Algebraic Semantics of Datalog with Equalities

Martin E. Bidlingmaier

Abstract

We discuss the syntax and semantics of relational Horn logic (RHL) and partial Horn logic (PHL). RHL is an extension of the Datalog programming language that allows introducing and equating variables in conclusions. PHL is an extension of RHL by partial functions and one of the many equivalent notions of essentially algebraic theory.

Our main contribution is a new construction of free models. We associate to RHL and PHL sequents *classifying morphisms*, which enable us to characterize logical satisfaction using lifting properties. We then obtain free and weakly free models using the *small object argument*. The small object argument can be understood as an abstract generalization of Datalog evaluation. It underpins the implementation of the *Eqlog* Datalog engine, which computes free models of PHL theories.

1 Introduction

Datalog (Ceri et al., 1989) is a programming language for logical inference from Horn clauses. Abstracting from concrete syntax, a Datalog program consists of the following declarations:

- A set of *sort symbols* s.
- A set of *relation symbols* and their *arities* $r : s_1 \times \cdots \times s_n$.
- A set of *sequents* (or *rules*, or *axioms*) of the form

  $$r_1(\bar{v}^1) \wedge \cdots \wedge r_n(\bar{v}^n) \Rightarrow r_{n+1}(\bar{v}^{n+1})$$

  where $\bar{v}^i = (v_1, \ldots, v_{k_i})$ is a sort-compatible list of variables for each i, and each variable in the conclusion also appears in the premise.

A *fact* is an expression of the form $r(c_1, \ldots, c_n)$ where each c_i is a constant symbol of the appropriate sort. Given a Datalog program and a set of input facts, a *Datalog engine* computes the set of facts that can be derived from the input facts by repeated application of sequents.

A typical example of a problem that can be solved using Datalog is the computation of the transitive closure of a (directed) graph. Graphs are given by a binary relation $E : V \times V$ of edges among a sort V of vertices. The only axiom of transitive graphs is the transitivity axiom

$$E(u, v) \wedge E(v, w) \Rightarrow E(u, w).$$

A set of input facts for this Datalog program is given by a set of of expressions $E(a, b)$, where a, b are constant symbols. We identify such data with the data of

a graph with vertices $V = \bigcup_{E(a,b)} \{a, b\}$ and edges $E = \{(a, b) \mid E(a, b)\}$. Every finite graph in which every vertex appears in some edge arises in this way, so we conflate such graphs and sets of facts. (Standard Datalog does not support constants that do not appear in a fact.)

Given the Datalog program for transitive graphs and a corresponding set of facts, a Datalog engine enumerates all matches of the premise of the transitivity axiom, i.e. all substitutions $u \mapsto a, v \mapsto b, w \mapsto c$ such that that the substituted conjuncts of the premise, $E(a, b)$ and $E(b, c)$, are in the set of input facts. For each such substitution, the Datalog engine then adds the substitution $E(a, c)$ of the conclusion to the set of facts. This process is repeated until the set of facts does not increase anymore, that is, until a fixed point has been reached. This final set of facts now corresponds to a transitive graph.

Datalog has seen renewed interest in recent years for the implementation of program analysis tools (Bravenboer and Smaragdakis, 2009; Whaley and Lam, 2004; Madsen et al., 2016). From a high-level point of view, one applies Datalog by encoding abstract syntax trees as tuples in relations: Each type of abstract syntax tree node translates into a relation. If the node type as n children, then the relation has $n + 1$ entries, one entry for each children and one entry representing the node itself. This encoding allows executing Datalog programs on inputs that are derived from source code. Of course, the abstract syntax tree need not be encoded faithfully if some of its features are not required for the analysis one is interested in.

Equality saturation has recently garnered interest as a program optimization technique (Willsey et al., 2021). The idea is to insert expressions that should be optimized into an e-graph, and then close the e-graph under a set of rewrite rules. E-graphs allow sharing nodes that occur as children more than once, so that a large number of expressions can be stored. Furthermore, e-graphs can be efficiently closed under congruence, i.e. equivalence can be propagated from subexpressions to their parents. After a suitable number of rewrite rules have been applied and the e-graph has been closed under congruence, one selects a suitable equivalent expression from the equivalence class of the expression one is interested in according to a cost function. Crucially, equality saturation makes considerations about the order of rewrites unnecessary.

In this paper, we study languages and corresponding semantics that combine and subsume both Datalog and the applications of e-graphs outlined above. To that end, we extend Datalog by *equality*, that is, the ability of enforce an equality $u \equiv v$ in the conclusion of a sequent. One example is the order-theoretic antisymmetry axiom

$$\mathrm{Le}(u, v) \wedge \mathrm{Le}(v, u) \Rightarrow u \equiv v$$

which is not valid Datalog due to the equality atom $u \equiv v$, but allowed in our extension. If during evaluation of RHL an equality among constants c_1 and c_2 is inferred, then we expect the system to not distinguish c_1 and c_2 from then on. In other words, inferred equality should behave as congruence with respect to relations. For example, the premise $E(u, v) \wedge E(v, w)$ of the premise of the transitivity axiom should match $(b, c_1), (c_2, d) \in E$ if the equality $c_1 \equiv c_2$ has been inferred earlier. In addition to a set of derived facts, we also expect evaluation to yield an equivalence relation on each sort, representing inferred equalities.

Relational Horn logic extends Datalog further by free variables matching any

element of a sort, and by variables that only occur in a conclusion. We interpret the latter as existentially quantified: If the premise of a sequent matches and the conclusion contains a variable that is not bound in the premise, then we expect the Datalog engine to create new identifiers of the given sort if necessary to ensure that the conclusion holds.

Partial Horn logic, originally due to Palmgren and Vickers (2007), is a layer of syntactic sugar on top RHL, i.e. a purely syntactic extension with the same descriptive power. PHL adds function symbols $f : s_1 \times \ldots s_n \to s$, which desugar into relations $f : s_1 \times \cdots \times s_n \times s$ representing the graph of the function and the functionality axiom

$$f(v_1, \ldots, v_n, u) \wedge f(v_1, \ldots, v_n, w) \Rightarrow u \equiv w.$$

In positions where RHL expects variables (e.g. arguments of predicates or in equations), PHL allows also composed terms. These composed terms are desugared into a fresh variable corresponding to the result of applying the function and an assertion about the graph of the function.

The features of PHL allow implementing more algorithms than standard Datalog, for example congruence closure (Downey et al., 1980), Steensgaard-style pointer analysis (Steensgaard, 1996) and Hindley-Milner type inference (Milner, 1978). In each case, evaluation (which will be described in follow-up work) of the PHL theory encoding the problem domain yields the same algorithm as the standard domain-specific algorithm. In general, we should expect problems that are typically solved by combining union-find data structures with fixed point computations to be instances of PHL evaluation.

Partial Horn logic is one of the equivalent notions of *essentially algebraic theory*. Essentially algebraic theories generalize the better-known algebraic theories of universal algebra by allowing functions to be partial. Crucially, the *free model theorem* of universal algebra continues to hold also for essentially algebraic theories. Free models are the basis of our semantics of PHL evaluation. We show that free models can be computed using the *small object argument*, which we shall come to understand as an abstract generalization of Datalog evaluation.

In brief, the relation of free models and Datalog evaluation can be understood for the transitivity Datalog program outlined above as follows. We have seen that input data for this Datalog program represent certain graphs $G = (V, E)$, while output data represent transitive graphs $G' = (V, E')$. The two graphs G and G' share the same set of vertices V, which is the set of constant symbols that appear in the set of input facts. Intuitively, G' arises from G by adding data that must exist due to the transitivity axiom but no more.

Let us rephrase the relation between G and G' using category theory. Denote by Graph the category of graphs: A morphism $f : (V_1, E_1) \to (V_2, E_2)$ between graphs is a map $f : V_1 \to V_2$ that preserves the edge relation. Thus if $(u, v) \in E_1$, then we must have $(f(u), f(v)) \in E_2$. The requirement that the output graph G' arises from the input G solely by application of the transitivity sequent can now be summarized as follows:

Proposition 1. *Let $G' = (V, E')$ be the output graph generated from evaluating the transitivity Datalog program on a finite input graph $G = (V, E)$. Then G' is the free transitive graph over G.*

Proof. First we must exhibit a canonical graph morphism $\eta : G \to G'$. As G and G' share the same set of vertices, we choose η simply as identity map on V. Note that the identity on V is indeed a graph morphism $(V, E) \to (V, E')$ because $E \subseteq E'$.

Now we must show that for all graph morphisms $f : G \to H$ where $H = (V_H, E_H)$ is a transitive graph, there exists a unique graph morphism $\bar{f} : G' \to H$ such that the following triangle commutes:

$$\begin{array}{ccc} G & \xrightarrow{f} & H \\ {\scriptstyle\eta}\downarrow & \nearrow_{\exists!\,\bar{f}} & \\ G' & & \end{array}$$

Because η is the identity map, it suffices to show that $f = \bar{f}$ also defines a graph morphisms $G' \to H$; uniqueness of $\bar{f}$ follows from surjectivity of η. Recall that G' arises from repeatedly matching the premise of the transitivity axiom and adjoining its conclusion. Thus there is a finite chain

$$E = E_0 \subseteq E_1 \subseteq \cdots \subseteq E_n = E'$$

where for each i there exist $a, b, c \in V$ such that

$$E_{i+1} = E_i \cup \{(a, c\} \qquad\qquad (a, b), (b, c) \in E_i. \tag{1}$$

By induction, it suffices to show for all i that f is a graph morphism $(V, E_{i+1}) \to H$, assuming that f is a graph morphisms $(V, E_i) \to H$. Choose a, b, c such that $(a, b), (b, c) \in E_i$ and (1) is satisfied. Because f is a graph morphism $(V, E_i) \to (V, E_{i+1})$, we have $(f(a), f(b)), (f(b), f(c)) \in E_H$. Because H is transitive, it follows that $(f(a), f(c)) \in E_H$. Thus f preserves the edge (a, c) and hence constitutes a graph morphism $(V, E_{i+1}) \to H$. □

Denote by TGraph the full subcategory of Graph given by the transitive graphs. The inclusion functor TGraph $\subseteq$ Graph has a left adjoint, a *reflector*, which is given by assigning a graph to its transitive hull. Thus Proposition 1 shows that the transitivity Datalog program computes the reflector. Our primary goal in this paper is to explore and extend a semantics of PHL along these lines.

Outline and Contributions. In Section 2, we review the *small object argument* (Hovey, 2007, Theorem 2.1.14) as a method of computing weak reflections into subcategories of injective objects. We introduce *strong* classes of morphisms, for which the small object argument specializes to the *orthogonal-reflection construction* (Adamek and Rosicky, 1994, Chapter 1.C).

In Section 3, we introduce *relational Horn logic (RHL)*. RHL extends Datalog with unbound variables, with variables that occur only in the conclusion, and with equations. Input data of Datalog programs generalize to finite *relational structures*, and output data generalize to *models*, i.e. relational structures that satisfy all sequents. We show that free models exist for *strong* RHL theories, which include all Datalog theories.

Our poof of the existence of free models associates to each RHL sequent a *classifying morphism* of relational structures. Satisfaction of the sequent can be characterized as lifting property against the classifying morphism. The small

object argument now shows the existence of free models for strong theories. From this perspective, we may thus understand the small object argument as an abstract formulation of Datalog evaluation.

In Section 4, we extend RHL by function symbols to obtain *partial Horn logic (PHL)*. By identifying each function symbol with a relation symbol representing its graph and adding a functionality axiom, every PHL theory gives rise to a relational Horn logic theory with equivalent semantics. For *epic* PHL theories, where all variables must be introduced in the premise of a sequent, the associated RHL theory is strong. Conversely, we show that the semantics of every strong RHL theory can be recovered as semantics of an epic PHL theory. This justifies the usage of epic PHL as an equally powerful but syntactically better-behaved language compared to strong RHL.

The results of this paper serve as semantics of *Eqlog*, a Datalog engine that computes free models of PHL theories. Eqlog's algorithm is based on an efficient implementation of the small object argument that combines optimized Datalog evaluation (semi-naive evaluation and indices) with techniques used in congruence closure algorithms. Independently of Eqlog and the work presented there, members of the Egg (Willsey et al., 2021) community have recently created the *Egglog* tool, which combines Datalog with e-graphs and is based on very similar ideas as those of Eqlog.

2 The Small Object Argument

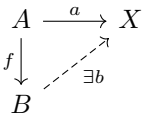
This section is a review of the small object argument, which we shall in later sections come to understand as an abstract description of Datalog evaluation. The concepts we discuss here are not new and are in fact widely known among homotopy theorists; see for example Hovey (2007) for a standard exposition. A minor innovation is our consideration of *strong* sets: Sets of morphisms for which injectivity coincides with orthogonality. For strong sets, the small object argument yields a reflection into the orthogonal subcategory where in general we would obtain only a weak reflection into the injective subcategory.

The related *orthogonal-reflection construction* (Adamek and Rosicky, 1994, Chapter 1.C) produces a reflection into the orthogonal subcategory for arbitrary sets of morphisms M. We show that every set of morphism M can be extended to a strong set N such that M and N induce the same orthogonality class. The small object argument for N now specializes to the orthogonal-reflection construction for M. Thus, the concept of strong morphisms can be used to understand the orthogonal-reflection construction as a specialized variation of the small object argument.

Fix a cocomplete category locally small $\mathcal{C}$ for the remainder of this section. We reserve the word *set* for a small set, while a *class* refers to a set in a larger set-theoretic universe that contains the collection of objects in $\mathcal{C}$. All colimits of set-indexed diagrams in $\mathcal{C}$ exist, while colimits of class-indexed diagrams need not exist.

