Relational completeness of query languages for annotated databases

Floris Geerts [a] Jan Van den Bussche [b]

[a] *University of Edinburgh*

[b] *Hasselt University/Transnational University Limburg*

Abstract

Annotated relational databases can be queried either by simply making the annotations explicitly available along the ordinary data, or by adapting the standard query operators so that they have an implicit effect also on the annotations. We compare the expressive power of these two approaches. As a formal model for the implicit approach we propose the color algebra, an adaptation of the relational algebra to deal with the annotations. We show that the color algebra is relationally complete: it is equivalent to the relational algebra on the explicit annotations. Our result extends a similar completeness result established for the query algebra of the MONDRIAN annotation system, from unions of conjunctive queries to the full relational algebra. We also show that the color algebra is non-redundant: no operator can be expressed in terms of the other operators. We also present a generalization of the color algebra that is relationally complete in the presence of built-in predicates on the annotations.

Key words: Annotated relational databases, expressive power, query languages

1 Introduction

Recently, much attention has been paid to annotated databases [18,6,3,8,12,11,9,5,13]. In querying annotated databases, there are two distinct approaches:

(1) In *annotation propagation* [18,6,10,3,9,5,13], queries are directed primarily at the ordinary data, not the annotations: the latter are merely propagated to the query results. For example, when joining two relations, the annotations of two joined tuples would become annotations of the new joint tuple.
(2) In *annotation querying* [12,11,8], queries can be directed to the annotations as well as to the ordinary data. For example, when joining two

relations, two tuples might be considered joinable only if they have a common annotation. Such join queries are outside the scope of annotation propagation.

Of course, these two approaches are not competing; it is simply that in some applications we want annotation propagation, while in other applications we want to really query on the basis of annotations. As a matter of fact, annotation propagation can be precisely characterized as that part of annotation querying that is invariant under arbitrary re-annotations, even those re-annotations that replace two different annotations by the same one [5].

In the present paper, we are concerned with full annotation querying, and here one can again distinguish two approaches: *explicit* and *implicit*.

(1) In explicit querying, we simply make the annotations explicitly available along with the ordinary data; any standard query language can then be used to query the database. For example, suppose we want to join annotated relations $R(A, B)$ and $S(A, C)$ not only on their common A-attribute, but also on common annotations. Then we simply model R as a relation $R(A, B, N)$, where N is an extra column holding the annotations, and likewise model S as $S(A, C, N)$, and write in SQL:

```
select R.*, S.*
from R, S
where R.A=S.A and R.N=S.N
```

A similar feature is provided by the ANNOT operator of the pSQL language in DBNotes [8], where we would write:

```
select R.*, S.*
from R, ANNOT(R) N1, S, ANNOT(S) N2
where R.A=S.A and N1=N2
```

(2) In implicit querying, which is more in the spirit of annotation propagation, annotations are not explicitly addressed in query formulations. Rather, the standard query operators are adapted so that they have an effect not only on the ordinary data but also on the annotations. For example, in the query algebra of MONDRIAN [12], one would write the above join query as

$$\mu\, \sigma_{R.A=S.A}(R \times S),$$

where

- the Cartesian product operator $\times$ is adapted so as to keep, for each joint tuple $r \cup s \in R \times S$ with $r \in R$ and $s \in S$, two sets of annotations: the annotations that r already had in R, and the annotations that s already had in S;
- the selection operator σ simply propagates these sets of annotations;
- the new merge operator μ intersects the two sets of annotations.

A natural question now arises as to the relative expressiveness of explicit versus implicit annotation querying. This question was already addressed for the MONDRIAN query algebra, which has been shown to be equivalent to the positive relational algebra on explicit annotations [12]. In the present paper, we continue this investigation and extend it to the full relational algebra (as opposed to its positive fragment, which does not have the difference operator). Recall that the relational algebra is much more powerful and complicated than its positive fragment [1]. For instance, in the positive algebra only unions of conjunctive queries can be expressed, and containment and equivalence of queries is decidable; in the full relational algebra, all first-order logic definable queries can be expressed, and equivalence (let alone containment) is undecidable.

