

Queries Determined by Views: Pack Your Views

Maarten Marx
Universiteit van Amsterdam,
the Netherlands
marx@science.uva.nl

ABSTRACT

A query Q is determined by a set of views V if, whenever $V(I1) = V(I2)$ for two database instances $I1, I2$ then also $Q(I1) = Q(I2)$. Does this imply that Q can be rewritten as a query Q' that only uses the views V ?

For first-order (FO) queries and view definitions over possibly infinite databases, the answer is yes, as follows from old results of Beth and Craig. We say that FO is complete for FO-to-FO rewritings. However, Nash, Segoufin and Vianu (2007) prove that if the query and the view definitions are given by conjunctive queries, then it might not be possible to formulate Q' as a conjunctive query. In other words, CQ is not complete for CQ-to-CQ rewritings.

Here we consider queries and view definitions in the packed fragment (PF) of first-order logic. This is a generalization of the guarded fragment, a fragment of particular interest to database theory. Gottlob et.al. 2002 show that the guarded conjunctive queries are exactly the acyclic queries. Leinders et.al. 2005 characterize the entire guarded fragment by the semijoin algebra.

We show that for both finite and unrestricted databases, PF is complete for PF-to-PF rewritings. The same holds for packed (unions of) conjunctive queries. In both cases, we provide algorithms for testing whether a query is determined by a set of views, and for actually rewriting Q to Q' . To compare: these problems are undecidable for full FO, and still open for conjunctive queries.

Categories and Subject Descriptors

H.1 [Information Systems]: Models and Principles

General Terms

Languages, Theory

Keywords

Views, Guarded Fragment, Query rewriting, Definability

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

PODS'07, June 11–13, 2007, Beijing, China.

Copyright 2007 ACM 978-1-59593-685-1/07/0006 ...\$5.00.

1. INTRODUCTION

Answering queries using another set of queries (usually called views) arises in numerous contexts in database theory and practice. Whatever the actual application, a natural question is how to formalize the intuitive notion that the views are sufficient to answer the query. This is closely related to the theory of definitions [26]. A first attempt at such formalization might be syntactic:

Query Q can be answered by the views V if there exists a query Q' in language L in which the V predicates are the only primitive symbols and Q and Q' are logically equivalent assuming the view definitions.

In this case, we say that there exists an L -rewriting of Q in terms of V . A second attempt might be semantic or information theoretic [24, 5]:

Query Q can be answered by the views V if $V(D_1) = V(D_2)$ guarantees that $Q(D_1) = Q(D_2)$, for all database instances D_1, D_2 .

In this case, we say that Q is *determined* by the views V .

If there exists an L -rewriting of Q in terms of V , then Q is also determined by V . The other direction may not hold, and depends of course on the choice of the rewriting language L .

The research program outlined in [24] can be briefly summarized as follows. For interesting query languages L , answer this question:

given a query Q and views V in L for which we know that Q is determined by V . What is the smallest query language extending L in which we can rewrite Q in terms of V ? In particular, is L itself sufficient?

If L itself is sufficient, we say that L *admits view-based rewritings*. This is of course the desired situation: the rewriting then is not more difficult than the views and the queries. Unfortunately, the conjunctive queries do *not* admit view-based rewritings [22]. For possibly infinite database, first order logic (FO) does admit view-based rewritings, by the theorems of Beth and Craig. But for finite databases, FO is not sufficient [24].

We contribute to this research program in two ways. First we provide a historical perspective, tracing the program back to, who else, Tarski, and indicating the relevance of the work of Beth and Craig. Second, we provide an example of

a practically relevant query language L which admits view-based rewritings, the packed fragment of first-order logic. This fragment has three properties which made it a good candidate for positive results: (1) it has the finite model property, (2) the counterexample of [22] mentioned in the abstract cannot be expressed in it, (3) it is decidable whether a query is determined by a set of views.

Our main results are

1. The packed fragment admits view-based rewritings, so do its restrictions to (unions of) conjunctive queries.
2. Determinacy is decidable for queries and views expressed in the packed fragment in 2EXPTIME .
3. For packed conjunctive queries and views, if the query is determined by the views, a rewriting can be obtained by a simple chase.

The paper is organized as follows. The next section provides the historical perspective. Section 3 recalls the packed fragment and its main properties. Sections 4 and 5 contain the main results —completeness of the packed fragment for rewritings and a complexity analysis. We conclude in Section 6.

2. HISTORICAL PERSPECTIVE AND RELATED WORK

Answering queries using views arises in numerous contexts in database theory and practice. Common is the basic problem that one is given a query Q in some signature σ , but one only has a set of σ -views V available to provide the answer to Q . Somehow we must create a query Q' which resembles Q as much as possible but is in the signature in which the views are the primitive symbols.

The most basic logical question is then the following: Suppose we know that the views V provide enough information to uniquely determine the answer to Q , what is the *minimal* language required to express Q *equivalently* in terms of V ? The answer to this question depends on the particular setting, i.e., the languages in which Q and V are expressed, and the kind of structures (databases) considered.

In the literature this setting is known as equivalent query rewriting in the case of exact view definitions, and is studied in [24, 5, 15]. Other settings are studied more frequently:

Sound views Here there is a difference between the view definition and its materialization available for query answering: the latter contains only a subset of the tuples in the answer.

Maximally contained rewritings In case the target language is fixed beforehand, one may require such a rewriting instead of an equivalent one (e.g., because the latter may not always exist).

In this work, as in [24], we focus on exact view definitions and equivalent rewritings. In this specific setting, the question has only been addressed in [5, 15], and then for specific query and view languages. For work within the other settings, see the related work sections in [5, 24].

To sum up, the same question has been addressed in similar situations in a number of places. As far as we know, Segoufin and Vianu [24] are the first within the database

community who generalize the results about individual systems into a research theme by formulating the basic logical question as a property of logical systems.

Here we relate this research theme to closely related theory in mathematical logic and in the field of knowledge representation.

