Proceedings of the 20th International Conference on Principles of Knowledge Representation and Reasoning
Main Track

Connecting Proof Theory and Knowledge Representation: Sequent Calculi and the Chase with Existential Rules

Tim S. Lyon[1] , **Piotr Ostropolski-Nalewaja**[1,2]
[1]Technische Universität Dresden
[2]University of Wrocław
{timothy_stephen.lyon, piotr.ostropolski-nalewaja}@tu-dresden.de

Abstract

Chase algorithms are indispensable in the domain of knowledge base querying, which enable the extraction of implicit knowledge from a given database via applications of rules from a given ontology. Such algorithms have proved beneficial in identifying logical languages which admit decidable query entailment. Within the discipline of proof theory, sequent calculi have been used to write and design proof-search algorithms to identify decidable classes of logics. In this paper, we show that the chase mechanism in the context of existential rules is in essence the same as proof-search in an extension of Gentzen's sequent calculus for first-order logic. Moreover, we show that proof-search generates universal models of knowledge bases, a feature also exhibited by the chase. Thus, we formally connect a central tool for establishing decidability proof-theoretically with a central decidability tool in the context of knowledge representation.

1 Introduction

Existential Rules and the Chase. The formalism of existential rules is a significant sub-discipline within the field of knowledge representation, offering insightful results within the domain of ontology-based query answering (Baget et al. 2009), data exchange and integration (Fagin et al. 2005), and serving a central role in the study of generic decidability criteria (Feller et al. 2023).[1] Ontology-based query answering is one of the principal problems studied within the context of existential rules, and asks if a query is logically entailed by a given knowledge base (KB) $\mathcal{K} = (\mathcal{D}, \mathcal{R})$, where $\mathcal{D}$ is a database and $\mathcal{R}$ is a finite set of existential rules (Baget et al. 2011). Databases generally consist of positive atomic facts such as $Female(Marie)$ or $Mother(Zuza, Marie)$, while existential rules—which are first-order formulae of the form $\forall \boldsymbol{x}\boldsymbol{y}\beta(\boldsymbol{x}, \boldsymbol{y}) \rightarrow \exists \boldsymbol{z}\alpha(\boldsymbol{y}, \boldsymbol{z})$ with β and α conjunctions of atoms—are used to encode a logical theory or ontology that permits the extraction of implicit knowledge from the encompassing KB.

The primary tool for studying query answering within this setting is the so-called *chase*, an algorithm that iteratively saturates a given database under applications of existential rules (Beeri and Vardi 1984). The chase is useful in that it generates a *universal model* satisfying exactly those queries entailed by a KB, and thus, allows for the reduction of query entailment to query checking over the constructed universal model (Deutsch, Nash, and Remmel 2008). In this paper, we show how the chase corresponds to proof-search in an extension of Gentzen's sequent calculus, establishing a connection between a central tool in the theory of existential rules with the primary decidability tool in proof theory.

Sequent Calculi and Proof-Search. Since its introduction, Gentzen's sequent formalism (Gentzen 1935a; Gentzen 1935b) has become one of the preferred proof-theoretic frameworks for the creation and study of proof calculi. A sequent is an object of the form $\Gamma \vdash \Delta$ such that Γ and Δ are finite (multi)sets of logical formulae, and a sequent calculus is a set of inference rules that operate over such. Sequent systems, and generalizations thereof, have proved beneficial in establishing (meta)logical properties with a diverse number of applications, being used to write decision algorithms (Dyckhoff 1992; Slaney 1997), to calculate interpolants (Maehara 1960; Lyon et al. 2020), and have even been applied in knowledge intergation scenerios (Lyon and Gómez Álvarez 2022).

It is well-known that *geometric implications*, i.e. first-order formula of the form $\forall \boldsymbol{x}(\varphi \rightarrow \exists \boldsymbol{y}_1\psi_1 \lor \cdots \lor \exists \boldsymbol{y}_n\psi_n)$ with φ and ψ_i conjunctions of atoms, can be converted into an inference rules in a sequent calculus (Simpson 1994, p. 24). Since such formulae subsume the class of existential rules, we may leverage this insight to extend Gentzen's sequent calculus for first-order logic with such rules to carry out existential rule reasoning. When we do so, we find that sequent systems mimic existential rule reasoning and proof-search (described below) simulates the chase.

Proof-search is the central means by which decidability is obtained with a sequent calculus, and usually operates by applying the inference rules of a sequent calculus bottom-up on an input sequent with the goal of constructing a proof thereof. If a proof of the input is found, the input is confirmed to be valid, and if a proof of the input is not found, a counter-model can typically be extracted witnessing the invalidity of the input. We make the novel observation that counter-models extracted from proof-search (in the context of existential rules) are universal, being homomorphically

[1]Existential rules are also referred to as a *tuple-generating dependencies* (Abiteboul, Hull, and Vianu 1995), *conceptual graph rules* (Salvat and Mugnier 1996), Datalog$^{\pm}$ (Gottlob 2009), and $\forall\exists$-*rules* (Baget et al. 2011) in the literature.

equivalent to the universal model generated by the chase.

