Computing Cores for Existential Rules with the Standard Chase and ASP

Markus Krötzsch
Knowledge-Based Systems Group, TU Dresden
markus.kroetzsch@tu-dresden.de

Abstract

To reason with existential rules (a.k.a. tuple-generating dependencies), one often computes universal models. Among the many such models of different structure and cardinality, the *core* is arguably the "best". Especially for finitely satisfiable theories, where the core is the unique smallest universal model, it has advantages in query answering, non-monotonic reasoning, and data exchange. Unfortunately, computing cores is difficult and not supported by most reasoners. We therefore propose ways of computing cores using practically implemented methods from rule reasoning and answer set programming. Our focus is on cases where the standard chase algorithm produces a core. We characterise this desirable situation in general terms that apply to a large class of cores, derive concrete approaches for decidable special cases, and generalise these approaches to non-monotonic extensions of existential rules.

Introduction

Existential rules are widely studied both in knowledge representation and in databases, where they are also known as *tuple-generating dependencies*. They are used as ontology language (Calì, Gottlob, and Pieris 2010; Baget et al. 2011; Calì, Gottlob, and Lukasiewicz 2012; Calì, Gottlob, and Pieris 2012; Cuenca Grau et al. 2013), declarative computing paradigm (Carral et al. 2019b; Bellomarini, Sallinger, and Gottlob 2018), or formalism for data exchange and integration (Fagin et al. 2005; Deutsch, Nash, and Remmel 2008; Calì, Gottlob, and Kifer 2008).

Reasoning with such rules is often based on *universal models*, the most general among all models, which are useful for solving diverse problems including query answering, data exchange, and rule entailment (Deutsch, Nash, and Remmel 2008). Universal models can be constructed by applying rules bottom-up – a process known as the *chase* and studied in many variations. Practical chase implementations were shown capable of solving a variety of reasoning problems at large scales (Geerts et al. 2014; Benedikt et al. 2017; Bellomarini, Sallinger, and Gottlob 2018; Urbani et al. 2018).

Nevertheless, universal models are neither unique nor logically equivalent, and different ways of performing the chase may lead to very different structures. For example, some might be finite when others are infinite, and chase termination is an important concern (see overviews by Cuenca Grau et al. (2013) and Krötzsch, Marx, and Rudolph (2019)). For positive queries, distinct universal models still entail the same results, but this is not true when allowing negation. As noticed by Baget et al. (2014), this leads to problems for adding (non-monotonic) negation to existential rules. The next example is adapted from Alviano, Morak, and Pieris (2017).

Example 1. *We consider existential rules with negation:*

$$\mathsf{human}(x) \rightarrow \exists v.\mathsf{hasFather}(x, v) \tag{1}$$

$$\mathsf{hasFather}(x, y) \rightarrow \mathsf{equals}(y, y) \tag{2}$$

$$\mathsf{hasFather}(x, y_1) \wedge \mathsf{hasFather}(x, y_2) \wedge \mathbf{not}\ \mathsf{equals}(y_1, y_2) \rightarrow \mathsf{twoFathers}(x) \tag{3}$$

Traditionally, non-monotonic semantics are based on interpretations over a fixed domain (the Herbrand universe*). We can achieve this by replacing existential variables with skolem terms, so that rule* (1) *becomes*

$$\mathsf{human}(x) \rightarrow \mathsf{hasFather}(x, f(x)). \tag{4}$$

The result is a stratified normal logic program*, which has a unique stable model. Now given an input database (fact set)* $\{\mathsf{human}(\mathsf{alice}), \mathsf{hasFather}(\mathsf{alice}, \mathsf{bob})\}$*, this model contains further facts* $\mathsf{hasFather}(\mathsf{alice}, f(\mathsf{alice}))$*,* $\mathsf{equals}(\mathsf{bob}, \mathsf{bob})$*,* $\mathsf{equals}(f(\mathsf{alice}), f(\mathsf{alice}))$*, and* $\mathsf{twoFathers}(\mathsf{alice})$*. This is unintuitive, since the statement that every human has some father* (1) *should not entail that Alice has two fathers.*

If we avoid skolemisation and use the standard chase on the original rules instead, then rule (1) *is not applicable (since Alice already has a father), and we derive* $\mathsf{equals}(\mathsf{bob}, \mathsf{bob})$ *as the only other fact.*

While the standard chase seems preferable here, it has the major drawback of producing different universal models depending on the order of rule applications. Negation can show these differences. The outputs of systems that combine (stratified) negation with this chase, such as VLog (Carral et al. 2019a), can therefore depend on implementation details.

As a possible solution, it has been suggested to focus on the special kind of universal models that are *cores*. Roughly, a core is a structure that cannot be embedded into a (simpler) substructure of itself. The skolemised model of Example 1 is not a core (if we admit that $f(\mathsf{Alice})$ is mapped to bob), whereas the standard chase model is. Among the most general models, core models can be viewed as the least redundant and most compact.

They have been characterised as the "best among all universal solutions" in data exchange (Fagin, Kolaitis, and Popa 2005), and suggested as preferable solutions in non-monotonic extensions to existential rules (Hernich 2011; Baget et al. 2014). Infinite interpretations can have several non-isomorphic cores, or none at all (Carral et al. 2018), but finitely satisfiable rule sets have a unique core, which can always be computed using the *core chase* (Deutsch, Nash, and Remmel 2008).

And yet, core models are hardly used in existential rule reasoning, and no major system seems to support the core chase. Indeed, the core chase is not a practical algorithm, since it involves the computation of homomorphisms from the whole chase into itself in an effort of finding strictly smaller cores. As a decision problem, this was shown to be complete for DP, i.e., the intersection of an NP-complete and a coNP-complete problem (Fagin, Kolaitis, and Popa 2005). Applying a rule in the standard chase is of – supposedly – higher complexity (namely Σ_2^{P} (Grahne and Onet 2018)), but the difference is that this is with respect to the size of a single rule, while the core computation depends on the size of the model that is computed. With models commonly containing millions of elements, this is not feasible in practice.

We therefore ask if core models can also be computed in simpler ways, using algorithms that are already shown to work in practice. To this end, we look for cases where the standard chase happens to produce a core. We derive a useful though undecidable semantic condition that is based on the absence of certain local homomorphisms, called *alternative matches*. Based on this abstract notion, we develop simplified conditions that can be decided in practice. Central to this approach is the analysis of interactions between rules, and the derivation of a rule application order based on the new concept of *core stratification*. Standard chases that abide by this order can be guaranteed to produce a core.

As an alternative approach, we consider the use of answer set programming (ASP) to compute core models. ASP is based on skolemisation, but we can use non-monotonic negation to express the condition on alternative matches. This leads to a correspondence of stable models and core models, and suggests that existential rule reasoning and data exchange might be interesting new application areas for ASP solvers.

We then return to the problem of non-monotonic negation and show that the *core-chase-stable models* of Baget et al. (2014) are not unique, even when restricting to stratified negation. By extending our previous notion of stratification for negation, we can identify cases where we can compute a *perfect core model* as a generalisation of stable models that is a core. We then discuss relations to other works in this area.

Finally, we study the complexity and expressivity of our logical fragments. We show that core-stratified rules on which the chase is guaranteed to terminate can express computations of non-elementary time complexity, and are therefore strictly more expressive than chase-terminating skolemised rules. Conversely, we also establish some general limits of our approach by showing that the (finite) core models of some rule sets cannot be computed by any standard chase.

We conclude by discussing open questions and potential further research that this work could enable.

Preliminaries

We briefly introduce key concepts and notations. The literature offers more thorough introductions (Abiteboul, Hull, and Vianu 1994; Ceri, Gottlob, and Tanca 1990). We construct expressions from countably infinite, mutually disjoint sets $\mathbf{V}$ of *variables*, $\mathbf{C}$ of *constants*, $\mathbf{N}$ of *labelled nulls*, and $\mathbf{P}$ of *predicate names*. Each predicate name $p \in \mathbf{P}$ has an *arity* $\mathsf{ar}(p) \geq 0$. *Terms* are elements of $\mathbf{V} \cup \mathbf{N} \cup \mathbf{C}$. We generally use $\boldsymbol{t}$ to denote a list $t_1, \ldots, t_{|\boldsymbol{t}|}$ of terms, and similar for special types of terms. An *atom* is an expression $p(\boldsymbol{t})$ with $p \in \mathbf{P}$, $\boldsymbol{t}$ a list of terms, and $\mathsf{ar}(p) = |\boldsymbol{t}|$. *Ground* terms or atoms contain neither variables nor nulls. An *interpretation* $\mathcal{I}$ is a set of atoms without variables. A *database* $\mathcal{D}$ is a finite set of ground atoms (i.e., a finite interpretation without nulls).

Rules An *existential rule* (or just *rule*) ρ is a formula

$$\rho = \forall \boldsymbol{x}, \boldsymbol{y}.\ \varphi[\boldsymbol{x}, \boldsymbol{y}] \rightarrow \exists \boldsymbol{z}.\ \psi[\boldsymbol{y}, \boldsymbol{z}], \tag{5}$$

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 $\mathsf{body}(\rho)$) and ψ the *head* (denoted $\mathsf{head}(\rho)$). We may treat conjunctions of atoms as sets, and we omit universal quantifiers in rules. We require that all variables in $\boldsymbol{y}$ do really occur in φ (*safety*).[1] A rule is *Datalog* if it has no existential quantifiers.