The theory of definitions (how to make the intuitive notion “the views V provide enough information to uniquely determine the answer to Q ” precise?) goes back to Aristoteles, cf., [30] for a historical overview. In 1935, Tarski [25]¹ gave the precise formulation that we still use today. The coincidence is striking enough to recall it here. The setting is as follows. There is a relation symbol Q , a set of relation symbols $V = \{V_1, V_2, \dots\}$ and a further set of relation symbols $R = \{R_1, R_2, \dots\}$. Further is given a finite set Δ of sentences in the signature $Q \cup V \cup R$. The conjunction of all sentences in Δ is represented as $A(Q; V_1; V_2; \dots; R_1; R_2; \dots)$. Theorem 2 in [25] gives the definition of the concept of *semantic or implicit definability*. It says that the relation Q is (semantically or implicitly) definable in terms of the relation symbols V by the theory Δ if the following second order sentence is a validity:

$$\begin{aligned} & \forall Q \forall Q' \forall V_1, V_2, \dots \forall R_1, R_2, \dots \forall R'_1, R'_2, \dots \\ & (A(Q; V_1; V_2; \dots; R_1; R_2; \dots) \wedge A(Q'; V_1; V_2; \dots; R'_1; R'_2; \dots)) \\ & \rightarrow Q \equiv Q'. \end{aligned}$$

We can just as well see this as a first order statement about two identical theories A which happen to be formulated in different signatures. So modern textbooks state: A implicitly defines Q in terms of V iff

$$\begin{aligned} & A(Q; V_1; V_2; \dots; R_1; R_2; \dots), \\ & A(Q'; V_1; V_2; \dots; R'_1; R'_2; \dots) \models Q \equiv Q'. \end{aligned} \quad (1)$$

Tarski [25] also introduces the notion of *completeness for definitions of a logical system L* , in analogy to the then well-known notion of deductive completeness. In modern terms, a logical system L (syntax and semantics) is said to be complete for definitions if whenever Δ implicitly defines the symbol Q in terms of the set of symbols V (all in L), then there exists an L formula ψ constructed from the V -symbols $V_1, V_2, \dots$ such that Q and ψ are equivalent given Δ .²

The Dutch logician Beth showed that first order logic is complete for definitions in 1953 [3]. Craig wrote a review of this paper and gave an alternative proof, based on his interpolation lemma³ (the first of the three uses of interpolation in [7], cf. also [6]). For an extensive historical review, see [30].

Beth was considered a sloppy writer and difficult to read. In fact, the notion of completeness referred to above is often called the *projective Beth property*. Here, *projective* refers to the fact that the definition is in terms of a set of parameters

¹Available in English translation in [26].

²We note that Δ may be *any* set of sentences. In DB-theory it is usually assumed that the views are given by explicit definitions in terms of the primitive relations. Tarski’s notion covers much more.

³Craig’s interpolation lemma says: whenever there are sentences A, B in signatures σ_1 and σ_2 , respectively such that $A \models B$, then there exists a sentence C in the signature $\sigma_1 \cap \sigma_2$ such that $A \models C \models B$. C is called an interpolant.

(in DB terminology: the view predicates). In handbooks as [6] a restricted simpler version of Beth's Theorem is usually presented in which the definition is without parameters, but just in the primitive symbols. The different formulations of Beth's property plus their algebraic counterparts are surveyed in [17].

Both Beth and Craig were concerned with *computing* the explicit definition ψ . They both indicated how, *given a proof of* (1), the definition can be calculated from the proof. [9] contains a modern exposition how to do this based on a tableaux proof (another invention of Beth). Beth presented this method only much later in [4].

Nowadays we speak of Craig interpolation and Beth definability (still with their capitals, though). Both are among the key properties investigated by logicians. Pigozzi [23] started a systematic study of equivalent algebraic formulations of these properties. In general, interpolation is strictly stronger than Beth definability. The guarded and packed fragments of first order logic are one of the few examples of a logical system which has Beth definability but lacks Craig interpolation [18].

A main focus in mathematical logic has been on logical systems being complete for definitions or having interpolation. The research theme as formulated by Segoufin and Vianu goes further because it relates fragments and asks for minimal extensions which are complete for these fragments. We mention a few examples which resemble this theme. First order logic restricted to two variables FO_2 does not have interpolation, nor Beth definability. Dov Gabbay asked whether a bound on the number of variables needed for the interpolant could be given. The answer is no [1]. [27] shows how to repair interpolation for Tarski's algebras of binary relations (cf. e.g., [29]) and for modal logic **K** expanded with the difference operator. For both systems, the answer can be seen as a characterization of first order logic from below: FO is the smallest expansion of these systems which has interpolation. Whether the same holds for Beth definability is much harder to prove and currently open.

In description logic, completeness for definitions is intimately related to the question whether cyclic theories (TBoxes) can be equivalently rewritten (in the same language) as acyclic theories [2, 28]. The latter are of interest because they have better computational properties. Because description logics and modal logics are often specialized fragments of FO, such results fall within the general research theme of Segoufin and Vianu.

Even if a query is not determined by a set of views, we might still be interested in finding a maximally contained rewriting in terms of the views. This relates to the notion of uniform interpolation [31]. The uniform pre-interpolant of a query and a set of view definitions exactly is the strongest query expressed in the view symbols which is contained in the original query. Examples of logics in which uniform interpolants always exist are modal logic **K**, the modal μ -calculus and second order logic. Uniform interpolation is used in [28] to obtain acyclic rewritings of cyclic TBoxes.

3. GUARDED AND PACKED FRAGMENTS

We introduce the guarded and packed fragments of first order logic, recall some established connections with database theory, and we list their main logical properties.

The guarded fragment was defined to explain and generalize the good logical properties of modal formalisms [1]. In

particular, the validity problem for the guarded fragment is decidable in 2EXPTIME [13] (even EXPTIME when restricted to a fixed signature or a fixed number of variables). The key intuition behind the decidability is the restricted quantification of the guarded fragment. Only quantified formulas of the form $\forall \bar{x}(G(\bar{x}, \bar{y}) \rightarrow \phi(\bar{x}, \bar{y}))$ are allowed, where G is an atomic relation symbol and $\phi(\bar{x}, \bar{y})$ is guarded as well.

The important restriction here is that *all* free variables $\bar{x}, \bar{y}$ occurring in the subformula $\phi(\bar{x}, \bar{y})$ *must* also occur in the guard $G(\bar{x}, \bar{y})$. For example, the following tuple generating dependency is guarded:

$$\forall xy(\text{employee}(x, y) \rightarrow \exists s \text{ salary}(x, y, s)),$$

while this key constraint is not guarded:

$$\begin{aligned} \forall xn(\text{soc_security_nr}(x, n) \rightarrow \\ \forall y(\text{soc_security_nr}(y, n) \rightarrow x = y)), \end{aligned}$$

because the quantified subformula is of the form $\forall y(G(y, n) \rightarrow \phi(x, y))$ and x, y do not occur together in a guard. Rewriting the key constraint does not help either. In fact we have a strong limit on what can be added to the guarded fragment. Useful ingredients of an undecidability proof like a transitive relation $\neg \forall xy(Rxy \rightarrow \forall z(Ryz \rightarrow Rxz))$ —or a partial function $\neg \forall xy(Rxy \rightarrow \forall z(Rxz \rightarrow y = z))$ —cannot be expressed in the guarded fragment, and indeed adding them would make the fragment undecidable [13].

