x) \quad (5)$$

$$f(x, y) \wedge \textbf{not}\, a(x) \wedge \textbf{not}\, o(x) \rightarrow t(y) \quad (6)$$

$$a(x) \rightarrow o(x) \quad (7)$$

We have (5) $\prec^\square$ (4)*,* (4) $\prec^+$ (5)*, and* (4) $\prec^+$ (6)*. There are no other relations, in particular* (7) $\not\prec^-$ (6) *since* (7) *is only applicable in cases where* (6) *is not applicable anyway.*

For a set of normal rules Σ, let R_Σ denote the set of restrained variables as of Definition 8, modified for normal rules as mentioned above. The set of R_Σ-influenced and core-safe positions is defined as before, ignoring negated atoms in rules. Then a rule $r \in \Sigma$ is *core-safe* if every variable in $\mathsf{body}^-(r)$ occurs on a core-safe position in $\mathsf{body}^+(r)$. A first simple observation highlights a case where the restricted chase can safely be applied to normal rules:

Proposition 3. *Let Σ be a set of normal rules such that (1) all rules in Σ are core-safe and (2) there is no negative reliance $r_1 \prec^- r_2$ between any rules $r_1, r_2 \in \Sigma$. Then every restricted chase sequence over Σ and any database $\mathcal{D}$ is generating and yields a model of Σ and $\mathcal{D}$. Moreover, all of these models are homomorphically equivalent.*

The full proof is included in our technical report (Ellmauthaler, Krötzsch, and Mennicke 2021).

Example 4. *Let Σ denote the set of rules in Example 3. We find that $R_\Sigma = \{v\}$ because of* (5) $\prec^\square$ (4)*, such that R_Σ-influenced positions are $\langle f, 2\rangle$, $\langle m, 1\rangle$, $\langle c, 1\rangle$, $\langle t, 1\rangle$. Hence,* (6) *is core-safe, and Proposition 3 applies.*

For a database $\mathcal{D} = \{p(A), f(A, B)\}$, we can obtain the restricted chases $\mathcal{U}_1 = \mathcal{D} \cup \{m(B), c(B, A), t(B)\}$

(by applying (5) *before* (4)*) and* $\mathcal{U}_2 = \mathcal{U}_1 \cup \{f(A,n), m(n), c(n,A), t(n)\}$ *(by applying* (4) *before* (5)*). Though distinct, they are homomorphically equivalent and* $\mathcal{U}_1$ *is their unique core.*

Example 4 also illustrates a case that is covered by Proposition 3 but is not in scope of the previously defined *full stratification*, which we recall and adapt next. As opposed to traditional stratifications, the definition of Krötzsch (2020) effectively allows some rules (esp. those that are not the target of any $\prec^-$ or $\prec^\Box$) to appear in multiple strata.

Definition 11. *For a set* Σ *of normal rules, a list* $\mathcal{S} = \langle \Sigma_1, \ldots, \Sigma_n \rangle$ *with* $\Sigma = \bigcup_{i=1}^{n} \Sigma_n$ *is a* quasi stratification *if, for all rules* $r_1 \in \Sigma_i$ *and* $r_2 \in \Sigma_j$*,*

1. *if* $r_1 \prec^+ r_2$ *then* $i \leq j$*,*
2. *if* $r_1 \prec^- r_2$ *then* $i < j$*,*
3. *if* $r_1 \prec^\Box r_2$ *then* $i \leq j$*.*

A quasi stratification $\mathcal{S}$ *is a* full stratification *if* $i < j$ *holds in case* (3)*; it is a* core-safe stratification *if all rules in* Σ_k *are core-safe for* Σ_k *for all* $k \in \{1, \ldots, n\}$*.*

Example 5. *The rules of Example 3 do not admit a full stratification due to the cycle* (5) $\prec^\Box$ (4) $\prec^+$ (5)*, but they can be a stratum in a core-safe stratification. We add further rules*

$$f(x,y) \to e(y,y) \tag{8}$$

$$f(x,y_1) \wedge f(x,y_2) \wedge \textbf{not}\, e(y_1,y_2) \to d(y_1,y_2) \tag{9}$$

