On the expressive power of semijoin queries

Dirk Leinders [a,*] Jerzy Tyszkiewicz [b,1] Jan Van den Bussche [a]

[a] *Limburgs Universitair Centrum, Universitaire Campus, 3590 Diepenbeek, Belgium*

[b] *Institute of Informatics, Warsaw University, ul. Banacha 2, 02-097 Warszawa, Poland*

Abstract

The semijoin algebra is the variant of the relational algebra obtained by replacing the join operator by the semijoin operator. We provide an Ehrenfeucht-Fraissé game, characterizing the discerning power of the semijoin algebra. This game gives a method for showing that queries are not expressible in the semijoin algebra.

Key words: database, relational algebra, semijoin, Ehrenfeucht-Fraissé game

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 [8], 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

* Corresponding author.

Email addresses: dirk.leinders@luc.ac.be (Dirk Leinders), jty@mimuw.edu.pl (Jerzy Tyszkiewicz), jan.vandenbussche@luc.ac.be (Jan Van den Bussche).

[1] 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.

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 full relational algebra obtained by replacing the join operator by the semijoin operator. We call this the semijoin algebra (SA). We will define an Ehrenfeucht-Fraissé game, that characterizes the discerning power of the semijoin algebra. Using this tool, we illustrate the borderline of expressibility of SA.

2 Preliminaries

In this section, we give a formal definition of the semijoin algebra.

From the outset, we assume a universe $\mathbb{U}$ of basic data values, over which a number of predicates or relations 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.

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 $R \in \mathbf{S}$ *belongs to SA.*
(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 $E_1 \in SA$ *has arity* n *and* X *is a subset of* $\{1, \ldots, n\}$*, then* $\pi_X(E_1)$ *belongs to SA and is of arity* $\#X$*.*
(4) If $E_1, E_2 \in SA$ *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 selection and the semijoin operator are as follows: $\sigma_{\theta_1}(E) := \{(a_1, \ldots, a_n) \in E \mid \theta_1(a_1, \ldots, a_n)$ *holds*$\}$*,* $E_1 \ltimes_{\theta_2} E_2 := \{(a_1, \ldots, a_n) \in E_1 \mid \exists (b_1, \ldots, b_m) \in E_2,\ \theta_2(a_1, \ldots, a_n, b_1, \ldots, b_m)$ *holds*$\}$*. The semantics of the other operators are well known.*

3 An Ehrenfeucht-Fraissé 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: $T_A := \bigcup_{R \in \mathbf{S}} \bigcup \left\{ \pi_{\mathrm{X}}(A(R)) \mid \mathrm{X} \subseteq \{1, \ldots, \mathrm{arity}(R)\} \right\}$, 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 $\overline{a} \in T_A$ and a tuple $\overline{b} \in T_B$. We will denote such a configuration by $(A, \overline{a}; B, \overline{b})$. The conditions for the duplicator to win the game with 0 rounds are:

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

In the game with $m \geq 1$ 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 2 (legal answer) *Suppose that at a certain moment in the semijoin game, the configuration is $(A, \overline{a}; B, \overline{b})$. If the spoiler takes a tuple $\overline{c} \in T_A$ in his next move, then the tuples $\overline{d} \in T_B$, for which the following conditions hold, are legal answers for the duplicator:*

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

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

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

We first state and prove

Proposition 3 *If the duplicator wins $G_m(A, \overline{a}; B, \overline{b})$, then for each semijoin*

expression E with $\leq m$ nested semijoins and projections, we have $\overline{a} \in E(A) \Leftrightarrow \overline{b} \in E(B)$.