Formally the guarded fragment of first order logic is defined as follows: it is the smallest set containing every first order atom over some relational signature σ , it is closed under the boolean connectives, and under the following rule for quantified formulas: if $\phi(\bar{x}, \bar{y})$ is a guarded formula, then also $\exists \bar{x}(G \wedge \phi(\bar{x}, \bar{y}))$ and $\forall \bar{x}(G \rightarrow \phi(\bar{x}, \bar{y}))$, provided that G is an atomic formula in which all variables $\bar{x}, \bar{y}$ are free.

From a database perspective, the guarded fragment coincides with the semijoin algebra, the result of replacing the product operator by the semijoin operator in Codd's relational algebra [20]. The idea of guarding corresponds to restricting the tuples that can be generated by queries to tuples whose elements belong together to a tuple in some atomic relation. A semijoin $G \ltimes_{\theta} \phi$ returns always a subset of G . A natural join on the other hand may create new tuples, as in e.g., a path query like $\exists z(Rxz \wedge Rzy)$.

The guarded fragment is characterized in terms of non-recursive stratified Datalog programs with acyclic rules in [10]; [12] characterizes the guarded conjunctive queries as the acyclic conjunctive queries.

EXAMPLE 3.1. Guarded views consist always of subtuples of a relation in the database. For instance, let $\forall xy(Vxy \leftrightarrow \exists uv(Gxyu \wedge Ruv))$ define the view V . This is syntactically not guarded, but equivalent to the guarded formula

$$\begin{aligned} \forall xy(Vxy \rightarrow \exists u(Gxyu \wedge \exists v Ruv)) \wedge \\ \forall xyu(Gxyu \rightarrow ((\exists v Ruv) \rightarrow Vxy)). \end{aligned}$$

For example, we *can* define the view **italian-person(name)** from the tables **person(name, id)** and **affiliation(id, country)** in the guarded fragment, but we *cannot* define the binary relation **habitat(name, country)** from these two tables.

3.1 Packed fragment

A useful generalization of the guarded fragment allows “safe products” as guards. A safe product is a conjunction $Q(x_1, \dots, x_n)$ of formulas of the form $\exists \bar{y} \text{Atom}(\bar{x}\bar{y})$, which has for all free variables x_i, x_j ($i \neq j$) an atom containing x_i and x_j . If we draw the hypergraph of $Q(x_1, \dots, x_n)$ then the set $x_1, \dots, x_n$ is a so-called *packed* set: every pair x_i, x_j belongs to a hyperedge. In case all relations are binary, the set $x_1, \dots, x_n$ is called a clique. This fragment is known as the loosely guarded, the clique guarded or the packed fragment.

The validity problem of the packed fragment has the same 2EXPTIME complexity as the guarded fragment, which also drops to EXPTIME for fixed finite variable fragments. It does not drop to EXPTIME for fixed signatures because in the packed fragment we can create arbitrary large guards with a single binary relation.

EXAMPLE 3.2. The query

$\exists \text{id}(\text{person}(\text{name}, \text{id}) \wedge \text{affiliation}(\text{id}, \text{country}))$

is no safe product because there is no atom connecting the variables **name** and **country**. But if we have the **habitat** relation as primitive, we can define a large table, for instance:

$\text{View}(\text{name}, \text{id}, \text{country}) \leftrightarrow$
 $\text{person}(\text{name}, \text{id}) \wedge \text{affiliation}(\text{id}, \text{country}) \wedge$
 $\text{habitat}(\text{name}, \text{country}).$

Note that in the hypergraph of this formula the three free variables indeed form a clique.

We now define the packed fragment.

DEFINITION 3.3. Consider a signature σ containing only relation symbols (of arbitrary arity).

- (i) A *guard* is a conjunction of formulas of the form $\exists \bar{y} \text{Atom}(\bar{x}\bar{y})$.
- (ii) A guard G *packs* the variables x_i, x_j if G contains an atom in which x_i and x_j both occur free.
- (iii) The packed fragment is the smallest set containing every first order atom over the signature σ , including equality; it is closed under the boolean connectives, and under the following rule for quantified formulas: if $\phi(\bar{x}, \bar{y})$ is a packed formula, then also $\exists \bar{x}(G \wedge \phi(\bar{x}, \bar{y}))$, provided that G is a guard which packs all different variables $x_i \in \bar{x}, y_i \in \bar{x} \cup \bar{y}$.

We will use $\forall \bar{x}(G \rightarrow \phi(\bar{x}, \bar{y}))$ as an abbreviation of $\neg \exists \bar{x}(G \wedge \neg \phi(\bar{x}, \bar{y}))$.

The formulas expressing transitivity and being a partial function are not in the packed fragment. The translation of the *until* connective from temporal logic is an example of a formula which is in the packed, but not in the guarded fragment:

$$\exists y(Rxy \wedge Py \wedge \forall z((Rxz \wedge Rzy) \rightarrow Qz)).$$

Note that $x = x$ may occur as a guard, for instance as in $\forall x(x = x \rightarrow \neg Px)$. Thus, as long as $\phi(x)$ is packed, $\forall x\phi(x)$ is packed as well.

The definition of the packed fragment can be extended to include constants without losing the good properties. We abstain from this for simplicity.

The main positive properties of the packed fragment are

1. Every satisfiable formula is satisfiable on a finite model [16].

2. The validity problem is 2EXPTIME complete, and EXPTIME complete when considering fixed finite variable fragments [13].
3. Every satisfiable formula with k variables is satisfiable on a model whose Gaifman graph has tree width at most $k + 1$ and is of bounded degree [13].
4. The packed fragment has a natural characterization as the fragment of first-order logic which is invariant for packed Ehrenfeucht–Fraïssé pebble games⁴ [1, 14, 21].