We will introduce *color relations* as a simple but general abstraction of annotated databases. A color relation is a standard database relation, where additionally every tuple is annotated by some set of "colors". Moreover, we will introduce the *color algebra (CA)*, an adaptation of the relational algebra to deal with color relations. CA is inspired by, but different from, the MONDRIAN query algebra. The operators of CA always produce color relations as output; in particular, in CA one cannot compute intermediate results that explicitly relate the colors of different tuples (by having *multiple* color columns). Nevertheless, we will prove that CA can still express any expression of the full relational algebra on explicit annotations, as long as the latter expression starts from color relations and finally ends up in color relations (relations with a *single* color column).

We also show that the color algebra, like the relational algebra [7], is nonredundant: no operator can be expressed in terms of the other operators.

We conclude the paper by extending the equivalence between explicit versus implicit querying in the presence of *built-in predicates* on annotations. Consider again the above explicit querying example. Suppose that annotations are equipped with a linear order. We may want to join annotated relations $R(A, B, N)$ and $S(A, C, N)$ on their common A-attribute, provided that the annotation in R is less than the annotation in S. Explicitly, this query can be expressed in SQL as follows:

```
select R.*, S.*
from R, S
where R.A=S.A and R.N < S.N
```

To express such annotation queries *implicitly*, we extend the color algebra with a *generalized color join* and show that the resulting generalized color algebra again is as powerful as the full relational algebra with built-in predicates on explicit annotations.

Our results, while answering natural questions, are mainly of theoretical interest. Yet, good theoretical underpinnings of new database management features, such as annotations, are important. We feel that our proposed formalisms are elegant and we hope they can serve as a guide to the understanding and design of annotation query languages.

2 Color relations

We assume as given an infinite set of *attributes*, an infinite set $\mathbb{D}$ of *data values*, and an infinite set $\mathbb{C}$ of *colors*. The sets $\mathbb{D}$ and $\mathbb{C}$ are disjoint; colors serve as an abstraction for annotation values.

(1) A *relation schema* is a finite set R of attributes.
(2) A *tuple* over R is a mapping $t\colon R \to \mathbb{D}$.
(3) A *relation* over R is a finite set of tuples over R.
(4) A *coloring* of a relation r is a subset r' of $r \times \mathbb{C}$, i.e., a set of tuple–color pairs where the tuples come from r, such that every tuple of r appears in r', i.e., every tuple of r gets at least one color.
(5) We call r the *underlying relation* of r'. We agree that whenever we denote a coloring by a primed letter, the unprimed letter stands for the underlying relation.
(6) Colorings of relations over R are also called *color relations over* R.
(7) A *database schema* $\mathcal{S}$ consists of a finite set of relation variables x, each with an associated relation schema $\mathcal{S}(x)$. (Relation variables will also be called relation names.)
(8) A *color database* D over $\mathcal{S}$ consists of a set of color relations $D(x)$, one for each relation variable x of $\mathcal{S}$, such that $D(x)$ is a color relation over $\mathcal{S}(x)$.

We can view a color relation r' alternatively as a mapping r' from r to $2^{\mathbb{C}}$, as follows:

$$r'(t) = \{c \mid (t, c) \in r'\}.$$

Note that, since every tuple gets at least one color, $r'(t)$ is never empty. For any subset $s \subseteq r$, the restriction of the mapping r' to s, which we denote by $r'|_s$, is of course a coloring of s. We will use this observation in the following section.

In our data model, we restrict attention to the coloring of entire tuples. In annotation systems such as DBNotes [3,8], not just tuples in relations can be colored, but also individual components of these tuples. We can model this by multiple color relations, one for each attribute. The system MONDRIAN [12,11] even allows the coloring of arbitrary subsets of projections of a relation. Even more generally, one can consider annotations of arbitrary combinations

of records and sets [5]. Such complex structures can always be decomposed in multiple flat relations, however, and since the focus of this paper is on expressive power, our model of color relations is sufficient.