Morphisms and cores Given a set of atoms $\mathcal{A}$ and an interpretation $\mathcal{I}$, a *homomorphism* $h : \mathcal{A} \rightarrow \mathcal{I}$ 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 $p \in \mathbf{P}$: if $p(\boldsymbol{t}) \in \mathcal{A}$, then $p(h(\boldsymbol{t})) \in \mathcal{I}$, where $h(\boldsymbol{t})$ is the list of h-images of the terms $\boldsymbol{t}$. We apply homomorphisms to a formula by applying them individually to all of its terms.

A homomorphism h is *strong* if $p(\boldsymbol{t}) \in \mathcal{A}$ iff $p(h(\boldsymbol{t})) \in \mathcal{I}$ for all $p \in \mathbf{P}$, and an *embedding* if it is strong and injective. An *isomorphism* is a bijective strong homomorphism (a surjective embedding). A homomorphism from a structure to itself is called *endomorphism*; an isomorphism that is an endomorphism is called *automorphism*.

$\mathcal{I}$ is a *core* if every endomorphism $h : \mathcal{I} \rightarrow \mathcal{I}$ is an embedding. On finite structures, it is equivalent to require that every endomorphism is an isomorphism or, alternatively, that it is merely surjective (Hell and Nešetřil 1992). On infinite structures these notions differ (Bauslaugh 1995).

Semantics of rules A match of a rule ρ in an interpretation $\mathcal{I}$ is a homomorphism $\mathsf{body}(\rho) \rightarrow \mathcal{I}$. A match h of ρ in $\mathcal{I}$ is *satisfied* if there is a homomorphism $h' : \mathsf{head}(\rho) \rightarrow \mathcal{I}$ that agrees with h on all variables that occur in body and head (i.e., variables $\boldsymbol{y}$ in (5)). Rule ρ is *satisfied* by $\mathcal{I}$, written $\mathcal{I} \models \rho$, if every match of ρ in $\mathcal{I}$ is satisfied. A set of rules Σ is satisfied by $\mathcal{I}$, written $\mathcal{I} \models \Sigma$, if $\mathcal{I} \models \rho$ for all $\rho \in \Sigma$. We may treat a database $\mathcal{D}$ as sets of rules with empty bodies (also called *facts*), and write, e.g., $\mathcal{I} \models \mathcal{D}, \Sigma$ to express that $\mathcal{I} \models \Sigma$ and $\mathcal{D} \subseteq \mathcal{I}$. In this case, $\mathcal{I}$ is a *model* of Σ and $\mathcal{D}$.

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

Universal models and the chase A model $\mathcal{I}$ of a rule set Σ and database $\mathcal{D}$ is *universal* if it admits a homomorphism $h : \mathcal{I} \to \mathcal{J}$ to every model $\mathcal{J}$ of Σ and $\mathcal{D}$. A *core model* is a universal model that is a core. Many algorithms for computing universal models exist. Their basic approach, known as the *chase*, is to construct models bottom-up by applying rules to facts. The version we consider here is the *standard chase* (also known as *restricted chase*).

Definition 1. *A* (standard) chase sequence *for a database $\mathcal{D}$ and a set of rules Σ is a potentially infinite sequence of interpretations $\mathcal{D}^0, \mathcal{D}^1, \ldots$ such that*

(1) $\mathcal{D}^0 = \mathcal{D}$;

(2) for every $\mathcal{D}^{i+1}$ with $i \geq 0$, there is a match h for some rule $\rho = \varphi[\boldsymbol{x}, \boldsymbol{y}] \to \exists \boldsymbol{z}.\psi[\boldsymbol{y}, \boldsymbol{z}] \in \Sigma$ in $\mathcal{D}^i$ such that

(a) h is an unsatisfied match in $\mathcal{D}^i$ (i.e., h cannot be extended to a homomorphism $\psi \to \mathcal{D}^i$), and

(b) $\mathcal{D}^{i+1} = \mathcal{D}^i \cup \psi[h'(\boldsymbol{y}), h'(\boldsymbol{z})]$, where $h' : \psi \to \mathcal{D}^{i+1}$ is such that $h'(y) = h(y)$ for all $y \in \boldsymbol{y}$, and for all $z \in \boldsymbol{z}$, $h'(z) \in \mathbf{N}$ is a distinct null not occurring in $\mathcal{D}^i$.

*(3) if h is a match for a rule $\rho \in \Sigma$ and $\mathcal{D}^i$ ($i \geq 0$), then there is $j > i$ such that h is satisfied in $\mathcal{D}^j$ (*fairness*).*

The (standard) chase *for such a chase sequence is $\bigcup_{i \geq 0} \mathcal{D}^i$.*

Under the conditions of (2), we say that rule ρ is *applicable* to $\mathcal{D}^i$ for match h, and that $\mathcal{D}^{i+1}$ was obtained by applying ρ for (extended) match h'. We use abbreviations $\mathsf{rule}[i+1] := \rho$, $\mathsf{hom}[i+1] := h'$, $\mathsf{head}[i+1] := \mathsf{hom}[i+1](\mathsf{head}(\mathsf{rule}[i+1]))$, and $\mathsf{body}[i+1] := \mathsf{hom}[i+1](\mathsf{body}(\mathsf{rule}[i+1]))$.

Cores from the Standard Chase

We now give a semantic characterisation of a broad class of standard chase sequences that are guaranteed to produce core models. In the standard chase, a rule can only be applied for a match that is not already satisfied in some alternative way, without using any of the fresh nulls that would be introduced when applying the rule. We generalise this by considering alternative matches that may also use some of those nulls.

Definition 2. *Let $\mathcal{I}_a \subseteq \mathcal{I}_b$ be interpretations such that $\mathcal{I}_a$ was obtained by applying rule ρ for match h. A homomorphism $h' : h(\mathsf{head}(\rho)) \to \mathcal{I}_b$ is an* alternative match *of h if*

(1) $h'(t) = t$ for all terms t in $h(\mathsf{body}(\rho))$, and

(2) there is a null n in $h(\mathsf{head}(\rho))$ that does not occur in $h'(h(\mathsf{head}(\rho)))$.

An alternative match for step k *of a chase $\mathcal{D}^\infty = \bigcup_{i \geq 0} \mathcal{D}^i$, is an alternative match for $\mathcal{I}_a = \mathcal{D}^k$, $\mathcal{I}_b = \mathcal{D}^\infty$, $\rho = \mathsf{rule}[k]$, and $h = \mathsf{hom}[k]$.*

Example 2. *Consider the rules*

$$\to \exists v, w.r(v, w) \wedge r(w, v) \tag{6}$$

$$r(x, y) \to s(y, x) \tag{7}$$

$$r(x, y) \to \exists u.s(y, u) \tag{8}$$

Applying rule (6) *on an empty database leads to $\mathcal{D}^1 = \{r(n_1, n_2), r(n_2, n_1)\}$. Further rule applications yield $s(n_2, n_1)$ (rule* (7)*), $s(n_1, n_3)$ (rule* (8)*), and $s(n_1, n_2)$ (rule* (7)*). Note that we prioritised* (7) *for $r(n_1, n_2)$, but* (8) *for $r(n_2, n_1)$. Then $h' = \{n_1 \mapsto n_1, n_3 \mapsto n_2\}$ is an alternative match for step 3.*

We could equivalently define alternative matches in a chase as homomorphisms that restrict to the identity mapping on $\mathsf{body}[i]$ but do not restrict to an automorphism on $\mathsf{head}[i]$. An alternative match for k can easily be extended to a homomorphism $\mathcal{D}^k \to \mathcal{D}^\infty$ that is the identity on $\mathcal{D}^{k-1}$ (rather than only on $\mathsf{body}[k] \subseteq \mathcal{D}^{k-1}$). We can further extend it to a homomorphism on $\mathcal{D}^\infty$ using the following result.

Lemma 1. *For a chase $\mathcal{D}^\infty = \bigcup_{i \geq 0} \mathcal{D}^i$, every homomorphism $h_k : \mathcal{D}^k \to \mathcal{D}^\infty$ for some $k \geq 0$ can be extended to a homomorphism $h : \mathcal{D}^\infty \to \mathcal{D}^\infty$ that agrees with h_k on $\mathcal{D}^k$.*

Proof. Since h_k is defined on $\mathsf{body}[k+1] \subseteq \mathcal{D}^k$, the mapping $h_k \circ \mathsf{hom}[k+1]$ is a match for $\mathsf{rule}[k+1]$.[2] This match is satisfied in $\mathcal{D}^\infty$, so it can be extended to a homomorphism $g : \mathsf{head}(\mathsf{rule}[k+1]) \to \mathcal{D}^\infty$. We extend h_k to h_{k+1} by defining $h_{k+1}(n_v) := g(v)$ for every fresh null n_v introduced in $\mathsf{head}[k+1]$ for an existential variable v. Since $h_{k+1}(\mathcal{D}^{k+1} \setminus \mathcal{D}^k) = g(\mathsf{head}(\mathsf{rule}[k+1]))$, h_{k+1} is a homomorphism $\mathcal{D}^{k+1} \to \mathcal{D}^\infty$; it also agrees with h_k on $\mathcal{D}^k$. Now h is obtained as the limit $\bigcup_{i \geq k} h_i$, which is a homomorphism since the relevant conditions are finitary (i.e., local). □

When the chase is clear from the context, we write $\mathsf{homchase}(h_k)$ for an arbitrarily chosen h as in Lemma 1. While Lemma 1 may seem intuitive, we remark that it would not hold when replacing *homomorphism* by *isomorphism*.

