The semijoin algebra and the guarded fragment

Dirk Leinders (dirk.leinders@luc.ac.be)
Limburgs Universitair Centrum, Universitaire Campus, Diepenbeek, Belgium

Maarten Marx (marx@science.uva.nl)
Informatics Institute, Universiteit van Amsterdam, The Netherlands

Jerzy Tyszkiewicz* (jty@mimuw.edu.pl)
Institute of Informatics, University of Warsaw, ul. Banacha 2, Warszawa, Poland

Jan Van den Bussche (jan.vandenbussche@luc.ac.be)
Limburgs Universitair Centrum, Universitaire Campus, Diepenbeek, Belgium

Abstract. In the 1970s Codd introduced the relational algebra, with operators selection, projection, union, difference and product, and showed that it is equivalent to first-order logic. In this paper, we show that if we replace in Codd's relational algebra the product operator by the "semijoin" operator, then the resulting "semijoin algebra" is equivalent to the guarded fragment of first-order logic. We also define a fixed point extension of the semijoin algebra that corresponds to μGF.

1. Introduction

In the 1970s Codd introduced the relational algebra, with operators selection, projection, union, difference and product (or join) [5]. This algebra is well known in computer science, specifically in the field of databases [1]. Of fundamental importance is the equivalence in expressive power between the relational algebra and first-order logic, called relational calculus in database theory [6].

In this paper, we show that if we replace in Codd's relational algebra the product operator by the "semijoin" operator, then the resulting "semijoin algebra" is equivalent to the guarded fragment of first-order logic. This fragment was introduced by Andreka, van Benthem and Nemeti [3] to extend modal logic from Kripke structures to arbitrary relational structures, while retaining the nice properties, such as the finite model property. Since its introduction, the guarded fragment has been studied extensively [15, 12, 14, 13, 18].

On the other hand, also the semijoin operator is well known in the field of databases. While computing project-join queries in general is NP-complete in the size of the query and the database, this can be done in polynomial time when the database schema is acyclic [24], a property

* This author has been partially supported by the European Community Research Training Network "Games and Automata for Synthesis and Validation" (GAMES), contract HPRN-CT-2002-00283.

known to be equivalent to the existence of a semijoin program [4]. Semijoins are often used as part of a query pre-processing phase where dangling tuples are eliminated, i.e., the database is resized to the part that is relevant for answering the query. Another interesting property is that the size of a relation resulting from a semijoin is always linear in the size of the input. Therefore, a query processor will try to use semijoins as often as possible when generating a query plan for a given query (a technique known as "pushing projections" [9]). Also in distributed query processing, semijoins have great importance, because when a database is distributed across several sites, they can help avoid the shipment of many unneeded tuples.

Interestingly, to the best of our knowledge, the semijoin algebra was never really considered before our work.

2. Preliminaries

In this section, we recall the definition of the guarded fragment and give a formal definition of its algebraization, which we call the semijoin algebra.

Throughout this paper we fix a *finite* relational vocabulary τ. We denote the arity of a relation symbol R in τ by $\text{arity}(R)$. The maximal arity of relation symbols in τ is denoted by m. In this paper, we will work with first-order logic formulas over τ with equality. Here, we consider the equality sign $=$ as a "logical symbol"; it is always interpreted by the identity predicate, and the sign $=$ is not part of τ.

Proviso. When φ stands for a first-order formula, then $\varphi(x_1, \ldots, x_k)$ indicates that all free variables of φ are among $x_1, \ldots, x_k$.

First, we recall the definition of the guarded fragment.

DEFINITION 1 (Guarded fragment, GF).

1. *All quantifier-free first-order formulas over τ are formulas of GF.*

2. *If φ and ψ are formulas of GF, then so are $\neg\varphi$, $\varphi \vee \psi$, $\varphi \wedge \psi$, $\varphi \rightarrow \psi$ and $\varphi \leftrightarrow \psi$.*

3. *Let $\varphi(\bar{x}, \bar{y})$ be a formula of GF and let $\alpha(\bar{x}, \bar{y})$ be a relation atom over τ (i.e., an atomic formula $R(\ldots)$ with $R \in \tau$). If all free variables of φ do actually occur in α then $\exists \bar{y}(\alpha(\bar{x}, \bar{y}) \wedge \varphi(\bar{x}, \bar{y}))$ is a formula of GF.*

As the guarded fragment is a fragment of first-order logic, the semantics of GF is that of first-order logic.

Now, we define the semijoin algebra.

DEFINITION 2 (Semijoin algebra, SA). *The syntax and semantics of the semijoin algebra are inductively defined as follows:*

1. *Each relation symbol* $R \in \tau$ *is a semijoin algebra expression. Its arity comes from* τ*.*

2. *If* $E_1, E_2 \in SA$ *have arity* n*, then also* $E_1 \cup E_2$*,* $E_1 - E_2$ *belong to SA and are of arity* n*.*

3. *If* $E \in SA$ *has arity* n *and* $i, j \in \{1, \ldots, n\}$*, then* $\sigma_{i=j}(E)$ *belongs SA and is of arity* n*.*

4. *If* $E \in SA$ *has arity* n *and* $i_1, \ldots, i_k$ *are distinct elements of* $\{1, \ldots, n\}$*, then* $\pi_{i_1,\ldots,i_k}(E)$ *belongs to SA and is of arity* k*.*