REMARK 3.4. As indicated in [20], there is an important difference between the semijoin algebra and the guarded fragment, which also plays a role in the packed fragment. Every quantifier-free first order formula is guarded (and packed) by definition, so arbitrary Cartesian products can be expressed in it, like $\{(x, y) \mid Sx \wedge Sy\}$. The tuples in this relation are not packed, and this relation cannot be expressed by a semijoin algebra expression. We want to restrict attention to those packed formulas in free variables whose output tuples are safe, i.e., are packed in every model. For that we use the notion of self-packed formulas from [19].

DEFINITION 3.5. A packed formula $\phi(\bar{x})$ is called *self-packed* if ϕ is of the form $(G \wedge \psi)$ with G a guard which packs the variables $\bar{x}$.

Self-packed formulas play an important role in view definitions.

PROPOSITION 3.6. Every sentence $\forall \bar{x}(V(\bar{x} \leftrightarrow \phi(\bar{x})))$ with V an atom and $\phi(\bar{x})$ a self-packed packed formula is equivalent to a packed sentence.

4. DETERMINACY IMPLIES REWRITING IN THE PACKED FRAGMENT

We will show that the packed fragment (*PF*) is complete for *PF*-to-*PF* rewritings and that the same holds when restricted to unions of conjunctive queries. What does this exactly mean?

DEFINITION 4.1 ([24]). Let L denote some fragment of first order logic. Let $\mathbf{Q}$ be a predicate symbol and $\mathbf{V} = \{V_1, \dots, V_n\}$ a set of predicate symbols ($\mathbf{Q}$ and the V_i may be of arbitrary arity). Let Σ be a set of sentences of the form $\forall \bar{x}(\mathbf{Q}(\bar{x}) \leftrightarrow \phi(\bar{x}))$ and $\forall \bar{x}(V(\bar{x}) \leftrightarrow \phi(\bar{x}))$, for each $V \in \mathbf{V}$, with $\phi \in L$.

We say that L is complete for L -to- L rewritings if whenever $\mathbf{Q}$ is determined by $\mathbf{V}$, there exists an L formula $\psi(\bar{x})$ constructed only from the view symbols $\mathbf{V}$ which is equivalent to $\mathbf{Q}$, i.e., $\Sigma \models \forall \bar{x}(\mathbf{Q}(\bar{x}) \leftrightarrow \psi(\bar{x}))$.

Finite and infinite models. In [24] the distinction is made between completeness with respect to possibly infinite and with respect to just finite models. By Beth’s Theorem, FO is complete for FO-to-FO rewritings for infinite models, but this is not the case for finite models: any complete language must be able to express all computable queries [24]. For

⁴Intuitively, this is an EF pebble game in which the universal player can move as many pebbles as he likes, provided that he always places them on a packed set. Definition 4.3 makes this notion precise.

fragments of languages which have the finite model property, this distinction vanishes however. A language L is said to have the finite model property if every satisfiable L -formula is also satisfiable on a finite model.

LEMMA 4.2. *Let L have the finite model property. Then L is complete for L -to- L rewritings on infinite models if and only if L is complete for L -to- L rewritings on finite models.*

PROOF. From left to right is immediate. The other direction follows by spelling out the definitions and applying the finite model property. $\square$

Because the packed fragment has the finite model property, by Lemma 4.2, we need not make the distinction between finite and infinite models, and we won't.

Before we can prove our results, we need some terminology and two technical lemmas. The first is a generalization of a result in modal logic which says that every two pointed Kripke models $(\mathcal{M}, m)$ and $(\mathcal{N}, n)$ with the same modal theory are pairwise bisimilar with models $(\mathcal{M}^+, m)$ and $(\mathcal{N}^+, n)$ which are elementary equivalent [1]. The second generalizes a similar lemma in [19].

4.1 Technical preliminaries

Let Z be a finite subset of a model $\mathcal{M}$. Z is called *live* in $\mathcal{M}$ if Z is either a singleton, or there exists a relation R and a set X such that $Z \subseteq X$ and the elements of X are R -related, in any order or multiplicity. E.g., if $\langle m_2, m_1, m_1, m_3 \rangle \in I^{\mathcal{M}}(R)$, then $\{m_2, m_3\}$ is live in $\mathcal{M}$. In this case we say that $\{m_2, m_3\}$ is *R -live* (in $\mathcal{M}$). We say that Z is *σ -live* if Z is R -live in $\mathcal{M}$, for some $R \in \sigma$. Z is called *σ -packed* in $\mathcal{M}$ if every pair $m, m' \in Z$ is σ -live in $\mathcal{M}$.

We use PF to denote the packed fragment and $(U)PCQ$ to denote the (unions of) packed $\exists\text{FO}_{\wedge+}$ formulas. The latter are all FO formulas generated from relation symbols and equality using only conjunction and existential quantification. We will call these conjunctive queries, though due to the syntax restrictions we cannot always write them in prenex notation⁵.

By a finite *partial σ -isomorphism* f we mean a finite one-to-one partial map between two models $\mathcal{M}, \mathcal{N}$ which preserves σ -relations in both ways. By the *image* of a map $f : X \rightarrow Y$ we understand the set $\{f(x) : x \in X\}$, and we refer to X as the *domain* of f .

DEFINITION 4.3. A non-empty set F of finite partial σ -isomorphisms between two models $\mathcal{M}$ and $\mathcal{N}$ is called a *packed σ -bisimulation* between $\mathcal{M}$ and $\mathcal{N}$ if for any $f \in F$ the following holds:

1. For any σ -packed set $Z \subseteq \mathcal{M}$ there is a $g \in F$ with domain Z such that g and f agree on the intersection of their domains. (The *zig*-condition)
2. For any σ -packed $W \subseteq \mathcal{N}$ there is a $g \in F$ with image W such that g^{-1} and f^{-1} agree on the intersection of their domains. (The *zag*-condition)

As we only use packed bisimulations in this paper, we just call them bisimulations. We say that models $(\mathcal{M}, \bar{m})$ and

⁵For instance, the packed conjunctive query $Q(x, y) \equiv \exists z(Rxyz \wedge \exists w(Syu \wedge Suv \wedge Svy))$.

$(\mathcal{N}, \bar{n})$ bisimulate if there exists an $f \in F$ which sends m_i to n_i for all m_i, n_i in $\bar{m}$ and $\bar{n}$, respectively.

Packed bisimulations are defined in such a way as to preserve packed formulas. That is, for a packed σ -formula $\varphi(v_1, \dots, v_k)$, a packed σ -bisimulation F between models $\mathcal{M}, \mathcal{N}$, $f \in F$, and $m_1, \dots, m_k \in \text{dom}(f)$ it is the case that $\mathcal{M} \models \varphi[m_1, \dots, m_k] \Leftrightarrow \mathcal{N} \models \varphi[f(m_1), \dots, f(m_k)]$.

