The semijoin algebra and the guarded fragment

Dirk Leinders
Jerzy Tyszkiewicz *
Jan Van den Bussche

Abstract

The semijoin algebra is the variant of the relational algebra obtained by replacing the join operator by the semijoin operator. We discuss some interesting connections between the semijoin algebra and the guarded fragment of first-order logic. We also provide an Ehrenfeucht-Fraïssé game, characterizing the discerning power of the semijoin algebra. This game gives a method for showing that certain queries are not expressible in the semijoin algebra.

1 Introduction

Semijoins are very important in the field of database query processing. 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 [12], a property known to be equivalent to the existence of a semijoin program [3]. Semijoins are often used as part of a query pre-processing phase where dangling tuples are eliminated. 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" [7]). 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.

Because of its practical importance, we would like to have a clear knowledge of the capabilities and the limitations of semijoins. For example, Bernstein, Chiu and Goodman [4, 5] have characterized the conjunctive queries computable by semijoin programs. In this paper, we consider the much larger class of queries computable in the variant of the relational algebra obtained by replacing the join operator by the semijoin operator. We call this the semijoin algebra (SA). A join of two relations combines all tuples satisfying a given condition, called

*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.

the join condition. A semijoin differs from a join in the sense that it selects only those tuples in the first relation that participate in the join. The semijoin algebra is a fragment of the relational algebra, which is known to be equivalent to first-order logic (called relational calculus in database theory [1]).

Interestingly, there is a fragment of first-order logic very similar to the semijoin algebra: it is the so called "guarded fragment" (GF) [2, 8, 9, 10], which has been studied in the field of modal logic. This is interesting because the motivations to study this fragment came purely from the field of logic and had nothing to do with database query processing. Indeed, the purpose was to extend propositional modal logic to the predicate level, retaining the good properties of modal logic, such as the finite model property. An important tool in the study of the expressive power of the GF is the notion of "guarded bisimulation", which provides a characterization of the discerning power of the GF.

We will show that when we allow only equalities to appear in the semijoin conditions, the semijoin algebra has essentially the same expressive power as the guarded fragment. When also nonequalities or other predicates are allowed, the semijoin algebra becomes more powerful. We will define a generalization of guarded bisimulation, in the form of an Ehrenfeucht-Fraïssé game, that characterizes the discerning power of the semijoin algebra. We will use this tool to show that certain queries are not expressible in SA.

2 Preliminaries

In this section, we give formal definitions of the semijoin algebra and the guarded fragment.

From the outset, we assume a universe $\mathbb{U}$ of basic data values, over which a number of predicates are defined. These predicates can be combined into quantifier-free first-order formulas, which are used in selection and semijoin conditions. The names of these predicates and their arities are collected in the vocabulary Ω. The equality predicate $(=)$ is always in Ω. A database schema is a finite set $\mathbf{S}$ of relation names, each associated with its arity. $\mathbf{S}$ is disjoint from Ω. A database D over $\mathbf{S}$ is an assignment of a finite relation $D(R) \subseteq \mathbb{U}^n$ to each $R \in \mathbf{S}$, where n is the arity of R.

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 define the Semijoin Algebra.

Definition 1 (Semijoin algebra, SA). Let $\mathbf{S}$ be a database schema. Syntax and semantics of the Semijoin Algebra is inductively defined as follows:

1. Each relation $R \in \mathbf{S}$ is a semijoin algebra expression.
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 $X \subseteq \{1, \ldots, n\}$, then $\pi_X(E)$ belongs to SA and is of arity $\#X$.

4. If $E_1, E_2 \in \mathrm{SA}$ have arities n and m, respectively, and $\theta_1(x_1, \ldots, x_n)$ and $\theta_2(x_1, \ldots, x_n, y_1, \ldots, y_m)$ are quantifier-free formulas over Ω, then also $\sigma_{\theta_1}(E_1)$ and $E_1 \ltimes_{\theta_2} E_2$ belong to SA and are of arity n.