Then (4) $\prec^+$ (8) *and* (8) $\prec^-$ (9)*. Rule* (9) *is not core-safe in the set of all rules. A possible core-safe stratification is* $\mathcal{S} = \langle \{(4),(5),(6),(7)\}, \{(8)\}, \{(9)\} \rangle$*.*

Full stratifications have been used to define the *perfect core model*, as the unique model obtained by conducting a (necessarily generating) chase that proceeds stratum by stratum. Core-safe stratification is strictly more general, since the stricter condition $i < j$ in case (3) implies that R_{Σ_k} is empty for every stratum Σ_k, so that its rules are core-safe. A suitable chase procedure for rule sets that are core-safe stratified is given next.

Definition 12. *Let* Σ *be a normal rule set with core-safe stratification* $\mathcal{S} = \langle \Sigma_1, \ldots, \Sigma_n \rangle$*, and let* $\mathcal{D}$ *be a database. The* core-safe chase sequence *for* $\mathcal{S}$ *and* $\mathcal{D}$ *is a list* $\mathcal{C}^0, \mathcal{C}^1, \ldots, \mathcal{C}^n$ *such that*

- $\mathcal{C}^0 = \mathcal{D}$*, and*
- *for every* $i \in \{1, \ldots, n\}$*,* $\mathcal{C}^i$ *is the core of a restricted chase over* Σ_i *and* $\mathcal{C}^{i-1}$*, provided that this core is finite.*

If such a sequence exists, $\mathcal{C}^n$ *is called the* core-safe chase *of* Σ *and* $\mathcal{D}$ *w.r.t.* $\mathcal{S}$*, and we denote it by* $\mathcal{S}(\mathcal{D})$*.*

Note that each stratum Σ_i satisfies the conditions of Proposition 3 since $\mathcal{S}$ is a core-safe stratification. Hence, the required restricted chase exists and, since it is finite, has a unique core. In practice, one can ensure the necessary finiteness by using acyclicity conditions that guarantee chase termination (Cuenca Grau et al. 2013).

Example 6. *Consider the stratification* $\mathcal{S}$ *of Example 5 and the database* $\mathcal{D} = \{p(A), f(A,B)\}$*. The core* $\mathcal{C}^1$ *of the restricted chase over the first stratum was computed as* $\mathcal{U}_1$ *in Example 4.* $\mathcal{C}^2$ *then is simply* $\mathcal{C}^0 \cup \{e(B,B)\}$*, and* $\mathcal{C}^3 = \mathcal{C}^2$ *is the resulting core-safe chase.*

Example 6 admits other core-safe stratifications, leading to different core-safe chase sequences, but the final result is the same for all of them. The main result of this section is that this is a general property of the core-safe chase, so that we obtain a unique model that provides a semantics of the underlying rule sets.

Theorem 5. *For rule set* Σ *and database* $\mathcal{D}$ *with core-safe stratifications* $\mathcal{S}$ *and* $\mathcal{S}'$*,* $\mathcal{S}(\mathcal{D})$ *is isomorphic to* $\mathcal{S}'(\mathcal{D})$*.*

In the proof in our technical report (Ellmauthaler, Krötzsch, and Mennicke 2021), we compare core-safe stratifications up to simple transformations. In particular, we consider splitting and merging of strata in a stratification to show that the resulting core-safe chases are isomorphic. The rest of the proof is concerned with showing that all pairs of core-safe stratifications can be transformed into one another by only sequences of splitting and merging operations. Moreover, we find that core-safe models generalise the perfect core models that are based on full stratifications:

Proposition 4. *If a rule set is fully stratified and has a finite perfect core model, then the latter is isomorphic to its core-safe model.*

Together with Theorem 5, the previous result justifies that we call our semantics for core-safe stratified rule sets the *perfect core semantics*, since it generalises the eponymous semantics of Krötzsch (2020) without giving up on the uniqueness of the model.