Example 3. *For the chase sequence in Example 2, $h_1 : \{n_1 \mapsto n_2, n_2 \mapsto n_1\}$ is an isomorphism on $\mathcal{D}^1$. Lemma 1 yields a homomorphism $\mathsf{homchase}(h_1) = \{n_1 \mapsto n_2, n_2 \mapsto n_1, n_3 \mapsto n_1\}$, but we cannot extend h_1 to any isomorphism.*

Theorem 2. *If a chase $\mathcal{D}^\infty = \bigcup_{i \geq 0} \mathcal{D}^i$ has no alternative matches, then $\mathcal{D}^\infty$ is a core model.*

Proof. For a contradiction, suppose some homomorphism $h_0 : \mathcal{D}^\infty \to \mathcal{D}^\infty$ is not an embedding. Note that h_0 is the identity mapping on $\mathcal{D}^0$, which contains no nulls or variables. For every chase step $i+1 > 0$, we construct a homomorphism $h_{i+1} : \mathcal{D}^\infty \to \mathcal{D}^\infty$ that is the identity mapping on $\mathcal{D}^{i+1}$.

Assume h_i was defined for $i \geq 0$. Let $\tilde{h}_i$ be the restriction of h_i to $\mathcal{D}^{i+1}$. Then $\tilde{h}_i$ is identity on $\mathsf{body}[i+1] \subseteq \mathcal{D}^i$ by the hypothesis. Since $\tilde{h}_i$ is no alternative match, $\tilde{h}_i$ is a bijection (and possibly identity) on the nulls in $\mathsf{head}[i+1]$, and therefore restricts to an automorphism $\tilde{h}_i : \mathsf{head}[i+1] \to \mathsf{head}[i+1]$. Since $\tilde{h}_i$ is the identity on $\mathcal{D}^i$, it is therefore an automorphism on $\mathcal{D}^{i+1}$ with inverse $\tilde{h}_i^- : \mathcal{D}^{i+1} \to \mathcal{D}^{i+1}$. We set $h_{i+1} := \mathsf{homchase}(\tilde{h}_i^-) \circ h_i$. Since $\tilde{h}_i^-$ and $\mathsf{homchase}(\tilde{h}_i^-)$ agree on $\mathcal{D}^{i+1}$, h_{i+1} is indeed the identity on $\mathcal{D}^{i+1}$.

Now since h_0 is not an embedding it is (a) not strong or (b) not injective. Hence there is a tuple $\boldsymbol{t}$ such that either (a) there is a fact of the form $p(h_0(\boldsymbol{t})) \in \mathcal{D}^\infty$ with $p(\boldsymbol{t}) \notin \mathcal{D}^\infty$ or (b) $\boldsymbol{t} = \langle t_1, t_2 \rangle$ with $h_0(t_1) = h_0(t_2)$. Either property is preserved for all h_i, which we can show by induction. For case (a), $p(h_i(\boldsymbol{t})) \in \mathcal{D}^\infty$ implies $p(h_{i+1}(\boldsymbol{t})) \in \mathcal{D}^\infty$ since $\mathsf{homchase}(\tilde{h}_i^-)$ is a homomorphism by Lemma 1. For case (b), $h_i(t_1) = h_i(t_2)$ immediately yields $h_{i+1}(t_1) = h_{i+1}(t_2)$.

Hence, all h_i satisfy either (a) or (b) for a fixed $\boldsymbol{t}$. However, for some $k \geq 0$, all terms of $\boldsymbol{t}$ are contained in $\mathcal{D}^k$. Since

[2] By $\circ$ we denote function composition: $(f \circ g)(x) = f(g(x))$.

h_k is the identity on $\mathcal{D}^k$, it satisfies neither (a) nor (b) – a contradiction. Hence, a non-embedding h_0 cannot exist. □

The converse of Theorem 2 is not true: even a chase that is a core can have alternative matches.

Example 4. *Applying the rules*

$$\rightarrow \exists v.p(v) \tag{9}$$
$$p(x) \rightarrow \exists w.r(x,w) \tag{10}$$
$$r(x,y) \rightarrow p(y) \wedge r(y,x) \tag{11}$$

to $\mathcal{D}^0 = \emptyset$ in any order yields a chase $\mathcal{D}^\infty = \{p(n_1), r(n_1,n_2), p(n_2), r(n_2,n_1)\}$, which is a core. However, there is an alternative match $\{n_1 \mapsto n_2\}$ for the first chase step (where we map the head of the first rule application $p(n_1)$).

Theorem 3. *If $\mathcal{D}^\infty = \bigcup_{i\geq 0} \mathcal{D}^i$ has an alternative match, then $\mathcal{D}^\infty$ admits an endomorphism that is not identity. In particular, if a chase with alternative matches yields a core, then it has a non-identity embedding.*

Proof. Let $h : \mathsf{head}[k] \rightarrow \mathcal{D}^\infty$ be an alternative match. We can extend h to a homomorphism $\hat{h} : \mathcal{D}^k \rightarrow \mathcal{D}^\infty$ by setting $h(t) = t$ for all terms in $\mathcal{D}^k$ that do not occur in $\mathsf{head}[k]$. The required endomorphism is $\mathsf{homchase}(\hat{h})$ of Lemma 1. □

This result suggests that core-producing chases with alternative matches are rare in practice, since non-trivial embeddings require a large amount of (non-redundant) symmetry that seems unlikely to occur in applications. We might therefore say that "most" cores that can be produced by a standard chase at all can be produced by a chase without alternative matches. However, as we will see later, there are core models (possibly without non-trivial embeddings) that cannot be obtained from any standard chase.

Core Stratification

We now turn the results of the previous section into a practically viable procedure for computing core models in the chase. Unfortunately, the general precondition of Theorem 2 does not lead to a decidable criterion.

Theorem 4. *It is undecidable if some chase (or all chases) of some rule set Σ and database $\mathcal{D}$ have an alternative match.*

This can be shown by reduction from the undecidable query answering problem over existential rules, using a rule that enforces an alternative match if its body (the query) is satisfied. To address this issue, we introduce a local criterion that can be used to detect situations when alternative matches could possibly occur. The next definition describes cases where the application of one rule might enable an alternative match for another rule that was applied earlier.

Definition 3. *A rule ρ_1* restrains *a rule ρ_2, written $\rho_1 \prec^\square \rho_2$, if there are interpretations $\mathcal{I}_a \subseteq \mathcal{I}_b$ such that*

(a) $\mathcal{I}_b$ is obtained by applying ρ_1 for match h_1,
(b) $\mathcal{I}_a$ is obtained by applying ρ_2 for match h_2,
(c) h_2 has an alternative match $h_2(\mathsf{head}(\rho_2)) \rightarrow \mathcal{I}_b$, and
(d) h_2 has no alternative match $h_2(\mathsf{head}(\rho_2)) \rightarrow \mathcal{I}_b \setminus h_1(\mathsf{head}(\rho_1))$.

Since Datalog rules cannot have alternative matches, they cannot be restrained. Also note that Definition 3 allows that $\rho_1 = \rho_2$ and $\mathcal{I}_a = \mathcal{I}_b$. Indeed, a rule may restrain itself:

Example 5. *Consider the rule*

$$\rho = b(x) \rightarrow \exists v, w.r(x,v,w) \wedge r(x,x,w) \wedge a(v).$$

We obtain $\rho \prec^\square \rho$ by setting $\mathcal{I}_a = \mathcal{I}_b = \{a(c), b(c), r(c,n_1,n_2), r(c,c,n_2), a(n_1)\}$ and $h_1 = h_2 = \{x \mapsto c, v \mapsto n_1, w \mapsto n_2\}$. The required alternative match is $\{c \mapsto c, n_1 \mapsto c, n_2 \mapsto n_2\}$. Note how the third parameter of r ensures that ρ is applicable to $\{x \mapsto c\}$, and also that ρ cannot be decomposed into two rules that produce the necessary and the redundant part of the head independently.

The interpretation $\mathcal{I}_a$ and $\mathcal{I}_b$ in Definition 3 can be restricted to contain at most as many facts as there are atoms in ρ_1 and ρ_2. Since the names of constants and nulls are irrelevant, we can therefore represent the interpretations polynomially. As observed by Grahne and Onet (2018), checking applicability of a rule is complete for Σ_2^P. We therefore obtain:

Proposition 5. *Deciding $\rho_1 \prec^\square \rho_2$ is Σ_2^P-complete.*

Computing $\prec^\square$ is therefore not harder than applying rules in the chase, but using only very small databases. Together with the next result, this suggests that $\prec^\square$ can serve as a practically computable approximation of the existence of alternative homomorphisms in the chase.

Lemma 6. *If a chase $\mathcal{D}^\infty = \bigcup_{i\geq 0} \mathcal{D}^i$ has an alternative match for step k, then there is $\ell \geq k$ with $\mathsf{rule}[\ell] \prec^\square \mathsf{rule}[k]$.*

Proof. Let $\ell \geq k$ be the smallest number such that there is an alternative match $\mathsf{head}[k] \rightarrow \mathcal{D}^{\ell+1}$. We set $\rho_1 = \mathsf{rule}[\ell]$, $\rho_2 = \mathsf{rule}[k]$, $\mathcal{I}_a = \mathcal{D}^{k+1}$, and $\mathcal{I}_b = \mathcal{D}^{\ell+1}$. Conditions (a) and (b) of Definition 3 are immediate from the chase. Condition (c) holds by our choice of ℓ. For (d), note that there is no alternative match $\mathsf{head}[k] \rightarrow \mathcal{D}^i$ for any $k < i \leq \ell$ by minimality of ℓ. There is also no alternative match $\mathsf{head}[k] \rightarrow \mathcal{D}^k$ since this would imply that $\mathsf{rule}[k]$ is not applicable. □

By Theorem 2 we can therefore obtain a core from a standard chase if we ensure that, whenever $\rho_1 \prec^\square \rho_2$, we only apply rule ρ_1 strictly before ρ_2. It is not enough to prioritise ρ_1 over ρ_2; before applying ρ_2, we also must ensure that ρ_1 will not become applicable ever again. To estimate if this is the case, we adapt a notion of positive dependency between rules due to Deutsch, Nash, and Remmel (2008).