Definition 2. Let $f : A \to B$ be a morphism and let X be an object. We write $f \pitchfork X$ and say that X is *injective* to f if for all maps $a : A \to X$ there exists a

map $b : B \to X$ such that



commutes. If furthermore b is unique for all a, then we write $f \perp X$ and say that X is *orthogonal* to f.

If M is a class of morphisms, then we write $M \pitchfork X$ if $f \pitchfork X$ for all $f \in M$, and $M \perp X$ if $f \perp X$ for all $f \in M$. The full subcategories given by the injective and orthogonal objects, respectively, are denoted by $M^{\pitchfork}$ and $M^{\perp}$. We call $M^{\pitchfork}$ the *injectivity class* of M and $M^{\perp}$ the *orthogonality class* of M.

Definition 3. A class M of morphisms is called *strong* if $M^{\pitchfork} = M^{\perp}$.

One of the main sources of strong sets is the following proposition:

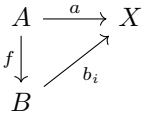
Proposition 4. *Let M be a class of epimorphisms. Then M is strong.* □

Proof. This follows immediately from right-cancellation. □

Another source of strong sets is the following proposition. It lets us reduce questions about orthogonality classes to strong injectivity classes.

Proposition 5. *Let M be a class of morphisms. Then there exists a superclass $N \supseteq M$ such that N is strong and $N^{\pitchfork} = M^{\perp}$. If M is a set, then N can be chosen as set.*

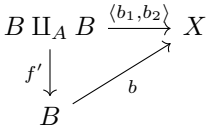
Proof. Let $f : A \to B$ be a morphism in M. Then for each object X, the data of a single map $a : A \to X$ and two maps $b_1, b_2 : B \to X$ such that



commutes for $i \in \{1, 2\}$ is in bijective correspondence to a map $\langle b_1, b_2 \rangle : B \amalg_A B \to X$. Let

$$f' : B \amalg_A B \to B.$$

be the canonical map that collapses the two copies of B into one. Then $b_1 = b_2$ if and only if there exists a map b such that



commutes. The map f' is an epimorphism. Thus if b exists, then it exists uniquely, and $b = b_1 = b_2$. It follows that X is orthogonal to f if and only if f is injective to both f and f'. The desired class N can thus be defined by $N = M \cup \{f' \mid f \in M\}$. □

Definition 6. A *sequence* of morphisms is a diagram of the form

$$X_0 \xrightarrow{f_0} X_1 \xrightarrow{f_1} \dots$$

for a countable set $(f_n)_{n\in\mathbb{N}}$ of morphisms. The *composition* of a sequence of morphisms $(f_n)_{n\in\mathbb{N}}$ is the canonical map

$$X_0 \to X_\infty = \operatorname{colim}_{n\geq 0} X_n$$

to the colimit of the sequence.

Note that the composition of a sequence of morphisms is uniquely determined only up to a choice of colimit.

Definition 7. Let M be a class of morphisms. The class $\mathrm{Cell}(M)$ of *relative M-cell complexes* is the least class of morphisms such that the following closure properties hold:

1. $M \subseteq \mathrm{Cell}(M)$.
2. $\mathrm{Cell}(M)$ is closed under coproducts. That is, if $(f_i : A_i \to B_i)_{i\in I}$ is a family of morphisms indexed by some set I and $f_i \in \mathrm{Cell}(M)$ for all $i \in I$, then
$$\coprod_{i\in I} f_i : \coprod_{i\in I} A_i \to \coprod_{i\in I} B_i$$
is in $\mathrm{Cell}(M)$.
3. $\mathrm{Cell}(M)$ is closed under pushouts. That is, if
$$\begin{array}{ccc} A & \xrightarrow{f} & B \\ \downarrow & & \downarrow \\ X & \xrightarrow{f'} & Y \end{array}$$
is a pushout square and $f \in \mathrm{Cell}(M)$, then $f' \in \mathrm{Cell}(M)$.
4. $\mathrm{Cell}(M)$ is closed under composition of sequences. That is, if
$$A_0 \xrightarrow{f_0} A_1 \xrightarrow{f_1} \dots$$
is a sequence of morphisms $f_n \in \mathrm{Cell}(M)$ with composition $f : A_0 \to A_\infty$, then $f \in \mathrm{Cell}(M)$.

Remark 8. Standard literature on factorization systems and the closely related small object argument (Hovey, 2007) considers usually not only countable sequences of morphisms but also arbitrary transfinite sequences, which are chains of morphisms indexed by an arbitrary ordinal number. In this more general setting, one then typically defines a relative M-cell complex to be a transfinite composition of pushouts of morphisms in M without mention of coproducts.

This more common notion of relative M-cell complex satisfies our closure properties 1 – 4. For 2, one chooses a well-ordering on the indexing set I, and then computes the coproduct as composition of a chain indexed by this well-ordering. Conversely, our definition of relative M-cell complex is closed

under arbitrary transfinite composition if all morphism in M have finitely presentable domains and codomains (Definition 14). Thus, whenever the domains and codomains of the morphisms in M are finitely presentable, the definition given here and the usual one agree.

Proposition 9. *Let M be a class of morphisms. Define classes of morphisms $M \subseteq M_1 \subseteq M_2 \subseteq M_3$ as follows:*

$$M_1 = M \cup \{f \mid f \text{ is a coproduct of morphisms in } M\}$$

$$M_2 = M_1 \cup \{f \mid f \text{ is a pushout of a morphism in } M_1\}$$

$$M_3 = M_2 \cup \{f \mid f \text{ is a composition of a sequence of morphisms in } M_2\}$$

Then $M_3 = \mathrm{Cell}(M)$.

Proof. Coproducts, pushouts and compositions of sequences are all defined via colimits. Because colimits commute with colimits, M_3 is closed under coproducts, pushouts and compositions of sequences. It follows that $\mathrm{Cell}(M) \subseteq M_3$, hence $\mathrm{Cell}(M) = M_3$. □

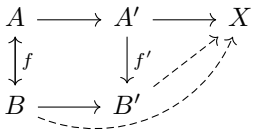
Proposition 10. *Let M be a class of morphisms. Then $\mathrm{Cell}(M)^{\pitchfork} = M^{\pitchfork}$ and $\mathrm{Cell}(M)^{\perp} = M^{\perp}$.*

Proof. If $M \supseteq N$ is an inclusion of classes of morphisms, then in general $M^{\pitchfork} \subseteq N^{\pitchfork}$ and $M^{\perp} \subseteq N^{\perp}$. This proves the inclusions $\subseteq$.

Conversely, it suffices to show for $X \in M^{\pitchfork}$ that the class

$$N = \{f \in \mathrm{Mor}\,\mathcal{C} \mid f \pitchfork X\}$$

satisfies the closure properties 1 – 4 of Definition 7, and similarly for orthogonality. This is routine. For example, closure under pushouts can be proved as follows. Let X be injective (orthogonal) to $f : A \to B$, let $f' : A' \to B'$ be a pushout of f, and let $A' \to X$ be an arbitrary morphism. The lift $B' \to X$ can then be obtained from a lift $B \to X$ and the universal property of the pushout in the following diagram:



If the lift $B \to X$ is unique, then also $B' \to X$ is unique by uniqueness of the morphism induced by the universal property of the pushout. □

Definition 11. Let $\mathcal{C}' \subseteq \mathcal{C}$ be a full subcategory. A *weak reflection* of an object $X \in \mathcal{C}$ is a map $\eta : X \to X'$ such that $X' \in \mathcal{C}'$ and every map $X \to Y$ with $Y \in \mathcal{C}'$ factors via η. If the factorization is unique for all $X \to Y$, then η is a *reflection*. A *(weak) reflector* consists of a functor $F : \mathcal{C} \to \mathcal{C}'$ and a natural transformation $\eta : \mathrm{Id} \to F$ such that η_X is a (weak) reflection for all $X \in \mathcal{C}$. The subcategory $\mathcal{C}'$ is *(weakly) reflective* in $\mathcal{C}$ if there exists a (weak) reflector.

Proposition 12. *Let M be a class of morphisms. Let $f : X \to Y$ be a relative M-cell complex, and let $g : X \to Z$ be a map with $Z \in M^{\pitchfork}$. Then there is a map $h : Y \to Z$ such that $hf = g$. If furthermore M is strong, then h is unique.*

Proof. This follows from the fact that the class of morphisms f for which the lemma holds satisfies properties 1 – 4 of Definition 7. □

Proposition 13. *Let M be a class of morphisms. Let $f : X \to Y$ be a relative M-cell complex such that $Y \in M^{\pitchfork}$. Then f is a weak reflection into $M^{\pitchfork}$. If M is strong, then f is a reflection.*

Proof. By Proposition 12. □

Definition 14. An object X is *finitely presentable* if the hom-functor $\mathrm{Hom}(X, -) : \mathcal{C} \to \mathrm{Set}$ preserves filtered colimits.

Proposition 15 (Small Object Argument: Property)**.** *Let M be a class of morphisms with finitely presentable domains and codomains. Let*

$$X_0 \xrightarrow{x_0} X_1 \xrightarrow{x_1} \dots$$

be a sequence in M such that the following holds:

1. *x_n is a relative M-cell complex for all n.*
2. *For all $f : A \to B$ in M, $n \geq 0$ and maps $a : A \to X_n$, there exists a map a map $b : B \to X_m$ for some $m \geq n$ such that*

$$\begin{array}{ccccccc} A & & & & & & B \\ {\scriptstyle a}\downarrow & & & & & & \downarrow{\scriptstyle b} \\ X_n & \xrightarrow{x_n} & X_{n+1} & \xrightarrow{x_{n+1}} & \dots & \xrightarrow{x_{m-1}} & X_m \end{array}$$

   commutes.

Then the transfinite composition $X_0 \to X_\infty$ of the x_n is a weak reflection into $M^{\pitchfork}$. If M is strong, then $X_0 \to X_\infty$ is a reflection.

Proof. Because $\mathrm{Cell}(M)$ is closed under infinite composition, the map $X_0 \to X_\infty$ is a relative M-cell complex. Thus by Proposition 13, it suffices to show that X_∞ is in $M^{\pitchfork}$.

Let $f : A \to B$ be in M and let $a : A \to X_\infty$. Because A is finitely presentable, there exists $n \in \mathbb{N}$ and $a_n : A \to X_n$ such that a factors as $A \to X_n \to X_\infty$. By assumption 2, there exist m and $b_m : B \to X_m$ that commutes with f, a_n and $x_{m-1} \circ \dots \circ x_n$. Thus if we define b as composition $B \to X_m \to X_\infty$, then $a = b \circ f$. □

Proposition 16 (Small Object Argument: Existence)**.** *Let M be a set of morphism with finitely presentable domains and codomains, and let X be an object. Then there exists a sequence*

$$X = X_0 \xrightarrow{x_0} X_1 \xrightarrow{x_1} \dots$$

satisfying the conditions of Proposition 15. In particular, $M^{\pitchfork}$ is a (weakly) reflective subcategory of $\mathcal{C}$.

Proof. It suffices to construct a relative M-cell complexes $X \to Y$ such that for every $f : A \to B$ in M and $a : A \to X$, there exists a commuting diagram

$$\begin{array}{ccc} A & \xrightarrow{f} & B \\ {\scriptstyle a}\downarrow & & \downarrow{\scriptstyle b} \\ X & \longrightarrow & Y. \end{array}$$

We then obtain the desired sequence by induction.

Let I be the set of pairs (f, a), where $f : A \to B$ is a morphism in M and $a : A \to X$. Note that I is a set because M is a set and $\mathrm{Hom}(A, X)$ is a set for all A. Now let $X \to Y$ be the map defined by the following pushout diagram:

$$\begin{array}{ccc} \coprod_{(f,a)\in I} \mathrm{dom}\, f & \longrightarrow & \coprod_{(f,a)\in I} \mathrm{cod}\, f \\ \downarrow & & \downarrow \\ X & \longrightarrow & Y \end{array}$$

Here the top map is the coproduct $\coprod_{(f,a)\in I} f$, and the left vertical map $\langle a \rangle_{(f,a)\in I}$ is induced by the universal property of coproducts. □

Proposition 17. *Let M be strong set of morphisms, and let $f : A \to B$. Denote by $\bar{f} : \bar{A} \to \bar{B}$ the reflection of f into $M^{\pitchfork}$. Then the following equations among injectivity and orthogonality classes hold:*

$$(M \cup \{f\})^{\perp} = (M \cup \{\bar{f}\})^{\perp} \qquad (M \cup \{f\})^{\pitchfork} = (M \cup \{\bar{f}\})^{\pitchfork}.$$

Proof. Let X be orthogonal (equivalently: injective) to M. Then there is a bijective correspondance between solutions to the following lifting problems:

$$\begin{array}{ccc} A & \xrightarrow{a} & X \\ {\scriptstyle f}\downarrow & \nearrow & \\ B & & \end{array} \qquad \begin{array}{ccc} \bar{A} & \xrightarrow{a'} & X \\ {\scriptstyle \bar{f}}\downarrow & \nearrow & \\ \bar{B}. & & \end{array}$$

Here a is an arbitrary map, and $a' : \bar{A} \to X$ is induced from a by the universal property of $\bar{A}$ and X being orthogonal to $\bar{f}$. □

3 Relational Horn Logic

Relational Horn Logic (RHL) is a superset of Datalog. Most notably, RHL allows equations, and in particular equations in conclusions. Our semantics of RHL are based on *relational structures*, which we introduce in Section 3.1. In Section 3.2, we then consider syntax and semantics of RHL. We show that RHL models can be characterized using lifting properties against *classifying morphisms*. This enables us to apply the small object argument to prove the existence of free models, in close analogy to Datalog evaluation. In Section 3.3, we prove a completeness result for the descriptive power of RHL: Every finitary orthogonality class of relational structures can be obtained as semantics of an RHL theory. In Section 3.4, we identify in detail the subset of RHL that corresponds to Datalog. We then explain how the computation of free RHL models can be reduced to evaluation of Datalog with minor extensions via the *setoid transformation*.