5. *If* $E_1, E_2 \in SA$ *have arities* n *and* m*, and* $\theta(x_1, \ldots, x_n, y_1, \ldots, y_m)$ *is a conjunction of equalities of the form* $\bigwedge_{l=1}^{s} x_{i_l} = y_{j_l}$*, then also* $E_1 \ltimes_\theta E_2$ *belongs to SA and is of arity* n*.*

Let E *be an SA expression over* τ *and let* $\mathcal{A}$ *be a* τ*-structure. Then the result of* E *on* $\mathcal{A}$*, denoted* $E(\mathcal{A})$*, is defined inductively as follows:*

1. $R(\mathcal{A}) := R^{\mathcal{A}}$*, where* $R^{\mathcal{A}}$ *is the interpretation of* R *in structure* $\mathcal{A}$*.*

2. $E_1 \cup E_2(\mathcal{A}) := E_1(\mathcal{A}) \cup E_2(\mathcal{A})$*,* $E_1 - E_2(\mathcal{A}) := E_1(\mathcal{A}) - E_2(\mathcal{A})$*.*

3. $\sigma_{i=j}E(\mathcal{A}) := \{\bar{a} \in E(\mathcal{A}) \mid a_i = a_j\}$*.*

4. $\pi_{i_1,\ldots,i_k}E(\mathcal{A}) := \{(a_{i_1}, \ldots, a_{i_k}) \mid (a_1, \ldots, a_n) \in E(\mathcal{A})\}$*.*

5. $E_1 \ltimes_\theta E_2(\mathcal{A}) := \{\bar{a} \in E_1(\mathcal{A}) \mid \exists \bar{b} \in E_2(\mathcal{A}) : \theta(\bar{a}, \bar{b})\}$*.*

3. Semijoin algebra versus guarded fragment

Before we prove that SA is subsumed by GF, we need a lemma that says that each tuple in the result of an SA expression E on a structure $\mathcal{A}$ is guarded in the following sense:

DEFINITION 3. *Let* $\mathcal{A}$ *be a* τ*-structure with universe* A*.*

– *A set* $X \subseteq A$ *is guarded in* $\mathcal{A}$ *if there exists a tuple* $(a_1, \ldots, a_k) \in R^{\mathcal{A}}$ *(for some* R *in* τ*) such that* $X = \{a_1, \ldots, a_k\}$*.*

– *A tuple* $(a_1, \ldots, a_n) \in A^n$ *is guarded in* $\mathcal{A}$ *if* $\{a_1, \ldots, a_n\} \subseteq X$ *for some guarded set* X *in* $\mathcal{A}$.

LEMMA 4. *For every SA expression* E*, for every* τ*-structure* $\mathcal{A}$ *and for every tuple* $\bar{a}$ *in* $E(\mathcal{A})$*,* $\bar{a}$ *is guarded.*

Proof. By structural induction on expression E. □

The set of guarded k-tuples in structures of vocabulary $\tau = \{R_1, \ldots, R_t\}$ can be defined by the following formula [14]:

$$\mathbb{G}_k(x_1, \ldots, x_k) := \bigvee_{i=1}^{t} \exists \bar{y} \left(R_i \bar{y} \wedge \bigwedge_{l=1}^{k} \bigvee_{j} x_l = y_j \right)$$

Note that this formula is syntactically *not* in GF. Nevertheless, it can be equivalently expressed in GF as follows. For any complete equality type on $\{x_1, \ldots, x_k\}$ specified by a quantifier-free formula $\eta(\bar{x})$ in the language of just $=$, let $\bar{x}^\eta$ be a subtuple of $\bar{x}$ comprising precisely one variable from each $=$-class specified by η. Let $\alpha(\bar{x}^\eta, \bar{y})$ be a τ-atom in which all variables in $\bar{x}^\eta$ actually occur and the $\bar{y}$ are new, i.e., disjoint from $\bar{x}$. It is clear that the formula

$$\bigvee_{\eta} \bigvee_{\alpha} \eta(\bar{x}) \wedge \exists \bar{y} \alpha(\bar{x}^\eta, \bar{y})$$

is in GF and is equivalent to $\mathbb{G}_k(x_1, \ldots, x_k)$. The following lemma is now clear. It will also be of use in the proof of Theorem 6.

LEMMA 5. *If* $\varphi(\bar{x}, \bar{y})$ *is in GF, then* $\exists \bar{y}(\mathbb{G}(\bar{x}, \bar{y}) \wedge \varphi(\bar{x}, \bar{y}))$ *and* $\forall \bar{y}(\mathbb{G}(\bar{x}, \bar{y}) \rightarrow \varphi(\bar{x}, \bar{y}))$ *can be equivalently expressed in GF.*

This lemma implies that, if we regard $\mathbb{G}_m$ as a relation symbol, with m the maximal arity of relation symbols in τ, each GF sentence is equivalent to a sentence of the guarded logic where we always use $\mathbb{G}_m$ as the guard. It is interesting to note that historically, GF has its roots in relativized cylindric algebras, where we indeed relativize all operations to a single relation [16, 22, 21].

We now prove that SA is subsumed by GF.

THEOREM 6. *For every SA expression* E *of arity* k*, there exists a GF formula* $\varphi_E(x_1, \ldots, x_k)$ *such that for every* τ*-structure* $\mathcal{A}$ *and for every tuple* $\bar{a}$ *over* $\mathcal{A}$*, we have* $\bar{a} \in E(\mathcal{A})$ *iff* $\mathcal{A} \models \varphi_E(\bar{a})$.