For the rest of the section, we fix some arbitrary relational signature σ . σ is the signature of our models. Besides the symbols in σ , we have an extra predicate symbol $\mathbf{Q}$ and a set of view predicate symbols $\mathbf{V} = \{V_1, V_2, \dots\}$. The models we consider are σ models expanded with the $\mathbf{Q}$ and V_i predicates. With the extra predicate symbols we can reason easily about σ -queries and σ -views.

LEMMA 4.4. *Let $(\mathcal{M}, \bar{m})$ and $(\mathcal{N}, \bar{n})$ be countable $\sigma \cup \{\mathbf{Q}\} \cup \mathbf{V}$ models such that $\bar{m}$ and $\bar{n}$ are $\mathbf{V}$ -packed and for all $PCQ \phi$ in the $\mathbf{V}$ -signature: $\mathcal{M} \models \phi(\bar{m}) \Leftrightarrow \mathcal{N} \models \phi(\bar{n})$. Then there exists $\sigma \cup \{\mathbf{Q}\} \cup \mathbf{V}$ models $(\mathcal{M}', \bar{m}')$ and $(\mathcal{N}', \bar{n}')$ such that*

1. $\text{dom}(\mathcal{M}') = \text{dom}(\mathcal{N}')$.
2. $I^{\mathcal{M}'}(V) = I^{\mathcal{N}'}(V)$, for all V in $\mathbf{V}$.
3. There are packed $\sigma \cup \{\mathbf{Q}\} \cup \mathbf{V}$ bisimulations between $(\mathcal{M}, \bar{m})$ and $(\mathcal{M}', \bar{m}')$, and $(\mathcal{N}, \bar{n})$ and $(\mathcal{N}', \bar{n}')$, respectively.

PROOF. We give a sketch. The proof first follows the proof technique used in [1] (Lemma 2.7.1), but adapted to the packed fragment. First, we unravel the two models into tree models by using the step-by-step construction in [13]. Then we use the multiplication technique whereby we copy each node in the two trees except the roots countably many times (cf. also [28]). The resulting countable models $(\mathcal{M}^+, \bar{m})$ and $(\mathcal{N}^+, \bar{n})$ can be shown to be elementarily equivalent in the $\mathbf{V}$ -signature, while $\sigma \cup \{\mathbf{Q}\} \cup \mathbf{V}$ -bisimulating with $(\mathcal{M}, \bar{m})$ and $(\mathcal{N}, \bar{n})$ respectively.

From these, the required models $\mathcal{M}', \mathcal{N}'$ can be obtained by the following result: for any two countable σ -elementarily equivalent models $\mathcal{M}^+, \mathcal{N}^+$, there exist countable models $\mathcal{M}', \mathcal{N}'$ which are σ -isomorphic in which $\mathcal{M}^+, \mathcal{N}^+$ can be elementarily embedded, respectively [8]. $\square$

DEFINITION 4.5. Let Σ be a set of packed sentences in the signature $\sigma \cup \{\mathbf{Q}\} \cup \mathbf{V}$. We say that Σ guarantees that $\mathbf{Q}$ is determined by $\mathbf{V}$ if for all $\sigma \cup \{\mathbf{Q}\} \cup \mathbf{V}$ models $\mathcal{M}, \mathcal{N}$ over the same domain D which satisfy Σ , if $I^{\mathcal{M}}(V) = I^{\mathcal{N}}(V)$, for all V in $\mathbf{V}$, then also $I^{\mathcal{M}}(\mathbf{Q}) = I^{\mathcal{N}}(\mathbf{Q})$.

LEMMA 4.6. *Let Σ be a set of packed sentences which guarantee that $\mathbf{Q}$ is determined by $\mathbf{V}$ and let $\mathcal{M}$ be a $\sigma \cup \{\mathbf{Q}\} \cup \mathbf{V}$ model of Σ . Then each $\mathbf{Q}$ -packed set $Y \subseteq \mathcal{M}$ is also $\mathbf{V}$ -packed.*

PROOF. Let $\mathcal{M}$ be a model of Σ . Suppose $Y_0 \subseteq \mathcal{M}$ is $\mathbf{Q}$ -packed but not $\mathbf{V}$ -packed. Thus there are $y, y' \in Y_0$ which are $\mathbf{Q}$ -live but not $\mathbf{V}$ -live. We derive a contradiction.

Let $\mathcal{M} \times 2$ denote the model with domain $M \times \{0, 1\}$ and the interpretation of the relations defined as follows. For every l -ary P and every $s = \langle s_1, \dots, s_l \rangle \in (M \times \{0, 1\})^l$, $s \in I^{\mathcal{M} \times 2}(P)$ iff

1. $\langle \pi_1(s_1), \dots, \pi_l(s_l) \rangle \in I^{\mathcal{M}}(P)$, and

2. if $\pi_1(s_i) = \pi_1(s_j)$ then also $\pi_2(s_i) = \pi_2(s_j)$.

As the reader can verify,

$$F_1 \stackrel{\text{def}}{=} \{\pi_1 : X \longrightarrow Y : X \text{ is packed in } \mathcal{M} \times \underline{2}\}$$

is bisimulation between $\mathcal{M} \times \underline{2}$ and $\mathcal{M}$, preserving $\mathbf{Q}$, $\mathbf{V}$ and all σ relations. (The second condition guarantees that every projection in F_1 is 1-1). Since $\mathcal{M} \models \Sigma$, we conclude that $\mathcal{M} \times \underline{2} \models \Sigma$.

Our aim is to modify the interpretation of $\mathbf{Q}$ on $\mathcal{M} \times \underline{2}$, while leaving the views the same. That yields our desired contradiction. Let $X_0 \stackrel{\text{def}}{=} \{\langle y, 0 \rangle, \langle y', 0 \rangle\}$. Thus X_0 is $\mathbf{Q}$ -live in $\mathcal{M} \times \underline{2}$, by the existence of the bisimulation. Let $(\mathcal{M} \times \underline{2})'$ be the model which differs from $\mathcal{M} \times \underline{2}$ only in that X_0 is *not* live at all. Thus, for every tuple $\bar{s}$ containing X_0 , $(\mathcal{M} \times \underline{2})' \not\models \mathbf{Q}(\bar{s})$, and $(\mathcal{M} \times \underline{2})' \not\models R(\bar{s})$, for any $R \in \sigma$. (By assumption, no tuple $\bar{s}$ containing y and y' is $\mathbf{V}$ -live).