PROOF. We prove this by induction on m. The base case $m = 0$ is clear. Now consider the case $m > 0$. Suppose that $\overline{a} \in E_1 \ltimes_\theta E_2$ but $\overline{b} \notin E_1 \ltimes_\theta E_2$. Then $\overline{a} \in E_1(A)$ and $\exists \overline{c} \in E_2(A) : \theta(\overline{a}, \overline{c})$, and either (*) $\overline{b} \notin E_1(B)$ or (**) $\neg \exists \overline{d} \in E_2(B) : \theta(\overline{b}, \overline{d})$. In situation (*), $\overline{a}$ and $\overline{b}$ 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 $\overline{c} \in E_2(A)$ with $\theta(\overline{a}, \overline{c})$, because each legal answer of the duplicator $\overline{d}$ has $\theta(\overline{b}, \overline{d})$ and therefore $\overline{d} \notin E_2(B)$. So, the spoiler now has a winning strategy in the game $G_{m-1}(A, \overline{c}; B, \overline{d})$. In case a projection distinguishes $\overline{a}$ and $\overline{b}$, a similar winning strategy for the spoiler exists. In case $\overline{a}$ and $\overline{b}$ 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 the text. This theorem concerns the game $G_\infty(A, \overline{a}; B, \overline{b})$, which we also abbreviate as $G(A, \overline{a}; B, \overline{b})$. We say that the duplicator wins $G(A, \overline{a}; B, \overline{b})$ 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 4 *The duplicator wins $G(A, \overline{a}; B, \overline{b})$ if and only if for each semijoin expression E, we have $\overline{a} \in E(A) \Leftrightarrow \overline{b} \in E(B)$.*

PROOF. The 'only if' direction of the proof follows directly from Proposition 3, because if the duplicator wins $G(A, \overline{a}; B, \overline{b})$, he wins $G_m(A, \overline{a}; B, \overline{b})$ for every $m \geq 0$. So, $\overline{a}$ and $\overline{b}$ are indistinguishable through all semijoin expressions. For the 'if' direction, it is sufficient to prove that if the duplicator loses, $\overline{a}$ and $\overline{b}$ are distinguishable. We therefore construct, by induction, a semijoin expression $E^r_{\overline{a}}$ such that (i) $\overline{a} \in E^r_{\overline{a}}(A)$, and (ii) $\overline{b} \in E^r_{\overline{a}}(B)$ iff the duplicator wins $G_r(A, \overline{a}; B, \overline{b})$. We define $E^0_{\overline{a}}$ as

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

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

Table 1
Queries delineating the expressive power of the semijoin algebra.

Expressible	Inexpressible
$R \times S \ \cap \ T$	$R \times S \subseteq T$
$T \subseteq R \times S$	$T = R \times S$
	$R \circ S \ \cap \ T$
	$T \subseteq R \circ S$
	$R \circ S \subseteq T$
$\exists$ path of length k	
$\exists$ simple path of length k $(k \leq 2)$	$\exists$ simple path of length k $(k \geq 3)$
$\exists$ cycle of length k $(k \leq 2)$	$\exists$ cycle of length k $(k \geq 3)$
	$\exists \geq k$ elements $(k \geq 3)$

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

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

In this expression, $\theta_{\overline{a},\overline{c}}$ is the atomic type of $\overline{a}$ and $\overline{c}$ 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 $\overline{a}$, and one with arity j. The notation E^{compl}, for an expression of arity k, is a shorthand for

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

4 The expressive power of the semijoin algebra

In this section, we present some queries that delineate the expressive power of the semijoin algebra. They are summarized in Table 1. The operation $R \circ S$ for binary relations R and S is a shorthand for $\pi_{1,4}\Big(\sigma_{2=3}(R \times S)\Big)$.