Discussion and Conclusion

We have investigated how to answer normal Boolean conjunctive queries (BNCQs) on sets of existential rules, and we proposed the use of core models as a semantic reference point for this task. Arguably, cores are both intuitive and mathematically appealing for defining a non-monotonic "negation as failure" semantics, since they satisfy existential rules but at the same time minimise the amount of inferences and avoid redundancies. Approaches of truth minimisation are the basis for most non-monotonic semantics, but are much less obvious when giving up the syntactic Herbrand semantics of logic programming.

Nevertheless, our approach also has limitations. One of them is the difficulty of computing cores in practice, which we have addressed by identifying cases where this can be avoided. This leads to practical procedures, which in fact have already been implemented, though implementers are often not aware that the method is not sound for arbitrary negative queries or stratified negation in existential rules (Carral et al. 2019). Our core-safe chase still requires certain core constructions (after each stratum), which might not be practical. A possible approach to address this would be to investigate in more detail whether the intermediate non-core structures are truly problematic for evaluating the following rules and queries.

A more general limitation is that cores only behave well if universal models are finite, whereas rule sets with infinite models may have several distinct universal cores or no core that is a universal model at all (Carral et al. 2018). In this sense, the question which semantic reference point to choose in general remains open.

Acknowledgments

This work is partly supported by Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) in project number 389792660 (TRR 248, Center for Perspicuous Systems), by the Bundesministerium für Bildung und Forschung (BMBF, Federal Ministry of Education and Research) in the Center for Scalable Data Analytics and Artificial Intelligence (ScaDS.AI), and by the Center for Advancing Electronics Dresden (cfaed).

References

Abiteboul, S.; Hull, R.; and Vianu, V. 1994. *Foundations of Databases*. Addison Wesley.

Alviano, M.; Morak, M.; and Pieris, A. 2017. Stable Model Semantics for Tuple-Generating Dependencies Revisited. In (Sallinger, den Bussche, and Geerts 2017), 377–388.

Baget, J.-F.; Garreau, F.; Mugnier, M.-L.; and Rocher, S. 2014. Revisiting Chase Termination for Existential Rules and their Extension to Nonmonotonic Negation. In Konieczny, S.; and Tompits, H., eds., *Proceedings of the 15th International Workshop on Non-Monotonic Reasoning (NMR 2014)*, volume 1843-14-01 of *INFSYS Research Report*, 176–184. TU Wien.

Baget, J.-F.; Leclère, M.; Mugnier, M.-L.; and Salvat, E. 2009. Extending Decidable Cases for Rules with Existential Variables. In Boutilier, C., ed., *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI'09)*, 677–682. IJCAI.

Bauslaugh, B. L. 1995. Core-like properties of infinite graphs and structures. *Discrete Math.*, 138(1): 101–111.

Bellomarini, L.; Sallinger, E.; and Gottlob, G. 2018. The Vadalog System: Datalog-based Reasoning for Knowledge Graphs. *Proceedings of the VLDB Endowment*, 11(9): 975–987.

Benedikt, M.; Konstantinidis, G.; Mecca, G.; Motik, B.; Papotti, P.; Santoro, D.; and Tsamoura, E. 2017. Benchmarking the Chase. In (Sallinger, den Bussche, and Geerts 2017), 37–52.

Calì, A.; Gottlob, G.; and Lukasiewicz, T. 2009. A general datalog-based framework for tractable query answering over ontologies. In Paredaens, J.; and Su, J., eds., *Proceedings of the 28th Symposium on Principles of Database Systems (PODS'09)*, 77–86. ACM.

Carral, D.; Dragoste, I.; González, L.; Jacobs, C.; Krötzsch, M.; and Urbani, J. 2019. VLog: A Rule Engine for Knowledge Graphs. In Ghidini et al., C., ed., *Proceedings of the 18th International Semantic Web Conference (ISWC'19, Part II)*, volume 11779 of *LNCS*, 19–35. Springer.