Proof. The proof is by structural induction on E.

– if E is R, then $\varphi_E(x_1, \ldots, x_k) := R(x_1, \ldots, x_k)$.

– if E is $E_1 \cup E_2$, then

$$\varphi_E(x_1, \ldots, x_k) := \varphi_{E_1}(x_1, \ldots, x_k) \vee \varphi_{E_2}(x_1, \ldots, x_k).$$

– if E is $E_1 - E_2$, then

$$\varphi_E(x_1, \ldots, x_k) := \varphi_{E_1}(x_1, \ldots, x_k) \wedge \neg\varphi_{E_2}(x_1, \ldots, x_k).$$

– if E is $\sigma_{i=j}(E_1)$, then $\varphi_E(x_1, \ldots, x_k) := \varphi_{E_1}(x_1, \ldots, x_k) \wedge x_i = x_j$.

– if E is $\pi_{i_1,\ldots,i_k}(E_1)$ with E_1 of arity n, then, by induction, we have a formula $\varphi_{E_1}(z_1, \ldots, z_n)$. Now replace in $\varphi_{E_1}(\bar{z})$, for $j = 1, \ldots, k$, each occurrence of z_{i_j} by x_j, and replace, for $l \notin \{i_j \mid j = 1, \ldots, k\}$, each occurrence of z_l by y_l. Let the resulting formula be $\psi(\bar{x}, \bar{y})$. By Lemma 4, $\psi(\bar{x}, \bar{y})$ is equivalent to the formula $\mathbb{G}_n(\bar{x}, \bar{y}) \wedge \psi(\bar{x}, \bar{y})$. Now, $\varphi_E(x_1, \ldots, x_k)$ is the formula

$$\exists \bar{y}(\mathbb{G}_n(\bar{x}, \bar{y}) \wedge \psi(\bar{x}, \bar{y}))$$

which can be written guarded by Lemma 5.

– if E is $E_1 \ltimes_\theta E_2$ with $\theta = \bigwedge_{l=1}^{s} x_{i_l} = y_{j_l}$ and E_2 of arity n, then, by induction, we have formulas $\varphi_{E_1}(x_1, \ldots, x_k)$ and $\varphi_{E_2}(z_1, \ldots, z_n)$. Now replace in $\varphi_{E_2}(\bar{z})$, for $l = 1, \ldots, s$, each occurrence of z_{j_l} by x_{i_l}, and replace, for $i \notin \{j_l \mid l = 1, \ldots, s\}$, each occurrence of z_i by y_i. Let the resulting formula be $\psi(\bar{x}, \bar{y})$. By Lemma 4, $\psi(\bar{x}, \bar{y})$ is equivalent to the formula $\mathbb{G}_n(\bar{x}, \bar{y}) \wedge \psi(\bar{x}, \bar{y})$. Now, $\varphi_E(x_1, \ldots, x_k)$ is the formula

$$\varphi_{E_1}(x_1, \ldots, x_k) \wedge \exists \bar{y}(\mathbb{G}_n(\bar{x}, \bar{y}) \wedge \psi(\bar{x}, \bar{y}))$$

which can be written guarded by Lemma 5. Note that condition θ is enforced by repetition of variables x_{i_l}. □

Note that our translation from SA to GF is effectively computable. Thus, any decidability result of GF carries over to SA. In particular, by the decidability of GF [3, 12], we obtain:

COROLLARY 7. *Satisfiability of SA expressions is decidable.*

By the finite model property of GF [2], we obtain:

COROLLARY 8. *The semijoin algebra has the finite model property.*

For a fixed finite vocabulary τ, the satisfiability problem for GF is in EXPTIME [12]. Note that our translation from SA to GF is exponential in general, so an EXPTIME complexity result for SA does not directly follow from Theorem 6. Nevertheless, we have the following:

THEOREM 9. *For every fixed finite vocabulary τ, the satisfiability problem for SA is in* EXPTIME.

Proof. Given an SA expression E of arity k over τ, we apply the same translation procedure as in Theorem 6, but we use a new k-ary relation symbol H_k instead of formula $\mathbb{G}_k$. From the translation it is clear that if $\mathbb{G}_k$ is used, then $k \leqslant m$, where m is the maximal arity of relation symbols in τ. The translation thus gives us a GF formula $\varphi'_E(x_1, \ldots, x_k)$ over $\tau' := \tau \cup \{H_1, \ldots, H_m\}$. Now consider the following sentence over τ':

$$\zeta := \bigwedge_{k=1}^{m} \forall \bar{x}(\mathbb{G}_k(\bar{x}) \rightarrow H_k(\bar{x})) \ \wedge \ \bigwedge_{k=1}^{m} \forall \bar{x}(H_k(\bar{x}) \rightarrow \mathbb{G}_k(\bar{x}))$$

By Lemma 5, ζ is in GF. We now prove the following: E is satisfiable if and only if $\varphi'_E(x_1, \ldots, x_k) \wedge \zeta$ is satisfiable.

Let $\bar{a} \in E(\mathcal{A})$. By Theorem 6, $\mathcal{A} \models \varphi_E(\bar{a})$. Define $\mathcal{A}'$ as the τ'-structure with $H_k(\mathcal{A}') = \mathbb{G}_k(\mathcal{A})$, for all k. On all τ-relations R, $R(\mathcal{A})$ and $R(\mathcal{A}')$ coincide. It is now clear that $\mathcal{A}' \models \varphi'_E(\bar{a}) \wedge \zeta$.