3.1 Relational Structures

Definition 18. A *relational signature* $\mathfrak{S}$ is given by the following data:

- A set S of *sort symbols*.
- A set R of *relation symbols*.
- A map that assigns to each relation symbol $r \in R$ an *arity*

$$r : s_1 \times \cdots \times s_n$$

of sort symbols $s_1, \ldots, s_n \in S$ for $n \geq 0$.

Definition 19. Let $\mathfrak{S} = (S, R)$ be a relational signature. A *relational structure* for $\mathfrak{S}$ consists of the following data:

- For each sort symbol $s \in S$, a *carrier set* X_s.
- For each relation symbol $r \in R$ with arity $r : s_1 \times \cdots \times s_n$, a relation $r_X \subseteq X_{s_1} \times \cdots \times X_{s_n}$.

A *morphism of relational structures* $f : X \to Y$ consists of functions $f_s : X_s \to Y_s$ for $s \in S$ that are compatible with the relations r_X and r_Y for all r. That is, we require that if $(x_1, \ldots, x_n) \in r_X$ for some relation symbol $r : s_1 \times \cdots \times s_n$, then $(f_{s_1}(x_1), \ldots, f_{s_n}(x_n)) \in r_Y$. The *category of relational structures* is denoted by $\mathrm{Rel}(\mathfrak{S})$.

When no confusion can arise, we suppress sort annotations. Thus if X is a relational structure, then we write $x \in X$ to mean that $x \in X_s$ for some $s \in S$. Similarly, if $f : X \to Y$ is a morphism of relational structures and $x \in X_s$, then we often denote the image of x under f by $f(x)$ instead of $f_s(x)$. If the signature $\mathfrak{S}$ is clear from context, we abbreviate $\mathrm{Rel}(\mathfrak{S})$ as Rel.

There is an evident forgetful functor $\mathrm{Rel}(\mathfrak{S}) \to \mathrm{Set}^S$ to the S-ary product of the category of sets, which is given by discarding the relations. When we mention the *carrier sets* of a relational structure, we mean the result of applying this forgetful functor.

Proposition 20. *Let $\mathfrak{S} = (S, R)$ and $\mathfrak{S}' = (S', R')$ be relational signatures such that $\mathfrak{S}'$ extends $\mathfrak{S}$, in the sense that $S \subseteq S', R \subseteq R'$ and $\mathfrak{S}$ and $\mathfrak{S}'$ assign the the same arities to relation symbols $r \in R$. Then the evident forgetful functor $\mathrm{Rel}(\mathfrak{S}') \to \mathrm{Rel}(\mathfrak{S})$ has left and right adjoints. Both adjoints are sections to the forgetful functor, that is, both composites*

$$\mathrm{Rel}(\mathfrak{S}) \rightrightarrows \mathrm{Rel}(\mathfrak{S}') \longrightarrow \mathrm{Rel}(\mathfrak{S})$$

are identity functors.

Proof. Let X be a relational $\mathfrak{S}$-structure. Let $s \in S' \setminus S$ and let $r : s_1 \times \cdots \times s_n$ be in $R' \setminus R$. The left adjoint extends X to a relational $\mathfrak{S}'$-structure Y by $Y_s = \emptyset$ and $r_Y = \emptyset$. The right adjoint extends X to a relational $\mathfrak{S}'$ structure Z such that $Z_s = \{*\}$ is a singleton set and $r_Z = Z_{s_1} \times \cdots \times Z_{s_n}$. □

Proposition 21. *Let $\mathfrak{S} = (S, R)$ be a relational signature. Then $\mathrm{Rel}(\mathfrak{S})$ is complete and cocomplete, and the forgetful functor $\mathrm{Rel}(\mathfrak{S}) \to \mathrm{Set}^S$ preserves limits and colimits.*

Proof. Limit and colimit preservation follows from Proposition 20, since the forgetful functor is induced by the inclusion $(S, \emptyset) \subseteq (S, R)$. Limits commute with other limits and in particular products. Thus, limits of relational structures can be constructed as limits of carriers endowed with the limits of relation sets.

Colimits are slightly more involved because products do not generally commute with quotients. Let $D : I \to \mathrm{Rel}(\mathfrak{S})$ be a diagram of relational structures. We define the carrier sets of our candidate colimit structure X by the colimit of carrier sets. That is,

$$X_s = \operatorname*{colim}_{i \in I} D(i)_s$$

for all $s \in S$. We obtain evident maps $(p_i)_s : D(i)_s \to X_s$ for all objects i in I and $s \in S$. Let $r : s_1 \times \cdots \times s_n$ be a relation symbol. Then we define r_X as union over the images of the $r_{D(i)}$. Thus,

$$r_X = \bigcup_{i \in I} p_i(r_{D(i)})$$

where $p_i(r_{D(i)}) = ((p_i)_{s_1} \times \cdots \times (p_i)_{s_n})(r_{D(i)})$. □

Definition 22. Let $\mathfrak{S} = (S, R)$ be a relational signature. A relational structure X for $\mathfrak{S}$ is *finite* if

$$\sum_{s \in S} |X_s| + \sum_{r \in R} |r_X| < \infty,$$

that is, if all the X_s and r_X are finite and almost always empty.

Proposition 23. *Let $\mathfrak{S} = (S, R)$ be a relational signature. Then a relational structure for $\mathfrak{S}$ is finite if and only if it is a finitely presentable object in* $\mathrm{Rel}(\mathfrak{S})$.

Proof. Let X be a finite relational structure and let

$$X \to Y = \operatorname{colim} D = \coprod_{i \in I} D(i) / \sim$$

be a map to a filtered colimit. Then the image of each element $x \in X$ is represented by some element $y_x \in D(i_x)$. Since X contains only finitely many elements and D is directed, we may assume that $i_x = i_{x'} = i$ is constant over all $x, x' \in X$. For each tuple $t = (x_1, \ldots, x_n) \in r_X$ for some $r \in R$ we have that $([y_{x_1}], \ldots, [y_{x_n}]) \in r_Y$. Since there are only finitely many t, we may again increase i so that $(y_{x_1}, \ldots, y_{x_n}) \in r_{D(i)}$. Now $X \to Y$ factors via $D(i)$.

Conversely, if a relational structure X is not finite, then there exists a strictly increasing sequence of relational substructures

$$X_0 \subset X_1 \subset X_2 \subset \cdots \subset X$$

such that $\bigcup_{n \geq 0} X_n = X$. Then the canonical map $X = \bigcup_{n \geq 0} X_n \cong \operatorname{colim}_{n \geq 0} X_n$ does not factor via any X_n, so X is not finitely presentable. □

3.2 Syntax and Semantics

Fix a relational signature $\mathfrak{S} = (S, R)$. We assume a countable supply of variable symbols v, each annotated with a sort $s \in S$.

Definition 24. An *RHL atom* is a statement of one of the following forms:

1. $r(v_1, \ldots, v_n)$, where $r : s_1 \times \cdots \times s_n$ is a relation symbol and the v_i are variables of sort s_i for all $i \in \{1, \ldots, n\}$.

2. $v \downarrow$ where v is a variable.

3. $u \equiv v$, where u and v are variables of the same sort.

A *relational formula* is a finite conjunction of flat atoms. A *relational sequent* is an implication of flat formulas. An *relational theory* is a set of relational sequents.

Definition 25. Let X be a relational structure. An *interpretation of a set of variables* V in X is a map I that assigns to each variable $v \in V$ of sort s an element $I(v) \in X_s$. An *interpretation of a relational atom* ϕ in X is an interpretation I of the variables occuring in ϕ such that one of the following conditions holds:

1. $\phi = r(v_1, \ldots, v_n)$ for some relation symbol r and $(I(v_1), \ldots, I(v_n)) \in r_X$.

2. $\phi = v \downarrow$ for some variable v, without further assumptions.

3. $\phi = u \equiv v$ for some variables u and v, and $I(u) = I(v)$.

An *interpretation of a relational formula* $\mathcal{F} = \phi_1 \wedge \cdots \wedge \phi_n$ in X is an interpretation of the variables occuring in $\mathcal{F}$ that restricts to an interpretation of ϕ_i for each $i \in \{1, \ldots, n\}$.

A relational structure X *satisfies* a relational sequent $\mathcal{F} \Rightarrow \mathcal{G}$ if each interpretation of $\mathcal{F}$ in X can be extended to an interpretation of $\mathcal{F} \wedge \mathcal{G}$ in X. A *model* of a theory T is a relational structure that satisfies all sequents in T. The *category of models* for a theory T is the full subcategory of relational structures given by the models of T and denoted by $\mathrm{Mod}(T)$.

Definition 26. We associate to each flat atom ϕ a *classifying relational structure* $[\phi]$ and a *generic interpretation* I_ϕ of ϕ in $[\phi]$ as follows:

1. If $\phi = r(v_1, \ldots, v_n)$ where $r : s_1 \times \cdots \times s_n$, then the carriers of $[\phi]$ are given by distinct elements $I_\phi(v_i) \in [\phi]_{s_i}$ and a single tuple $(I_\phi(v_1), \ldots, I_\phi(v_n)) \in r_{[\phi]}$. The relations $r'_{[\phi]}$ for $r \neq r'$ are empty.

2. If $\phi = v \downarrow$, where v has sort s, then $[\phi]_s$ contains a single element $I_\phi(v)$. All other carrier sets and all relations are empty.

3. If $\phi = v_1 \equiv v_2$, where v_1 and v_2 have sort s, then $[\phi]_s$ contains a single element $I_\phi(v_1) = I_\phi(v_2)$. All other carrier sets and all relations are empty.

Let $\mathcal{F} = \phi_1 \wedge \cdots \wedge \phi_n$ be a relational formula. The classifying relational struture $[\mathcal{F}]$ of $\mathcal{S}$ is the quotient

$$([\phi_1] \amalg \cdots \amalg [\phi_n])/\sim$$

where $\sim$ is the relation given by

$$I_{\phi_i}(v) \sim I_{\phi_j}(v)$$

for all $i, j \in \{1, \ldots, n\}$ and variables v occuring in both ϕ_i and ϕ_j, and the generic interpretation $I_\mathcal{F}$ is the amalgamation of the interpretations I_{ϕ_i}.

Proposition 27. *Let $\mathcal{F}$ be a relational formula and let X be a relational structure. Then there is a bijection between interpretations of $\mathcal{F}$ in X and maps $[\mathcal{F}] \to X$.*

Proof. If $f : [\mathcal{F}] \to X$, then $f \circ I_{\mathcal{F}}$ is an interpretation of $\mathcal{F}$ in X. For the converse, let $\mathcal{F} = \phi_1 \wedge \cdots \wedge \phi_n$ for relational atoms ϕ_i. Then every interpretation I of $\mathcal{F}$ restricts to an interpretation of ϕ_i for each i. The carrier sets of $[\phi_i]$ are defined using the variables of ϕ_i, which defines an evident map $[\phi_i] \to X$. Since the restrictions of I to the variables in each ϕ_i agree on variables that occur in simultaneously in two atoms, the individual maps $[\phi_i] \to X$ glue to a map $[\mathcal{F}] \to X$. □

Definition 28. Let $\mathcal{S} = \mathcal{F} \Rightarrow \mathcal{G}$ be a relational sequent. The *classifying morphism* of $\mathcal{S}$ is the map $[\mathcal{S}] : [\mathcal{F}] \to [\mathcal{F} \wedge \mathcal{G}]$ that is induced by the canonical interpretation of $\mathcal{F}$ in $[\mathcal{F} \wedge \mathcal{G}]$.

Proposition 29. *Let $\mathcal{S}$ be a relational sequent and let X be a relational structure. Then X satisfies $\mathcal{S}$ if and only if X is injective to $[\mathcal{S}]$.*

Proof. Let $\mathcal{S} = \mathcal{F} \Rightarrow \mathcal{G}$. By Proposition 27, interpretations I of the premise $\mathcal{F}$ correspond to maps $\langle I \rangle : [\mathcal{F}] \to X$, and interpretations J of $\mathcal{F} \wedge \mathcal{G}$ correspond to maps $\langle J \rangle : [\mathcal{F} \wedge \mathcal{G}] \to X$. The map $[\mathcal{S}] : [\mathcal{F}] \to [\mathcal{F} \wedge \mathcal{G}]$ can be obtained by restriction of the generic interpretation of $\mathcal{F} \wedge \mathcal{G}$ in $[\mathcal{F} \wedge \mathcal{G}]$ to the variables occurring in $\mathcal{F}$. It follows that

$$\begin{array}{ccc} [\mathcal{F}] & \xrightarrow{\langle I \rangle} & X \\ {\scriptstyle [\mathcal{S}]}\downarrow & \nearrow_{\langle J \rangle} & \\ [\mathcal{F} \wedge \mathcal{G}] & & \end{array}$$

commutes if and only if I is a restriction of J. □

Proposition 30. *Let T be a relational theory. Then* $\mathrm{Mod}(T) \subseteq \mathrm{Rel}(\mathfrak{S})$ *is a weakly reflective category. If $M = \{[\mathcal{S}] \mid \mathcal{S} \in T\}$ is strong, then* $\mathrm{Mod}(T)$ *is a reflective subcategory.*

Proof. By application of the small object argument (Propositions 15 and 16) to $M = \{[\mathcal{S}] \mid \mathcal{S} \in T\}$. □

Remark 31. Let us reflect on the similarities between the proof of Proposition 30 and Datalog evaluation. Note that Datalog is a strict subset of RHL, so we have to specialize Proposition 30 to Datalog theories T in order to compare. Thus, we assume that the conclusions of sequents in T only contain atoms $r(v_1, \ldots, v_n)$ for variables that occur in the premise (see Section 3.4 for detailed discussion of fragments of RHL).