Contributions. Our contributions in this paper are as follows: (1) We establish a strong connection between tools in the domain of existential rules with that of proof theory; in particular, we show how to transform derivations with existential rules into sequent calculus proofs and vice versa. (2) We establish a correspondence between the chase and sequent-based proof-search, and (3) we recognize that proof-search, like the chase, generates universal models for knowledge bases, which is a novel, previously unknown insight regarding the capability of sequent systems.

Organization. The preliminaries are located in Section 2. In Section 3, we present the sequent calculus framework and write a proof-search algorithm that simulates the chase. Correspondences between existential rule reasoning and sequent-based reasoning are explicated in Section 4, and in Section 5, we conclude and discuss future research. The reader may consult the appended version of this paper (Lyon and Ostropolski-Nalewaja 2023) for proofs of claims.

2 Preliminaries and Existential Rules

Formulae and Syntax. We let $\mathbf{C}$ and $\mathbf{V}$ be two disjoint denumerable sets of *constants* and *variables*. We use $a, b, c, \dots$ to denote constants and $x, y, z, \dots$ to denote variables. We define the set of *terms* to be $\mathbf{T} = \mathbf{C} \cup \mathbf{V}$, and we denote terms by t and annotated versions thereof. Moreover, we let $\mathbf{P} = \{p, q, r, \dots\}$ be a denumerable set of *predicates* containing denumerably many predicates of each arity $n \in \mathbb{N}$, and use $ar(p) = n$ to denote that $p \in \mathbf{P}$ is of arity n. An *atom* is a formula of the form $p(t_1, \dots, t_n)$ such that $t_1, \dots, t_n \in \mathbf{T}$ and $ar(p) = n$. We will often write atoms as $p(\boldsymbol{t})$ with $\boldsymbol{t} = t_1, \dots, t_n$. The *first-order language* $\mathcal{L}$ is defined via the following grammar in Backus–Naur form:

$$\varphi ::= p(\boldsymbol{t}) \mid \neg\varphi \mid \varphi \wedge \varphi \mid \exists x \varphi$$

such that $p \in \mathbf{P}$, $\boldsymbol{t} \in \mathbf{T}$, and $x \in \mathbf{V}$. We use φ, ψ, χ, ... to denote *formulae* from $\mathcal{L}$, and define $\varphi \vee \psi := \neg(\neg\varphi \wedge \neg\psi)$, $\varphi \rightarrow \psi := \neg\varphi \vee \psi$, and $\forall x \varphi := \neg\exists x \neg\varphi$. The occurrence of a variable is *free* in a formula φ when it does not occur within the scope of a quantifier. We let $\varphi(t/x)$ represent the formula obtained by substituting the term t for every free occurrence of the variable x in φ. We use Γ, Δ, Σ, ... to denote sets of formulae from $\mathcal{L}$, let $\mathbf{V}(\Gamma)$ denote the set of free variables in the formulae of Γ, and let $\mathbf{T}(\Gamma)$ denote the set of free variables and constants occurring in the formulae of Γ. We let $i \in [n]$ represent $1 \leq i \leq n$, and define a *ground atom* to be an atom $p(t_1, \dots, t_n)$ such that for each $i \in [n]$, $t_i \in \mathbf{C}$. An *instance* $\mathcal{I}$ is defined to be a (potentially infinite) set of atoms, and a *database* $\mathcal{D}$ is defined to be a finite set of ground atoms. We let $\top$ be a special unary predicate and define $\mathcal{I}^{\top} = \mathcal{I} \cup \{\top(c) \mid c \in \mathbf{C}\}$. An instance $\mathcal{I}$ is referred to as an *interpretation iff* $\mathcal{I}^{\top} = \mathcal{I}$.

Substitutions. A *substitution* σ is defined to be a partial function over $\mathbf{T}$. A *homomorphism* from an instance $\mathcal{I}$ to an instance $\mathcal{J}$ is a substitution π from the terms of $\mathcal{I}$ to the terms of $\mathcal{J}$ such that (1) if $p(t_1, \dots, t_n) \in \mathcal{I}$, then $p(\pi(t_1), \dots, \pi(t_n)) \in \mathcal{J}$, and (2) $\pi(a) = a$, for each $a \in \mathbf{C}$. We say that an instance $\mathcal{I}$ *homomorphically maps* into an instance $\mathcal{J}$ *iff* a homomorphism exists from $\mathcal{I}$ to $\mathcal{J}$. Two instances $\mathcal{I}$ and $\mathcal{J}$ are defined to be *homomorphically equivalent*, written $\mathcal{I} \equiv \mathcal{J}$, *iff* each instance can be homomorphically mapped into the other. An $\mathcal{I}$*-assignment* is defined to be a substitution μ such that (1) $\mu(x) \in \mathbf{T}(\mathcal{I})$, for each $x \in \mathbf{V}$, and (2) $\mu(a) = a$, for each $a \in \mathbf{C}$. For an $\mathcal{I}$-assignment μ, we let $\mu(\varphi)$ denote the formula obtained by replacing each free variable of φ with its value under μ, and we let $\mu[\boldsymbol{t}/\boldsymbol{x}]$ be the same as μ, but where the variables $\boldsymbol{x}$ are respectively mapped to $\boldsymbol{t} \in \mathbf{T}$.