For the other direction, let $\mathcal{A}' \models \varphi'_E(\bar{a}) \wedge \zeta$. From the definition of ζ, it follows that $H_k(\mathcal{A}') = \mathbb{G}_k(\mathcal{A}')$, for all k. Therefore, $\mathcal{A} \models \varphi_E(\bar{a})$ and thus, by Theorem 6, $\bar{a} \in E(\mathcal{A})$.

Note that ζ depends only on τ and is thus constant, and that φ'_E is computable from E in polynomial time. We have thus reduced the satisfiability problem for SA in polynomial time to the satisfiability problem for GF. □

The literal converse statement of Theorem 6 is not true, because the guarded fragment contains all quantifier-free first-order formulas, so that one can express arbitrary cartesian products in it, such as $\{(x, y) \mid S_1(x) \wedge S_2(y)\}$. Cartesian products, of course, can not be expressed in the semijoin algebra. Nevertheless, the result of any GF query restricted to guarded k-tuples, where $k \leqslant m$, is always expressible in SA.

It is clear that for every τ-structure $\mathcal{A}$, for every $k \leqslant m$, the set of guarded k-tuples in $\mathcal{A}$ equals $G_k(\mathcal{A})$, where G_k is the SA expression

$$\bigcup_{R \in \tau} \{\pi_{i_1, \ldots, i_k} R \mid 1 \leqslant i_1, \ldots, i_k \leqslant \text{arity}(R)\}.$$

We now prove

THEOREM 10. *For every GF formula $\varphi(x_1, \ldots, x_k)$ with $k \leqslant m$, there exists an SA expression $E_\varphi^{(x_1, \ldots, x_k)}$ such that for every structure $\mathcal{A}$ and for every guarded tuple $\bar{a}$ in $\mathcal{A}$, we have $\mathcal{A} \models \varphi(\bar{a})$ iff $\bar{a} \in E_\varphi^{(x_1, \ldots, x_k)}(\mathcal{A})$.*

Proof. By structural induction on φ, we construct the desired semijoin expression $E_\varphi^{(x_1, \ldots, x_k)}$.

- if $\varphi(x_1,\ldots,x_k)$ is $R(x_{i_1},\ldots,x_{i_l})$ then $E_\varphi^{(x_1,\ldots,x_k)} := G_k \ltimes_\theta R$, where θ is $(x_{i_1}=y_1)\wedge(x_{i_2}=y_2)\wedge\ldots\wedge(x_{i_l}=y_l)$;

- if $\varphi(x_1,\ldots,x_k)$ is $(x_i=x_j)$ then $E_\varphi^{(x_1,\ldots,x_k)} := \sigma_{i=j}(G_k)$;

- if $\varphi(x_1,\ldots,x_k)$ is $\psi(x_1,\ldots,x_k)\vee\xi(x_1,\ldots,x_k)$ then $E_\varphi^{(x_1,\ldots,x_k)} := E_\psi^{(x_1,\ldots,x_k)} \cup E_\xi^{(x_1,\ldots,x_k)}$;

- if $\varphi(x_1,\ldots,x_k)$ is $\neg\psi(x_1,\ldots,x_k)$ then $E_\varphi^{(x_1,\ldots,x_k)} := G_k - E_\psi^{(x_1,\ldots,x_k)}$;

- suppose $\varphi(x_1,\ldots,x_k)$ is $\exists z_1,\ldots,z_p(\alpha(\bar{x},\bar{z})\wedge\psi(\bar{x},\bar{z}))$. First, note that not every x_i may effectively occur in α. So, let $x_{i_1},\ldots,x_{i_r}$ be the variables among $x_1,\ldots,x_k$ that effectively occur in α. By induction, we have expressions $E_\alpha^{(x_{i_1},\ldots,x_{i_r},z_1,\ldots,z_p)}$ and $E_\psi^{(x_{i_1},\ldots,x_{i_r},z_1,\ldots,z_p)}$. Note that we can use the induction hypothesis on $\psi(\bar{x},\bar{z})$, because all variables in ψ must effectively occur in α and therefore $|\bar{x}|+|\bar{z}| \leqslant m$. Now, let θ_1 be $\bigwedge_{i=1}^r x_i = y_i$ and let θ_2 be $\bigwedge_{j=1}^r x_{i_j} = y_j$. Then, $E_\varphi^{(x_1,\ldots,x_k)}$ is
$$G_k \ltimes_{\theta_2} (E_\alpha^{(x_{i_1},\ldots,x_{i_r},z_1,\ldots,z_p)} \ltimes_{\theta_1} E_\psi^{(x_{i_1},\ldots,x_{i_r},z_1,\ldots,z_p)}).$$

□

Note that our translation from GF to SA is linear, so we can transfer lower complexity bounds known for GF. But some care has to be taken because we consider *finite* vocabularies, and Grädel's proof of EXPTIME-hardness for GF [12] considers an infinite (though bounded-arity) vocabulary. In fact, it is not hard to see that satisfiability for the guarded fragment with only unary predicates is NP-complete.