Now, unfolding the small object argument, we see that the reflection of a relational structure X into the category of models is given by the colimit of a chain

$$X = X_0 \longrightarrow X_1 \longrightarrow X_2 \longrightarrow \ldots$$

of relational structures. The relational structure X corresponds to the set of input facts to the Datalog program, and each X_i represents the total set of derived facts after the ith iteration of Datalog evaluation. Because T contains

Datalog sequents only, the transition maps $X_i \to X_{i+1}$ are bijective on carriers. The data of the sequence 31 is thus equivalent to a sequence of inclusions

$$r_X = r_{X_0} \subseteq r_{X_1} \subseteq \dots$$

on the carrier of X for all relation symbols r, mirroring the monotonically growing relations during Datalog evaluation.

Unfolding our existence proof of the small object argument (Proposition 16) and the universal property of classifying structures (Proposition 27), we see that X_{i+1} is obtained from X_i via the following pushout square:

$$\begin{array}{ccc} \coprod_{(\mathcal{S},I)\in K}[\mathcal{F}_\mathcal{S}] & \longrightarrow & \coprod_{(\mathcal{S},a)\in K}[\mathcal{G}_\mathcal{S}] \\ \downarrow & \ulcorner & \downarrow \\ X_i & \longrightarrow & X_{i+1} \end{array}$$

Here K is the set of of pairs of sequents $\mathcal{S} = \mathcal{F}_\mathcal{S} \Rightarrow \mathcal{G}_\mathcal{S}$ and interpretations $I : [\mathcal{F}_\mathcal{S}] \to X_i$. Thus the left vertical map corresponds to the set of matches of premises among the facts established after the ith iteration of Datalog evaluation. Defining X_{i+1} using the pushout square above has the effect of adjoining matches of the conclusion for each match of the premise. Since T is a Datalog theory, the conclusions are relation atoms, hence X_{i+1} is obtained from X_i by adjoining new tuples to relations.

Remark 32. Semi-naive evaluation is an optimized version of Datalog evaluation, where we consider only matches of premises at the ith stage that have not been present already in the $(i-1)$th stage. This does not change the result of Datalog evaluation since conclusions of matches that have been found in a previous iteration have already been adjoined. In terms of the small object argument, this optimization can be understood as a more economic choice of the set K: In Diagram 31, we can replace K by the set of interpretations $I : [\mathcal{F}_\mathcal{S}] \to X_i$ that do not factor via X_{i-1}.

Remark 33. Still, there are properties of Datalog evaluation that Proposition 30 does not entirely capture. First, the result of Datalog evaluation is determined uniquely via fixed point semantics, whereas Proposition 30 guarantees uniqueness (up to ismorphism) only in the case of strong theories. Since classifying morphisms of Datalog sequents are surjective, all Datalog theories are strong (Proposition 4). Thus, Proposition 30 does indeed determine the result of Datalog computation uniquely. However, not all strong theories are Datalog theories or even contain epimorphisms only. For example, Proposition 5 allows extending every RHL theory to a strong theory. A syntactic characterization of strong RHL theories is the main purpose of Section 4, where we discuss partial Horn logic.

A second feature of Datalog evaluation we have not discussed is that it always terminates: Since each Datalog evaluation monotonically increases the size of relations on a fixed carrier, we reach a fixed point after a finite number of iterations. This is not generally true for RHL theories, since the carrier sets change during evaluation. We can, however, prove termination for *surjective* theories, which subsume and generalize Datalog theories (Corollary 38). In general, however, termination of RHL evaluation is undecidable.

3.3 Completeness Results

Definition 34. Let $f : X \to Y$ be a map of relational structures. We say that f is *surjective* if $f_s : X_s \to Y_s$ is a surjective map for all $s \in S$.

Proposition 35. *Let $\mathfrak{S} = (S, R)$ be a relational signature. Let $f : X \to Y$ be a map of relational structures.*

1. *f is an epimorphism if and only if f is surjective.*
2. *f is an effective epimorphism if and only if f is surjective and furthermore the induced maps $f_r : r_X \to r_Y$ are surjective for all $r \in R$.*

Proof. 1. Note that, for every morphism $f : X \to Y$ in a cocomplete category $\mathcal{C}$, f is an epimorphism if and only if

$$\begin{array}{ccc} A & \xrightarrow{f} & A \\ {\scriptstyle f}\downarrow & & \downarrow \\ A & \longrightarrow & A \end{array}$$

is a pushout square. Because the forgetful functor from relational structures to S-indexed families of sets preserves colimits, it follows that it preserves epimorphisms, i.e. that every epimorphism of relational structures must be surjective. The same forgetful functor is faithful, hence reflects epimorphisms.

2. This follows from the construction of colimits in Proposition 21. □

Proposition 36. *The image of (surjective, effectively epic) relational sequents under the assignment $\mathcal{S} \mapsto [\mathcal{S}]$ can be described as follows:*

1. *If $\mathcal{S}$ is a relational sequent, then $[\mathcal{S}]$ is a map finite relational structures. Conversely, every map of finite relational structures is isomorphic to a map of the form $[\mathcal{S}]$ for a relational sequent $\mathcal{S}$.*
2. *If $\mathcal{S}$ is surjective, then $[\mathcal{S}]$ is a surjection of finite relational structures. Conversely, every surjection of finite relational structures is isomorphic to a map of the form $[\mathcal{S}]$ for a surjective relational sequent $\mathcal{S}$.*
3. *If $\mathcal{S}$ is effectively epic, then $[\mathcal{S}]$ is an effective epimorphism of finite relational structures. Conversely, every effective epimorphism of finite relational structures is isomorphic to a map of the form $[\mathcal{S}]$ for an effectively epic relational sequent $\mathcal{S}$.*

Proof. 1. Relational sequents $\mathcal{S}$ are by definition finite formulas, so it follows from the definition of the map $[\mathcal{S}]$ that its domain and codomain are finite.

Conversely let $f : A \to B$ be a map of finite relational structures. Choose distinct variables v_x of sort s for all sorts s and $x \in A_s$. The formulas

$$\mathcal{F}_{\mathrm{car}} = \bigwedge_{x \in A} v_x \downarrow \qquad \mathcal{F}_{\mathrm{rel}} = \bigwedge_{\substack{r \in R \\ (x_1, \ldots, x_n) \in r_A}} r(v_{x_1}, \ldots, v_{x_n})$$

are finite (and hence well-defined) because A is finite. The formula $\mathcal{F}_{\mathrm{car}}$ encodes the carrier sets of A, and $\mathcal{F}_{\mathrm{rel}}$ encodes the relations. Thus if $\mathcal{F} = \mathcal{F}_{\mathrm{car}} \wedge \mathcal{F}_{\mathrm{rel}}$, then $A \cong [\mathcal{F}]$.

Define similarly variables v_y for each $y \in B$ and formulas $\mathcal{G}_{\mathrm{car}}, \mathcal{G}_{\mathrm{rel}}$ for B. Set

$$\mathcal{G}_{\mathrm{eq}} = \bigwedge_{x \in A} v_x \equiv v_{f(x)}$$

and let $\mathcal{G} = \mathcal{G}_{\mathrm{car}} \wedge \mathcal{G}_{\mathrm{rel}} \wedge \mathcal{G}_{\mathrm{eq}}$. Then

$$B \cong [\mathcal{G}_{\mathrm{car}} \wedge \mathcal{G}_{\mathrm{rel}}] \cong [\mathcal{G}] \cong [\mathcal{G} \wedge \mathcal{F}]$$

and it can be verified using the universal property of $[\mathcal{F}]$ (Proposition 27) that

$$\begin{array}{ccc} [\mathcal{F}] & \longrightarrow & [\mathcal{F} \wedge \mathcal{G}] \\ \cong\downarrow & & \downarrow\cong \\ A & \xrightarrow{f} & B \end{array}$$

commutes.

2. The carrier of a relational structure $[\mathcal{F}]$ obtained from a formula $\mathcal{F}$ is a quotient of the set of variables occuring in $\mathcal{F}$. Thus if every variable in the conclusion $\mathcal{G}$ of a sequent $\mathcal{F} \Rightarrow \mathcal{G}$ occurs also in the premise $\mathcal{F}$, then the resulting map $[\mathcal{F} \Rightarrow \mathcal{G}] : [\mathcal{F}] \to [\mathcal{F} \wedge \mathcal{G}]$ is surjective. Conversely, let $f : A \to B$ be a surjective map of finite relational structures. Let v_x for $x \in A$ and $\mathcal{F} = \mathcal{F}_{\mathrm{car}} \wedge \mathcal{F}_{\mathrm{rel}}$ be as in the proof of 1, so that $[\mathcal{F}] \cong A$. Set

$$\mathcal{G}_{\mathrm{eq}} = \bigwedge_{\substack{x,y \in A \\ f(x)=f(y)}} v_x \equiv v_y \qquad \mathcal{G}_{\mathrm{rel}} = \bigwedge_{\substack{r \in R \\ x_1,\ldots,x_n \in A \\ (f(x_1),\ldots,f(x_n)) \in r_B}} r(v_{x_1}, \ldots, v_{x_n})$$

and $\mathcal{G} = \mathcal{G}_{\mathrm{eq}} \wedge \mathcal{G}_{\mathrm{rel}}$. Because all the v_x occur in $\mathcal{F}$, all the variables of $\mathcal{G}$ occur in $\mathcal{F}$. By definition of $\mathcal{G}_{\mathrm{eq}}$, the map $[\mathcal{F} \wedge \mathcal{G}_{\mathrm{eq}}] \to B$ is an isomorphism on carrier sets, which then implies that $[\mathcal{F} \wedge \mathcal{G}] \cong B$ by definition of $\mathcal{G}_{\mathrm{rel}}$. Thus $[\mathcal{F} \Rightarrow \mathcal{G}] \cong f$.

3. If $\mathcal{S}$ is effectively epic and $f = [\mathcal{S}] : A \to B$, then the maps $f_r : r_A \to r_B$ are surjective for all relation symbols r. Thus by Proposition 35, f is an effective epimorphism.

Conversely, let $f : A \to B$ be an effective epimorphism of finite relational structures. Let v_x for $x \in A$ and $\mathcal{F} = \mathcal{F}_{\mathrm{car}} \wedge \mathcal{F}_{\mathrm{rel}}$ be as in the proof of 1, so that $[\mathcal{F}] \cong A$. Let

$$\mathcal{G} = \mathcal{G}_{\mathrm{eq}} = \bigwedge_{\substack{x,y \in A \\ f(x)=f(y)}} v_x \equiv v_y.$$

By the same argument as in the proof of 2, it follows that $[\mathcal{F} \wedge \mathcal{G}] \to B$ is an isomorphism on carrier sets. Let r be a relation symbol and let $(y_1, \ldots, y_n) \in r_B$. Then because f is an effective epimorphism, there exist $x_1, \ldots, x_n \in A$ such $f(x_i) = y_i$ for all i and $(x_1, \ldots, x_n) \in r_A$. Thus $\mathcal{F}$ contains the atom $r(v_{x_1}, \ldots, v_{x_n})$, hence $(y_1, \ldots, y_n)$ is in the image of $r_{[\mathcal{F} \wedge \mathcal{G}]} \to B$. We conclude $[\mathcal{F} \wedge \mathcal{G}] \cong B$, hence $f \cong [\mathcal{F} \Rightarrow \mathcal{G}]$. □

Proposition 37. *Let M be a finite set of epimorphisms of finite relational structures. Let*

$$X_0 \xrightarrow{x_0} X_1 \xrightarrow{x_1} \ldots$$

be any sequence of maps of relational structures satisfying the conditions of Proposition 15 such that furthermore X_0 is finite. Then the sequence is eventually stationary, in the sense that x_n is an isomorphism for all sufficiently large n.

Proof. Since all maps in M are surjective and colimits of relational structures commute with colimits on carrier sets, it follows that all maps in $\mathrm{Cell}(M)$ are surjective. Thus the cardinality of the carriers of the X_n decreases monotonically with n. Since X is finite, the carriers X_s are empty for almost all sorts s. Eventually, the sum of the cardinalities of the carriers of X_n must thus become stable, say after $n_0 \in \mathbb{N}$. Without loss of generality, we may assume that x_n is the identity map on carriers for $n \geq n_0$. Let $r \in R$. Then for all $n \geq n_0$, we have that

$$r_{X_n} \subseteq r_{X_{n+1}} \subseteq (X_{n_0})_{s_1} \times \cdots \times (X_{n_0})_{s_n}$$

and the latter is a finite set. Thus, we eventually have $r_{X_n} = r_{X_{n+1}}$. Even when R is infinite, we have $r_X = r_{X_n}$ for all n and almost all r, since only finitely many relations are non-empty in any of the involved relational structures (X or a domain or codomain of a map in M). For sufficiently large $n \geq n_1$ we thus have $r_{X_n} = r_{X_{n+1}}$ for all r and hence $X_n = X_{n+1}$. □

Corollary 38. *Let T be an RHL theory containing only surjective sequents. Then the reflection of a finite relational structure into* $\mathrm{Mod}(T)$ *is a finite relational structure.* □

Remark 39. Proposition 37 can likely be generalized to locally finitely presentable categories (perhaps with further conditions on the cardinality of M) as such categories are always co-wellpowered: For every object X, there exists up to isomorphism only a *set* of epimorphism $X \twoheadrightarrow Y$. We would now have to identify objects for which this set is finite.

3.4 Datalog and Relational Horn Logic

Definition 40. Let $\mathcal{S}$ be an RHL sequent.