The semantics of the projection, the selection and the semijoin operator are as follows: $\pi_X(E) := \{(a_i)_{i \in X} \mid (a_1, \ldots, a_n) \in E\}$, $\sigma_{\theta_1}(E) := \{(a_1, \ldots, a_n) \in E \mid \theta_1(a_1, \ldots, a_n) \text{ holds}\}$, $E_1 \ltimes_{\theta_2} E_2 := \{(a_1, \ldots, a_n) \in E_1 \mid \exists (b_1, \ldots, b_m) \in E_2, \theta_2(a_1, \ldots, a_n, b_1, \ldots, b_m) \text{ holds}\}$. The semantics of the other operators are well known.

Now, we recall the definition of the guarded fragment.

Definition 2 (Guarded fragment, GF). Let $\mathbf{S}$ be a database schema.

1. All quantifier-free first-order formulas over $\mathbf{S}$ 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. If $\varphi(\bar{x}, \bar{y})$ is a formula of GF and $\alpha(\bar{x}, \bar{y})$ is an atomic formula such that 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, interpreted over the active domain of the database [1].

3 Semijoin algebra versus guarded fragment

In this section, $\Omega = \{=\}$ consists only of the equality predicate. Suppose furthermore that we only allow conjunctions of equalities to be used in the semijoin conditions; selection conditions can be arbitrary quantifier-free formulas over Ω. We will denote the semijoin algebra with this restriction on the semijoin conditions by $\mathrm{SA}^=$. Before we prove that $\mathrm{SA}^=$ is subsumed by GF, we need a lemma.

Lemma 3. *For every $SA^=$ expression E of arity k, for every database A and for every tuple $\bar{z} = (z_1, \ldots, z_k)$ in $E(A)$, there exists R in $\mathbf{S}$, an injective function $f : \{1, \ldots, k\} \rightarrow \{1, \ldots, \mathrm{arity}(R)\}$, and a tuple $\bar{t}$ in $A(R)$ such that $\bigwedge_{i=1}^{k} z_i = t_{f(i)}$.*

Proof. By structural induction on expression E. □

Theorem 4. *For every $SA^=$ expression E of arity k, there exists a GF formula φ_E such that for every database D, $E(D) = \{\bar{d} \in D \mid \varphi_E(\bar{d})\}$.*

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 $\sigma_\theta(E_1)$, then $\varphi_E(x_1, \ldots, x_k) := \varphi_{E_1}(x_1, \ldots, x_k) \wedge \theta(x_1, \ldots, x_k)$.
- if E is $\pi_{i_1,\ldots,i_k}(E_1)$ with E_1 of arity n, then, by induction, $\varphi_{E_1}(z_1, \ldots, z_n)$ defines all tuples in $E_1(D)$. By Lemma 3, $\varphi_{E_1}(\bar{z})$ is equivalent to the formula obtained by replacing in $\psi :=$

$$\bigvee_{R \in \mathbf{S}} \bigvee_{\substack{f:\{1,\ldots,n\}\to \\ \{1,\ldots,\mathrm{arity}(R)\}}} \exists (t_j)_{j \in Q} \big(R(\bar{t}) \ \wedge \ \varphi_{E_1}(t_{f(1)}, \ldots, t_{f(n)}) \big)$$

  each $t_{f(i)}$ by z_i, $i = 1, \ldots, n$. In this formula, Q is a shorthand for the set $\{1, \ldots, \mathrm{arity}(R)\} - f(\{1, \ldots, n\})$. Formula φ_E should now only select components $i_1, \ldots, i_k$ out of this formula. To this end, we modify ψ such that in each disjunct it quantifies over $(t_j)_{j \in Q'}$ with $Q' = \{1, \ldots, \mathrm{arity}(R)\} - f(\{i_1, \ldots, i_k\})$ and in each disjunct $t_{f(i_l)}$ is replaced by x_l, $l = 1, \ldots, k$. Now $\varphi_E(x_1, \ldots, x_k)$ is obtained.
- 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, $\varphi_{E_1}(x_1, \ldots, x_k)$ and $\varphi_{E_2}(z_1, \ldots, z_n)$ define all tuples in $E_1(D)$ and $E_2(D)$ respectively. By Lemma 3, $\varphi_E(x_1, \ldots, x_k)$ is obtained by replacing in formula $\chi :=$