Claim. $F_1' \stackrel{\text{def}}{=} \{\pi_1 : X \longrightarrow Y : X \text{ is packed}\}$ is a bisimulation between $(\mathcal{M} \times \underline{2})'$ and $\mathcal{M}$.

Proof of claim. By construction, X_0 is not live in $(\mathcal{M} \times \underline{2})'$. Thus X_0 is not a subset of any packed set X in $(\mathcal{M} \times \underline{2})'$. But for each such set X , $\mathcal{M} \times \underline{2}_{\upharpoonright X} = (\mathcal{M} \times \underline{2})'_{\upharpoonright X}$.

This shows that each $\pi_1 \in F_1'$ is a partial isomorphism. The zig-condition holds by construction. For the zag-condition, let $\pi_1 : X \longrightarrow Y$ in F_1' , and W be a packed set in $\mathcal{M}$. Define $Z \stackrel{\text{def}}{=} \pi_1^{-1}[Y \cap W] \cup \{\langle m, 1 \rangle : m \in W \setminus Y\}$. Then Z is a packed set in $\mathcal{M} \times \underline{2}$. Now $X_0 \not\subseteq X$, thus $X_0 \not\subseteq Z \cap X$. By construction, $X_0 \cap Z \setminus X = \emptyset$. Whence $X_0 \not\subseteq Z$, and $(\mathcal{M} \times \underline{2})'_{\upharpoonright Z} = \mathcal{M} \times \underline{2}_{\upharpoonright Z}$. Thus Z is a packed set in $(\mathcal{M} \times \underline{2})'$, and we are done.

End of proof of claim.

Now we have our desired pair of models: $(\mathcal{M} \times \underline{2})'$ and $\mathcal{M} \times \underline{2}$ have the same domain. By the claim $(\mathcal{M} \times \underline{2})' \models \Sigma$. And by construction they do agree on $\mathbf{V}$ but do *not* agree on $\mathbf{Q}$. $\square$

4.2 Query rewritings in the packed fragment

We are now ready to prove our main results.

First we show Beth's property for the packed fragment. As a corollary we obtain that *PF* is complete for *PF*-to-*PF* rewritings. Then we show that the same holds for (unions of) packed conjunctive queries.

THEOREM 4.7. *Let Σ be a set of packed *PF* sentences which guarantees that $\mathbf{Q}$ is determined by $\mathbf{V}$. Then there exists some $\varphi \in \text{PF}$ in the $\mathbf{V}$ signature which is equivalent to $\mathbf{Q}$ on every Σ model.*

PROOF. Let Σ be as in the theorem. Define $\Theta(\mathbf{V})$ as

$$\{\phi(\bar{x}) \in \text{PF} \text{ and in the } \mathbf{V} \text{ signature} \mid \Sigma \models \mathbf{Q}(\bar{x}) \rightarrow \phi(\bar{x})\}.$$

If $\Theta(\mathbf{V})$ is inconsistent, $\perp$ is equivalent to $\mathbf{Q}$. Otherwise, let $(\mathcal{M}, \bar{m}) \models \Theta(\mathbf{V})$, for $(\mathcal{M}, \bar{m})$ an arbitrary $\sigma \cup \{\mathbf{Q}\} \cup \mathbf{V}$ model which satisfies Σ . We want to show that $\mathcal{M} \models \mathbf{Q}(\bar{m})$, because then, by compactness, we have found the $\mathbf{V}$ -rewriting of $\mathbf{Q}$. (Recall that by Lemma 4.2, we may just as well reason on infinite models.)

Let $\Theta(\bar{m}) = \{\phi(\bar{x}) \in \text{PF} \text{ and in the } \mathbf{V} \text{ signature} \mid \mathcal{M} \models \phi(\bar{m})\}$. We claim that $\Sigma \cup \Theta(\bar{m}) \cup \{\mathbf{Q}(\bar{x})\}$ is consistent. If not, then by compactness, $\Sigma \models \mathbf{Q}(\bar{x}) \rightarrow \neg\theta(\bar{x})$, for some $\theta(\bar{x}) \in \Theta(\bar{m})$. But then both $\theta(\bar{x})$ and its negation belong to $\Theta(\bar{m})$, which is absurd.

Now, let $(\mathcal{N}, \bar{n})$ be a Σ model which satisfies $\Theta(\bar{m}) \cup \{\mathbf{Q}(\bar{x})\}$ at $\bar{n}$. By Lemma 4.6, $\bar{n}$ is $\mathbf{V}$ -packed. As $(\mathcal{M}, \bar{m})$ and $(\mathcal{N}, \bar{n})$ agree on all $\mathbf{V}$ -formulas, also $\bar{m}$ is $\mathbf{V}$ -packed. The models $(\mathcal{M}, \bar{m})$ and $(\mathcal{N}, \bar{n})$ are constructed in such a way that Lemma 4.4 applies to them. But then there exist two models $(\mathcal{M}', \bar{m}')$ and $(\mathcal{N}', \bar{n}')$ over the same domain which agree on the views, and which bisimulate with $(\mathcal{M}, \bar{m})$ and $(\mathcal{N}, \bar{n})$, respectively. Thus $\mathcal{M}$ and $\mathcal{N}$ both satisfy Σ , and $\mathcal{N}' \models \mathbf{Q}(\bar{m}')$. Because $\mathbf{Q}$ is determined by the views $\mathbf{V}$, also $\mathcal{M}' \models \mathbf{Q}(\bar{m}')$. But then, by the bisimulation between $(\mathcal{M}', \bar{m}')$ and $(\mathcal{M}, \bar{m})$, $\mathcal{M} \models \mathbf{Q}(\bar{m})$, which was to be shown. $\square$