1. $\mathcal{S}$ is a *Datalog sequent* if all atoms in $\mathcal{S}$ are of the form $r(v_1, \ldots, v_n) \downarrow$ and all variables in the conclusion of $\mathcal{S}$ also occur in the premise.
2. $\mathcal{S}$ is a *Datalog sequent with sort quantification* if all atoms in $\mathcal{S}$ are of the form $r(v_1, \ldots, v_n) \downarrow$ or $v \downarrow$, and all variables in the conclusion of $\mathcal{S}$ also occur in the premise.
3. $\mathcal{S}$ is a *Datalog sequent with choice* if all atoms in $\mathcal{S}$ are of the form $r(v_1, \ldots, v_n) \downarrow$ or $v \downarrow$.

Note that in standard Datalog, usually only sequents with a single atom as conclusion are allowed. However, our generalized Datalog sequents have the same descriptive power as standard Datalog, since a single sequent with n conclusions can equivalently be replaced by n sequents with single conclusions. The name *Datalog with choice* in 3 alludes to the choice construct in Souffle (Hu et al., 2021) with similar semantics.

Definition 41. An element $x \in X$ in a relational structure is *unbound* if it does not appear in any tuple $t \in r_X$ for all $r \in R$.

Definition 42. A morphism $f : X \to Y$ of relational structures is *injective* if $f_s : X_s \to Y_s$ is an injective map for all $s \in S$.

Proposition 43. *The classifying morphisms of Datalog sequents can be characterized up to isomorphism as follows:*

1. *The classifying morphisms of Datalog sequents are precisely the injective surjective morphisms of finite relational structures that do not contain unbound variables.*
2. *The classifying morphisms of Datalog sequents with sort quantification are precisely the injective surjective morphisms of finite relational structures.*
3. *The classifying morphisms of Datalog sequents with choice are precisely the injective surjective morphisms of finite relational structures.*

□

Definition 44. A *setoid* consists of a set X and an equivalence relation $\sim_X$ on X. A morphism $f : X \to Y$ is a map of underlying sets that respects the equivalence relations. Two morphisms $f, g : X \to Y$ of setoids are equal if $f(x) \sim_Y g(x)$ for all $x \in X$. The category of setoids is denoted by Setoid.

Proposition 45. *The categories* Setoid *and* Set *are equivalent. An equivalence is given by the functor* Setoid $\to$ Set *defined by* $(X, \sim_X) \mapsto X/\sim_X$ *and the functor* Set $\to$ Setoid *defined by* $X \mapsto (X, \{(x, x) \mid x \in X\}$. □

Definition 46. The *setoid transformation* of an RHL theory T defined on a signature $\mathfrak{S} = (S, R)$ is a Datalog with choice theory T' defined on a relational signature $\mathfrak{S}'$ as follows. The signature $\mathfrak{S}'$ extends $\mathfrak{S}$ by a relation symbol $\mathrm{Eq}_s : s \times s$ for each sort $s \in S$. The sequents of T' are given as follows:

1. For each sort s, sequents asserting that Eq_s is an equivalence relation:

$$x! \Rightarrow \mathrm{Eq}_s(x) \qquad \mathrm{Eq}_s(x, y) \Rightarrow \mathrm{Eq}_s(y, x)$$

$$\mathrm{Eq}_s(x, y) \wedge \mathrm{Eq}_s(y, z) \Rightarrow \mathrm{Eq}_s(x, z)$$

2. For each relation $r : s_1 \times \cdots \times s_n$, a sequent asserting that the equivalence relations Eq_s behave as congruences with respect to r:

$$r(v_1, \ldots, v_n) \wedge \mathrm{Eq}_{s_1}(v_1, u_1) \wedge \cdots \wedge \mathrm{Eq}_{s_n}(v_n, u_n) \Rightarrow r(u_1, \ldots, u_n)$$

3. For each sequent $\mathcal{S}$ in T, the sequent which is obtained from $\mathcal{S}$ by replacing each equality atom $u \equiv v$ with the atom $\mathrm{Eq}_s(u, v)$, where s is the sort of u and v.

The *category of setoid models* $\mathrm{Mod}_{\mathrm{Setoid}}(\mathfrak{S}, T)$ is given by the models of $(\mathfrak{S}', T')$, where we consider morphisms $f, g : X \to Y$ as equal if $f_s, g_s : (X_s, \mathrm{Eq}_s) \to (Y_s, \mathrm{Eq}_s)$ are equal as setoid morphisms for all sorts s.

Proposition 47. *Let* $(\mathfrak{S}, T)$ *be an RHL theory. Then* $\mathrm{Mod}_{\mathrm{Setoid}}(\mathfrak{S}, T)$ *and* $\mathrm{Mod}(\mathfrak{S}, T)$ *are equivalent categories.*

An equivalence is given as follows. The functor $F : \mathrm{Mod}(\mathfrak{S}, T) \to \mathrm{Mod}_{\mathrm{Setoid}}(\mathfrak{S}, T)$ *extends a relational* $\mathfrak{S}$*-structure* X *to a relational* $\mathfrak{S}'$*-structure* $F(X)$ *on the same carrier by* $(\mathrm{Eq}_s)_{F(X)} = \{(x, x) \mid x \in X\}$ *for all sorts* $s \in S$*. The functor* $G : \mathrm{Mod}_{\mathrm{Setoid}}(\mathfrak{S}, T)$ *assigns to a setoid mdel* Y *the relational* $\mathfrak{S}$*-structure with carriers* $X_s = Y_s/\mathrm{Eq}_s$ *and relations* $r_X = \{([y_1], \ldots, [y_n]) \mid (y_1, \ldots, y_n) \in r_Y\}$.

Proof. We must first verify that F and G are well-defined, i.e. that the relational structures in their images are indeed models of the respective theories. This is clear for F.

Let $Y = G(X)$ for $X \in \mathrm{Mod}_{\mathrm{Setoid}}(\mathfrak{S}, T)$. Let $\mathcal{F}$ be a formula for the signature $\mathfrak{S}$ and let I be an interpretation of $\mathcal{F}$ in Y. Since the carriers of Y are defined as quotients of the carriers of X, every interpretation I of $\mathcal{F}$ in Y lifts to an interpretation I' of the same set of variables in X, so that we have $I(v) = [I'(v)]$ for all variables v. Note that I' is an interpretation of the set of variables of $\mathcal{F}$, but not always of the formula $\mathcal{F}$. Let $\mathcal{F}'$ be the formula obtained from $\mathcal{F}$ by replacing every equality atom $u \equiv v$ by the atom $\mathrm{Eq}_s(u, v)$, where s is the sort of u and v. We claim that I' is an interpretation of $\mathcal{F}'$. To show this, it suffices to consider the case where $\mathcal{F}$ is an atom:

- If $\mathcal{F} = u \equiv v$, then $I(u) = I(v)$, so $[I'(u)] = I(u) = I(v) = [I'(v)]$. Thus $I'(u)$ and $I'(v)$ are in the same equivalence class, that is, $(I'(u), I'(v)) \in \mathrm{Eq}_s$.
- If $\mathcal{F} = r(v_1, \ldots, v_n)$ for some relation symbol r, then $(I(v_1), \ldots, I(v_n)) \in r_Y$. By definition of r_X, there exist $x_1, \ldots, x_n \in X$ such that $[x_i] = I(v_i)$ and $(x_1, \ldots, x_n) \in r_X$. Thus $[I'(v_i)] = [x_i]$, so we have $(I'(v_i), x_i) \in \mathrm{Eq}_{s_i}$ for all i. Since X satisfies the congruence sequents 2, r_X is closed under equivalence in each argument, hence $(I'(v_1), \ldots, I'(v_n)) \in r_X$. Thus I' is an interpretation of $\mathcal{F}'$.
- The case $\mathcal{F} = v\downarrow$ is trivial.

Conversely, every interpretation I' of $\mathcal{F}'$ in X descends to an interpretation of I of $\mathcal{F}$ in Y by setting $I'(v) = [I(v)]$.

Now, let $\mathcal{F} \Rightarrow \mathcal{G}$ be a sequent in T, and let I be an interpretation of $\mathcal{F}$ in $G(Y)$. We have just shown that I' lifts to an interpretation of $\mathcal{F}'$ in X. Because X satisfies $\mathcal{F}' \Rightarrow \mathcal{G}'$, we can extend I' to an interpretation J' of $\mathcal{G}'$ in X, and then J' descends to an interpretation of $\mathcal{G}$ in Y that extends I. Thus G is well-defined.

The composition $G \circ F$ is equivalent to the identity functor since a quotient by the diagonal does not change the original set. As for $F \circ G$, note that there is a canonical map $f : X \to G(F(X))$ for all setoid models X. The restriction f_s of f to a setoid carrier (X_s, Eq_s) is an isomorphism of setoids for all sorts s, with inverses g_s given by a choice of representative in each equivalence class. Since the relations of X are closed under equivalence in each argument, it follows that g is a morphism of relational structures. Thus f and g are isomorphisms. □

Corollary 48. *Let $(\mathfrak{S}, T)$ be an RHL theory with setoid transformation $(\mathfrak{S}', T')$. Then the reflection $\mathrm{Rel}(\mathfrak{S}) \to \mathrm{Mod}(\mathfrak{S}, T)$ can be computed as composite*

$$\mathrm{Rel}(\mathfrak{S}) \xrightarrow{F_1} \mathrm{Rel}(\mathfrak{S}') \xrightarrow{F_2} \mathrm{Mod}(\mathfrak{S}', T') \xrightarrow{F_3} \mathrm{Mod}(\mathfrak{S}, T)$$

where

- *F_1 is the functors that extends relational $\mathfrak{S}$-structures X to relational $\mathfrak{S}'$-structures with empty relations Eq_s,*
- *F_2 is the free T'-model functor, and*

- *F_3 is one half of the equivalence constructed in Proposition 47.*

Proof. F_1 and F_2 are left adjoints and F_3 is an equivalence. The composite of the respective right adjoints is the inclusion $\mathrm{Mod}(\mathfrak{S}, T) \subseteq \mathrm{Rel}(\mathfrak{S})$, so the composite of the F_i is the reflection into $\mathrm{Mod}(\mathfrak{S}, T)$. □

Remark 49. Proposition 47 shows that RHL can be reduced to Datalog (with minor extensions). In practice, however, using the resulting Datalog programs to compute free models is often unfeasible even for small inputs.

One issue is that storing the equivalence relations Eq_s naively requires quadratic memory with respect to the size of equivalence classes. This problem can be largely addressed by using a union-find data structure, which only requires linear memory. Union-find data structures are available in the Souffle Datalog engine (Nappa et al., 2019).

A more significant issue are the congruence axioms 2, which result in an exponential increase in memory requirements with respect to the arity of relations. Every equality inferred during evaluation can significantly increase the total size of the relational structure in the next stage. Semantically, however, every inferred equality should in fact *reduce* the size of the relational structure at the next stage. The Eqlog engine, which evaluates RHL theories directly, takes advantage of this observation by maintaining a union-find data structure on each sort, allowing for a canonical representative of each equivalence class. The relations then contain entries only for these canonical representatives. An inferred equality results in a merge of two equivalence classes, with one of the canonical representatives ceasing to be representative. Eqlog then *canonicalizes* all tuples by replacing each occurrence of the old representative with the new representative. Since relations are stored without duplicates, this often results in a decrease in the size of the relations, so that later stages can be computed faster.

Remark 50. The following alternative *sparse* setoid transformation $(\mathfrak{S}', T'')$ of an RHL theory $(\mathfrak{S}, T)$ can result in a more efficient Datalog program. The signature $\mathfrak{S}'$ of the sparse setoid transformation is the same as in the standard setoid transformation (Definition 46), so $\mathfrak{S}'$ contains additional equivalence relations Eq_s for all sorts s. As before, T'' contains the equivalence relation axioms 1. However, the congruence axioms 2 are omitted.

Instead, we modify the premise of each sequent $\mathcal{F} \Rightarrow \mathcal{G}$ in T so that the different occurences of a variable can be interpreted by distinct but equivalent elements. As before, we replace equality atoms $u \equiv v$ by atoms $\mathrm{Eq}_s(u, v)$ in premise and conclusion. Next, for each variable v that occurs $n > 0$ times in the premise $\mathcal{F}$, we choose a list $v = v^1, v^2, \ldots, v^n$ of variables of the same sort, where $v^2, \ldots, v^n$ are fresh. We now replace the ith occurence of v in $\mathcal{F}$ with v^i, and add the atoms $\mathrm{Eq}(v, v^i)$ for all $i = 2, \ldots, n$ to $\mathcal{F}$.

The transformed premises $\mathcal{F}''$ have the following property: If X is a relational $\mathfrak{S}'$-structure that satisfies the equivalence relation axioms and X' is the relational structure over X that furthermore satisfies the congruence axioms, then maps $[\mathcal{F}'] \to X'$ are in bijection to maps $[\mathcal{F}''] \to X$ up to setoid morphism equality. From this it follows that Corollary 48 holds also for the sparse transformation $(\mathfrak{S}', T'')$.

The sparse transformation avoids duplication in many cases, but data that can be infered twice for different but equivalent elements is still duplicated. Fur-

thermore, the transformed premises $\mathcal{F}''$ can be more computationally expensive to match because all elements in an equivalence class must be considered for every occurence of a variable in the original premise $\mathcal{F}$.

4 Partial Horn Logic

Partial Horn logic is one of the many equivalent notions of essentially algebraic theory. It was initially defined by Palmgren and Vickers (Palmgren and Vickers, 2007), who proved its equivalence to essentially algebraic theories. Here we shall understand PHL as a syntactic extension, as *syntactic sugar*, over RHL. The advantages of PHL, then, are entirely syntactical.