Our model of color relations is not sufficient, however, to capture the "intensional" coloring of tuples in the result of queries [14,15]. In this approach, instead of coloring tuples of a relation, one associates colors with *views* of the relations. The querying of such color views involves the rewriting of queries over views, which is outside the scope of this paper.

3 The color algebra

We are familiar with the classical relational algebra operations on relations: union ($\cup$), difference ($-$), natural join ($\bowtie$), renaming (ρ), selection (σ), and projection (π). We now define a number of analogous operations on color relations. The result of these operations is again a color relation. Some of the (less obvious) operations are illustrated in Figure 1.

Let r' and s' be two color relations over the same relation scheme R.

Union: $r' \cup s'$ is the standard set-theoretic union. This is a coloring of $r \cup s$.

Full difference: $r' - s'$ is the standard set-theoretic difference. It is a coloring not of $r - s$, but of

$$(r - s) \cup \{t \in r \cap s \mid r'(t) \not\subseteq s'(t)\}\,.$$

For the definition of the next two operations, s' no longer needs to be over the same relation scheme as r'.

Tuple join: $r' \boxtimes s'$ equals [1]

$$\{(t_1 \cup t_2, c) \mid t_1 \cup t_2 \in r \bowtie s \text{ and } c \in r'(t_1) \cup s'(t_2)\}\,.$$

It is a coloring of $r \bowtie s$.

Full join: $r' \bowtie s'$ is defined in the same way as $r \boxtimes s$, except that now we take the intersection $r'(t_1) \cap s'(t_2)$ rather than the union. It is thus a coloring not of $r \bowtie s$, but of

$$\{t_1 \cup t_2 \in r \bowtie s \mid r'(t_1) \cap s'(t_2) \neq \emptyset\}\,.$$

[1] Note the union $t_1 \cup t_2$ of two tuples t_1 and t_2. This is well-typed since tuples are defined as mappings, and mappings formally are sets of pairs. Moreover, since $t_1 \cup t_2 \in r_1 \bowtie r_2$, the result of the union is again a tuple (mapping).

r'

A	B		
a_1	b_1	$\mapsto$	$\{\text{red}, \text{green}, \text{blue}\}$
a_2	b_2	$\mapsto$	$\{\text{blue}\}$
a_3	b_3	$\mapsto$	$\{\text{green}\}$

s'

A	B		
a_1	b_1	$\mapsto$	$\{\text{red}, \text{yellow}\}$
a_2	b_2	$\mapsto$	$\{\text{red}\}$
a_4	b_4	$\mapsto$	$\{\text{green}, \text{blue}\}$

$r' \cup s'$

A	B		
a_1	b_1	$\mapsto$	$\{\text{red}, \text{green}, \text{blue}, \text{yellow}\}$
a_2	b_2	$\mapsto$	$\{\text{blue}, \text{red}\}$
a_3	b_3	$\mapsto$	$\{\text{green}\}$
a_4	b_4	$\mapsto$	$\{\text{green}, \text{blue}\}$

$r' - s'$

A	B		
a_1	b_1	$\mapsto$	$\{\text{green}, \text{blue}\}$
a_3	b_3	$\mapsto$	$\{\text{green}\}$

$r' \boxtimes s'$

A	B		
a_1	b_1	$\mapsto$	$\{\text{red}, \text{green}, \text{blue}, \text{yellow}\}$
a_2	b_2	$\mapsto$	$\{\text{blue}, \text{red}\}$

$r' \bowtie s'$

A	B		
a_1	b_1	$\mapsto$	$\{\text{red}\}$

Fig. 1. Example of the color algebra operators union ($\cup$), full difference ($-$), tuple join ($\boxtimes$) and full join ($\bowtie$) on two color relations r' and s'.