Models and Satisfaction. Given an interpretation $\mathcal{I}$ and an $\mathcal{I}$-assignment μ, we recursively define satisfaction $\models$ as:

(1) $\mathcal{I}, \mu \models p(t_1, \dots, t_n)$ *iff* $p(\mu(t_1), \dots, \mu(t_n)) \in \mathcal{I}$;
(2) $\mathcal{I}, \mu \models \neg\varphi$ *iff* $\mathcal{I}, \mu \not\models \varphi$;
(3) $\mathcal{I}, \mu \models \varphi \wedge \psi$ *iff* $\mathcal{I}, \mu \models \varphi$ and $\mathcal{I}, \mu \models \psi$;
(4) $\mathcal{I}, \mu \models \exists x \varphi$ *iff* $t \in \mathbf{T}(\mathcal{I})$ exists and $\mathcal{I}, \mu[t/x] \models \varphi$.

We say that $\mathcal{I}$ is a *model* of Γ and write $\mathcal{I} \models \Gamma$ *iff* for every $\varphi \in \Gamma$ and $\mathcal{I}$-assignment μ, we have $\mathcal{I}, \mu \models \varphi$. We define an instance $\mathcal{I}$ to be a *universal model* of Γ *iff* for any model $\mathcal{J}$ of Γ there exists a homomorphism from $\mathcal{I}$ to $\mathcal{J}$.

Existential Rules. An *existential rule* is a first-order formula $\rho = \forall \boldsymbol{x}\boldsymbol{y}\, \beta(\boldsymbol{x}, \boldsymbol{y}) \rightarrow \exists \boldsymbol{z}\, \alpha(\boldsymbol{y}, \boldsymbol{z})$ such that $\beta(\boldsymbol{x}, \boldsymbol{y}) = \mathrm{body}(\rho)$ (called the body) and $\alpha(\boldsymbol{y}, \boldsymbol{z}) = \mathrm{head}(\rho)$ (called the head) are conjunctions of atoms over constants and the variables $\boldsymbol{x}, \boldsymbol{y}$ and $\boldsymbol{y}, \boldsymbol{z}$, respectively. We call a finite set $\mathcal{R}$ of existential rules a *rule set*. We define Γ to be $\mathcal{R}$*-valid iff* for every interpretation $\mathcal{I}$, if $\mathcal{I} \models \mathcal{R}$, then $\mathcal{I} \models \Gamma$.

Derivations and the Chase. We say that an existential rule ρ is *applicable* to an instance $\mathcal{I}$ *iff* there exists an $\mathcal{I}$-assignment μ such that $\mu(\beta(\boldsymbol{x}, \boldsymbol{y})) \subseteq \mathcal{I}$, and when this is the case, we say that $\tau = (\rho, \mu)$ is a *trigger* in $\mathcal{I}$. Given a trigger $\tau = (\rho, \mu)$ in $\mathcal{I}$ we define an *application* of the trigger τ to the instance $\mathcal{I}$ to be the instance $\tau(\mathcal{I}) = \mathcal{I} \cup \alpha(\mu(\boldsymbol{y}), \boldsymbol{z})$ where $\boldsymbol{z}$ is a tuple of fresh variables. We define a *chase derivation* $(\mathcal{I}_i, \tau_i)_{i \in [n]}$ to be a sequence $(\mathcal{I}_1, \tau_1), \dots, (\mathcal{I}_n, \tau_n), (\mathcal{I}_{n+1}, \emptyset)$ such that for every $i \in [n]$, τ_i is a trigger in $\mathcal{I}_i$ and $\tau_i(\mathcal{I}_i) = \mathcal{I}_{i+1}$. For an instance $\mathcal{I}$ and a rule set $\mathcal{R}$, we define the *one-step chase* to be:

$$\mathbf{Ch}_1(\mathcal{I}, \mathcal{R}) = \bigcup_{\tau \text{ is a trigger in } \mathcal{I}} \tau(\mathcal{I}).$$

We let $\mathbf{Ch}_0(\mathcal{I}, \mathcal{R}) = \mathcal{I}$ as well as let $\mathbf{Ch}_{n+1}(\mathcal{I}, \mathcal{R}) = \mathbf{Ch}_1(\mathbf{Ch}_n(\mathcal{I}, \mathcal{R}), \mathcal{R})$. Finally, we define the *chase* to be $\mathbf{Ch}_\infty(\mathcal{I}, \mathcal{R}) = (\bigcup_{i \in \mathbb{N}} \mathbf{Ch}_i(\mathcal{I}, \mathcal{R}))^{\top}$, which serves as a universal model of $\mathcal{I} \cup \mathcal{R}$ (Deutsch, Nash, and Remmel 2008).[2]