Observe that it cannot be read off from the individual sequents whether or not an RHL theory is strong or not. Instead, one has to consider the interplay between the different sequents of the theory. As a result, it is computationally undecidable whether or not a given RHL theory is strong.

Our application for the semantics developed in this paper are tools that allow computations based on the small object argument. Such tools compute (fragments of) free models of user-defined theories that encode problem domains. It is highly desirable that these theories are strong, since otherwise the result of the computation is not uniquely determined. As strong RHL theories are difficult to recognize for both humans and computers, we argue that RHL is not directly suitable as an input theory language for this purpose. What is needed, then, is a language with the same expressive power of RHL, but where an easily recognizable subset allows axiomatizing all strong theories.

PHL is indeed such a language: Proposition 79 shows that every strong RHL theory is equivalent to a PHL theory containing *epic* sequents only. PHL sequents are epic if no new variables are introduced in the conclusion, which is a criterion that can be easily checked separately for each sequent without regard for the theory the sequent appears in. If tools wish to allow only strong theories, they can use PHL as input language and reject non-epic sequents. Note that there exist PHL theories containing non-epic sequents which are nevertheless strong, but such theories can be equivalently axiomatized as epic PHL theories. Thus, no generality is lost compared to general strong theories when rejecting non-epic PHL theories.

4.1 Algebraic Structures

Definition 51. An *algebraic signature* is a relational signature (S, R) equipped with a partition $R = P \sqcup F$ of the set of relation symbol into disjoint sets P of *predicate symbols* and F of *function symbols* such that the arity of every function symbol is non-empty. If $f \in F$ is a function symbol, then we write $f : s_1 \times \cdots \times s_n \to s$ if the arity of f as a relation symbol is $f : s_1 \times \cdots \times s_n \times s$.

Definition 52. Let $\mathfrak{S} = (S, P \sqcup F)$ be an algebraic signature. An *algebraic structure* for $\mathfrak{S}$ is a relational structure X for $\mathfrak{S}$ such that f_X is the graph of a partial function for all $f \in F$. Thus if $(x_1, \ldots, x_n, y) \in f_X$ and $(x_1, \ldots, x_n, z) \in f_X$, then $y = z$. We use $f_X(x_1, \ldots, x_n)$ to denote the unique element y such that $(x_1, \ldots, x_n, y) \in f_X$, and we write $f_X(x_1, \ldots, x_n)\downarrow$ to say that such an element y exists. A *morphism of algebraic structures* is a mor-

phism of underlying relational structures. The *category of algebraic structures* is denoted by $\mathrm{Alg}(\mathfrak{S})$.

If the algebraic signature $\mathfrak{S}$ is clear from context, we abbreviate $\mathrm{Alg}(\mathfrak{S})$ as Alg.

Proposition 53. *Let $\mathfrak{S} = (S, P \sqcup F)$ be an algebraic signature and let X be a relational structure. Then X is an algebraic structure if and only if it satisfies the relational sequent*

$$f(v_1, \ldots, v_n, u_0) \wedge f(v_1, \ldots, v_n, u_1) \Rightarrow u_0 \equiv u_1 \tag{2}$$

for each function symbol $f : s_1 \times \cdots \times s_n \to s$.

Proof. The relational structure X satisfies the sequent (2) if and only if f_X is right-unique, i.e. the graph of a partial function. □

Corollary 54. *Let $\mathfrak{S} = (S, P \sqcup F)$ be an algebraic signature. The category of algebraic structures is a reflective subcategory of the category of relational structures. The reflections $X \to X'$ of relational structure X into* Alg *are effective epimorphisms.*

Proof. That every reflection is an effective epimorphisms follows from the fact that the classifying morphisms of the functionality axioms (2) are effective epimorphisms, hence so are all coproducts, pushouts and (infinite) compositions thereof. □

We denote the free algebraic structure functor by $\mathrm{FAlg} : \mathrm{Rel} \to \mathrm{Alg}$.

Corollary 55. *Let $\mathfrak{S} = (S, P \sqcup F)$ be an algebraic signature. The category of algebraic structures is complete and cocomplete.*

Proof. This follows from general facts about reflective subcatgories: They are stable under limits, and colimits are computed by reflecting colimits of the ambient category. □

4.2 Syntax and Semantics

Definition 56. Let $\mathfrak{S} = (S, P \sqcup F)$ be an algebraic signature. The set of *terms* and a sort assigned to each term is given by the following recursive definition:

1. If v is a variable of sort s, then v is a term of sort s.
2. If $f : s_1 \times \cdots \times s_n \to s$ is a function symbol and $t_1, \ldots, t_n$ are terms such that t_i has sort s_i for all $i = 1, \ldots, n$, then $f(t_1, \ldots, t_n)$ is a term of sort s.

Algebraic atoms, formulas and sequents are defined as in Definition 24, but with two changes:

1. In each type of atom, also composite terms of the same sort are allowed in place of only variables.
2. An algebraic atom $r(t_1, \ldots, t_n)$ is valid only if $r = p$ is a predicate symbol, but not if r is a function symbol.

Definition 57. Let $\mathfrak{S} = (S, P \sqcup F)$ be an algebraic signature and let X be an algebraic structure. An *interpretation of a term* t in X is an interpretation I of the variables occuring in t such that the following recursive extension of I is well-defined on t:

$$I(f(t_1, \ldots, t_n)) = f_X(I(t_1), \ldots, I(t_n)).$$

Note that the right-hand side might not be defined; in this case also the left-hand side is undefined.

An *interpretation of an algebraic atom* in X is defined analogously to the interpretation of a relational atom, but with the additional condition that the interpretation is defined on all (possibly composite) terms occuring in the atom.

An *interpretation of an algebraic formula* $\mathcal{F} = \phi_1 \wedge \cdots \wedge \phi_n$ is an interpretation of the variables occuring in $\mathcal{F}$ that restricts to an interpretation of ϕ_i for each $i \in \{1, \ldots, n\}$. An algebraic structure X *satisfies* an algebraic sequent $\mathcal{F} \Rightarrow \mathcal{G}$ if each interpretation of $\mathcal{F}$ in X can be extended to an interpretation of $\mathcal{F} \wedge \mathcal{G}$ in X.

Definition 58. Let $\mathfrak{S} = (S, P \sqcup F)$ be an algebraic signature, and let t be a term. The *flattening* of t consists of a relational formula $\mathrm{Flat}(t)$ and a result variable $v_{\mathrm{Flat}}(t)$. Flattening is defined recursively as follows:

1. If $t = v$ is a variable, then $\mathrm{Flat}(t) = \top$ is the empty conjunction and $v_{\mathrm{Flat}}(t) = v$.
2. If $t = f(t_1, \ldots, t_n)$, then

   $$\mathrm{Flat}(t) = \mathrm{Flat}(t_1) \wedge \cdots \wedge \mathrm{Flat}(t_n) \wedge f(v_{\mathrm{Flat}}(t_1), \ldots, v_{\mathrm{Flat}}(t_n), u)$$

   where $u =: v_{\mathrm{Flat}}(t)$ is a fresh variable.

Let ϕ be an algebraic atom. The flattening $\mathrm{Flat}(\phi)$ is a relational formula which is defined depending on the type of ϕ as follows:

1. If $\phi = p(t_1, \ldots, t_n)$ for some predicate p, then

   $$\mathrm{Flat}(\phi) = \mathrm{Flat}(t_1) \wedge \cdots \wedge \mathrm{Flat}(t_n) \wedge p(v_{\mathrm{Flat}}(t_1), \ldots, v_{\mathrm{Flat}}(t_n)).$$

2. If $\phi = t\downarrow$ for some term t, then

   $$\mathrm{Flat}(\phi) = \mathrm{Flat}(t) \wedge v_{\mathrm{Flat}}(t)\downarrow .$$

3. If ϕ is of the form $t_1 \equiv t_2$ for terms t_1, t_2, then

   $$\mathrm{Flat}(\phi) = \mathrm{Flat}(t_1) \wedge \mathrm{Flat}(t_2) \wedge v_{\mathrm{Flat}}(t_1) \equiv v_{\mathrm{Flat}}(t_2).$$

The flattening of an algebraic formula is the conjunction of the flattenings of each atom making up the formula. The flattening of an algebraic sequent is given by flattening premise and conclusion.

Remark 59. The flattening of a composite term $t = f(t_1, \ldots, t_n)$ involves the choice of a "fresh" variable u. This notion can be made precise as follows: The flattening operations take as additional parameter a sequence $(u_n)_{n\in\mathbb{N}}$ of

variables, such that the variables u_i do not occur in the syntactic objects that should be flattened. Choosing a "fresh" variable u now means that we set $u = u_0$, and for all further flattening operations we pass the sequence $(u_{n+1})_{n\in\mathbb{N}}$.

Note that a term t that appears twice in the same formula $\mathcal{F}$ will be flattened twice with different choices of fresh variables. For example, if f is a binary function symbol and x_1, x_2 are variables, then the flattening of the algebraic formula

$$\mathcal{F} = f(x_1, x_2) = x_1 \wedge f(x_1, x_2) = x_2$$

is the relational formula

$$\mathrm{Flat}(\mathcal{F}) = f(x_1, x_2, u_0) \wedge u_0 \equiv x_1 \wedge f(x_1, x_2, u_1) \equiv x_2$$

where $u_0 \neq u_1$.

Proposition 60. *Let $\mathfrak{S} = (S, P \sqcup F)$ be an algebraic signature and let X be an algebraic structure.*

1. *Let t be a term and let I be an interpretation of the variables of t in X. Then I can be extended to the term t if and only if I can be extended to an interpretation of the relational formula* $\mathrm{Flat}(t)$. *In either case, if an extension J exists, then it exists uniquely, and $J(v_{\mathrm{Flat}}(t)) = I(t)$.*

2. *Let ϕ be an algebraic atom and let I be an interpretation of the variables of ϕ in X. Then I is an interpretation of ϕ if and only if I can be extended to an interpretation J of the relational formula* $\mathrm{Flat}(\phi)$. *If J exists, then it exists uniquely.*

3. *Let $\mathcal{S}$ be an algebraic sequent. Then X satisfies $\mathcal{S}$ if and only if it satisfies the relational sequent* $\mathrm{Flat}(\mathcal{S})$.

Proof. By construction. □

Definition 61. Let $\mathfrak{S} = (S, P \sqcup F)$ be an algebraic signature. We associate to each algebraic formula $\mathcal{F}$ the following *classifying algebraic structure*:

$$[\mathcal{F}] = \mathrm{FAlg}([\mathrm{Flat}(\mathcal{F})])$$

The composition of the generic interpretion $I_{\mathrm{Flat}(\mathcal{F})}$ with the reflection into Alg induces an interpretation of $\mathrm{Flat}(\mathcal{F})$ in $[\mathcal{F}]$, which then restricts to an interpretation $I_{\mathcal{F}}$ of $\mathcal{F}$ in $[\mathcal{F}]$. We call $I_{\mathcal{F}}$ the *generic interpretation* of $\mathcal{F}$.

Proposition 62. *Let $\mathfrak{S} = (S, P \sqcup F)$ be an algebraic signature. Let $\mathcal{F}$ be an algebraic formula and let X be an algebraic structure. Then there is a bijection between interpretations of $\mathcal{F}$ in X and maps $[\mathcal{F}] \to X$.*

Proof. This follows by combining Proposition 60, Proposition 62 and the universal property of the free algebraic structure functor. □

Definition 63. Let $\mathcal{S} = \mathcal{F} \Rightarrow \mathcal{G}$ be an algebraic sequent. The *classifying morphism* of $\mathcal{S}$ is the map $[\mathcal{S}] : [\mathcal{F}] \to [\mathcal{F} \wedge \mathcal{G}]$ that is induced by the canonical interpretation of $\mathcal{F}$ in $[\mathcal{F} \wedge \mathcal{G}]$.

Proposition 64. *Let $\mathcal{S}$ be an algebraic sequent and let X be an algebraic structure. Then X satisfies $\mathcal{S}$ if and only if X is injective to $[\mathcal{S}]$.*

Proof. Analogous to the proof of Proposition 29. □

Proposition 65. *Let T be an algebraic theory. Denote the functionality sequent 2 for function symbols $f \in F$ by f_{func}. Then* $\text{Mod}(T)$ *is equivalent to the following injectivity classes:*

1. $(M_1)^{\pitchfork} \subseteq \text{Alg}(\mathfrak{S})$, *where* $M_1 = \{[\mathcal{S}] \mid \mathcal{S} \in T\}$.
2. $(M_2)^{\pitchfork} \subseteq \text{Rel}(\mathfrak{S})$, *where* $M_2 = \{[f_{\text{func}}] \mid f \in F\} \cup \{[\mathcal{S}] \mid \mathcal{S} \in T\}$.
3. $(M_3)^{\pitchfork} \subseteq \text{Rel}(\mathfrak{S})$, *where* $M_3 = \{[f_{\text{func}}] \mid f \in F\} \cup \{[\text{Flat}(\mathcal{S})] \mid \mathcal{S} \in T\}$.

Here $[\mathcal{S}]$ in the definition of M_2 denotes the classifying morphism of the algebraic sequent $\mathcal{S}$, which we regard as a morphism of relational structures, and $[\text{Flat}(\mathcal{S})]$ *denotes the classifying morphism of the relational sequent* $\text{Flat}(\mathcal{S})$.

In particular, $\text{Mod}(T)$ *is a weakly reflective subcategory of both* $\text{Rel}(\mathfrak{S})$ *and* $\text{Alg}(\mathfrak{S})$. *If any one of M_1, M_2 or M_3 are strong, then all of them are strong, and* $\text{Mod}(T)$ *is a reflective subcategory of* $\text{Rel}(\mathfrak{S})$ *and* $\text{Alg}(\mathfrak{S})$.