Renaming: if $A \in R$ and B is an attribute not in R, then $\rho_{A/B}(r')$ equals

$$\{(\rho_{A/B}(t), c) \mid (t, c) \in r'\},$$

with $\rho_{A/B}(t) = t|_{R-A} \cup \{(B, t(A))\}$ the classical renaming of a tuple. It is thus a coloring of $\rho_{A/B}(r)$.

Selection: if $A, B \in R$, then $\sigma_{A=B}(r')$ equals $r'|_{\sigma_{A=B}(r')}$.

Color selection: if $k \geq 2$ is a natural number, then $\sigma_{\text{color} \geq k}(r')$ equals $r'|_u$, where

$$u = \{t \in r \mid |r'(t)| \geq k\},$$

with $|r'(t)|$ denoting the cardinality of $r'(t)$, i.e., the number of distinct colors of t in r'.

Projection: if $X \subseteq R$, then $\pi^{\text{col}}_X(r')$ equals

$$\{(t|_X, c) \mid (t, c) \in r'\}.$$

This concludes the definition of the operations of the *color algebra*, abbreviated CA. We remark that most of the operators in CA are intuitive, except, perhaps, for the color selection $\sigma_{\text{color} \geq k}$. This operator is necessary, however, to show the relational completeness of CA.

$r' \rhd s'$

A	B	
a_1	b_1	$\mapsto \{\text{red}, \text{green}, \text{blue}\}$
a_2	b_2	$\mapsto \{\text{blue}\}$

$s' \rhd r'$

A	B	
a_1	b_1	$\mapsto \{\text{red}, \text{yellow}\}$
a_2	b_2	$\mapsto \{\text{red}\}$

$r' \boxminus s'$

A	B	
a_3	b_3	$\mapsto \{\text{green}\}$

Fig. 2. Example of the derived color algebra operators $\rhd$ (Example 2) and tuple difference ($\boxminus$) (Example 4) on the two color relations r' and s' given in Fig. 1.

Example 1 *Consider the CA-expression*

$$x \bowtie \mathsf{green},$$

where green *is a color relation over the empty relation scheme. When evaluated on a color relation r' (for x) and the color relation over the empty relation scheme consisting of a single 'green'-colored (empty) tuple (for* green*), this expression returns all tuples in r' that are colored 'green'.* □

Example 2 *Let us introduce the following derived CA operator: $x \rhd y$ is an abbreviation for $x \bowtie (x \boxtimes y)$. The reader is invited to verify that $r' \rhd s'$, for color relations r' and s', equals*

$$\{(t_1 \cup t_2, c) \mid t_1 \cup t_2 \in r \bowtie s \text{ and } (t_1, c) \in r'\}.$$

Examples illustrating this operator are provided in Figure 2. The CA-expression

$$(x \rhd y) - (y \rhd x)$$

applied to color relations r' and s', returns joint tuples $t_1 \cup t_2$ from the natural join of the underlying relations r and s (with $t_1 \in r$ and $t_2 \in s$); these joint tuples are colored by the colors t_1 has in r', except for the colors t_2 has in s'. In particular, if t_1 has only colors that t_2 has too, then the joint tuple $t_1 \cup t_2$ is not returned at all, since in color relations, each tuple must have at least one color. □

Example 3 *The expression*

$$x - \sigma_{\text{color} \geq 3}(x)$$

returns all tuples in x that have at most two colors. □

Example 4 *We introduce the derived CA operator $x \boxminus y$ that is an abbreviation for $x - (y \boxtimes \pi^{\text{col}}_{\emptyset}(x))$ and which we call the* tuple difference*. Note that $r' \boxminus s'$, for color relations r' and s', equals $r'|_{r-s}$. It is thus a coloring of $r - s$. An example illustrating this operator is provided in Figure 2.* □