COROLLARY 4.8. *PF is complete for PF-to-PF rewritings.*

THEOREM 4.9. (i) *PCQ is complete for PCQ-to-PCQ rewritings.*

(ii) *UPCQ is complete for UPCQ-to-UPCQ rewritings.*

PROOF. We start with (i). We copy the proof of Theorem 4.7. Define

$$\Theta(\mathbf{V}) = \{\phi(\bar{x}) \mid \Sigma \models \mathbf{Q}(\bar{x}) \rightarrow \phi(\bar{x})\},$$

as in the proof of Theorem 4.7 but now $\phi(\bar{x})$ must be a PCQ in the $\mathbf{V}$ -signature. Let $(\mathcal{M}, \bar{x})$ be a *free model*: that is $\Theta(\mathbf{V}) = \{\phi(\bar{x}) \in \text{PCQ}(\mathbf{V}) \mid \mathcal{M} \models \phi(\bar{x})\}$. We may obtain $(\mathcal{M}, \bar{x})$ by chasing $\Theta(\mathbf{V})$ together with the view definitions. By Lemma 4.6 $\bar{x}$ must be $\mathbf{V}$ -packed. Now let $(\mathcal{N}, \bar{x})$ be the *frozen body* [22] of $\mathbf{Q}(\bar{x})$. That is, $(\mathcal{N}, \bar{x})$ is the model obtained by chasing $\mathbf{Q}(\bar{x})$. Minimally expand $(\mathcal{N}, \bar{x})$ with the view predicates according to the definitions in Σ . Then, because $(\mathcal{N}, \bar{x})$ is a free model, we have that in the expansion $\bar{x} \models \phi(\bar{x})$ iff $\Sigma \models \mathbf{Q}(\bar{x}) \rightarrow \phi(\bar{x})$. Thus by the definition of $\Theta(\mathbf{V})$,

$$\begin{aligned} \{\phi(\bar{x}) \in \text{PCQ}(\mathbf{V}) \mid \mathcal{M} \models \phi(\bar{x})\} &= \\ \Theta(\mathbf{V}) &= \\ \{\phi(\bar{x}) \in \text{PCQ}(\mathbf{V}) \mid \mathcal{N} \models \phi(\bar{x})\}. \end{aligned}$$

Thus $\bar{x}$ is $\mathbf{V}$ -packed in both models. Whence we can apply Lemma 4.4 again, and the proof continues as before.

(ii) follows from (i) by compactness. Let $\mathbf{Q} = \bigcup_i \mathbf{Q}_i$. For each $\mathbf{Q}_i$, define $\Theta_i(\mathbf{V})$ as in part (i). Continuing as in part (i), we show that $\Theta_i(\mathbf{V}) \models \mathbf{Q}_i$. Thus by compactness, some $\theta_i \in \Theta_i(\mathbf{V})$ is equivalent to $\mathbf{Q}_i$. But then $\mathbf{Q}$ is equivalent to $\bigcup_i \theta_i$. $\square$

5. ALGORITHMS AND COMPLEXITY

The determinacy and validity problems reduce to each other, as explained in Section 2. For full FO, determinacy is undecidable, both for unrestricted and finite databases, which follows from the undecidability of the validity problem. For conjunctive queries, the problem is open⁶, both for the unrestricted and the finite case. For unions of conjunctive queries it is undecidable to know whether $\mathbf{V}$ determines $\mathbf{Q}$ [24].

For the packed fragment, the story is different. First of all, there is no distinction between the finite and the unrestricted case because of the finite model property. Decidability of determinacy follows immediately from decidability

⁶Note that [24] claims it is decidable for conjunctive queries in the unrestricted case, but the mistake in the proof is indicated in [22].

of validity of the packed fragment [13]. Complexity bounds are also easy to obtain:

THEOREM 5.1. *For $\mathbf{Q}$ and $\mathbf{V}$ in the packed fragment, the problem whether $\mathbf{Q}$ is determined by $\mathbf{V}$ is 2EXPTIME complete.*

PROOF. This follows immediately from the upper and lower bounds given in [13]. $\square$

Now we consider the complexity of obtaining the rewriting. We have to be careful what we consider known and what is part of the input. To explain this, consider FO-to-FO rewritings in the unrestricted case. If we *know* that $\mathbf{Q}$ is determined by $\mathbf{V}$ then there exists an algorithm that produces the rewriting: by the completeness theorem of FO tableaux there must exist a tableaux proof showing that $\mathbf{Q}$ is determined by $\mathbf{V}$. The algorithm searches through an enumeration of all proofs until it finds the one. From the proof we can calculate the rewriting in linear time by Craig’s algorithm (cf. [9]).

Thus, if the tableaux proof T is part of the input, rewriting is in linear time. Unfortunately, there is (by the undecidability of FO validity) no bound on T given $\mathbf{Q}$ and $\mathbf{V}$. The situation is even worse, in FO, not only is there no recursive bound on the length of the smallest explicit definition, there is not even a recursive bound on the minimal number of quantifier alternations in the explicit definition [11].

For the full packed fragment, the complexity of finding a rewriting given that we only know that $\mathbf{Q}$ determines $\mathbf{V}$ remains open. We conjecture that this can be done in 2EXPTIME, using an algorithm which computes the interpolant from a pseudo-model (called a “saturated set of mosaics” in [21]). Pseudo-models resemble tableaux expansions, but they remain bounded because “packed parts of a branch” (a mosaic) can be reused in other parts. This would yield a 2EXPTIME upper bound on the size of the rewriting. There exists an example in the description logic $\mathcal{ALC}$ which requires an exponential size definition [28].

For the conjunctive queries in the packed fragment, the story is slightly better, thanks to the completeness for rewriting and the result of [22] Proposition 1.(iv). If $\mathbf{Q}$ is determined by $\mathbf{V}$, it has a PCQ rewriting by Theorem 4.9. Then by [22] Proposition 1.(iv), $\mathbf{Q}_\mathbf{V}$ is a rewriting. $\mathbf{Q}_\mathbf{V}$ is the conjunctive query in the $\mathbf{V}$ signature obtained from adding the view predicates to the frozen body of $\mathbf{Q}$. $\mathbf{Q}_\mathbf{V}$ can be obtained from the frozen body of $\mathbf{Q}$ by chasing it with the view definitions. The size of the query $\mathbf{Q}_\mathbf{V}$ is linear in the number of variables in $\mathbf{Q}$ and exponential in the maximal arity of the view predicates. This is optimal, as can be seen by the following example:

$$\begin{aligned} Q(x, y) &\equiv Rxx \wedge Ryy \wedge Rxy \wedge Ryx \\ V(x_1, \dots, x_k) &\equiv \bigwedge_{1 \leq i, j \leq k} R x_i x_j. \end{aligned}$$

The view definition is quadratic in k . The domain of the frozen body of Q is $\{x, y\}$. And $Q_\mathbf{V}$ contains 2^k V -tuples. Of course in this example there exists a much smaller rewriting of Q in terms of V . Whether this is always the case for packed conjunctive queries remains open.