Queries and Entailment. A *Boolean conjunctive query (or, BCQ)* is a formula $\exists \boldsymbol{x} q(\boldsymbol{x})$ such that $q(\boldsymbol{x})$ is a conjunction of atoms over the variables $\boldsymbol{x}$ and constants. We define a *knowledge base (or, KB)* to be an ordered pair $\mathcal{K} = (\mathcal{D}, \mathcal{R})$ with $\mathcal{D}$ a database and $\mathcal{R}$ a rule set, and let $\mathcal{I}$ be a *model* of $\mathcal{K}$, written $\mathcal{I} \models \mathcal{K}$, *iff* $\mathcal{I} \models \mathcal{D} \cup \mathcal{R}$. We write $\mathcal{K} \models \exists \boldsymbol{x} q(\boldsymbol{x})$ to mean that for every $\mathcal{I}$, if $\mathcal{I} \models \mathcal{K}$, then $\mathcal{I} \models \exists \boldsymbol{x} q(\boldsymbol{x})$. A chase derivation $(\mathcal{I}_i, \tau_i)_{i \in [n]}$ *witnesses* $(\mathcal{D}, \mathcal{R}) \models \exists \boldsymbol{x} q(\boldsymbol{x})$

[2] We use a *restricted* variant of the chase; cf. (Fagin et al. 2005).

$$\frac{}{\Gamma, p(\boldsymbol{t}) \vdash p(\boldsymbol{t}), \Delta}\ (id) \qquad \frac{\Gamma \vdash \varphi, \Delta}{\Gamma, \neg\varphi \vdash \Delta}\ (\neg_L) \qquad \frac{\Gamma, \varphi \vdash \Delta}{\Gamma \vdash \neg\varphi, \Delta}\ (\neg_R) \qquad \frac{\Gamma, \varphi, \psi \vdash \Delta}{\Gamma, \varphi \wedge \psi \vdash \Delta}\ (\wedge_L)$$

$$\frac{\Gamma \vdash \varphi, \Delta \qquad \Gamma \vdash \psi, \Delta}{\Gamma \vdash \varphi \wedge \psi, \Delta}\ (\wedge_R) \qquad \frac{\Gamma, \varphi(y/x) \vdash \Delta}{\Gamma, \exists x\varphi \vdash \Delta}\ (\exists_L)\ y \text{ fresh} \qquad \frac{\Gamma \vdash \exists x\varphi, \varphi(t/x), \Delta}{\Gamma \vdash \exists x\varphi, \Delta}\ (\exists_R)\ t \in \mathbf{T}$$

Figure 1: The sequent calculus G3 for first-order logic.

iff $\mathcal{I}_1 = \mathcal{D}$, only rules from $\mathcal{R}$ are applied, and there exists an $\mathcal{I}_{n+1}$-assignment μ such that $\mu(q(\boldsymbol{x})) \subseteq \mathcal{I}_{n+1}$.

3 Sequent Systems and Proof-Search

We define a *sequent* to be an object of the form $\Gamma \vdash \Delta$ such that Γ and Δ are *finite* sets of formulae from $\mathcal{L}$. Typically, multisets are used in sequents rather than sets, however, we are permitted to use sets in the setting of classical logic; cf. (Kleene 1952). For a sequent $\Gamma \vdash \Delta$, we call Γ the *antecedent* and Δ the *consequent*. We define the *formula interpretation* of a sequent to be $f(\Gamma \vdash \Delta) = \bigwedge \Gamma \rightarrow \bigvee \Delta$.

The sequent calculus G3 (Kleene 1952) for first-order logic is defined to be the set of inference rules presented in Figure 1. It consists of the *initial rule* (id) along with *logical rules* that introduce complex logical formulae in either the antecedent or consequent of a sequent. The $(\exists_L)$ rule is subject to a side condition, stating that the rule is applicable only if y is *fresh*, i.e. y does not occur in the surrounding context Γ, Δ. The $(\exists_R)$ rule allows for the bottom-up instantiation of an existentially quantified formula with a term t. An *application* of a rule is obtained by instantiating the rule with formulae from $\mathcal{L}$. We call an application of a rule *top-down* (*bottom-up*) whenever the conclusion (premises) is (are) obtained from the premises (conclusion).

It is well-known that every *geometric implication*, which is a formula of the form $\forall \boldsymbol{x}(\varphi \rightarrow \exists \boldsymbol{y}_1\psi_1 \vee \cdots \vee \exists \boldsymbol{y}_n\psi_n)$ with φ and ψ_i conjunctions of atoms, can be converted into an inference rule; see (Simpson 1994, p. 24) for a discussion. We leverage this insight to transform existential rules (which are special instances of geometric implications) into inference rules that can be added to the sequent calculus G3. For an existential rule $\rho = \forall \boldsymbol{x}\boldsymbol{y}\beta(\boldsymbol{x}, \boldsymbol{y}) \rightarrow \exists \boldsymbol{z}\alpha(\boldsymbol{y}, \boldsymbol{z})$, we define its corresponding *sequent rule* $s(\rho)$ to be:

$$\frac{\Gamma, \beta(\boldsymbol{x}, \boldsymbol{y}), \alpha(\boldsymbol{y}, \boldsymbol{z}) \vdash \Delta}{\Gamma, \beta(\boldsymbol{x}, \boldsymbol{y}) \vdash \Delta}\ s(\rho)\ \boldsymbol{z} \text{ fresh}$$