$$\phi_{E_1}(x_1, \ldots, x_k) \wedge \bigvee_{R \in \mathbf{S}} \bigvee_{\substack{f:\{1,\ldots,n\}\to \\ \{1,\ldots,\mathrm{arity}(R)\}}} \exists (t_j)_{j \in Q''} \big(R(\bar{t}) \ \wedge \ \varphi_{E_2}(t_{f(1)}, \ldots, t_{f(n)}) \big)$$

  each $t_{f(j_l)}$ by x_{i_l}, $l = 1, \ldots, s$. Note that condition θ is enforced by repetition of variables x_{i_l}. In this formula, $Q'' = \{1, \ldots, \mathrm{arity}(R)\} - f(\{j_1, \ldots, j_s\})$.

□

By the decidability of GF, we obtain:

Corollary 5. *Satisfiability of $SA^=$ expressions is decidable.*

With decidability of SA expressions, we always mean finite satisfiability, because a database is finite by definition.

The literal converse statement of Theorem 4 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 a single relation by a semijoin is always expressible in $\mathrm{SA}^=$:

Theorem 6. *For every GF formula $\varphi(x_1, \ldots, x_k)$, for every relation R (with arity n), for every injective function $f : \{1, \ldots, k\} \to \{1, \ldots, n\}$, the query $\{\bar{x} \mid \varphi(\bar{x})\} \ltimes_\theta R$ in which θ is $\bigwedge_{i=1}^{k} x_i = y_{f(i)}$, is expressible in $SA^=$.*

Proof. By structural induction on φ, we construct the desired semijoin expression $E^{f,R}_{\varphi,k}$.

- if $\varphi(x_1,\ldots,x_k)$ is $T(x_{i_1},\ldots,x_{i_l})$ then $E^{f,R}_{\varphi,k} := \pi_{f(1),\ldots,f(k)}(R) \ltimes_\theta T$, 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 $E^{f,R}_{\varphi,k} := \sigma_{i=j}(\pi_{f(1),\ldots,f(k)}(R))$;
- if $\varphi(x_1,\ldots,x_k)$ is $\psi(x_1,\ldots,x_k) \vee \xi(x_1,\ldots,x_k)$ then $E^{f,R}_{\varphi,k} := E^{f,R}_{\psi,k} \cup E^{f,R}_{\xi,k}$;
- if $\varphi(x_1,\ldots,x_k)$ is $\neg\psi(x_1,\ldots,x_k)$ then $E^{f,R}_{\varphi,k} := \pi_{f(1),\ldots,f(k)}(R) - E^{f,R}_{\psi,k}$;
- suppose $\varphi(x_1,\ldots,x_k)$ is $\exists\bar{z}(\alpha(\bar{x},\bar{z}) \wedge \psi(\bar{x},\bar{z}))$, where α is atomic with relation name T. Let $x_{i_1},\ldots,x_{i_r}$ be the different occurrences of variables among $x_1,\ldots,x_k$ in α. Now, $E^{f,R}_{\varphi,k} := \pi_{f(1),\ldots,f(k)}(R) \ltimes_\theta E^{g,T}_{\psi,r+l}$ where θ is $(x_{i_1} = y_1) \wedge (x_{i_2} = y_2) \wedge \ldots \wedge (x_{i_r} = y_r)$ and g is the function that maps $j \in \{1,\ldots,r\}$ to the position of x_{i_j} in α and that maps $j \in \{r+1,\ldots,r+l\}$ to the position of z_{j-r} in α.

□

Taking $k = 0$ and R equal to any nonempty relation in the above theorem, we obtain:

Corollary 7. *Over the class of nonempty databases GF sentences and 0-ary SA$^=$ expressions have equal expressive power.*

Here, a database is said to be empty if all its relations are empty.

Let us now allow arbitrary semijoin conditions (still over equality only). Specifically, nonequalities are now allowed. We will denote the semijoin algebra over $\Omega = \{=\}$ by SA$^{\neq}$. Then, GF no longer subsumes SA$^{\neq}$. A counterexample is the query that asks whether there are at least two distinct elements in a single unary relation S. This is expressible in 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 [2, 10].

Unfortunately, these nonequalities in semijoin conditions make SA undecidable.