Definition 4. *A rule ρ_2* positively relies on *a rule ρ_1, written $\rho_1 \prec^+ \rho_2$, if there are interpretations $\mathcal{I}_a \subseteq \mathcal{I}_b$ and a function h_2 such that*

(a) $\mathcal{I}_b$ is obtained from $\mathcal{I}_a$ by applying ρ_1 for the match h_1,
(b) h_2 is an unsatisfied match for ρ_2 on $\mathcal{I}_b$, and
(c) h_2 is not a match for ρ_2 on $\mathcal{I}_a$.

Intuitively, $\rho_1 \prec^+ \rho_2$ means that applying ρ_1 might sometimes enable new applications of ρ_2.

Definition 5. *Consider a rule set Σ and a rule $\rho \in \Sigma$. The set $\rho{\downarrow^\square}$ consists of all rules $\rho' \in \Sigma$ that stand in the relation ρ' $(\prec^+)^* \cdot \prec^\square$ ρ, where $(\prec^+)^*$ is the reflexive-transitive closure of $\prec^+$, and $\cdot$ is relation composition: $R \cdot S = \{\langle x,y\rangle \mid x\ R\ z \text{ and } z\ S\ y \text{ for some } z\}$.*

A rule set Σ *is* core-stratified *if* $\rho \notin \rho{\downarrow^{\square}}$ *for all* $\rho \in \Sigma$. *A chase sequence is core-stratified if, for all chase steps* i, j, $\mathsf{rule}[i] \prec^{\square} \mathsf{rule}[j]$ *implies* $i < j$.

Traditional stratification (with respect to negation) is often used to group rules in linearly ordered *strata*, obtained as a topological ordering of the strongly connected components in the relation $\prec^{\square} \cup \prec^{+}$. The rules are then applied bottom-up stratum by stratum. Our notion of core-stratified chase is more permissive, since we allow rules from higher strata to be applied if they are not restrained by any rule that might still be applied. In particular, Datalog rules can always be applied. Nevertheless, core-stratified rule sets do not always admit a core-stratified chase.

Theorem 7. *Deciding if* Σ *is core-stratified is* Σ_2^{P}*-complete. Whether* Σ *admits a core-stratified chase is undecidable.*

While Σ_2^{P}-completeness is immediate from Proposition 5 (and analogous observations for $\prec^{+}$), the undecidability might surprise. Given a core-stratified rule set, one can enforce a rule application strategy that prioritises rules in $\rho{\downarrow^{\square}}$ over ρ, and this will ensure that $\mathsf{rule}[i] \prec^{\square} \mathsf{rule}[j]$ implies $i < j$, but it does not necessarily result in a chase sequence. Indeed, fairness will be violated if the chase does not terminate on some stratum that is not maximal. Chase termination has been studied extensively, and many decidable criteria exist for detecting it (Baget et al. 2011; Cuenca Grau et al. 2013; Carral, Dragoste, and Krötzsch 2017). Any of these works can be applied to detect the preconditions of the next result.

Theorem 8. *Consider a rule set* Σ *and a database* $\mathcal{D}$. *If* Σ *is core-stratified and if the chase on* Σ *and* $\mathcal{D}$ *terminates for all strategies (or for all strategies that prioritise Datalog rules), then there is a core-stratified chase over* Σ *and* $\mathcal{D}$*, which produces the finite core model of* Σ *and* $\mathcal{D}$.

Proof. We can use any rule application order that exhaustively applies rules in $\rho{\downarrow^{\square}}$ before a rule ρ. A Datalog rule ρ may be applied at any time as long as no rule was applied that is restrained by ρ – after this point, ρ will no longer be applicable due to the assumed stratification. Therefore, applicable Datalog rules can always be prioritised. The chase is guaranteed to be finite by the preconditions.

Suppose for a contradiction that the resulting chase has alternative matches. By Lemma 6, we find $\ell \geq k$ with $\mathsf{rule}[\ell] \prec^{\square} \mathsf{rule}[k]$. With the chosen chase strategy, this implies that $\mathsf{body}[\ell] \not\subseteq \mathcal{D}^{k-1}$, although $\mathsf{body}[\ell] \subseteq \mathcal{D}^{\ell-1}$. By the definition of $\prec^{+}$, all rules that contributed to deriving atoms of $\mathsf{body}[\ell]$ are in $\mathsf{rule}[\ell]{\downarrow^{\square}}$. Using the assumption that matches of these rules were prioritised over $\mathsf{rule}[k]$, we can show that $\mathsf{rule}[\ell]$ would be applicable with $\mathsf{hom}[\ell]$ at step k, contradicting our assumptions.

Hence, the chase has no alternative matches, so it yields a core by Theorem 2. □

One could further generalise this by admitting non-termination in maximal strata, which would allow for infinite chase sequences that yield cores. Though of limited practical utility, this is interesting since infinite core models are difficult or impossible to obtain in general (Carral et al. 2018). Finally, we point out a trade-off between core stratification and termination.

Definition 6. *Given a rule* ρ *as in* (5)*, its* decomposition *consists of the two rules* $\rho_{\mathsf{body}} = \varphi[\boldsymbol{x}, \boldsymbol{y}] \rightarrow \exists \boldsymbol{z}.r_\rho(\boldsymbol{y}, \boldsymbol{z})$ *and* $\rho_{\mathsf{head}} = r_\rho(\boldsymbol{y}, \boldsymbol{z}) \rightarrow \psi[\boldsymbol{y}, \boldsymbol{z}]$*, where* r_ρ *is a new predicate.*

Since r_ρ is new, every match of ρ_{body} is unsatisfied unless the rule was applied to some match that agrees on $\boldsymbol{y}$ – the rule behaves as if skolemised and cannot be "blocked" by other rules. Indeed, ρ_{body} is not restrained by any rule. As a consequence, sets of decomposed rules are always core-stratified, but might lead to larger (possibly infinite) cores. Interestingly, decomposition can also be used selectively to remove individual $\prec^{\square}$ relations, which may preserve termination while ensuring core stratification. We use this idea to demonstrate the expressive power of core-stratified rule sets in Theorem 18 below.

Core Models from ASP

We now show how to obtain answer sets that encode core models. Answer set programming (ASP) is a successful paradigm for declarative rule-based computation with applications in non-monotonic inference, constraint satisfaction, and optimisation (Lifschitz 2019). Its main reasoning task is the computation of *stable models*, known as *answer sets*, and of query results over (all) such models.

Existential quantifiers and named nulls are not allowed in ASP, but we can use function symbols instead: a signature of ASP consists of *variable names* $\mathbf{V}$, *constants* $\mathbf{C}$, *predicate names* $\mathbf{P}$, and *function symbols* $\mathbf{F}$. We define *ASP atoms* and *ASP interpretations* as for existential rules, but using terms that may contain functions rather than named nulls. A *(normal) logic program* P is a set of rules of the form

$$B_1 \wedge \cdots \wedge B_n \wedge \mathbf{not}\, N_1 \wedge \cdots \wedge \mathbf{not}\, N_m \rightarrow H_1 \wedge \cdots \wedge H_\ell$$

where B_i, N_i, and H_i are atoms that may contain variables and function symbols (including constants), and where each variable of the rule occurs in some atom B_i.

Definition 7. *Let* $\mathcal{D}$ *be a database and* P *a logic program. The (possibly infinite) program* $\mathrm{ground}(P \cup \mathcal{D})$ *is the union of* $\mathcal{D}$ *with the set of all rules that can be obtained from some rule in* P *by uniform replacement of each variable with a ground term over symbols in* $P \cup \mathcal{D}$*. For an ASP interpretation* $\mathcal{I}$*, the* reduct $\mathrm{red}_{\mathcal{I}}(P \cup \mathcal{D})$ *is obtained from* $\mathrm{ground}(P \cup \mathcal{D})$ *by (1) deleting every negated atom* $\mathbf{not}\, N$ *with* $N \notin \mathcal{I}$ *and (2) deleting every rule with a negated atom* $\mathbf{not}\, N$ *with* $N \in \mathcal{I}$*.* $\mathcal{I}$ *is a* stable model *of* $P \cup \mathcal{D}$ *if it is the least model of* $\mathrm{red}_{\mathcal{I}}(P \cup \mathcal{D})$.

Functions can simulate existential quantifiers by *skolemisation*: given a rule $\varphi[\boldsymbol{x}, \boldsymbol{y}] \rightarrow \exists \boldsymbol{z}.\, \psi[\boldsymbol{y}, \boldsymbol{z}]$, we replace each existential variable $z \in \boldsymbol{z}$ by a term $f_z(\boldsymbol{y})$, where f_z is a fresh ("skolem") function symbol of arity $|\boldsymbol{y}|$. Skolemised existential rules are therefore a special case of the logic programs that ASP solvers support, and indeed many ASP systems essentially perform the so-called *skolem chase* (Marnette 2009) during the grounding phase of solving.