THEOREM 11. *For every fixed finite vocabulary τ with at least one relation symbol of arity two, the satisfiability problem for SA is* EXPTIME-*hard.*

Proof. We give a sketch only. EXPTIME-hardness of the guarded fragment can be shown by an encoding of the local-global satisfiability problem for modal logic **K**. (Given two formulas ϕ and ψ, is ϕ satisfiable in a Kripke model in which ψ holds in every world? [21].) Every modal formula is locally equivalent to the guarded formula obtained by the standard translation. Whence we obtain EXPTIME-hardness for vocabularies with an unbounded number of unary predicates and one binary predicate. Using a technique described by Halpern [17] we can reduce the number of propositional variables to just one and obtain an equisatisfiable formula. In the equivalent guarded formula we can now replace the unary predicate Px by $R(x, x)$, and again obtain an equisatisfiable formula. Whence the result. □

Combining Theorems 9 and 11, we obtain

THEOREM 12. *For every fixed finite vocabulary τ with at least one relation symbol of arity two, the satisfiability problem for SA is* EXPTIME-*complete.*

4. Fixed point extensions

In this section we define the fixed point extension μSA of SA and show that it corresponds to μGF in the same way that SA corresponds to GF. We recall the definition of guarded fixed point logic μGF [15]. For background on fixed point logics, we refer to Ebbinghaus and Flum [7].

DEFINITION 13 (μGF). *The guarded fixed point logic μGF is obtained by adding to GF the following rules for constructing fixed-point formulae:*

Let W be a k-ary relation variable and let $\bar{x} = (x_1, \ldots, x_k)$ be a k-tuple of distinct variables. Further, let $\psi(W, \bar{x})$ be a guarded formula where W appears only positively and not in guards. Moreover we require that all the free variables of $\psi(W, \bar{x})$ are contained in $\bar{x}$. For such a formula $\psi(W, \bar{x})$ we can build the formula $[\mathrm{LFP}\ W\bar{x}.\psi](\bar{x})$.

The semantics of the fixed point formulae is the usual one: Given a structure $\mathcal{A}$ and a valuation χ for the free second-order variables in ψ, other than W, the formula $\psi(W, \bar{x})$ defines an operator on k-ary relations $W \subseteq A^k$, namely $\psi^{\mathcal{A},\chi} := \{\bar{a} \in A^k \mid \mathcal{A}, \chi \models \psi(W, \bar{a})\}$. Since W occurs only positively in ψ, this operator is monotone and therefore has a least fixed point $\mathrm{LFP}(\psi^{\mathcal{A},\chi})$. Now, the semantics of a least fixed point formula is defined by $\mathcal{A}, \chi \models [\mathrm{LFP}\ W\bar{x}.\psi(W, \bar{x})](\bar{a})$ iff $\bar{a} \in \mathrm{LFP}(\psi^{\mathcal{A},\chi})$.

Correspondingly, we will now define the fixed point extension μSA of SA. We assume a relational vocabulary V disjoint from τ. The relation names in V will be called relation variables. For each μSA expression E, we also define the set $F(E)$ of free relation variables in E and the sets $\mathsf{pos}(E)$ and $\mathsf{neg}(E)$ that contain the relation variables that occur positively and negatively in E, respectively.

DEFINITION 14 (μSA). *The syntax and semantics of μSA are inductively defined as follows:*

1. *Each relation symbol $R \in \tau$ is in μSA. $F(R) = \emptyset$, $\mathsf{pos}(R) = \{R\}$ and $\mathsf{neg}(R) = \emptyset$. Its arity comes from τ.*

2. *Each relation variable $X \in V$ is in μSA. $F(X) = \{X\}$, $\mathsf{pos}(X) = \{X\}$ and $\mathsf{neg}(R) = \emptyset$. Its arity comes from V.*

3. *If $E_1, E_2 \in \mu SA$ have arity n, then also $E := E_1 \cup E_2$ belongs to μSA and is of arity n. $F(E) = F(E_1) \cup F(E_2)$, $\mathsf{pos}(E) = \mathsf{pos}(E_1) \cup \mathsf{pos}(E_2)$ and $\mathsf{neg}(E) = \mathsf{neg}(E_1) \cup \mathsf{neg}(E_2)$.*

4. *If $E_1, E_2 \in \mu SA$ have arity n, then also $E := E_1 - E_2$ belongs to μSA and is of arity n. $F(E) = F(E_1) \cup F(E_2)$, $\mathsf{pos}(E) = \mathsf{pos}(E_1) \cup \mathsf{neg}(E_2)$ and $\mathsf{neg}(E) = \mathsf{neg}(E_1) \cup \mathsf{pos}(E_2)$.*

5. *If $E \in \mu SA$ has arity n, $i, j \in \{1, \ldots, n\}$, and $i_1, \ldots, i_k$ are distinct elements of $\{1, \ldots, n\}$, then $E' := \sigma_{i=j}(E)$ and $E'' := \pi_{i_1,\ldots,i_k}(E)$ belong μSA and are of arity n and k respectively. $F(E') = F(E'') = F(E)$, $\mathsf{pos}(E') = \mathsf{pos}(E'') = \mathsf{pos}(E)$ and $\mathsf{neg}(E') = \mathsf{neg}(E'') = \mathsf{neg}(E)$.*