Theorem 8. *Satisfiability of SA$^{\neq}$ expressions is undecidable.*

Proof. Grädel [8, 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 SA$^{\neq}$ as $D \ltimes_{x_1 = y_1 \wedge x_2 \neq y_2} D = \emptyset$, it follows that SA$^{\neq}$ is undecidable. □

In the next section, we will generalize guarded bisimulation to the semijoin algebra, with arbitrary quantifier-free formulas over Ω as semijoin conditions.

4 An Ehrenfeucht-Fraïssé game for the semijoin algebra

In this section, we describe an Ehrenfeucht-Fraïssé game that characterizes the discerning power of the semijoin algebra.

Let A and B be two databases over the same schema $\mathbf{S}$. The *semijoin game* on these databases is played by two players, called the spoiler and the duplicator. They, in turn, choose tuples from the tuple spaces T_A and T_B, which are defined as follows: $T_A := \bigcup_{R \in \mathbf{S}} \bigcup \{\pi_{\mathrm{X}}(A(R)) \mid \mathrm{X} \subseteq \{1, \ldots, \mathrm{arity}(R)\}\}$, and T_B is defined analogously. So, the players can pick tuples from the databases and projections of these.

At each stage in the game, there is a tuple $\bar{a} \in T_A$ and a tuple $\bar{b} \in T_B$. We will denote such a configuration by $(A, \bar{a}; B, \bar{b})$. The conditions for the duplicator to win the game with 0 rounds are:

1. $\forall R \in \mathbf{S}, \forall \mathrm{X} \subseteq \{1, \ldots, \mathrm{arity}(R)\} : \bar{a} \in \pi_{\mathrm{X}}(A(R)) \Leftrightarrow \bar{b} \in \pi_{\mathrm{X}}(B(R))$
2. for every atomic formula (equivalently, for every quantifier-free formula) θ over Ω, $\theta(\bar{a})$ holds iff $\theta(\bar{b})$ holds.

In the game with $m \geqslant 1$ rounds, the spoiler will be the first one to make a move. Therefore, he first chooses a database (A or B). Then he picks a tuple in T_A or in T_B respectively. The duplicator then has to make an "analogous" move in the other tuple space. When the duplicator can hold this for m times, no matter what moves the spoiler takes, we say that the duplicator wins the m-round semijoin game on A and B. The "analogous" moves for the duplicator are formally defined as legal answers in the next definition.

Definition 9 (legal answer). Suppose that at a certain moment in the semijoin game, the configuration is $(A, \bar{a}; B, \bar{b})$. If the spoiler takes a tuple $\bar{c} \in T_A$ in his next move, then the tuples $\bar{d} \in T_B$, for which the following conditions hold, are legal answers for the duplicator:

1. $\forall R \in \mathbf{S}, \forall \mathrm{X} \subseteq \{1, \ldots, \mathrm{arity}(R)\} : \bar{d} \in \pi_{\mathrm{X}}(B(R)) \Leftrightarrow \bar{c} \in \pi_{\mathrm{X}}(A(R))$
2. for every atomic formula θ over Ω, $\theta(\bar{a}, \bar{c})$ holds iff $\theta(\bar{b}, \bar{d})$ holds.

If the spoiler takes a tuple $\bar{d} \in T_B$, the legal answers $\bar{c} \in T_A$ are defined identically.

In the following, we denote the semijoin game with initial configuration $(A, \bar{a}; B, \bar{b})$ and that consists of m rounds, by $G_m(A, \bar{a}; B, \bar{b})$.

We first state and prove

Proposition 10. *If the duplicator wins $G_m(A, \bar{a}; B, \bar{b})$, then for each semijoin expression E with $\leqslant m$ nested semijoins and projections, we have $\bar{a} \in E(A) \Leftrightarrow \bar{b} \in E(B)$.*