We now discuss the results presented in the table. The semijoin algebra lacks the cartesian product operator, but nevertheless one can check if $T \subseteq R \times S$. Indeed, $T \subseteq R \times S$ iff $T - (T \ \cap \ R \times S) = \emptyset$, and $T \ \cap \ R \times S = (T \ltimes_{x_1=y_1 \wedge x_2=y_2} R) \ltimes_{x_3=y_1 \wedge x_4=y_2} S$. Conversely, it is impossible to check if $T \supseteq R \times S$. In Figure 1, two databases A and B are shown that are indistinguishable through semijoin expressions because the duplicator has an obvious winning strategy. But A

$A(R)$
a
b

$A(S)$
1
2

$A(T)$	
a	1
a	2
b	1
b	2

$B(R)$
a
b
c

$B(S)$
1
2
3

$B(T)$	
a	1
a	2
b	2
b	3
c	1
c	3

Fig. 1. In A, $T = R \times S$, but not in B.

$A(R)$	
1	a
3	b

$A(S)$	
a	2
b	4

$A(T)$	
1	2
3	4

$B(R)$	
1	a
3	b

$B(S)$	
b	2
a	4

$B(T)$	
1	2
3	4

Fig. 2. In A, $T = R \circ S$, but in B neither $T \subseteq R \circ S$ nor $T \supseteq R \circ S$.

satisfies $T \supseteq R \times S$ and B does not. The same databases actually show that it is impossible to check if $T = R \times S$.

Although one can check in SA if a relation is contained in a *cartesian product*, it is impossible to check if a relation is contained in or subsumed by a *join*. Using our semijoin game, one can show that databases A and B in Figure 2 satisfy the same semijoin expressions. But A satisfies $T = R \circ S$, while B satisfies neither $T \subseteq R \circ S$ nor $T \supseteq R \circ S$. Note that a binary relation R is transitive if and only if $R \circ R \subseteq R$. This is a special case of $R \circ S \subseteq T$; yet, a similar argument shows that transitivity is also inexpressible in the semijoin algebra.

The existence of a path of length k can be checked with the following inductively defined semijoin expression:

$$\begin{cases} \text{path}(1) := R \\ \text{path}(k) := R \ltimes_{x_2=y_1} \Big(\text{path}(k-1)\Big) \end{cases}$$

Problems arise when we require the path to be simple. Let $D^{(k)}$ be the structure $\{(1,2),(2,3),\ldots,(k-1,k),(k,1)\}$ over the schema $\mathbf{S}$ containing a single edge relation R. Then, the duplicator has a winning strategy in the infinite game played on $D^{(k)}$ and $D^{(k+1)}$ where $k \geq 4$. To see this, note that only three types of moves are possible here: next tuple (change only first component of

pebbled tuple), previous tuple (change only second component) and other tuple (change both components). The duplicator can answer every type of move of the spoiler. But $D^{(k+1)}$ contains a simple path of length k and $D^{(k)}$ does not. For $k = 3$, note that $D^{(3)}$ and $D^{(4)}$ *are* distinguishable. Nevertheless, existence of a simple path of length 3 is still inexpressible because $D^{(4)}$ is indistinguishable from the structure consisting of two disjoint copies of $D^{(3)}$. For $k = 2$, the existence of a path of length 2 is expressible as $R \ltimes_{x_2=y_1 \wedge x_2 \neq x_1 \wedge y_2 \neq x_2} R$.

Another property that is inexpressible in SA is the existence of a cycle of length k. For $k \geq 4$, the inexpressibility result follows because $D^{(k)}$ contains a cycle of length k and $D^{(k+1)}$ does not. For $k = 3$, that the structure consisting of two disjoint copies of $D^{(3)}$ contains a cycle of length 3, but $D^{(4)}$ does not.

A last example of a query that is inexpressible in SA is the query that asks if there are at least k elements in a unary relation S, where $k \geq 3$. This property is inexpressible because the duplicator has a winning strategy in the infinite game played on two relations, one with 2 and one with k distinct elements.

5 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 order. This is very interesting because in first-order logic, 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]).