Note that we take the body $\beta(\boldsymbol{x}, \boldsymbol{y})$ and head $\alpha(\boldsymbol{y}, \boldsymbol{z})$ to be sets of atoms, rather than conjunctions of atoms, and we note that $\boldsymbol{x}, \boldsymbol{y}$ may be instantiated with terms in rule applications. Also, $s(\rho)$ is subject to the side condition that the rule is applicable only if all variables $\boldsymbol{z}$ are fresh. We define the sequent calculus $\mathsf{G3}(\mathcal{R}) = \mathsf{G3} \cup \{s(\rho) \mid \rho \in \mathcal{R}\}$. We define a *derivation* to be any sequence of applications of rules in $\mathsf{G3}(\mathcal{R})$ to arbitrary sequents, define an $\mathcal{R}$-*derivation* to be a derivation that only applies $s(\rho)$ rules, and define a *proof* to be a derivation starting from applications of the (id) rule. An example of a proof is shown on the left side of Figure 2.

Theorem 1 (Soundness and Completeness)**.** *$f(\Gamma \vdash \Delta)$ is $\mathcal{R}$-valid iff there exists a proof of $\Gamma \vdash \Delta$ in $\mathsf{G3}(\mathcal{R})$.*

Algorithm 1 The proof-search algorithm `Prove`.

Input: A sequent $\Gamma \vdash \Delta$.
Output: A Boolean `True` and `False`.

If no rule is applicable to $\Gamma \vdash \Delta$, **Return** `False`;
If there exists a $p(\boldsymbol{t}) \in \Gamma \cap \Delta$, **Return** `True`;
If $\varphi \wedge \psi \in \Delta$, but $\varphi, \psi \notin \Delta$,
  Set $\Delta_1 := \varphi, \Delta$ and $\Delta_2 := \psi, \Delta$;
  If `Prove`$(\Gamma \vdash \Delta_i) =$ `False` for some $i \in \{1, 2\}$,
    Return `False`;
  Else
    Return `True`;
If $\exists x\varphi \in \Delta$ and $t \in \mathbf{T}(\Gamma)$, but $\varphi(t/x) \notin \Delta$,
  Set $\Delta := \varphi(t/x), \Delta$; **Return** `Prove`$(\Gamma \vdash \Delta)$;
Let $\rho = \forall \boldsymbol{x}\boldsymbol{y}\ \beta(\boldsymbol{x}, \boldsymbol{y}) \rightarrow \exists \boldsymbol{z}\ \alpha(\boldsymbol{y}, \boldsymbol{z})$ be the next rule according to $\prec$ (if no rule has yet been picked, choose one in $\mathcal{R}$);
If a Γ-assignment μ exists such that $\mu(\beta(\boldsymbol{x}, \boldsymbol{y})) \subseteq \Gamma$, but no terms $\boldsymbol{t} \in \mathbf{T}(\Gamma)$ exist such that $\mu[\boldsymbol{t}/\boldsymbol{z}](\alpha(\boldsymbol{y}, \boldsymbol{z})) \subseteq \Gamma$;
  Set $\Gamma := \alpha(\mu(\boldsymbol{y}), \boldsymbol{z}), \Gamma$ with $\boldsymbol{z}$ fresh;
  Return `Prove`$(\Gamma \vdash \Delta)$.

We now define a proof-search algorithm that decides (under certain conditions) if a BCQ is entailed by a knowledge base. The algorithm `Prove` (shown above) takes a sequent of the form $\mathcal{D} \vdash \exists \boldsymbol{x}q(\boldsymbol{x})$ as input and bottom-up applies inference rules from $\mathsf{G3}(\mathcal{R})$ with the goal of constructing a proof thereof. Either, `Prove` returns a proof witnessing that $(\mathcal{D}, \mathcal{R}) \models \exists \boldsymbol{x}q(\boldsymbol{x})$, or a counter-model to this claim can be extracted from failed proof search. Due to the shape of the input $\mathcal{D} \vdash \exists \boldsymbol{x}q(\boldsymbol{x})$, only (id), $(\wedge_R)$, $(\exists_R)$, and $s(\rho)$ rules are applicable during proof search. We note that a rule during proof-search is applicable to a sequent whenever a bottom-up application of the rule will introduce a new formula into the sequent. Moreover, we let $\prec$ be an arbitrary cyclic order over $\mathcal{R} = \{\rho_1, \ldots, \rho_n\}$, that is, $\rho_1 \prec \rho_2 \cdots \rho_{n-1} \prec \rho_n \prec \rho_1$. We use $\prec$ to ensure the *fair application* of $s(\rho)$ rules during proof-search, meaning that no bottom-up rule application is delayed indefinitely.