Proof. We prove this by induction on m. The base case $m = 0$ is clear. Now consider the case $m > 0$. Suppose that $\bar{a} \in E_1 \ltimes_\theta E_2(A)$ but $\bar{b} \notin E_1 \ltimes_\theta E_2(B)$. Then $\bar{a} \in E_1(A)$ and $\exists \bar{c} \in E_2(A) : \theta(\bar{a}, \bar{c})$, and either (*) $\bar{b} \notin E_1(B)$ or (**) $\neg \exists \bar{d} \in E_2(B) : \theta(\bar{b}, \bar{d})$. In situation (*), $\bar{a}$ and $\bar{b}$ are distinguished by an expression with $m-1$ semijoins or projections, so the spoiler has a winning strategy; in situation (**), the spoiler has a winning strategy by choosing this $\bar{c} \in E_2(A)$ with $\theta(\bar{a}, \bar{c})$, because each legal answer of the duplicator $\bar{d}$ has $\theta(\bar{b}, \bar{d})$ and therefore $\bar{d} \notin E_2(B)$. So, the spoiler now has a winning strategy in the game $G_{m-1}(A, \bar{c}; B, \bar{d})$. In case a projection distinguishes $\bar{a}$ and $\bar{b}$, a similar winning strategy for the spoiler exists. In case $\bar{a}$ and $\bar{b}$ are distinguished by an expression that is neither a semijoin, nor a projection, there is a simpler expression that distinguishes them, so the result follows by structural induction. □

We now come to the main theorem of this section. This theorem concerns the game $G_\infty(A, \bar{a}; B, \bar{b})$, which we also abbreviate as $G(A, \bar{a}; B, \bar{b})$. We say that the duplicator wins $G(A, \bar{a}; B, \bar{b})$ if the spoiler has no winning strategy. This means that the duplicator can keep on playing forever, choosing legal answers for every move of the spoiler.

Theorem 11. *The duplicator wins $G(A, \bar{a}; B, \bar{b})$ if and only if for each semijoin expression E, we have $\bar{a} \in E(A) \Leftrightarrow \bar{b} \in E(B)$.*

Proof. The 'only if' direction of the proof follows directly from Proposition 10, because if the duplicator wins $G(A, \bar{a}; B, \bar{b})$, he wins $G_m(A, \bar{a}; B, \bar{b})$ for every $m \geqslant 0$. So, $\bar{a}$ and $\bar{b}$ are indistinguishable through all semijoin expressions. For the 'if' direction, it is sufficient to prove that if the duplicator loses, $\bar{a}$ and $\bar{b}$ are distinguishable. We therefore construct, by induction, a semijoin expression $E^r_{\bar{a}}$ such that (i) $\bar{a} \in E^r_{\bar{a}}(A)$, and (ii) $\bar{b} \in E^r_{\bar{a}}(B)$ iff the duplicator wins $G_r(A, \bar{a}; B, \bar{b})$. We define $E^0_{\bar{a}}$ as

$$\sigma_{\theta_{\bar{a}}}\Big(\bigcap_{R \in \mathbf{S}} \bigcap_{\{\mathrm{X} \subseteq Z \mid \bar{a} \in \pi_{\mathrm{X}}(A(R))\}} \pi_{\mathrm{X}}(R)\Big) - \bigcup_{R \in \mathbf{S}} \bigcup_{\{\mathrm{X} \subseteq Z \mid \bar{a} \notin \pi_{\mathrm{X}}(A(R))\}} \pi_{\mathrm{X}}(R)$$

In this expression, Z is a shorthand for $\{1, \ldots, \mathrm{arity}(R)\}$ and $\theta_{\bar{a}}$ is the *atomic type* of $\bar{a}$ over Ω, i.e., the conjunction of all atomic and negated atomic formulas over Ω that are true of $\bar{a}$.

We now construct $E^r_{\bar{a}}$ in terms of $E^{r-1}_{\bar{a}}$:

$$\bigcap_{\bar{c} \in T_A} \big(E^0_{\bar{a}} \ltimes_{\theta_{\bar{a},\bar{c}}} E^{r-1}_{\bar{c}}\big) \cap \Big(E^0_{\bar{a}} - \bigcup_{j=1}^{s} \bigcup_{\theta} \big(E^0_{\bar{a}} \ltimes_\theta \bigcap_{\substack{\bar{c} \in T_A \\ \theta(\bar{a},\bar{c})}} (E^{r-1}_{\bar{c}})^{\mathrm{compl}}\big)\Big)$$