Some inexpressible queries presented in Section 4 remain inexpressible on ordered databases. An example is the query $R \times S \subseteq T$. Indeed, consider the following databases A and B: $A(R) = B(R) = \{1, 2, \ldots, m\}$, $A(S) = B(S) = \{m+1, m+2, \ldots, 2m\}$, $A(T) = A(R) \times A(S)$ and $B(T) = A(T) - \{(\frac{m+1}{2}, m + \frac{m+1}{2})\}$. We will show that when $m = 2n + 1$, the duplicator has a winning strategy in the n-round semijoin game $G_n(A, \langle\rangle; B, \langle\rangle)$ with $\Omega = \{=, <\}$. From Lemma 3, it then follows that the query $R \times S \subseteq T$ is inexpressible in SA. The duplicator's winning strategy consists of playing exact match until the spoiler chooses $\overline{c}$ to be the special tuple $(\frac{m+1}{2}, m + \frac{m+1}{2})$ in A. In that case we must distinguish five possibilities for the previous tuple $\overline{a}$: (1) $a_1 = \frac{m+3}{2}$, (2)

$a_1 = \frac{m-1}{2}$, (3) $a_1 = \frac{m+1}{2}$ and $a_2 = m + \frac{m+3}{2}$, (4) $a_1 = \frac{m+1}{2}$ and $a_2 = m + \frac{m-1}{2}$ and (5) all other cases. The duplicator chooses $\overline{d}$ equal to $(\frac{m-1}{2}, m + \frac{m+1}{2})$ in case 1, $(\frac{m+3}{2}, m + \frac{m+1}{2})$ in case 2, $(\frac{m+1}{2}, m + \frac{m-1}{2})$ in case 3, $(\frac{m+1}{2}, m + \frac{m+3}{2})$ in case 4, and $(\frac{m-1}{2}, m + \frac{m+1}{2})$ in case 5. Let us assume case 1 applies; cases 2 to 5 are analogous. Then, there are two possibilities. First, if the spoiler chooses a value $c_1 \neq a_1 - 1$ or if he chooses a value $d_1 \neq b_1 + 1$ in some next round, the duplicator can play exact match and the game starts over. Second, if the spoiler chooses in each next round $c_1 = a_1 - 1$ or $d_1 = b_1 + 1$, the duplicator answers $d_1 = b_1 - 1$ or $c_1 = a_1 + 1$, respectively. The duplicator can follow this strategy for at least $\frac{m-3}{2} = n - 1$ rounds. Counting from the round where the spoiler chose the special tuple, we thus see that the duplicator wins the game $G_n(A, \langle\rangle; B, \langle\rangle)$.

Exactly the same argument shows that also the query $R \times S = T$ is inexpressible in SA with order.

Another query from Table 1 that remains inexpressible in SA with order is $R \circ S \subseteq T$. Therefore, consider the following databases A and B: $A(R) = B(R) = \{1, \ldots, m\} \times \{2m+1\}$, $A(S) = B(S) = \{2m+1\} \times \{m+1, \ldots, 2m\}$, $A(T) = A(R) \circ A(S) = \{1, \ldots, m\} \times \{m+1, \ldots, 2m\}$ and $B(T) = B(R) \circ B(S) - \{(\frac{m+1}{2}, m + \frac{m+1}{2})\}$. A similar argument as in the previous paragraph shows that when $m = 2n+1$, the duplicator wins $G_n(A, \langle\rangle; B, \langle\rangle)$. Again, this also shows that $R \circ S = T$ is inexpressible in SA with order.

For the remaining SA-inexpressible queries in Table 1, the question whether they become expressible in SA with order remains open.

6 Concluding remarks

Interestingly, there is a fragment of first-order logic very similar to the semijoin algebra: it is the so called "guarded fragment"(GF) [2], 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.

When we only allow conjunctions of equalities to be used in the semijoin conditions, SA is subsumed by GF, and conversely, every GF sentence is expressible in SA.

When negations of equalities are allowed in semijoin conditions, however, SA is no longer subsumed by GF. 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 as $S \ltimes_{x_1 \neq y_1} S$, but it is not expressible in GF. Proofs of the claims presented in this section will be presented in a separate paper.

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