6. *If $E_1, E_2 \in \mu SA$ have arities n and m, and $\theta(x_1, \ldots, x_n, y_1, \ldots, y_m)$ is a conjunction of equalities of the form $\bigwedge_{l=1}^{s} x_{i_l} = y_{j_l}$, then also $E := E_1 \ltimes_\theta E_2$ belongs to μSA and is of arity n. $F(E) = F(E_1) \cup F(E_2)$, $\mathsf{pos}(E) = \mathsf{pos}(E_1) \cup \mathsf{pos}(E_2)$ and $\mathsf{neg}(E) = \mathsf{neg}(E_1) \cup \mathsf{neg}(E_2)$.*

7. *If E is a μSA expression such that $X \notin \mathsf{neg}(E)$ and $X \in F(E)$, then also $E' := [\mathrm{LFP}\ X.E]$ is a μSA expression. $F(E') = F(E) - \{X\}$, $\mathsf{pos}(E') = \mathsf{pos}(E) - \{X\}$ and $\mathsf{neg}(E') = \mathsf{neg}(E)$.*

Let E be a μSA expression and let $\mathcal{A}$ be a structure over $\tau \cup F(E)$. Then the result of E on $\mathcal{A}$, denoted $E(\mathcal{A})$, is defined inductively as follows:

1. *For R in $\tau \cup F(E)$, $R(\mathcal{A}) := R^{\mathcal{A}}$ where $R^{\mathcal{A}}$ is the interpretation of R in structure $\mathcal{A}$.*

2. $E_1 \cup E_2(\mathcal{A}) := E_1(\mathcal{A}) \cup E_2(\mathcal{A})$, $E_1 - E_2(\mathcal{A}) := E_1(\mathcal{A}) - E_2(\mathcal{A})$.

3. $\sigma_{i=j}E(\mathcal{A}) := \{\bar{a} \in E(\mathcal{A}) \mid a_i = a_j\}$.

4. $\pi_{i_1,\ldots,i_k}E(\mathcal{A}) := \{(a_{i_1}, \ldots, a_{i_k}) \mid (a_1, \ldots, a_n) \in E(\mathcal{A})\}$.

5. $E_1 \ltimes_\theta E_2(\mathcal{A}) := \{\bar{a} \in E_1(\mathcal{A}) \mid \exists \bar{b} \in E_2(\mathcal{A}) : \theta(\bar{a}, \bar{b})\}$.

Let $\mathcal{A}$ be a structure over $\tau \cup F(E) - \{X\}$ with universe A. Let k be the arity of X. Then $[\mathrm{LFP}\ X.E](\mathcal{A})$ is defined as the least fixed *point of the operator $E^{\mathcal{A}}$ on k-ary relations on A, defined as follows: $E^{\mathcal{A}}(r) := E(\mathcal{A}, r)$. Here, by $(\mathcal{A}, r)$ we denote the structure $\mathcal{A}'$ over $\tau \cup F(E)$ defined by*

$$\begin{cases} R^{\mathcal{A}'} &= R^{\mathcal{A}} \text{ if } R \in \tau \\ Y^{\mathcal{A}'} &= Y^{\mathcal{A}} \text{ if } Y \in V,\ Y \neq X \\ X^{\mathcal{A}'} &= r \end{cases}$$

This least fixed point always exists because $E^{\mathcal{A}}$ is monotone, as shown in Lemma 15.

LEMMA 15. *Let E be a μSA expression such that $X \in F(E)$. Let $\mathcal{A}$ be a structure over $\tau \cup F(E) - \{X\}$. If $X \notin \mathsf{neg}(E)$, then the operator $E^{\mathcal{A}}$ is monotone; if $X \notin \mathsf{pos}(E)$, then $E^{\mathcal{A}}$ is anti-monotone.*

Proof. The proof is by structural induction on E. Suppose $X \notin \mathsf{neg}(E)$. The base case where $E = X$ is clear. Suppose the lemma is true for E_1 and E_2, then the lemma also holds for $\sigma_{i=j}E_1$, $\pi_{i_1,\dots,i_k}E_1$, $E_1 \cup E_2$ and $E_1 \ltimes_\theta E_2$ because selection, projection, union and semijoin are monotone operators. If $E = E_1 - E_2$ and $X \notin \mathsf{neg}(E)$, then $X \notin \mathsf{neg}(E_1)$ and $X \notin \mathsf{pos}(E_2)$, so $E_1^{\mathcal{A}}$ is monotone and $E_2^{\mathcal{A}}$ is anti-monotone by induction. Then, clearly $E^{\mathcal{A}}$ is monotone. The case where $X \notin \mathsf{pos}(E)$ is analogous. □

We now prove that μSA and μGF are equivalent in the same way as the logics without fixed point extensions: μSA is subsumed by μGF, and conversely, the result of any μGF query restricted to guarded tuples is always expressible in μSA.

THEOREM 16. *For every μSA expression E of arity k, there exists a μGF formula $\varphi_E(x_1, \dots, x_k)$ such that for every structure $\mathcal{A}$ and for every tuple $\bar{a}$ over $\mathcal{A}$, we have $\bar{a} \in E(\mathcal{A})$ iff $\mathcal{A} \models \varphi_E(\bar{a})$.*

Proof. The proof is by structural induction. All cases except least fixed point are handled as in the proof of Theorem 6. In particular, if $X \in F(E_1)$, then X does not appear in a guard of φ_{E_1}; if $X \notin \mathsf{neg}(E_1)$, then X is positive in φ_{E_1}. Consider now the case where E is of the form [LFP $X.E_1$]. Now, [LFP $X\bar{x}.\varphi_{E_1}(\bar{x})](\bar{x})$ is a well-defined μGF formula equivalent to E. □

To go from μGF to μSA, the following lemma proved by Grädel et al. [14] is particularly instrumental:

LEMMA 17. *Any formula of μGF is logically equivalent to one in which all fixed points are of the form* [LFP $W\bar{x}.(\psi(\bar{x}) \wedge \mathbb{G}(\bar{x}))](\bar{x})$.