The skolem chase leads to universal models, just like the standard chase, but the latter produces a finite model for strictly more inputs. This is an important concern, since ASP solvers require answer sets to be finite. Unfortunately, when restricting to rules for which the chase is guaranteed to be finite, using the skolem chase imposes far greater expressive

limitations than using the standard chase (Krötzsch, Marx, and Rudolph 2019). Moreover, even when the skolem chase is finite, the result is rarely a core model of the original rules, and this is not influenced by the order of rule applications.

Rather than relying on the skolem chase alone, we therefore incorporate the non-monotonic reasoning capabilities of ASP to guarantee the general conditions of Theorem 2.

Definition 8. *Let Σ be a set of existential rules. The normal logic program* $\mathrm{ASP}(\Sigma)$ *consists of the following rules for every rule $\rho = \varphi[\boldsymbol{x}, \boldsymbol{y}] \to \exists \boldsymbol{z}.\, \psi[\boldsymbol{y}, \boldsymbol{z}]$ in Σ:*

(a) a rule

$$\varphi[\boldsymbol{x}, \boldsymbol{y}] \wedge \mathbf{not}\, \mathsf{block}_\rho(\boldsymbol{y}) \to \psi[\boldsymbol{y}, \boldsymbol{f}_{\boldsymbol{z}}(\boldsymbol{y})] \wedge \bigwedge\nolimits_{z \in \boldsymbol{z}} \mathsf{null}^z_\rho(f_z(\boldsymbol{y}), \boldsymbol{y})$$

where $\psi[\boldsymbol{y}, \boldsymbol{f}_{\boldsymbol{z}}(\boldsymbol{y})]$ is the skolemisation of $\psi[\boldsymbol{y}, \boldsymbol{z}]$;

(b) for every $v \in \boldsymbol{z}$, a rule

$$\varphi[\boldsymbol{x}, \boldsymbol{y}] \wedge \psi[\boldsymbol{y}, \boldsymbol{z}] \wedge \bigwedge_{z \in \boldsymbol{z}} \mathbf{not}\, \mathsf{null}^v_\rho(z, \boldsymbol{y}) \to \mathsf{block}_\rho(\boldsymbol{y}),$$

using fresh predicate names block_ρ and null^v_ρ.

Given an interpretation $\mathcal{I}$, we define $\mathcal{I}^- := \{p(\boldsymbol{t}) \in \mathcal{I} \mid p$ not of the form block_ρ or $\mathsf{null}^v_\rho\}$ to be the set of atoms that do not use these auxiliary predicates.

Rules of the form (b) are applicable to matches of $\boldsymbol{y}$ that extend to an alternative match of rule ρ, since they require that at least one "null" (skolem term) is not used in the match for the variables $\boldsymbol{z}$. Note that Datalog rules do not lead to any rules of form (b), since they contain no $v \in \boldsymbol{z}$, and are therefore never blocked.

Theorem 9. *Consider a rule set Σ and a database $\mathcal{D}$.*

(1) For every stable model $\mathcal{I}$ of $\mathrm{ASP}(\Sigma) \cup \mathcal{D}$, $\mathcal{I}^-$ is a core model of Σ and $\mathcal{D}$.

(2) Every core model $\mathcal{J}$ of Σ and $\mathcal{D}$ that is the result of a chase without alternative matches is isomorphic to $\mathcal{I}^-$ for some stable model $\mathcal{I}$ of $\mathrm{ASP}(\Sigma) \cup \mathcal{D}$.

Proof. Consider a stable model $\mathcal{I}$ as in (1). Let $R := \mathrm{red}_{\mathcal{I}}(\mathrm{ASP}(\Sigma) \cup \mathcal{D})$. The reduced ground rules in R are of two types, depending on whether they originate from rules of the form (a) or (b) in Definition 8, and atoms in $\mathcal{I}^-$ are only entailed by type (a) rules. Being the least model, $\mathcal{I}$ can be computed by a sequence of chase-like applications of rules in R. We construct a corresponding chase sequence $\mathcal{D}^0, \mathcal{D}^1, \ldots$ and an isomorphism $\iota : \mathcal{D}^\infty \to \mathcal{I}^-$.

Initially, $\mathcal{D} = \mathcal{D}^0 \subseteq \mathcal{I}$, and ι is the identity mapping on constants in $\mathcal{D}$. Assume that the chase and ι was constructed until $\mathcal{D}^i$, and that $\rho_i = \varphi[\boldsymbol{t}, \boldsymbol{s}] \to \psi[\boldsymbol{s}, \boldsymbol{f}_{\boldsymbol{z}}(\boldsymbol{s})] \wedge \bigwedge_{z \in \boldsymbol{z}} \mathsf{null}^z_\rho(f_z(\boldsymbol{s}), \boldsymbol{s})$ is the ith type (a) rule applied when computing $\mathcal{I}$. By the induction hypothesis, $\varphi[\iota^-(\boldsymbol{t}), \iota^-(\boldsymbol{s})] \in \mathcal{D}^i$, so the chase has a match h for the original rule $\rho = \varphi[\boldsymbol{x}, \boldsymbol{y}] \to \exists \boldsymbol{z}.\, \psi[\boldsymbol{y}, \boldsymbol{z}]$. Match h is not satisfied in $\mathcal{D}^i$: otherwise, there would be terms $\boldsymbol{r}$ in $\mathcal{D}^i$ for which a type (b) rule $\varphi[\boldsymbol{t}, \boldsymbol{s}] \wedge \psi[\boldsymbol{s}, \iota(\boldsymbol{r})] \to \mathsf{block}_\rho(\boldsymbol{s})$ applies, which would entail $\mathsf{block}_\rho(\boldsymbol{s}) \in \mathcal{I}$ and contradict the assumption $\rho_i \in R$. Hence, ρ is applicable in the chase, introducing a fresh null n_z for each $z \in \boldsymbol{z}$. We set $\iota(n_z) := f_z(\boldsymbol{s})$.

This yields a (possibly infinite) chase. To see that it is fair, consider a match h for some $\rho \in \Sigma$ and $\mathcal{D}^i$. Case (i): there is a corresponding type (a) rule $\rho_j \in R$ where body variables are instantiated by $h \circ \iota$; then ρ_j will eventually be applied in the computation of $\mathcal{I}$ and the corresponding chase step. Case (ii): there is no corresponding type (a) rule in R, since it was deleted during the reduction; then there is a fact $\mathsf{block}_\rho(\iota(h(\boldsymbol{y}))) \in \mathcal{I}$ that must be entailed by a suitable type (b) rule, witnessing that rule match h is satisfied in $\mathcal{I}$.

Moreover, the constructed chase has no alternative matches. Every mapping h that satisfies (2) of Definition 2 corresponds to a type (b) rule in R. When applicable, the rule derives a block-atom that ensures that R contains no type (a) rule for a corresponding match. Therefore, by Theorem 2, the chase yields a core, finishing the proof of (1).

Now consider a core $\mathcal{J}$ as in (2). Using similar arguments as above, we obtain a sequence of applications of type (a) rules in $\mathrm{ASP}(\Sigma)$ that corresponds to the chase. Let $\mathcal{I}$ be the set of atoms that contains (i) all atoms inferred by this sequence of type (a) rules, including null^z_ρ-atoms, and (ii) all block-atoms that can be inferred using a match of a type (b) rule whose negated body atoms of form $\mathbf{not}\, \mathsf{null}^v_\rho(t, \boldsymbol{s})$ are never derived under (i). Then $\mathrm{red}_{\mathcal{I}}(\mathrm{ASP}(\Sigma) \cup \mathcal{D})$ contains rules corresponding to each type (a) rule used in the construction since there are no alternative matches for the chase, so that none of their negated block-atoms is derived. Hence $\mathcal{I}$ is the required stable model, and $\mathcal{I}^-$ is isomorphic to $\mathcal{J}$. □

Note that Theorem 9 still allows cores to exist when $\mathrm{ASP}(\Sigma) \cup \mathcal{D}$ has no stable model, but these cores either cannot be obtained by any standard chase sequence, or admit non-trivial embeddings (Theorem 3).

Theorem 9 also applies to infinite stable models, but we can use known chase-termination criteria to focus on finite cases. Together with (the proof of) Theorem 8, we get:

Corollary 10. *If Σ is core-stratified and if the chase on Σ and $\mathcal{D}$ terminates for all strategies (or for all strategies that prioritise Datalog rules), then $\mathrm{ASP}(\Sigma) \cup \mathcal{D}$ has a unique stable model $\mathcal{I}$, such that $\mathcal{I}^-$ is isomorphic to the unique core model of Σ and $\mathcal{D}$.*

Non-monotonic Negation

As discussed in the introduction, cores are also appealing as minimal structures for defining the semantics of non-monotonic negation. We now develop a semantics based on a notion of *perfect core model* that relies on this intuition.

An *existential rule with negation* is a formula

$$\rho = \forall \boldsymbol{x}, \boldsymbol{y}.\, \varphi[\boldsymbol{x}, \boldsymbol{y}] \wedge \bar{\chi}[\boldsymbol{x}, \boldsymbol{y}] \to \exists \boldsymbol{z}.\, \psi[\boldsymbol{y}, \boldsymbol{z}], \tag{12}$$

where $\bar{\chi}[\boldsymbol{x}, \boldsymbol{y}]$ denotes a conjunction of negated atoms of the form $\mathbf{not}\, p(\boldsymbol{t})$ that may use variables from $\boldsymbol{x}$ and $\boldsymbol{y}$, and all other details are as in (5). To define an ASP-like semantics for such rules, Baget et al. (2014) suggest a characterisation based on *generating rules*, which was first proposed for ASP by Konczak, Linke, and Schaub (2006).