Proof. 1 follows from Proposition 62, and then 2 and 3 follow from Proposition 17. □

Proposition 66. *Let $\mathfrak{S} = (S, P \sqcup F)$ be an algebraic signature. Let ϕ be a relational formula. Then there exists an algebraic formula* $\text{Unflat}(\phi)$ *with the following properties:*

1. *Every variable occurs in ϕ if and only if it occurs in* $\text{Unflat}(\phi)$.
2. *Let I be an interpretation of the variables of ϕ in an algebraic structure X. Then I is an interpretation of the algebraic formula ϕ if and only if it is an interpretation of the relational formula* $\text{Unflat}(\phi)$.

Proof. Replace every relational atom of the form $f(x_1, \ldots, x_n, x)$ for some function symbol $f : s_1 \times \cdots \times s_n \to s$ with the algebraic atom $f(x_1, \ldots, x_n) \equiv x$. □

Proposition 67. *Let $\mathcal{S} = \mathcal{F} \Rightarrow \mathcal{G}$ and $\mathcal{T} = \mathcal{G} \Rightarrow \mathcal{H}$ be algebraic sequents. Then*

$$[\mathcal{F} \wedge \mathcal{G} \Rightarrow \mathcal{H}] \circ [\mathcal{F} \Rightarrow \mathcal{G}] = [\mathcal{F} \Rightarrow \mathcal{G} \wedge \mathcal{H}].$$

Proof. This follows from Proposition 62. □

4.3 Completeness Results

Definition 68. Let $\mathcal{S}$ be an algebraic sequent. We say that $\mathcal{S}$ is *epic* if every variable in the conclusion of $\mathcal{S}$ also occurs in the premise. We say that $\mathcal{S}$ is *effectively epic* if every atom in the conclusion of $\mathcal{S}$ is of the form $t_1 \equiv t_2$ for terms t_1, t_2 that also occur in the premise.

Definition 69. Let f be a function symbol. The *totality sequent* $f \downarrow$ is given by

$$v_1 \downarrow \wedge \cdots \wedge v_n \downarrow \Rightarrow f(v_1, \ldots, v_n) \downarrow .$$

We denote by

$$\text{Tot} = \{[f \downarrow] \mid f \in F\}$$

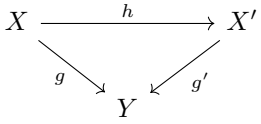
the set of classifying morphisms of totality sequents.

Proposition 70. *Let f be a function symbol.*

1. *An algebraic structure X satisfies $f\downarrow$ if and only if f_X is a total function.*
2. *$[f\downarrow]$ is an epimorphism of algebraic structures.* □

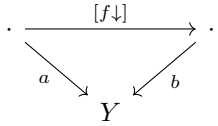
Definition 71. Let $g : X \to Y$ be a map of algebraic structures. We say that X is *total over* Y (with respect to g) if and only if for all function symbols f and elements $x_1, \ldots, x_n \in X$, if $f_Y(g(x_1), \ldots, g(x_n))$ is defined, then $f_X(x_1, \ldots, x_n)$ is defined.

Proposition 72. *Let $g : X \to Y$ be a map of algebraic structures. Then there exists a factorization*



such that h is a relative Tot*-cell complex and X' is total over Y. Moreover, the triple (X', h, g') is uniquely determined by g up to unique isomorphism.*

Proof. Consider the set Tot_Y of all triples (f, a, b) corresponding to commuting triangles



Tot_Y is a set of epimorphisms in the slice category $\mathrm{Alg}_{/Y}$. It follows that Tot_Y is strong, so $(\mathrm{Tot}_Y)^{\pitchfork}$ is a reflective subcategory of $\mathrm{Alg}_{/Y}$. Partial algebras over Y are total over Y if and only if they are injective to M. It follows that the triple (X', h, g) exists and is unique up to unique isomorphism, and that h is a relative Tot_Y-cell complex.

It remains to show that h is a relative Tot-cell complex. By definition, the class of relative M-cell complexes is obtained from M by closure under certain classes of colimits. The forgetful functor $\mathrm{Alg}_{/Y} \to \mathrm{Alg}$ preserves colimits and maps Tot_Y into Tot. From this it follows that the image of a relative Tot_Y-cell complex in Alg is a relative Tot-complex. In particular, h is a relative Tot-cell complex. □

Definition 73. Let $g : X \to Y$ be a map of algebraic structures. We call the unique factorization $g = g'h$ as in Proposition 72 the *relative totalization* of X.

Proposition 74. *Let $g : X \to Y$ be a map of algebraic structures such that X is total over Y. Then g is an epimorphism in* Alg *if and only if it is an epimorphism in* Rel*.*

Proof. The free algebraic structure functor $\mathrm{Rel} \to \mathrm{Alg}$ preserves epimorphisms, which shows necessity of the condition.

Note that $\mathrm{Im}_{\mathrm{Alg}}\, g = \mathrm{Im}_{\mathrm{Rel}}\, g$, that is, the image algebraic structure can be computed as image of relational structures. Because g is an epimorphism in Rel or Alg if and only if the inclusion of the image in the corresponding category is an epimorphism, we may assume that g is a monomorphism, i.e. injective on carrier sets.

Consider the pushout $Z = Y \amalg_X^{\mathrm{Rel}} Y$ in Rel. There are inclusions $Y \cong Y_0 \subseteq Z$, $Y \cong Y_1 \subseteq Z$ corresponding to the two components of Z such that $Y_0 \cup Y_1 = Z$, and an inclusions $X \subseteq Y_0, X \subseteq Y_1$. We have $(Y_0)_s \cap (Y_1)_s = X_s$ for all sorts s, but note that the analogous equation does not hold for relations: Epimorphisms of relational structures need not be surjective on relations.

We claim that Z is an algebraic structure. Thus let f be a function symbol, and let $z_1, \ldots, z_n, z, z' \in Z$ such that $\bar{z} = (z_1, \ldots, z_n, z) \in f_Z$ and $\bar{z}' = (z_1, \ldots, z_n, z') \in f_Z$. We need to show that $z = z'$.

If $\bar{z}, \bar{z}' \in Y_0$ or $\bar{z}, \bar{z}' \in Y_1$ this follows from the fact that $Y_0 \cong Y \cong Y_1$ is an algebraic structure. We may thus (by symmetry) assume that $\bar{z} \in Y_0$ and $\bar{z}' \in Y_1$. Because the first n projections of $\bar{z}$ and $\bar{z}'$ agree, we have $z_i \in Y_0 \cap Y_1$, hence $z_i \in X$ for $i \in \{1, \ldots, n\}$. Because X is total over the Y_i with respect to the inclusions $X \subseteq Y_i$, we have $f_X(z_1, \ldots, z_n) = z''$ for some $z'' \in X$. It follows that $z = z''$ and $z' = z''$, hence $z = z'$.

As in every category with colimits, g is an epimorphism if and only if the two maps $Y \to Y \amalg_X^{\mathrm{Alg}} Y =: Z'$ agree. But we have just shown that $Z' = Z$, so also the two maps $Y \to Y \amalg_X^{\mathrm{Rel}} Y$ agree. Thus g is an epimorphism in Rel. □

Proposition 75. *Let $g : X \to Y$ be a map of algebraic structures. Let $g'h = g$ be the relative totalization of X. Then the following holds:*

1. *g is an epimorphism in* Alg *if and only if g' is an epimorphism in* Rel.
2. *g is an effective epimorphism in* Alg *if and only if it is an effective epimorphism in* Rel.

Proof. 1. Every morphism in Tot is an epimorphism. Since epimorphisms are stable under pushouts and transfinite compositions, every relative Tot-cell complex and in particular h is an effective epimorphism. Thus g is an epimorphism in Alg if and only if g' is an epimorphism in Alg. We conclude with Proposition 74.

2. This follows from general properties of colimits and reflective subcategories. □

Definition 76. An algebraic sequent $\mathcal{S}$ is called *epic* if every variable that occurs in the conclusion of $\mathcal{S}$ also occurs in the premise. $\mathcal{S}$ is called *effectively epic* if every atom in the conclusion of $\mathcal{S}$ is of the form $t_1 \equiv t_2$, where t_1 and t_2 are terms that occur in the premise of $\mathcal{S}$.

Proposition 77. *Let $\mathfrak{S} = (S, P \sqcup F)$ be an algebraic signature. Let $\mathcal{F}$ be an algebraic formula and let V be the set of variables that occur in $\mathcal{F}$. Let* $\mathrm{Flat}(\mathcal{F}) = \phi_1 \wedge \cdots \wedge \phi_n$ *for relational atoms ϕ_i. Let $\phi = \phi_i$ for some $i \in \{1, \ldots, n\}$ and let*

$$V = \{v \mid v \text{ occurs in } \mathcal{F} \text{ or in } \phi_j \text{ for some } j < i\}.$$

Then ϕ has one of the following forms:

1. *$\phi = p(v_1, \ldots, v_m)$, where p is a predicate symbol and $v_1, \ldots, v_m \in V$.*
2. *$\phi = f(v_1, \ldots, v_m, v)$, where $f : s_1 \times \cdots \times s_m \to s$ is a function symbol, $v_1, \ldots, v_m \in V$ and $v \notin V$.*
3. *$\phi = v\!\downarrow$, where $v \in V$.*

4. $\phi = v_1 \equiv v_2$*, where* $v_1, v_2 \in V$*.*

Proof. Follows inductively from the definition of flattening; see also Remark 59. □

Proposition 78. *Let* $\mathfrak{S} = (S, P \sqcup F)$ *be an algebraic signature. The image of (epic, effectively epic) algebraic sequents under the assignment* $\mathcal{S} \mapsto [\mathcal{S}]$ *can be described as follows:*

1. *If* $\mathcal{S}$ *is an algebraic sequent, then* $[\mathcal{S}]$ *is a map of finite algebraic structures. Conversely, every map of finite relational structures is isomorphic to a map of the form* $[\mathcal{S}]$ *for some algebraic sequent* $\mathcal{S}$*.*

2. *If* $\mathcal{S}$ *is an epic algebraic sequent, then* $[\mathcal{S}]$ *is an epimorphism of finite algebraic structures. Conversely, every epimorphism of finite relational structures is isomorphic to a map of the form* $[\mathcal{S}]$ *for some epic algebraic sequent* $\mathcal{S}$*.*

3. *If* $\mathcal{S}$ *is a quotient algebraic sequent, then* $[\mathcal{S}]$ *is an effective epimorphism of finite algebraic structures. Conversely, every effective epimorphism of finite relational structures is isomorphic to a map of the form* $[\mathcal{S}]$ *for some quotient algebraic sequent* $\mathcal{S}$*.*

Proof. 1. If $\mathcal{S}$ is an algebraic sequent, then by Proposition 36, $[\mathrm{Flat}(\mathcal{S})]$ is a map of finite relational structures, hence $\mathrm{FAlg}([\mathrm{Flat}(\mathcal{S})])$ is a map of finite algebraic structures.

Conversely, let $g : X \to Y$ be a map of finite algebraic structures. Then g is, in particular, a map of finite relational structures. By Proposition 36, there exists a relational sequent $\mathcal{S}$ such that $[\mathcal{S}] \cong g$. We now invoke Proposition 66 to obtain an algebraic sequent $\mathrm{Unflat}(\mathcal{S})$ such that $[\mathrm{Unflat}(\mathcal{S})] \cong g$.

3. Let $\mathcal{S} = \mathcal{F} \Rightarrow \mathcal{G}$ be an effectively epic algebraic sequent. By Proposition 67, we may assume by induction that the conclusion $\mathcal{G}$ consists of a single atom $t_1 \equiv t_2$ for terms t_1, t_2 that also occur in the premise $\mathcal{F}$. Let $u_1 = v_{\mathrm{Flat}}(t_1)$ and $u_2 = v_{\mathrm{Flat}}(t_2)$ be two variables corresponding to flattenings of t_1 and t_2 in the premise. Then for the effectively epic relational sequent $\mathcal{S}' = \mathrm{Flat}(\mathcal{F}) \Rightarrow v_{\mathrm{Flat}}(v_1) \equiv v_{\mathrm{Flat}}(v_2)$ we have $\mathrm{FAlg}([\mathcal{S}']) \cong [\mathcal{S}]$. By Proposition 36, $[\mathcal{S}']$ is an effective epimorphism of relational structures. Because the free algebra functor is a left adjoint, it preserves effective epimorphisms. Thus $\mathrm{FAlg}(\mathcal{S}') \cong [\mathcal{S}]$ is an effective epimorphism of algebraic structures.

Conversely, let $g : X \to Y$ be an effective epimorphism of algebraic structures. Then g is also effectively epic as morphism of relational structures. By clause 3 of Proposition 36, we find an effectively epic relational sequent $\mathcal{S}$ such that $[\mathcal{S}] \cong g$. Then $\mathcal{S}' = \mathrm{Unflat}(\mathcal{S})$ is an effectively epic algebraic sequent such that $[\mathcal{S}'] = g$ as desired.

2. Let $\mathrm{Flat}(\mathcal{S}) = \mathcal{F} \Rightarrow \mathcal{G}$ be the flattening of an epic algebraic formula. We need to show that $g = \mathrm{FAlg}([\mathcal{F} \Rightarrow \mathcal{G}])$ is an epimorphism in Alg.

By Proposition 67, we may without loss of generality assume that $\mathcal{G} = \phi$ is a relational atom of one of the types as in Proposition 77, where ϕ is the set of variables in $\mathcal{F}$. In cases 1, 3 and 4, the morphism $[\mathcal{F} \Rightarrow \phi]$ of relational structures is surjective. Because the free algebraic structure functor preserves epimorphisms, this implies that $g = \mathrm{FAlg}([\mathcal{F} \Rightarrow \phi])$ is an epimorphism in Alg. In case 1, $[\mathcal{F} \Rightarrow \phi]$ is a pushout of $[f\downarrow]$ in Rel. Since $[f\downarrow] = \mathrm{FAlg}([f\downarrow])$ is an

epimorphism in Alg and pushouts preserve epimorphisms, it follows that also in this case g is an epimorphism.