In this expression, $\theta_{\bar{a},\bar{c}}$ is the atomic type of $\bar{a}$ and $\bar{c}$ over Ω; s is the maximal arity of a relation in $\mathbf{S}$; θ ranges over all atomic Ω-types of two tuples, one with the arity of $\bar{a}$, and one with arity j. The notation E^{compl}, for an expression of arity k, is a shorthand for

$$\Big(\bigcup_{R \in \mathbf{S}} \bigcup_{\substack{X \subseteq \{1, \ldots, \mathrm{arity}(R)\} \\ \#X = k}} \pi_{\mathrm{X}}(R)\Big) - E$$

□

5 Queries inexpressible in the semijoin algebra

Grädel [8] already showed that transitivity is not expressible in the guarded fragment. We will now show that transitivity is still inexpressible in the more powerful semijoin algebra.

Theorem 12. *Transitivity is inexpressible in the semijoin algebra.*

Proof. We will give two databases A and B over the schema $\mathbf{S}$ containing a single relation R, that are indistinguishable by semijoin expressions, and with the property that R is transitive in A and not in B. These databases are shown graphically in Figure 1.

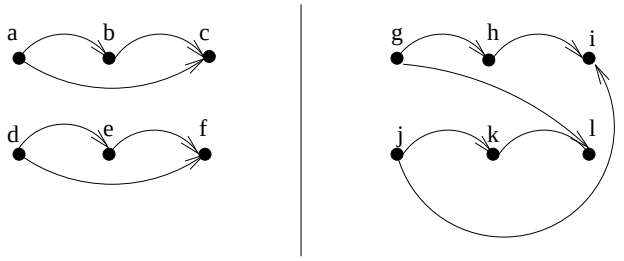

Figure 1: Databases A (left) and B (right) that imply inexpressibility of transitivity in the semijoin algebra

In this figure the edges represent the relation R. A moment's inspection reveals that the duplicator has a winning strategy in the semijoin game $G(A, \langle\rangle; B, \langle\rangle)$. For the sake of completeness we give here the formal strategy. We do this by using the following bijections from tuple space T_A to T_B.

$f : T_A \to T_B$	
$a \mapsto g$	$ab \mapsto gh$
$b \mapsto h$	$bc \mapsto hi$
$c \mapsto i$	$de \mapsto jk$
$d \mapsto j$	$ef \mapsto kl$
$e \mapsto k$	$ac \mapsto gl$
$f \mapsto l$	$df \mapsto ji$

$g : T_A \to T_B$	
$a \mapsto j$	$ab \mapsto jk$
$b \mapsto k$	$bc \mapsto kl$
$c \mapsto l$	$de \mapsto gh$
$d \mapsto g$	$ef \mapsto hi$
$e \mapsto h$	$ac \mapsto ji$
$f \mapsto i$	$df \mapsto gl$

When the spoiler makes his first move, the duplicator has a legal answer by taking the image or pre-image of the spoiler's chosen tuple under bijection f. The duplicator now continues answering each spoiler move by applying f or f^{-1} to the chosen tuple, until:

- in configuration $(A, ac; B, gl)$ the spoiler chooses bc or kl, or
- in configuration $(A, bc; B, hi)$ the spoiler chooses ac or ji, or

- in configuration $(A, df; B, ji)$ the spoiler chooses ef or hi, or
- in configuration $(A, ef; B, kl)$ the spoiler chooses df or gl.

In either case, the duplicator answers with the tuple obtained from applying g or g^{-1} to the chosen tuple, and from then, he follows strategy function g. Following g, he switches back to strategy function f whenever:

- in configuration $(A, ac; B, ji)$ the spoiler chooses bc or hi, or
- in configuration $(A, bc; B, kl)$ the spoiler chooses ac or gl, or
- in configuration $(A, df; B, gl)$ the spoiler chooses ef or kl, or
- in configuration $(A, ef; B, hi)$ the spoiler chooses df or ji.