Definition 9. *Let $\mathcal{I}$ be an interpretation and let ρ be a rule as in* (12)*. A match h for ρ is* generating *with respect to $\mathcal{I}$ if, $p(h(\boldsymbol{t})) \notin \mathcal{I}$ for all $\mathbf{not}\, p(\boldsymbol{t}) \in \bar{\chi}$.*

Let Σ be a set of rules with negation and let $\mathcal{D}$ be a database. A generating (standard) chase sequence *for Σ and $\mathcal{D}$ is a chase sequence $\mathcal{D}^0, \mathcal{D}^1, \ldots$ as in Definition 1 that*

- *is based on the set of rules Σ^+ obtained from Σ by omitting the negative body $\bar{\chi}$ in each rule, and*
- *in each step $\mathcal{D}^i$, considers only matches for a rule ρ^+ on $\mathcal{D}^i$ if ρ is generating with respect to $\mathcal{D}^\infty$.*

An interpretation $\mathcal{I}$ is a generated model *of Σ and $\mathcal{D}$ if it is of the form $\mathcal{I} = \mathcal{D}^\infty$ for some generating chase sequence.*

In other words, a rule can only be applied in a generating chase if its negative body will never be satisfied in this chase, and fairness is only required for such rule applications. When adopted to rules with (skolem) function symbols, generated models coincide with the stable models of Definition 7 (Konczak, Linke, and Schaub 2006). Baget et al. (2014) observe that the generating chase leads to non-equivalent results for different chase variants and rule application orders, but argue that "[t]his problem does not arise with core chase." This might be too optimistic, as illustrated by the next example.

Example 6. *Consider $\mathcal{D} = \{H(a), F(a, b)\}$ and the rules:*

$$H(x) \rightarrow \exists v.F(x, v) \wedge M(v) \tag{13}$$
$$F(x, y) \rightarrow M(y) \tag{14}$$
$$F(x, y) \rightarrow E(y, y) \tag{15}$$
$$F(x, y_1) \wedge F(x, y_2) \wedge \textbf{not}\, E(y_1, y_2) \rightarrow D(y_1, y_2) \tag{16}$$

We can obtain a generated model $\mathcal{I}_1$ consisting of $\mathcal{D}$ and atoms $M(b)$ (from (14)*) and $E(b, b)$ (from* (15)*). No match for rule* (16) *is generating for this model.*

However, there is another generated model $\mathcal{I}_2$ where we extend $\mathcal{D}$ by deriving facts in the order $F(a, n), M(n)$ (13)*, $E(b, b), E(n, n)$* (15)*, $D(b, n), D(n, b)$* (16)*, and $M(b)$* (14)*. Match $\{x \mapsto a, y_1 \mapsto b, y_2 \mapsto n\}$ of* (16) *is generating now.*

Interestingly, both $\mathcal{I}_1$ and $\mathcal{I}_2$ are cores. In fact, all intermediate structures $\mathcal{D}^i$ in either chase sequence are cores as well, so even an interleaved core construction (used in the core chase) does not change the result. $\mathcal{I}_2$ is isomorphic to the answer set we obtain when skolemising existential quantifiers, but $\mathcal{I}_1$ arguably is the intended model (where Alice has just one father, cf. Example 1).

The previous example highlights a conflict between cores and non-monotonicity. The applicability of rules with negation is not preserved under homomorphisms, hence cores may have fewer generating matches (as in $\mathcal{I}_1$, where (16) has no generating match). Conversely, applying non-monotonic rules may prevent core constructions (as in $\mathcal{I}_2$, where facts $D(b, n), D(n, b)$ prevent endomorphisms with $n \mapsto b$).

Since generating matches are not closed under homomorphisms, Lemma 1 fails for rules with negation, and the proof of Theorem 2 is not applicable. We re-establish the relevance of alternative matches by focussing on finite models.

Theorem 11. *If a generating chase $\mathcal{D}^\infty = \bigcup_{i \geq 0} \mathcal{D}^i$ is finite and has no alternative matches, then $\mathcal{D}^\infty$ is a core model.*

Proof. Suppose for a contradiction that $\mathcal{D}^\infty$ is not a core. Then it has an endomorphism that is not an embedding. For finite structures, this is equivalent to the existence of an endomorphism $h : \mathcal{D}^\infty \rightarrow \mathcal{D}^\infty$ such that $h(\mathcal{D}^\infty)$ is the (unique) core of $\mathcal{D}^\infty$ and $h(x) = x$ for all x in $h(\mathcal{D}^\infty)$. Since h is not the identity on $\mathcal{D}^\infty$, there is a smallest i such that h is the identity on $\mathcal{D}^i$ but not on $\mathcal{D}^{i+1}$. But then h restricts to an alternative match for chase step $i + 1$, since our assumptions on h do not allow it to be an automorphism on $\mathsf{head}[i+1]$. □

Example 7. *The generating chase for $\mathcal{I}_1$ in Example 6 does not have alternative matches and indeed is a core. On the other hand, the chase for $\mathcal{I}_2$ has an alternative match that maps the rule head $F(a, n), M(n)$ to $F(a, b), M(b)$, and yet it is also a core. In contrast to the positive case, where such alternative matches were associated with non-identity embeddings (Theorem 3), this alternative match cannot be extended to any endomorphism of $\mathcal{I}_2$.*

It remains open how "good" answer sets for existential rules with negation can be defined in general, but we present an approach that covers certain well-behaved cases, including Example 6, and that leads to practical algorithms. To this end, we adapt the notion of core-stratification to rules with negation, and combine it with a suitable form of stratification of negation. We slightly restrict the relationships $\prec^\square$ and $\prec^+$ to consider only cases that also satisfy the negative body of the involved rules, and we further introduce a relation $\prec^-$ to detect if a rule could produce a conclusion that makes (a match for) another rule non-generating.

Definition 10. *Let ρ_1 and ρ_2 be rules with negation, and let ρ_1^+ and ρ_2^+ be obtained by removing any negated body atoms. Then $\rho_1 \prec^\square \rho_2$ (resp. $\rho_1 \prec^+ \rho_2$) is true if the conditions of Definition 3 (resp. Definition 4) hold for ρ_1^+ and ρ_2^+, and the match h_i is generating for ρ_i with respect to $\mathcal{I}_b$ ($i \in \{1, 2\}$).*

Rule ρ_2 negatively relies on *ρ_1, written $\rho_1 \prec^- \rho_2$, if there are interpretations $\mathcal{I}_a \subseteq \mathcal{I}_b$ such that*

(a) $\mathcal{I}_b$ is obtained by applying ρ_1^+ for match h_1,
(b) $\mathcal{I}_a$ is obtained by applying ρ_2^+ for match h_2,
(c) h_2 is not generating for ρ_2 with respect to $\mathcal{I}_b$, and
(d) h_2 is generating for ρ_2 with respect to $\mathcal{I}_b \setminus h_1(\mathsf{head}(\rho_1))$.

Using $\prec^-$, it is possible to define a notion of stratification along the lines of Definition 5, and this is a refinement of the classical stratification in logic programming. Normal logic programs that are stratified in this sense have a unique stable model, also called the *perfect model*. Example 6 shows that this is not true for generating models on existential rules: both chase sequences follow the stratification by exhaustively applying rule (15) before rule (16). To obtain "perfect models" for existential rules, we incorporate core stratification.

Definition 11. *Consider a set Σ of rules with negation and a rule $\rho \in \Sigma$. Let $\rho{\downarrow^\square}$ be defined as in Definition 5, and let $\rho{\downarrow^-}$ be the set of all rules $\rho' \in \Sigma$ with $\rho' \, (\prec^+)^* \cdot \prec^- \rho$.*

A rule set Σ is fully stratified *if $\rho \notin \rho{\downarrow^\square} \cup \rho{\downarrow^-}$ for all $\rho \in \Sigma$. A* fully stratified chase sequence *is a standard chase that is based on the set of rules Σ^+ obtained from Σ by omitting the negative body $\bar{\chi}$ in each rule, and such that for all steps i*

(a) match $\mathsf{hom}[i]$ for $\mathsf{rule}[i]$ is generating w.r.t. $\mathcal{D}^{i-1}$,
(b) no rule in $(\mathsf{rule}[i]){\downarrow^\square} \cup (\mathsf{rule}[i]){\downarrow^-}$ is applicable (i.e., having a generating and unsatisfied match in $\mathcal{D}^{i-1}$),

where $\mathsf{rule}[i]$ *denotes the rule with negation that corresponds to the positive rule used in step* i.

As before, the relations $\prec^{\Box}$, $\prec^{+}$, and $\prec^{-}$ can be decided in Σ_2^{P}, leading to the following generalisation of Theorem 7.

Theorem 12. *Deciding if* Σ *is fully stratified is* Σ_2^{P}*-complete.*

There is a subtle difference between the notions of stratified chase in Definition 5 and Definition 11: the former merely imposes a condition on rules that were actually applied, while the latter requires certain rules to be applied exhaustively. The latter is therefore more restrictive, and we use this to show the uniqueness of fully stratified chase results below (Theorem 16). For positive rules, uniqueness follows from the uniqueness of cores, which is no longer given here.