Conversely, let $g : X \to Y$ be an epimorphism of finite algebraic structures. Let $g = g'h$ be the relative totalization of X over Y. Since h is a relative Tot-cell complex, there exists by Proposition 9 a sequence of pushout squares

$$\begin{array}{ccc} [v_1 \downarrow \wedge \cdots \wedge v_n \downarrow] & \xrightarrow{[f_n\downarrow]} & [f_n(v_1, \ldots, v_n) \downarrow] \\ \downarrow{\scriptstyle a} & & \downarrow{\scriptstyle b} \\ X_n & \xrightarrow{\quad h_n \quad} & X_{n+1} \end{array}$$

for a totality sequent $f_n \downarrow$ for all n such that h is the infinite composition of the h_n. Note that, a priori, Proposition 9 implies only that h_n is a pushout of a *coproduct* of totality sequents. However, finiteness of X and Y implies inductively that these coproducts can be chosen to be finite, and then a single pushout of a finite coproduct can equivalently be written as a finite composition of pushout.

We claim that $h_n \cong [\mathcal{S}]$ for some epic algebraic sequent $\mathcal{S}$ for all $n \geq 0$. To verify this, choose first an algebraic formula $\mathcal{F}$ such that $[\mathcal{F}] \cong X_n$. A formula $\mathcal{F}$ with this property exists by clause 1 above because the identity on X_n is a map of finite algebraic structures. The map a corresponds to elements $x_1, \ldots, x_n \in X$, and these elements are the interpretations of terms $t_1, \ldots, t_n$ that occur in $\mathcal{F}$. Now

$$h_n \cong [\mathcal{F} \Rightarrow f_n(t_1, \ldots, t_n) \downarrow].$$

Let $g'_n : X_n \to \operatorname{colim}_i X_i \xrightarrow{g'} Y$ for $n \geq 0$. Because Y is finite, the chain

$$\operatorname{Im} g'_0 \subseteq \operatorname{Im} g'_1 \subseteq \cdots \subseteq Y$$

is eventually stationary, say for $n \geq n_0$, and then $\operatorname{Im} g'_{n_0} = \operatorname{Im} g'$. g is an epimorphism, hence g' is a surjection by Proposition 75, hence also $g'_{n_0} : X_{n_0} \to Y$ is a surjection. Thus $g'_{n_0} \cong [\mathcal{S}] \cong [\mathrm{Unflat}(\mathcal{S})]$ for some surjective relational sequent $\mathcal{S}$. Note that the algebraic sequent $\mathrm{Unflat}(\mathcal{S})$ is epic because the relational sequent $\mathcal{S}$ is surjective.

We have thus decomposed g into a composition

$$X = X_0 \xrightarrow{h_0} X_1 \xrightarrow{h_1} X_2 \xrightarrow{h_2} \ldots \xrightarrow{h_{n_0-1}} X_{n_0} \xrightarrow{g_{n_0}} Y$$

in which each map is isomorphic to $[\mathcal{S}]$ for some epic algebraic sequent $\mathcal{S}$. If $\mathcal{F} \Rightarrow \mathcal{G}$ is an epic algebraic sequent and $\mathcal{F} \wedge \mathcal{G} \Rightarrow \mathcal{H}$ is an epic algebraic sequent for algebraic formulas $\mathcal{F}, \mathcal{G}, \mathcal{H}$, then also $\mathcal{F} \Rightarrow \mathcal{H}$ is epic. Thus by Proposition 67, $g \cong [\mathcal{S}]$ for some epic algebraic sequent $\mathcal{S}$. □

Proposition 79. *Let $\mathfrak{S} = (S, R)$ be a relational signature. Let M be a set of morphisms of finite relational $\mathfrak{S}$-structures. Then there exists an algebraic signature $\mathfrak{S}' = (S, P \sqcup F)$ on the same set of sorts S such that $P = R$ and a set M' of epimorphisms of finite algebraic $\mathfrak{S}'$-structures such that the forgetful functor $\mathrm{Alg}(\mathfrak{S}') \to \mathrm{Rel}(\mathfrak{S})$ restricts to an equivalence $M^{\perp} \simeq (M')^{\pitchfork}$.*

Proof. Choosing a relational sequent $\mathcal{S}_g : \mathcal{F}_g \Rightarrow \mathcal{G}_g$ such that $[\mathcal{S}_g] \cong g$ for each $g \in M$, we identify M with a set of relational sequents.

Our set of function symbols F is given by

$$F = \{f_{\mathcal{S},v} \mid \mathcal{S} = \mathcal{F} \Rightarrow \mathcal{G} \text{ is in } M \text{ and } v \in \mathrm{Var}(\mathcal{G}) \setminus \mathrm{Var}(\mathcal{F})\}.$$

Let $\mathcal{S} = \mathcal{F} \Rightarrow \mathcal{G}$ be in M. Let $v_1, \ldots, v_n$ be an enumeration of the variables in $\mathcal{F}$, let s_i be the sort of v_i and let s be the sort of v. Let $v \in \mathrm{Var}(\mathcal{G}) \setminus \mathrm{Var}(\mathcal{F})$. Then the signature of $f_{\mathcal{S},v}$ is given by $f_{\mathcal{S},v} : s_1 \times \cdots \times s_n \to s$.

Note that each relation symbol in $\mathfrak{S}$ corresponds to a predicate symbol in $\mathfrak{S}'$. We thus implicitly coerce relational $\mathfrak{S}$-sequents to algebraic $\mathfrak{S}'$-sequents (without invoking Unflat).

Let M' be the set containing the following algebraic sequents, for all $\mathcal{S} = \mathcal{F} \Rightarrow \mathcal{G}$ in M:

1. The sequent $\mathcal{F} \Rightarrow \mathcal{G}'$, where $\mathcal{G}'$ is obtained from $\mathcal{G}$ by replacing each variable v in $\mathcal{G}$ with $f_{\mathcal{S},v}(v_1, \ldots, v_n)$.

2. The sequents
$$f_{\mathcal{S},v}(v_1, \ldots, v_n) \downarrow \Rightarrow \mathcal{F}$$
for all variables v in $\mathcal{G}$.

3. The sequent
$$\mathcal{F} \wedge \mathcal{G} \Rightarrow \bigwedge_{v \in \mathrm{Var}(\mathcal{G}) \setminus \mathrm{Var}(\mathcal{F})} v \equiv f_{\mathcal{S},v}(v_1, \ldots, v_n).$$

Clearly all algebraic sequents in M' are epic. Let $G : \mathrm{Alg}(\mathfrak{S})' \to \mathrm{Rel}(\mathfrak{S})$ be the forgetful functor.

We first show that G maps algebraic structures $X \in (M')^{\pitchfork}$ to $M^{\perp}$. Let $\mathcal{S} = \mathcal{F} \Rightarrow \mathcal{G}$ be a sequent in M and let $V = \{v_1, \ldots, v_n\}$ be the enumeration of variables in $\mathcal{F}$ that we chose in the definition of the signature of the function symbols $f_{\mathcal{S},v}$. Let I be an interpretation of $\mathcal{F}$ in $G(X)$. As mentioned earlier, we implicitly treat $\mathcal{F}$ also as an algebraic sequent for the signature $\mathfrak{S}'$, and under this identification we can view I as an interpretation of the algebraic sequent $\mathcal{F}$ in the algebraic structure X. Because X satisfies the sequent 1, it follows that I is also an interpretation of $\mathcal{G}'$ in X. By definition of $\mathcal{G}'$, it follows that we obtain an interpretation J of $\mathcal{F} \wedge \mathcal{G}$ by setting $J(v) = I(v)$ if v occurs in $\mathcal{F}$ and $J(v) = I(f_{\mathcal{S},v}(v_1, \ldots, v_n))$ if v does not occur in $\mathcal{F}$.

Thus $G(X)$ satisfies the sequent $\mathcal{F} \Rightarrow \mathcal{G}$. Two interpretations of $\mathcal{F} \wedge \mathcal{G}$ in $G(X)$ that agree on the variables in $\mathcal{F}$ agree also on the variables that occur in $\mathcal{G}$ because of sequent 3. We have thus shown that G restricts to a functor $(M')^{\pitchfork} \to M^{\perp}$.

We now construct an inverse functor $H : M^{\perp} \to (M')^{\pitchfork}$. Thus, let $Y \in M^{\perp}$ and let us construct $X = H(Y) \in (M')^{\pitchfork}$ such that $G(X) = Y$. This leaves us no choice but to define X on the same carrier sets as Y with $r_X = r_Y$ for $r \in R$. Let $\mathcal{S} = \mathcal{F} \Rightarrow \mathcal{G}$ be in M, and let $v_1, \ldots, v_n$ be the enumeration of the variables in $\mathcal{F}$ that we chose earlier. Then we set

$$(f_{\mathcal{S},v})_X(J(v_1), \ldots, J(v_n)) = J(v) \tag{3}$$

whenever J is an interpretation of $\mathcal{F} \wedge \mathcal{G}$ in Y. Since Y satisfies $\mathcal{F} \Rightarrow \mathcal{G}$ uniquely, two interpretations of $\mathcal{F} \wedge \mathcal{G}$ are equal as soon as they agree on the variables in $\mathcal{F}$. Thus the sequent (3) defines partial functions as required. By construction, X satisfies the sequent 1 and the sequents 2. Because Y satisfies $\mathcal{F} \Rightarrow \mathcal{G}$ uniquely, X satisfies the sequents 3.

It remains to show that H is functorial. Thus let $g : Y_0 \to Y_1$ be a map of relational structures $Y_0, Y_1 \in M^{\perp}$, and let $X_i = H(Y_i)$ for $i \in \{0, 1\}$. We need to show that the action of g on carrier sets is also a morphism $X_0 \to X_1$, i.e. that it preserves partial functions. This follows from the definition of the partial functions (3) because if J is an interpretation of $\mathcal{F} \wedge \mathcal{G}$ in Y_0, then $g \circ J$ is an interpretation of $\mathcal{F} \wedge \mathcal{G}$ in Y_1. □

References

J. Adamek and J. Rosicky. *Locally Presentable and Accessible Categories*. Cambridge University Press, mar 1994. doi: 10.1017/cbo9780511600579. URL https://doi.org/10.1017/cbo9780511600579.

M. Bravenboer and Y. Smaragdakis. Strictly declarative specification of sophisticated points-to analyses. In *Proceeding of the 24th ACM SIGPLAN conference on Object oriented programming systems languages and applications - OOPSLA 09*. ACM Press, 2009. doi: 10.1145/1640089.1640108. URL https://doi.org/10.1145/1640089.1640108.

S. Ceri, G. Gottlob, and L. Tanca. What you always wanted to know about datalog (and never dared to ask). 1(1):146–166, mar 1989. doi: 10.1109/69.43410. URL https://doi.org/10.1109/69.43410.

P. J. Downey, R. Sethi, and R. E. Tarjan. Variations on the common subexpression problem. *Journal of the ACM*, 27(4):758–771, oct 1980. doi: 10.1145/322217.322228. URL https://doi.org/10.1145/322217.322228.

M. Hovey. *Model categories*. Number 63. American Mathematical Soc., 2007.

X. Hu, J. Karp, D. Zhao, A. Zreika, X. Wu, and B. Scholz. The choice construct in the soufflé language. In *Programming Languages and Systems*, pages 163–181. Springer International Publishing, 2021. doi: 10.1007/978-3-030-89051-3_10. URL https://doi.org/10.1007/978-3-030-89051-3_10.

M. Madsen, M.-H. Yee, and O. Lhoták. From datalog to flix: a declarative language for fixed points on lattices. *ACM SIGPLAN Notices*, 51(6):194–208, jun 2016. doi: 10.1145/2980983.2908096. URL https://doi.org/10.1145/2980983.2908096.

R. Milner. A theory of type polymorphism in programming. *Journal of Computer and System Sciences*, 17(3):348–375, dec 1978. doi: 10.1016/0022-0000(78)90014-4. URL https://doi.org/10.1016/0022-0000(78)90014-4.

P. Nappa, D. Zhao, P. Subotic, and B. Scholz. Fast parallel equivalence relations in a datalog compiler. In *2019 28th International Conference on Parallel*

Architectures and Compilation Techniques (PACT). IEEE, sep 2019. doi: 10.1109/pact.2019.00015. URL https://doi.org/10.1109/pact.2019.00015.

E. Palmgren and S. Vickers. Partial horn logic and cartesian categories. *Annals of Pure and Applied Logic*, 145(3):314–353, mar 2007. doi: 10.1016/j.apal.2006.10.001. URL https://doi.org/10.1016/j.apal.2006.10.001.

B. Steensgaard. Points-to analysis in almost linear time. In *Proceedings of the 23rd ACM SIGPLAN-SIGACT symposium on Principles of programming languages - POPL '96*. ACM Press, 1996. doi: 10.1145/237721.237727. URL https://doi.org/10.1145/237721.237727.

J. Whaley and M. S. Lam. Cloning-based context-sensitive pointer alias analysis using binary decision diagrams. *ACM SIGPLAN Notices*, 39(6):131–144, jun 2004. doi: 10.1145/996893.996859. URL https://doi.org/10.1145/996893.996859.

M. Willsey, C. Nandi, Y. R. Wang, O. Flatt, Z. Tatlock, and P. Panchekha. Egg: Fast and extensible equality saturation. *Proc. ACM Program. Lang.*, 5 (POPL), jan 2021. doi: 10.1145/3434304. URL https://doi.org/10.1145/3434304.