Theorem 2. *Let $\mathcal{R}$ be a rule set, $\mathcal{D}$ be a database, and $\exists \boldsymbol{x}q(\boldsymbol{x})$ be a BCQ. Then,*

1. *If* `Prove`$(\mathcal{D} \vdash \exists \boldsymbol{x}q(\boldsymbol{x})) =$ `True`*, then a proof in* $\mathsf{G3}(\mathcal{R})$ *can be constructed witnessing that* $(\mathcal{D}, \mathcal{R}) \models \exists \boldsymbol{x}q(\boldsymbol{x})$*;*
2. *If* `Prove`$(\mathcal{D} \vdash \exists \boldsymbol{x}q(\boldsymbol{x})) \neq$ `True`*, then a universal model can be constructed witnessing that* $(\mathcal{D}, \mathcal{R}) \not\models \exists \boldsymbol{x}q(\boldsymbol{x})$.

We refer to the universal model of $(\mathcal{D}, \mathcal{R})$ stated in the second claim of Theorem 2 as the *witnessing counter-model*.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma \vdash \exists x(\mathtt{A}(x,a) \wedge \mathtt{F}(x)), \mathtt{A}(c,a)}\,(id) \qquad \dfrac{}{\Gamma \vdash \exists x(\mathtt{A}(x,a) \wedge \mathtt{F}(x)), \mathtt{F}(c)}\,(id)}{\Gamma \vdash \exists x(\mathtt{A}(x,a) \wedge \mathtt{F}(x)), \mathtt{A}(c,a) \wedge \mathtt{F}(c)}\,(\wedge_R)}{\mathtt{M}(b,a), \mathtt{A}(b,a), \mathtt{F}(b), \mathtt{M}(c,b), \mathtt{A}(c,b), \mathtt{F}(c), \mathtt{A}(c,a) \vdash \exists x(\mathtt{A}(x,a) \wedge \mathtt{F}(x))}\,(\exists_R)}{\mathtt{M}(b,a), \mathtt{A}(b,a), \mathtt{F}(b), \mathtt{M}(c,b), \mathtt{A}(c,b), \mathtt{F}(c) \vdash \exists x(\mathtt{A}(x,a) \wedge \mathtt{F}(x))}\,s(\rho_2)}{\mathtt{M}(b,a), \mathtt{A}(b,a), \mathtt{F}(b), \mathtt{M}(c,b) \vdash \exists x(\mathtt{A}(x,a) \wedge \mathtt{F}(x))}\,s(\rho_1)}{\mathtt{M}(b,a), \mathtt{M}(c,b) \vdash \exists x(\mathtt{A}(x,a) \wedge \mathtt{F}(x))}\,s(\rho_1)$$

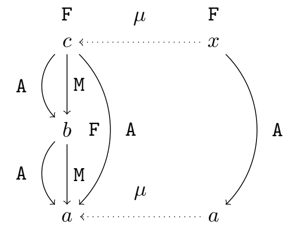

Figure 2: Above left is a proof in $\mathsf{G3}(\mathcal{R})$ witnessing that $\mathcal{K} \models \exists x(\mathtt{A}(x,a) \wedge \mathtt{F}(x))$, where $\mathcal{K} = (\mathcal{D}, \mathcal{R})$ is as defined in Example 10 and $\Gamma = \mathtt{M}(b,a), \mathtt{A}(b,a), \mathtt{F}(b), \mathtt{M}(c,b), \mathtt{A}(c,b), \mathtt{F}(c), \mathtt{A}(c,a)$. Above right is an illustration showing that the BCQ $\exists x(\mathtt{A}(x,a) \wedge \mathtt{F}(x))$ (to the right) can be mapped into the chase $\mathbf{Ch}_\infty(\mathcal{D}, \mathcal{R})$ (to the left) via the $\mathbf{Ch}_\infty(\mathcal{D}, \mathcal{R})$ -assignment μ (dotted arrows).

4 Simulations and Equivalences

We present a sequence of results which culminate in the establishment of two main theorems: (1) Theorem 8, which confirms that chase derivations are mutually transformable with certain proofs in $\mathsf{G3}(\mathcal{R})$, and (2) Theorem 9, which confirms an equivalence between `Prove` and the chase. We end the section by providing an example illustrating the latter correspondence between proofs and the chase.

Observation 3. *Let $\mathcal{R}$ be a rule set. If $\rho \in \mathcal{R}$, then any application of $(\wedge_R)$ and $(\exists_R)$ permute above $s(\rho)$.*

Proof. It is straightforward to confirm the permutation of such rules as the $s(\rho)$ rules operate on the antecedent of a sequent, and $(\wedge_R)$ and $(\exists_R)$ operate on the consequent. □

Observation 4. *If $\mathcal{I}$ is an instance, then only $s(\rho)$ rules of $\mathsf{G3}(\mathcal{R})$ can be bottom-up applied to $\mathcal{I} \vdash \emptyset$. Moreover, such an application yields a sequent $\mathcal{I}' \vdash \emptyset$ with $\mathcal{I}'$ an instance.*