Lemma 13. *In every fully stratified chase, if* $\mathsf{rule}[i]$ *is generating with respect to* $\mathcal{D}^{i-1}$ *then* $\mathsf{rule}[i]$ *is generating with respect to* $\mathcal{D}^{\infty}$*. Hence the chase is generating.*

Proof. Otherwise, if there would be some smallest $j \geq i$ such that $\mathsf{rule}[i]$ is not generating with respect to $\mathcal{D}^j$, then we would find $\mathsf{rule}[j] \prec^{-} \mathsf{rule}[i]$ using $\mathcal{I}_a = \mathcal{D}^i$ and $\mathcal{I}_b = \mathcal{D}^j$ in Definition 10. Using the definition of $\prec^{+}$, this can be shown to contradict Definition 11 (b). □

Lemma 14. *A fully stratified chase has no alternative matches.*

Proof. The proof is similar to that of Lemma 6. The added requirements that rules are generating for $\prec^{\Box}$ follow from Lemma 13. Assuming the existence of alternative matches therefore implies $\mathsf{rule}[i] \prec^{\Box} \mathsf{rule}[j]$ for some $j \leq i$. Using $\prec^{+}$, one can again show a contradiction to Definition 11 (b). □

The next result shows that the models produced by a fully stratified chase are in some sense most general among all generated models without alternative matches.

Lemma 15. *Let* $\mathcal{D}_1^{\infty}$ *be a fully stratified chase for* Σ *and* $\mathcal{D}$*, and let* $\mathcal{D}_2^{\infty}$ *be a generating chase for* Σ *and* $\mathcal{D}$ *without alternative matches. Then there is an embedding* $\mathcal{D}_1^{\infty} \to \mathcal{D}_2^{\infty}$.

Proof. We assume w.l.o.g. that the choice of fresh nulls in both chases is determined in the same way from the values of the head-variables in the match that first introduces them. With this assumption, we can show $\mathcal{D}_1^{\infty} \subseteq \mathcal{D}_2^{\infty}$, which yields the required embedding. Clearly, $\mathcal{D}_1^0 = \mathcal{D}_2^i = \mathcal{D} \subseteq \mathcal{D}_2^{\infty}$.

Now suppose for a contradiction that $\mathcal{D}_1^i \not\subseteq \mathcal{D}_2^{\infty}$ for some $i \geq 1$. Let k be the smallest such i. Then the match $\mathsf{hom}_1[k]$ of $\rho = \mathsf{rule}_1[k]$ is never applied with ρ in the second chase. Since $\mathcal{D}_1^{k-1} \subseteq \mathcal{D}_2^{\infty}$, $\mathsf{hom}_1[k]$ is a match for ρ in the second chase. Therefore, ρ either (i) is not generating with respect to $\mathcal{D}_2^{\infty}$ or (ii) is satisfied in $\mathcal{D}_2^{\infty}$ without being applied. Since $\mathcal{D}_1^{k-1} \subseteq \mathcal{D}_2^{\infty}$, there is a smallest $j > k$ such that either $\mathsf{rule}_2[j] \prec^{-} \rho$ for (i) or $\mathsf{rule}_2[j] \prec^{\Box} \rho$ for (ii). In particular, $\mathsf{rule}_2[j] \in (\mathsf{rule}[k])\!\downarrow^{\Box} \cup (\mathsf{rule}[k])\!\downarrow^{-}$.

We recursively define a set J of chase steps that were relevant for applying $\mathsf{rule}_2[j]$ as follows: $j \in J$ and, if $i \in J$ and there is an atom $\alpha \in \mathsf{body}_2[i]$ that was first introduced in $\mathcal{D}_2^{i'}$, then $i' \in J$. Then, for all $i \in J$, we find: (1) $\mathsf{rule}_2[i] \prec^{+*} \mathsf{rule}_2[j]$, hence (2) $\mathsf{rule}_2[i] \in (\mathsf{rule}[k])\!\downarrow^{\Box} \cup (\mathsf{rule}[k])\!\downarrow^{-}$; moreover (3) match $\mathsf{hom}_2[i]$ for $\mathsf{rule}_2[i]$ is generating with respect to $\mathcal{D}_2^{\infty}$ by Definition 9; (4) since $\mathcal{D}_1^{k-1} \subseteq \mathcal{D}_2^{\infty}$, match $\mathsf{hom}_2[i]$ is also generating in $\mathcal{D}_1^k$.

By Definition 11 (b), no rule in $(\mathsf{rule}[k])\!\downarrow^{\Box} \cup (\mathsf{rule}[k])\!\downarrow^{-}$ is applicable at step k (‡), which includes all rules $\mathsf{rule}_2[i]$, $i \in J$ by (2) above. We can now show that $\mathsf{head}_2[i] \subseteq \mathcal{D}_1^{k-1}$ for all $i \in J$. This is achieved by induction along the chase sequence restricted to rule applications J, which starts from $\mathcal{D}_2^0 = \mathcal{D}$. All relevant matches are generating by (4) above. By induction hypothesis, we have $\mathsf{body}_2[i] \subseteq \mathcal{D}_1^{k-1}$, i.e., that $\mathsf{hom}_2[i]$ is a match on $\mathcal{D}_1^{k-1}$. By (‡), $\mathsf{hom}_2[i]$ must be satisfied. Since $\mathcal{D}_1^{k-1} \subseteq \mathcal{D}_2^{\infty}$ and $\mathcal{D}_2^{\infty}$ has no alternative matches, $\mathsf{hom}_2[i]$ is satisfied by $\mathsf{head}_2[i] \subseteq \mathcal{D}_1^{k-1}$.

Therefore, we find that $\mathsf{head}_2[j] \subseteq \mathcal{D}_1^{k-1}$, contradicting the assumption that the application of $\mathsf{rule}_2[j]$ produces an atom that makes $\mathsf{rule}_1[k]$ (i) non-generating or (ii) satisfied in $\mathcal{D}_2^{\infty}$ but not in $\mathcal{D}_1^{\infty}$. □

Theorem 16. *All fully stratified chase sequences produce isomorphic results, which are generated models without alternative matches that can be embedded into any other chase with these properties. When finite, the result is a core.*

Proof. Whereas mutual embeddings do not guarantee isomorphism on infinite structures, the $\subseteq$ relation shown in the proof of Lemma 15 shows the claimed isomorphism. The other properties further follow from Lemmas 13 and 14, and Theorem 11. □

We may therefore call the unique, finite result of a fully stratified terminating chase sequence a *perfect core model*. To ensure the termination of the chase in the presence of negation, one can use any criterion for skolem-chase termination on the rules obtained by deleting negated atoms. Other criteria are applicable if they ensure standard chase termination on all subsets of a set of rules. Some criteria of this type can also take negation into account (Magka, Krötzsch, and Horrocks 2013; Baget et al. 2014).

Corollary 17. *Consider a set* Σ *of rules with negation and a database* $\mathcal{D}$*. If* Σ *is fully stratified and if the chase on* $\mathcal{D}$ *and any subset of* Σ^{+} *terminates for all strategies, then* Σ *has a fully stratified chase over* $\mathcal{D}$*, producing a perfect core model.*

Proof. A fully stratified chase can be constructed bottom up by applying only rules that satisfy the conditions in Definition 11. The rest of the claim follows from Theorem 16. □

Related Work Existential rules with negation are also called *normal tuple-generating dependencies* in the literature. Several prior works on non-monotonic negation for existential rules have used skolemisation to adopt an existing logic programming semantics. Calì, Gottlob, and Lukasiewicz (2012) use a notion of *indefinite grounding*, which leads to results that are isomorphic to skolemisation. Magka, Krötzsch, and Horrocks (2013) define relations similar to our $\prec^{+}$ and $\prec^{-}$ to identify decidable cases where stable models are finite or unique. Gottlob et al. (2014) study decidability for guarded rules, where stable models might be infinite. These

works face the general issues with skolemisation, as discussed in Example 1. Another approach of this type are FDNC rules, which impose a fixed syntactic form to ensure decidability (Eiter and Simkus 2010).

Several other works investigate approaches that avoid the interpretation of existential rules by skolem terms that are inherently distinct. Baget et al. (2014) introduce C-stable models that correspond to generated models based on a chase C. Both models in Example 6 are core-chase-stable and standard-chase-stable, but only $\mathcal{I}_1$ is a perfect core model.

Alviano, Morak, and Pieris (2017) use circumscription to generalise stable models to existential rules. The results are different from the entailments with perfect core models. For example, the authors argue that the rules in Example 6 with the only input fact $H(a)$ should still have $\mathcal{I}_1$ as a stable model (cf. Example 4 in their paper), whereas the unique perfect core model in this case would be $\{H(a), F(a,n), E(n,n), M(n)\}$. The latter entails **not** $F(a,b)$ while the former does not. Alviano et al. argue that this is desirable – we cannot know for sure that Bob is not the father of Alice –, yet it seems to be at odds with the usual approach to non-monotonic logic, where a statement is false "by default" unless there is some evidence in its favour. In this sense, their approach re-introduces some of the open-world reasoning typical for classical logics.

For the case of well-founded semantics, Gottlob et al. (2012) considered the option of allowing distinct constants and nulls to be equal. The result differs from our work in many aspects, and in particular does not seem to suggest any obvious correspondence even on fully stratified rule sets.

Expressivity and Complexity

We now discuss the expressive scope of the classes of rules that our approach encompasses, and the complexity of computing cores in these frameworks.