□

Another example of a query inexpressible in the semijoin algebra is the following:

Theorem 13. *The query $R = \pi_1(R) \times \pi_2(R)$? about a binary relation R is inexpressible in the semijoin algebra.*

Proof. In Figure 2, two databases A and B are shown where A satisfies the query and B does not. The duplicator has a winning strategy in the semijoin game $G(A, \langle\rangle; B, \langle\rangle)$. □

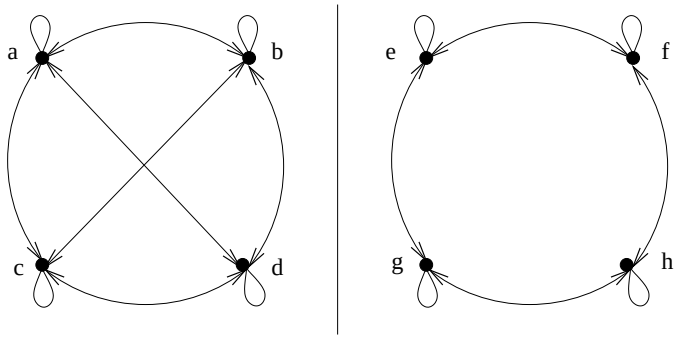

Figure 2: Databases A (left) and B (right) that imply inexpressibility of the query of Theorem 13.

6 Impact of order

In this section, we investigate the impact of order. On ordered databases (where Ω now also contains a total order on the domain), the query that asks if there

are at least k elements in a unary relation S becomes expressible as at_least(k), which is inductively defined as follows:

$$\begin{cases} \text{at\_least}(1) & := & S \\ \text{at\_least}(k) & := & S \ltimes_{x_1<y_1} \big(\text{at\_least}(k-1)\big) \end{cases}$$

Note that this query is independent of the chosen order. This parallels the situation in first-order logic, where there also exists an order-invariant query that is expressible with but inexpressible without order ([1, Exercise 17.27] and [6, Proposition 2.5.6]).

Transitivity remains inexpressible in the semijoin algebra even on ordered databases. Consider the following databases A and B over a single binary relation R: $A(R)$ is the union of X, Y and Z, where $X = \{1, \ldots, m\} \times \{2m+1\}$, $Y = \{2m+1\} \times \{m+1, \ldots, 2m\}$, and $Z = \{1, \ldots, m\} \times \{m+1, \ldots, 2m\}$; $B(R) = A(R) - \{(\frac{m+1}{2}, m + \frac{m+1}{2})\}$. Clearly, R is transitive in A, but not in B. We have shown elsewhere [11] that when $m = 2n+1$, the duplicator has a winning strategy in the n-round semijoin game $G_n(A, \langle\rangle; B, \langle\rangle)$. By Proposition 10, transitivity is not expressible in SA with order.

References

[1] S. Abiteboul, R. Hull, and V. Vianu. *Foundations of databases*. Addison-Wesley, 1995.

[2] H. Andreka, I. Nemeti, and J. van Benthem. Modal languages and bounded fragments of predicate logic. *Journal of Philosophical Logic*, 27(3):217–274, 1998.

[3] 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.

[4] P.A. Bernstein and D.W. Chiu. Using semi-joins to solve relational queries. *Journal of the ACM*, 28(1):25–40, 1981.

[5] P.A. Bernstein and N. Goodman. Power of natural semijoins. *SIAM Journal on Computing*, 10(4):751–771, 1981.

[6] H.-D. Ebbinghaus and J. Flum. *Finite model theory*. Springer, 1999.

[7] H. Garcia-Molina, J.D. Ullman, and J. Widom. *Database systems: the complete book*. Prentice Hall, 2000.

[8] E. Grädel. On the restraining power of guards. *Journal of Symbolic Logic*, 64(4):1719–1742, 1999.

[9] E. Grädel, C. Hirsch, and M. Otto. Back and forth between guarded and modal logics. In *Proc. 15th IEEE Symp. on Logic in Computer Science*. IEEE Computer Society Press, 2000.

[10] C. Hirsch. *Guarded logics: Algorithms and bisimulation.* PhD thesis, RWTH Aachen, 2002.

[11] D. Leinders, J. Tyszkiewicz, and J. Van den Bussche. On the expressive power of semijoin queries. *Information Processing Letters*, 91(2):93–98, 2004.

[12] M. Yannakakis. Algorithms for acyclic database schemes. In *Proc. of Intl. Conf. on Very Large Data Bases*, pages 82–94. IEEE Press, 1981.