Carral, D.; Krötzsch, M.; Marx, M.; Ozaki, A.; and Rudolph, S. 2018. Preserving Constraints with the Stable Chase. In Kimelfeld, B.; and Amsterdamer, Y., eds., *Proceedings of the 21st International Conference on Database Theory (ICDT'18)*, volume 98 of *LIPIcs*, 12:1–12:19. Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik.

Cuenca Grau, B.; Horrocks, I.; Krötzsch, M.; Kupke, C.; Magka, D.; Motik, B.; and Wang, Z. 2013. Acyclicity Notions for Existential Rules and Their Application to Query Answering in Ontologies. *Journal of Artificial Intelligence Research*, 47: 741–808.

Deutsch, A.; Nash, A.; and Remmel, J. B. 2008. The Chase Revisited. In Lenzerini, M.; and Lembo, D., eds., *Proceedings of the 27th Symposium on Principles of Database Systems (PODS'08)*, 149–158. ACM.

Ebbinghaus, H.-D.; Flum, J.; and Thomas, W. 1994. *Semantics of First-Order Languages*, 27–57. New York, NY: Springer New York. ISBN 978-1-4757-2355-7.

Ellmauthaler, S.; Krötzsch, M.; and Mennicke, S. 2021. Answering Queries with Negation over Existential Rules. arXiv:2112.07376.

Fagin, R.; Kolaitis, P. G.; Miller, R. J.; and Popa, L. 2005. Data exchange: semantics and query answering. *Theoretical Computer Science*, 336(1): 89–124.

Fagin, R.; Kolaitis, P. G.; and Popa, L. 2005. Data exchange: Getting to the core. *ACM Trans. Database Syst.*, 30(1): 174–210.

Gottlob, G.; Hernich, A.; Kupke, C.; and Lukasiewicz, T. 2014. Stable Model Semantics for Guarded Existential Rules and Description Logics. In Baral, C.; De Giacomo, G.; and Eiter, T., eds., *Proceedings of the 14th International Conference on Principles of Knowledge Representation and Reasoning (KR'14)*, 258–267. AAAI Press.

Hell, P.; and Nešetřil, J. 1992. The core of a graph. *Discrete Math.*, 109: 117–126.

Krötzsch, M. 2020. Computing Cores for Existential Rules with the Standard Chase and ASP. In Calvanese, D.; Erdem, E.; and Thielscher, M., eds., *Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning (KR 2020)*, 603–613. IJCAI.

Krötzsch, M.; and Rudolph, S. 2011. Extending Decidable Existential Rules by Joining Acyclicity and Guardedness. In Walsh, T., ed., *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI'11)*, 963–968. AAAI Press/IJCAI.

Magka, D.; Krötzsch, M.; and Horrocks, I. 2013. Computing Stable Models for Nonmonotonic Existential Rules. In Rossi, F., ed., *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI'13)*, 1031–1038. AAAI Press/IJCAI.

Nenov, Y.; Piro, R.; Motik, B.; Horrocks, I.; Wu, Z.; and Banerjee, J. 2015. RDFox: A Highly-Scalable RDF Store. In Arenas, M.; Corcho, Ó.; Simperl, E.; Strohmaier, M.; d'Aquin, M.; Srinivas, K.; Groth, P. T.; Dumontier, M.; Heflin, J.; Thirunarayan, K.; and Staab, S., eds., *Proceedings of the 14th International Semantic Web Conference (ISWC'15), Part II*, volume 9367 of *LNCS*, 3–20. Springer.

Sallinger, E.; den Bussche, J. V.; and Geerts, F., eds. 2017. *Proceedings of the 36th Symposium on Principles of Database Systems (PODS'17)*. ACM.

Urbani, J.; Krötzsch, M.; Jacobs, C. J. H.; Dragoste, I.; and Carral, D. 2018. Efficient Model Construction for Horn Logic with VLog: System description. In Galmiche, D.; Schulz, S.; and Sebastiani, R., eds., *Proceedings of the 9th International Joint Conference on Automated Reasoning (IJCAR'18)*, volume 10900 of *LNCS*, 680–688. Springer.