Many of our methods inherit complexity bounds from the standard chase. Deciding whether a rule has an applicable match is Σ_2^{P}-complete in this case (Grahne and Onet 2018), and remains the same if we check that rule matches are generating as in Definition 11. Nevertheless, there are rule sets for which the chase is finite on all databases, but of non-elementary size (Krötzsch, Marx, and Rudolph 2019). This result can be extended to the case of core-stratified rules.

Theorem 18. *There is a core-stratified rule set Σ that admits a finite, core-stratified chase on every database $\mathcal{D}$, but for which the size of the core model is not bounded by any elementary function in the size of $\mathcal{D}$.*

Proof. We adopt the rule set presented by Krötzsch, Marx, and Rudolph (2019, Theorem 14). Their solution consists of two parts: (1) a set of eight rules (labelled (27)–(34)) that generate a chase of non-elementary size, and (2) a set of Datalog rules that entails all possible atoms if a cycle is found in the database (a technique called "flooding"). Rules (2) are prioritised to ensure that existential rules from (1) can never be applied on cyclic inputs, since their matches will already be satisfied. On cycle-free inputs, rules (1) always lead to a finite chase, where every match is unsatisfied until it is applied. In other words: rules from (2) can block rules from (1), but rules from (1) never need to block other rules from (1). We can therefore apply the decomposition of Definition 6 to all rules in (1) and augment rules (2) to include the fresh predicates in the flooding step to obtain a core-stratified rule set that produces a chase of non-elementary size. □

In this sense, core-stratified rule sets are highly expressive, and the same applies to rules with negation under the perfect core model semantics, which generalise this case. Nevertheless, there are negation-free rule sets with finite cores that cannot be computed by a standard chase on any set of rules.

Theorem 19. *There is a rule set Σ that has a finite core model on every database, such that there is no rule set Σ' for which the same core models can be computed using a standard chase. This remains true even if Σ' can use auxiliary predicates that are ignored in the result.*

Proof. The rules used in Theorem 18 generate a linear order of non-elementary length, which we can use to simulate a non-elementary time-bounded Turing machine (TM) computation (Krötzsch, Marx, and Rudolph 2019). If the TM accepts, we use Datalog rules to derive all possible atoms for all terms and predicates, and an additional fact $\mathsf{Accept}(t)$ for all terms t. If it rejects, we do the same but for facts $\mathsf{Reject}(t)$. In either case, the core of the model contains only the elements in the given database (no nulls), marked by Accept or Reject. A standard chase that derives this result could only use Datalog rules, which can be done exhaustively in polynomial time with respect to the size of the database (Dantsin et al. 2001). This would decide non-elementary TM acceptance in PTime, which is not possible. □

Theorem 18 also shows the expressive power of the ASP encoding of Corollary 10. However, many ASP solvers use an exhaustive, skolemisation-based grounding phase, and it is known that skolemised rules that terminate on arbitrary inputs terminate after polynomially many steps (Marnette 2009). Therefore, ASP solvers that rely on full grounding can only compute core models that are polynomial with respect to the input, or must accept that non-termination may occur for some databases. ASP engines that apply partial grounding or other methods might be able to surpass this boundary.

Conclusions

We have shed new light on some of the properties of core models and their intricate relationship to the chase and non-monotonic negation. Surprisingly, the "local" computation process of the standard chase can often achieve the global optimality of a core. Exact characterisations and further practical criteria for this desirable situation are promising goals for future research. Similarly, while our results show a way of using these ideas in non-monotonic reasoning, they also raise many questions on how our perfect core models can be further characterised, generalised, or computed in ASP.

From a practical perspective, the main results are Theorem 8 and Corollaries 10 and 17, since they lead to concrete approaches that can be put into practice in modern reasoners. This opens the door to empirical studies of the practicality of these approaches in various systems, which might further inspire refined syntactic criteria and algorithms.

Acknowledgements The author thanks the anonymous reviewers of the 17th International Conference on Principles of Knowledge Representation and Reasoning (KR 2020) for detailed and insightful technical comments, as well as Stephan Mennicke and Ali Elhalawati for their feedback; all remaining errors are the author's. This work is partly supported by Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) in projects number 389792660 (TRR 248, Center for Perspicuous Systems) and KR 4381/1-1 (Emmy Noether grant DIAMOND), and 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).

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 et al. (2017), 377–388.

Baget, J.-F.; Leclère, M.; Mugnier, M.-L.; and Salvat, E. 2011. On rules with existential variables: Walking the decidability line. *Artificial Intelligence* 175(9–10):1620–1654.

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., *Proc. 15th Int. Workshop on Non-Monotonic Reasoning (NMR 2014)*, volume 1843-14-01 of *INFSYS Research Report*, 176–184. TU Wien.

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. *Proc. 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 et al. (2017), 37–52.

Calì, A.; Gottlob, G.; and Kifer, M. 2008. Taming the infinite chase: Query answering under expressive relational constraints. In Brewka, G., and Lang, J., eds., *Proc. 11th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR'08)*, 70–80. AAAI Press.

Calì, A.; Gottlob, G.; and Lukasiewicz, T. 2012. A general Datalog-based framework for tractable query answering over ontologies. *J. Web Semant.* 14:57–83.

Calì, A.; Gottlob, G.; and Pieris, A. 2010. Query answering under non-guarded rules in Datalog+/-. In Hitzler, P., and Lukasiewicz, T., eds., *Proc. 4th Int. Conf. on Web Reasoning and Rule Systems (RR 2010)*, volume 6333 of *LNCS*, 1–17. Springer.

Calì, A.; Gottlob, G.; and Pieris, A. 2012. Towards more expressive ontology languages: The query answering problem. *J. of Artif. Intell.* 193:87–128.

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., *Proc. 21st Int. Conf. on Database Theory (ICDT'18)*, volume 98 of *LIPIcs*, 12:1–12:19. Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik.

Carral, D.; Dragoste, I.; González, L.; Jacobs, C.; Krötzsch, M.; and Urbani, J. 2019a. VLog: A rule engine for knowledge graphs. In Ghidini et al., C., ed., *Proc. 18th Int. Semantic Web Conf. (ISWC'19, Part II)*, volume 11779 of *LNCS*, 19–35. Springer.

Carral, D.; Dragoste, I.; Krötzsch, M.; and Lewe, C. 2019b. Chasing sets: How to use existential rules for expressive reasoning. In Kraus, S., ed., *Proc. 28th Int. Joint Conf. on Artificial Intelligence (IJCAI'19)*, 1624–1631. ijcai.org.

Carral, D.; Dragoste, I.; and Krötzsch, M. 2017. Restricted chase (non)termination for existential rules with disjunctions. In Sierra, C., ed., *Proc. 26th Int. Joint Conf. on Artificial Intelligence (IJCAI'17)*, 922–928. ijcai.org.

Ceri, S.; Gottlob, G.; and Tanca, L. 1990. *Logic Programming and Databases*. Springer.

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. *J. of Artificial Intelligence Research* 47:741–808.

Dantsin, E.; Eiter, T.; Gottlob, G.; and Voronkov, A. 2001. Complexity and expressive power of logic programming. *ACM Computing Surveys* 33(3):374–425.

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

Eiter, T., and Simkus, M. 2010. FDNC: decidable nonmonotonic disjunctive logic programs with function symbols. *ACM Trans. Comput. Log.* 11(2):14:1–14:50.

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.

Geerts, F.; Mecca, G.; Papotti, P.; and Santoro, D. 2014. That's all folks! LLUNATIC goes open source. *PVLDB* 7(13):1565–1568.

Gottlob, G.; Hernich, A.; Kupke, C.; and Lukasiewicz, T. 2012. Equality-friendly well-founded semantics and applications to description logics. In Hoffmann, J., and Selman, B., eds., *Proc. 26th AAAI Conf. on Artificial Intelligence (AAAI'12)*. AAAI Press.

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., *Proc. 14th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR'14)*, 258–267. AAAI Press.

Grahne, G., and Onet, A. 2018. Anatomy of the chase. *Fundam. Inform.* 157(3):221–270.

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

Hernich, A. 2011. Answering non-monotonic queries in relational data exchange. *Logical Methods in Computer Science* 7(3).

Konczak, K.; Linke, T.; and Schaub, T. 2006. Graphs and colorings for answer set programming. *Theory Pract. Log. Program.* 6(1-2):61–106.

Krötzsch, M.; Marx, M.; and Rudolph, S. 2019. The power of the terminating chase. In Barceló, P., and Calautti, M., eds., *Proc. 22nd Int. Conf. on Database Theory (ICDT'19)*, volume 127 of *LIPIcs*, 3:1–3:17. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik.

Lifschitz, V. 2019. *Answer Set Programming*. Springer.

Magka, D.; Krötzsch, M.; and Horrocks, I. 2013. Computing stable models for nonmonotonic existential rules. In Rossi, F., ed., *Proc. 23rd Int. Joint Conf. on Artificial Intelligence (IJCAI'13)*, 1031–1038. AAAI Press/IJCAI.

Marnette, B. 2009. Generalized schema-mappings: from termination to tractability. In Paredaens and Su (2009), 13–22.

Paredaens, J., and Su, J., eds. 2009. *Proc. 28th Symposium on Principles of Database Systems (PODS'09)*. ACM.

Sallinger, E.; den Bussche, J. V.; and Geerts, F., eds. 2017. *Proc. 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., *Proc. 9th Int. Joint Conf. on Automated Reasoning (IJCAR'18)*, volume 10900 of *LNCS*, 680–688. Springer.