THEOREM 18. *For every μGF formula $\varphi(x_1, \ldots, x_k)$ with $k \leqslant m$, there exists a μSA expression E_φ such that for every structure $\mathcal{A}$ and for every guarded tuple $\bar{a}$ in $\mathcal{A}$, we have $\mathcal{A} \models \varphi(\bar{a})$ iff $\bar{a} \in E_\varphi(\mathcal{A})$.*

Proof. The proof is by structural induction. All cases except least fixed point are handled as in the proof of Theorem 10. Consider now the case where $\varphi(x_1, \ldots, x_k)$ is of the form [LFP $W\bar{x}.\psi(\bar{x})](\bar{x})$. By Lemma 17, we may assume that $\psi(\bar{x})$ is of the form $\chi(\bar{x}) \wedge \mathbb{G}(\bar{x})$. By induction we have that [LFP $X.E_\chi^{(\bar{x})}$] is equivalent to $\varphi(\bar{x})$. □

Using an argument similar to that of Theorem 9, and given that for any fixed vocabulary, satisfiability for μGF is in EXPTIME [15], we obtain that satisfiability for μSA is in EXPTIME. It is actually EXPTIME-complete, since satisfiability for SA is already EXPTIME-hard (Theorem 11).

5. Generalizations of GF and SA

In Codd's relational algebra (RA), mentioned in the Introduction, the semijoin operator is replaced by the product operator. Syntax and semantics of the other operators remain unchanged. The syntax and semantics of the product operator are as follows. If $E_1, E_2 \in$ RA have arities n and m, then $E_1 \times E_2$ belongs to RA and has arity $n + m$; $E_1 \times E_2(\mathcal{A}) := \{(\bar{a}, \bar{b}) \mid \bar{a} \in E_1(\mathcal{A}), \bar{b} \in E_2(\mathcal{A})\}$. As mentioned in the Introduction, this relational algebra is equivalent to full first-order logic [6].

The semijoin operator can be seen as a relativized version of the product operator; thus, SA is a relativized version of RA. Indeed, let $\mathcal{I}$ be a function mapping pairs $(\mathcal{A}, k)$, where $\mathcal{A}$ is a τ-structure and k is a natural number, to relations, such that $\mathcal{I}(\mathcal{A}, k)$ is a k-ary relation on $\mathcal{A}$. Define the syntax and semantics of the relational algebra relativized to $\mathcal{I}$, as follows:

- The syntax is that of the relational algebra;
- The semantics of the selection, projection, union and difference operator are the same as in the relational algebra. The semantics of the product operator relativized to $\mathcal{I}$ is defined as follows:

$$E_1 \times E_2(\mathcal{A}) := \{(\bar{a}, \bar{b}) \mid \bar{a} \in E_1(\mathcal{A}), \bar{b} \in E_2(\mathcal{A}), \\ (\bar{a}, \bar{b}) \in \mathcal{I}(\mathcal{A}, \text{arity}(\bar{a}) + \text{arity}(\bar{b}))\}$$

We denote RA relativized to $\mathcal{I}$ by RA$^{\mathcal{I}}$. Then, if we define $\mathcal{I}^{\text{GF}}(\mathcal{A}, k) := G_k(\mathcal{A})$, for all $\mathcal{A}$ and k, it is clear that RA$^{\mathcal{I}^{\text{GF}}}$ is equivalent to SA (and thus also to GF).

In literature, generalizations of GF based on loosening the guards have been considered [23, 11, 20]. In the packed fragment for example [20], all quantifications are relative to the set of packed tuples. A tuple $\bar{a}$ is *packed* in a τ-structure $\mathcal{A}$ if each a_i and a_j appear together in some tuple $\bar{d} \in R(\mathcal{A})$. If we define $\mathcal{I}^{\mathrm{PF}}(\mathcal{A}, k) := P_k(\mathcal{A})$, where P_k returns all packed k-tuples in $\mathcal{A}$, then it is easy to adapt our proofs of Theorem 6 and 10 and show that $\mathrm{RA}^{\mathcal{I}^{\mathrm{PF}}}$ is equivalent to the packed fragment.

6. Evaluation complexity

For a fixed finite vocabulary τ, we can consider the evaluation problem for SA, defined as follows:

Input: A finite τ-structure $\mathcal{A}$, a SA expression E and a tuple $\bar{a} \in \mathcal{A}$.

Decide: Is $\bar{a} \in E(\mathcal{A})$?

It is known that the corresponding problem for GF is decidable in linear time on a RAM (Random Access Machine), provided a suitable array-based representation is used to represent finite structures [10]. Actually, in that article, this linear evaluation complexity was shown for a language called Datalog LIT, and it is an easy matter to provide a linear translation from SA to Datalog LIT. We can thus conclude:

THEOREM 19. *For every fixed finite vocabulary τ, the evaluation problem for SA can be solved in linear time.*

7. Discussion

Our characterization of the guarded fragment by using semijoins suggests generalizations of GF in directions other than those considered up to now, based on loosening the guards. Specifically, we can allow other semijoin conditions than just conjunctions of equalities.