Observation 5. *The inference shown below left is a correct application of $s(\rho)$ iff the inference shown below right is:*

$$\dfrac{\Gamma' \vdash \emptyset}{\Gamma \vdash \emptyset}\, s(\rho) \qquad \dfrac{\Gamma' \vdash \Delta}{\Gamma \vdash \Delta}\, s(\rho)$$

Observation 6. *Let $\mathcal{I}$ and $\mathcal{I}'$ be instances with $\tau = (\rho, \mu)$ a trigger on $\mathcal{I}$. Then, $(\mathcal{I}, \tau), (\mathcal{I}', \emptyset)$ is a chase derivation iff the following is a correct application of $s(\rho)$:*

$$\dfrac{\mathcal{I}' \vdash \emptyset}{\mathcal{I} \vdash \emptyset}\, s(\rho)$$

Lemma 7. *For every rule set $\mathcal{R}$, $n \in \mathbb{N}$, and instances $\mathcal{I}_1, \ldots, \mathcal{I}_n$ there exists a chase derivation $(\mathcal{I}_i, \tau_i)_{i \in [n-1]}$ iff there exists an $\mathcal{R}$-derivation of $\mathcal{I}_1 \vdash \emptyset$ from $\mathcal{I}_n \vdash \emptyset$.*

To prove the following theorem, one shows that every chase derivation can be transformed into a proof in $\mathsf{G3}(\mathcal{R})$ and vice-versa.

Theorem 8. *Let $\mathcal{R}$ be a rule set. A chase derivation $(\mathcal{I}_i, \tau_i)_{i \in [n]}$ witnessing $(\mathcal{D}, \mathcal{R}) \models \exists \boldsymbol{x} q(\boldsymbol{x})$ exists iff a proof in $\mathsf{G3}(\mathcal{R})$ of $\mathcal{D} \vdash \exists \boldsymbol{x} q(\boldsymbol{x})$ exists.*

Leveraging Theorems 2 and 8, it is straightforward to prove the first claim of the theorem below. The second claim is immediate as $\mathcal{I}$ and $\mathbf{Ch}_\infty(\mathcal{D}, \mathcal{R})$ are universal models.

Theorem 9. *Let $\mathcal{R}$ be a rule set, $\mathcal{D}$ be a database, and $\exists \boldsymbol{x} q(\boldsymbol{x})$ be a BCQ. Then,*

1. $\mathtt{Prove}(\mathcal{D} \vdash \exists \boldsymbol{x} q(\boldsymbol{x})) = \mathtt{True}$ *iff there is an $n \in \mathbb{N}$ such that $\mathbf{Ch}_n(\mathcal{D}, \mathcal{R}) \models \exists \boldsymbol{x} q(\boldsymbol{x})$ iff $\mathbf{Ch}_\infty(\mathcal{D}, \mathcal{R}) \models \exists \boldsymbol{x} q(\boldsymbol{x})$;*
2. *If* $\mathtt{Prove}(\mathcal{D} \vdash \exists \boldsymbol{x} q(\boldsymbol{x})) \neq \mathtt{True}$ *, then $\mathcal{I} \equiv \mathbf{Ch}_\infty(\mathcal{D}, \mathcal{R})$ with $\mathcal{I}$ the witnessing counter-model.*

Example 10. We provide an example demonstrating the relationship between a proof and the chase. We read $\mathtt{F}(x)$ as 'x is female', $\mathtt{M}(x,y)$ as 'x is the mother of y' and $\mathtt{A}(x,y)$ as 'x is the ancestor of y'. We let $\mathcal{K} = (\mathcal{D}, \mathcal{R})$ be a knowledge base such that $\mathcal{D} = \{\mathtt{M}(b,a), \mathtt{M}(c,b)\}$, $\mathcal{R} = \{\rho_1, \rho_2\}$, and

$$\rho_1 = \forall xy(\mathtt{M}(x,y) \rightarrow \mathtt{A}(x,y) \wedge \mathtt{F}(x));$$
$$\rho_2 = \forall xy(\mathtt{A}(x,y) \wedge \mathtt{A}(y,z) \rightarrow \mathtt{A}(x,z)).$$

In Figure 2, $\mathcal{K} \models \exists x(\mathtt{A}(x,a) \wedge \mathtt{F}(x))$ is witnessed and verified by the proof shown left. The graph shown right demonstrates that the BCQ $\exists x(\mathtt{A}(x,a) \wedge \mathtt{F}(x))$ (to the right) can be mapped into the chase $\mathbf{Ch}_\infty(\mathcal{D}, \mathcal{R})$ (to the left) via a $\mathbf{Ch}_\infty(\mathcal{D}, \mathcal{R})$-assignment μ (depicted as dotted arrows). (NB. We have omitted the points $\{\top(c) \mid c \in \mathbf{C}\}$ in the picture of $\mathbf{Ch}_\infty(\mathcal{D}, \mathcal{R})$ for simplicity.)