6. CONCLUSIONS

We have put the research theme of Segoufin and Vianu into an historical context, and gave a useful class of queries and views for which positive results could be obtained. Looking at the counterexample given in [22] it seems that the packed fragment is maximal in its good properties. [22] provide two other well-behaved classes of CQ queries and views, but they are orthogonal to the packed fragment.

The packed fragment was an obvious choice to try because (1) it has the finite model property, so there is no distinction between the (easier) unrestricted and the (hard) finite case, (2) it was known that a restricted form of Beth definability holds for it.

Interesting open problems related to the present paper mainly concern finding good rewritings. We have succinctness questions: How large are the rewritings? There are questions of implementing algorithms and their complexity: the algorithm using tableaux generates a rewriting, but it is highly dependent on the actual tableaux proof, and often contains a lot of “tautological junk”. Can we easily remove that? Is there some canonical tableaux proof which we can try? [22] presents an algorithm for computing the rewriting if it exists, based on a certain strategy of applying the tableaux expansion rules. It is interesting to see whether their strategy yields the same rewriting as the algorithm described in [9].

Acknowledgments

Thanks are due to Balder ten Cate, Luc Segoufin and the PODS reviewers for extensive and helpful feedback.

7. REFERENCES

- [1] H. Andréka, J. van Benthem, and I. Németi. Modal languages and bounded fragments of predicate logic. *Journal of Philosophical Logic*, 27(3):217–274, 1998.
- [2] F. Baader and W. Nutt. Basic description logics. In *The description logic handbook: theory, implementation, and applications*, pages 43–95. Cambridge University Press, New York, NY, USA, 2003.
- [3] E. Beth. On Padoa’s method in the theory of definition. *Indagationes Mathematicae*, 15:330–339, 1953.
- [4] E. Beth. *Formal methods, An introduction to symbolic logic and to the study of effective operations in arithmetic and logic*. Number 4 in Synthese Library. Reidel, Dordrecht, 1962.
- [5] D. Calvanese, G. De Giacomo, M. Lenzerini, and M. Vardi. Lossless regular views. In *Proceedings PODS*, pages 247–258, 2002.
- [6] C. Chang and H. Keisler. *Model Theory*. North-Holland, 1973.
- [7] W. Craig. Three uses of the Herbrand–Gentzen theorem in relation to model theory and proof theory. *Journal of Symbolic Logic*, 22:269–285, 1957.
- [8] K. Doets. *Basic Model Theory*. CSLI Publications, Stanford, 1996.
- [9] M. Fitting. *First Order Logic and Automated Theorem Proving (second edition)*. Springer Verlag, 1996.
- [10] J. Flum, M. Frick, and M. Grohe. Query evaluation via tree-decompositions. In *Proceedings ICDT*, pages 22–38, 2001.

- [11] H. Friedman. The complexity of explicit definitions. *Advances in Mathematics*, 20(1):18–29, 1976.
- [12] G. Gottlob, N. Leone, and F. Scarcello. Robbers, marshals, and guards: game theoretic and logical characterizations of hypertree width. *J. Comput. Syst. Sci.*, 66(4):775–808, 2003.
- [13] E. Grädel. On the restraining power of guards. *Journal of Symbolic Logic*, 64(4):1719–1742, 1999.
- [14] E. Grädel. Guarded fixed point logics and the monadic theory of countable trees. *Theor. Comput. Sci.*, 288(1):129–152, 2002.
- [15] S. Grumbach and L. Tininini. On the content of materialized aggregate views. In *Proceedings PODS*, pages 47–57, 2000.
- [16] I. Hodkinson. Loosely guarded fragment of first-order logic has the finite model property. *Studia Logica*, 70(2):205–240, 2002.
- [17] E. Hoogland. Algebraic characterizations of various beth definability properties. *Studia Logica*, 65(1):91–112, 2000.
- [18] E. Hoogland and M. Marx. Interpolation in the guarded fragment. *Studia Logica*, 70(3):273–409, 2002.
- [19] E. Hoogland, M. Marx, and M. Otto. Beth definability for the guarded fragment. In H. Ganzinger, D. McAllester, and A. Voronkov, editors, *Logic for Programming and Automated Reasoning, 6th International Conference LPAR99, Tbilisi, Georgia*, volume 1705 of *LNAI*, pages 273–285. Springer, 1999.
- [20] D. Leinders, M. Marx, J. Tysiewicz, and J. van den Bussche. The semijoin algebra and the guarded fragment. *Journal of Logic, Language and Information*, 14:331–343, 2005.
- [21] M. Marx and Y. Venema. Local variations on a loose theme: modal logic and decidability. In E. Grädel, P. Kolaitis, L. Libkin, M. Marx, J. Spencer, M. Vardi, Y. Venema, and S. Weinstein, editors, *Finite Model Theory and its Applications*. Springer-Verlag, 2007.
- [22] A. Nash, L. Segoufin, and V. Vianu. Determinacy and rewriting of conjunctive queries using views: a progress report. In *Proceedings ICDT 2007*, to appear.
- [23] D. Pigozzi. Amalgamation, congruence extension and interpolation properties in algebras. *Algebra Universalis*, 1:269–349, 1972.
- [24] L. Segoufin and V. Vianu. Views and queries: determinacy and rewriting. In *Proceedings PODS*, pages 49–60, 2005.
- [25] A. Tarski. Einige methodologische Untersuchungen über die Definierbarkeit der Begriffe. *Erkenntnis*, 5:80–100, 1935.
- [26] A. Tarski. *Logic, semantics and metamathematics*. Clarendon, Oxford, 1956.
- [27] B. ten Cate. Interpolation for extended modal languages. *Journal of Symbolic Logic*, 70(1):223–234, 2005.
- [28] B. ten Cate, W. Conradi, M. Marx, and Y. Venema. Definitorially complete description logics. In *Proceedings KR 2006*, pages 79–89, 2006.
- [29] J. Van den Bussche. Applications of Alfred Tarski’s ideas in database theory. *Lecture Notes in Computer Science*, 2142:20–37, 2001.
- [30] P. van Ulsen. *E.W. Beth als logicus*. PhD thesis, University of Amsterdam, 2000.
- [31] A. Visser. Uniform interpolation and layered bisimulation. In *Gödel ’96 (Brno, 1996)*, volume 6 of *Lecture Notes Logic*, pages 139–164. Springer, Berlin, 1996.