But, as the following example shows, such generalizations are not innocent. For instance, let us allow nonequalities in semijoin conditions. This variant of the semijoin algebra, denoted $\mathrm{SA}^{\neq}$, is strictly more expressive than GF. Consider for example the query that asks whether there are at least two distinct elements in a single unary relation S. This is expressible in $\mathrm{SA}^{\neq}$ as $S \ltimes_{x_1 \neq y_1} S$, but is not expressible in GF. Indeed, a set with a single element is "guarded bisimilar" to a set with two elements [3].

Unfortunately, it follows from a result by Grädel that these nonequalities in semijoin conditions make SA undecidable.

THEOREM 20. *Satisfiability of $SA^{\neq}$ expressions is undecidable.*

Proof. Grädel [12, Theorem 5.8] shows that GF with functionality statements in the form of functional[D], saying that the binary relation D is the graph of a partial function, is a conservative reduction class. Since functional[D] is expressible in $\mathrm{SA}^{\neq}$ as $D \ltimes_{x_1=y_1 \wedge x_2 \neq y_2} D = \emptyset$, it follows that $\mathrm{SA}^{\neq}$ is undecidable. □

A generalization of guarded bisimilarity to the semijoin algebra with arbitrary semijoin conditions has been proposed by us in a previous paper [19].

We note that it has already been observed that boolean acyclic non-recursive stratified datalog (NRSD) programs have the same expressive power as GF sentences [8, 10]. Each rule in such a program is an acyclic join query. By the well-known correspondence between acyclic join queries and semijoin programs [4], these acyclic NRSD programs also correspond to SA. Hence, the correspondence we have shown between SA and GF could also have been derived by combining these previous results. Nevertheless, the equivalence proof we give is direct and elementary.

References

1. S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases*. Addison-Wesley, 1995.
2. H. Andreka, I. Hodkinson, and I. Németi. Finite algebras of relations are representable on finite sets. *Journal of Symbolic Logic*, 64(1):243–267, 1999.
3. H. Andréka, I. Németi, and J. van Benthem. Modal languages and bounded fragments of predicate logic. *Journal of Philosophical Logic*, 27(3):217–274, 1998.
4. C. Beeri, R. Fagin, D. Maier, and M. Yannakakis. On the desirability of acyclic database schemes. *Journal of the ACM*, 30(3):479–513, 1983.
5. E.F. Codd. A relational model of data for large shared data banks. *Communications of the ACM*, 13(6):377–387, 1970.
6. E.F. Codd. Relational completeness of data base sublanguages. In R. Rustin, editor, *Data Base Systems*, pages 65–98. Prentice-Hall, 1972.
7. H.-D. Ebbinghaus and J. Flum. *Finite Model Theory*. Springer, 1999.
8. J. Flum, M. Frick, and M. Grohe. Query evaluation via tree-decompositions. *Journal of the ACM*, 49(6):716–752, 2002.
9. H. Garcia-Molina, J.D. Ullman, and J. Widom. *Database Systems: the Complete Book*. Prentice Hall, 2000.
10. G. Gottlob, E. Grädel, and H. Veith. Datalog lite: a deductive query language with linear time model checking. *ACM Transactions on Computational Logic*, 3(1):42–79, 2002.
11. E. Grädel. Decision procedures for guarded logics. In *Proceedings of the 16th International Conference on Automated Deduction*, volume 1632. Springer-Verlag, 1999.

12. E. Grädel. On the restraining power of guards. *Journal of Symbolic Logic*, 64(4):1719–1742, 1999.
13. E. Grädel. Guarded fixed point logics and the monadic theory of countable trees. *Theoretical Computer Science*, 288(1):129–152, 2002.
14. E. Grädel, C. Hirsch, and M. Otto. Back and forth between guarded and modal logics. *ACM Transactions on Computational Logic*, 3(3):418–463, 2002.
15. E. Grädel and I. Walukiewicz. Guarded fixed point logic. In *Proceedings of the 14th IEEE Symposium on Logic in Computer Science LICS '99*, pages 45–54, 1999.
16. P. Halmos. *Algebraic Logic*. American Mathematical Society, 1962.
17. J.Y. Halpern. The effect of bounding the number of primitive propositions and the depth of nesting on the complexity of modal logic. *Artificial Intelligence*, 75(2):361–372, 1995.
18. E. Hoogland and M. Marx. Interpolation and definability in guarded fragments. *Studia Logica*, 70(3):373–409, April 2002.
19. D. Leinders, J. Tyszkiewicz, and J. Van den Bussche. On the expressive power of semijoin queries. *Information Processing Letters*, 91(2):93–98, 2004.
20. M. Marx. Tolerance logic. *Journal of Logic, Language and Information*, 10(3):353–374, 2001.
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, 2004.
22. I. Németi. A fine-structure analysis of first-order logic. In M. Marx, L. Pólos, and M. Masuch, editors, *Arrow Logic and Multimodal Logics*, Studies in Logic, Language and Information, pages 221–247. CSLI Publications, Stanford, 1995.
23. J. van Benthem. Dynamic bits and pieces. Technical Report LP-1997-01, ILLC, 1997.
24. M. Yannakakis. Algorithms for acyclic database schemes. In *Proceedings of the 7th International Conference on Very Large Data Bases*, pages 82–94. IEEE Computer Society, 1981.