5 Concluding Remarks

We have formally established an equivalence between existential rule reasoning and sequent calculus proofs, effectively showing that proof-search simulates the chase. This work is meaningful as it uncovers and connects two central reasoning tasks and tools in the domain of existential rules and proof theory. Moreover, we have found that the counter-models extracted from failed proof-search are universal, implying their homomorphic equivalence to the chase—a previously unrecognized observation.

For future work, we aim to examine the relationship between the *disjunctive chase* (Bourhis et al. 2016) and proof-search in sequent calculi with disjunctive inference rules. It may additionally be worthwhile to investigate if our sequent systems can be adapted to facilitate reasoning with non-classical variants or extensions of existential rules. For example, we could merge our sequent calculi with those of (Lyon and Gómez Álvarez 2022) for *standpoint logic*—a modal logic used in knowledge integration to reason with diverse and potentially conflicting knowledge sources (Gómez Álvarez and Rudolph 2021). Finally, as this paper presents a sequent calculus for querying with existential rules, we plan to further explore its utility; e.g. by identifying admissible rules or applying loop checking techniques to uncover new classes of existential rules with decidable query entailment.

Acknowledgments

Work supported by the European Research Council (ERC) Consolidator Grant 771779 (DeciGUT).

References

Abiteboul, S.; Hull, R.; and Vianu, V. 1995. *Foundations of Databases*. Addison-Wesley.

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. Morgan Kaufmann.

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):1620–1654.

Beeri, C., and Vardi, M. Y. 1984. A proof procedure for data dependencies. *Journal of the ACM* 31(4):718–741.

Bourhis, P.; Manna, M.; Morak, M.; and Pieris, A. 2016. Guarded-based disjunctive tuple-generating dependencies. *ACM Trans. Database Syst.* 41(4).

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

Dyckhoff, R. 1992. Contraction-free sequent calculi for intuitionistic logic. *The Journal of Symbolic Logic* 57(3):795–807.

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. Database Theory.

Feller, T.; Lyon, T. S.; Ostropolski-Nalewaja, P.; and Rudolph, S. 2023. Finite-cliquewidth sets of existential rules: Toward a general criterion for decidable yet highly expressive querying. In *Proceedings of the 26th International Conference on Database Theory (ICDT 2023)*, LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik. To appear. Preprint available via https://arxiv.org/abs/2209.02464.

Gentzen, G. 1935a. Untersuchungen über das logische schließen. i. *Mathematische zeitschrift* 39(1):176–210.

Gentzen, G. 1935b. Untersuchungen über das logische schließen. ii. *Mathematische Zeitschrift* 39(1):405–431.

Gómez Álvarez, L., and Rudolph, S. 2021. Standpoint logic: Multi-perspective knowledge representation. In Neuhaus, F., and Brodaric, B., eds., *Proceedings of the 12th International Conference on Formal Ontology in Information Systems*, volume 344 of *FAIA*, 3–17. IOS Press.

Gottlob, G. 2009. Datalog+/-: A unified approach to ontologies and integrity constraints. In Antonellis, V. D.; Castano, S.; Catania, B.; and Guerrini, G., eds., *Proceedings of the 17th Italian Symposium on Advanced Database Systems, (SEBD'09)*, 5–6. Edizioni Seneca.

Kleene, S. C. 1952. *Introduction to Metamathematics*. American Elsevier Publishing Company, INC. - New York.

Lyon, T. S., and Gómez Álvarez, L. 2022. Automating Reasoning with Standpoint Logic via Nested Sequents. In *Proceedings of the 19th International Conference on Principles of Knowledge Representation and Reasoning*, 257–266.

Lyon, T. S., and Ostropolski-Nalewaja, P. 2023. Connecting proof theory and knowledge representation: Sequent calculi and the chase with existential rules. arXiv, Paper can be found via https://arxiv.org/abs/2306.02521.

Lyon, T.; Tiu, A.; Goré, R.; and Clouston, R. 2020. Syntactic interpolation for tense logics and bi-intuitionistic logic via nested sequents. In Fernández, M., and Muscholl, A., eds., *28th EACSL Annual Conference on Computer Science Logic, CSL 2020, January 13-16, 2020, Barcelona, Spain*, volume 152 of *LIPIcs*, 28:1–28:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

Maehara, S. 1960. On the interpolation theorem of craig. *Sûgaku* 12(4):235–237.

Salvat, E., and Mugnier, M.-L. 1996. Sound and complete forward and backward chainings of graph rules. In Eklund, P. W.; Ellis, G.; and Mann, G., eds., *Proceedings of the 4th International Conference on Conceptual Structures (ICCS'96)*, volume 1115 of *LNCS*, 248–262. Springer.

Simpson, A. K. 1994. *The proof theory and semantics of intuitionistic modal logic*. Ph.D. Dissertation, University of Edinburgh. College of Science and Engineering. School of Informatics.

Slaney, J. K. 1997. Minlog: A minimal logic theorem prover. In McCune, W., ed., *Automated Deduction - CADE-14, 14th International Conference on Automated Deduction, Townsville, North Queensland, Australia, July 13-17, 1997, Proceedings*, volume 1249 of *Lecture Notes in Computer Science*, 268–271. Springer.