4 CA and the relational algebra

Let us reserve a special attribute *col* and agree that it is never used in the relation schemes of color relations. For any relation scheme R, we define the relation scheme $\bar{R} = R \cup \{col\}$. We can naturally view a color relation r over R as a relation over $\bar{R}$, as follows:

$$\{t \cup \{(col, c)\} \mid (t, c) \in r\}.$$

Conversely, any relation r over $\bar{R}$ can be viewed as a color relation as follows:

$$\{(t|_R, t(col)) \mid t \in r\}.$$

Beware that when we regard r as a *color* relation, it is a color relation over R, i.e., r's relation scheme is just R, because the color attribute is implicit in color relations. Indeed, this is exactly the main feature of the color algebra: that colors are handled automatically. When we regard r as an ordinary relation, however, it is a relation over $\bar{R}$ and the color attribute becomes explicitly visible.

Under the view of color relations as ordinary relations, we can apply classical relational algebra operations to color relations, and consider relational algebra expressions with $\bar{R}$ as result relation scheme to be producing color relations over R. It then becomes apparent that the classical relational algebra can actually simulate the color algebra. The simulation is given in Table 1. The table shows the simulation of the individual operations; the simulation of more complex expressions can be obtained using composition.

More interestingly, the converse simulation holds as well: every operation on color relations that is definable in the relational algebra is already definable in CA. More formally, to every color database schema $\mathcal{S}$ we can associate the relational database schema $\bar{\mathcal{S}}$ which has precisely the same relation variables, but when relation variable x has relation scheme R in $\mathcal{S}$, then x has relation scheme $\bar{R}$ in $\bar{\mathcal{S}}$. We will establish:

Theorem 5 *For every relational algebra expression over $\bar{\mathcal{S}}$ whose result relation scheme is of the form $\bar{R}$ for some relation scheme R, there exists an equivalent CA-expression over $\mathcal{S}$.*

In proving this theorem, one cannot hope for a simple bottom-up syntax-directed translation from relational algebra to CA, such as we had with Table 1 for the other direction. For instance, consider in that table the line for $\sigma_{\text{color} \geq k}(x)$, but now read from right to left. More generally, the challenge is how to deal with relational algebra expressions that produce relations as intermediate results that explicitly relate colors from different tuples in the database. We will give the proof of Theorem 5 in Section 6.

Table 1
Simulation of CA by relational algebra (RA). In the cases of $x \boxtimes y$ and $\sigma_{\text{color}\geq k}(x)$, the letter R (S) refers to the relation scheme of the color relation x (y). Moreover, in the simulation of $\sigma_{\text{color}\geq k}(x)$, the auxiliary attributes col_i are chosen such that they do not appear in R.

CA	$\mapsto$	RA
$x \cup y$	$\mapsto$	$x \cup y$
$x - y$	$\mapsto$	$x - y$
$x \boxtimes y$	$\mapsto$	$(x \bowtie \pi_S(y)) \cup (\pi_R(x) \bowtie y)$
$x \bowtie y$	$\mapsto$	$x \bowtie y$
$\rho_{A/B}(x)$	$\mapsto$	$\rho_{A/B}(x)$
$\sigma_{A=B}(x)$	$\mapsto$	$\sigma_{A=B}(x)$
$\sigma_{\text{color}\geq k}(x)$	$\mapsto$	$\pi_{\bar{R}}\sigma_{\bigwedge_{i\neq j} col_i \neq col_j}(\rho_{col/col_1}(x) \bowtie \cdots \bowtie \rho_{col/col_k}(x))$
$\pi^{\text{col}}_X(x)$	$\mapsto$	$\pi_{X\cup\{col\}}(x)$

5 Nonredundancy of CA

In this section we show that no operator of CA can be expressed in terms of the other operators:

Proposition 6 *The color algebra CA is nonredudant.*

PROOF. We first show that each operator $\mathsf{op} \in \{\cup, -, \rho_{A/B}, \sigma_{A=B}, \pi^{\text{col}}_X\}$ is non-redundant by reduction to the well-known fact that their classical relational counterparts are non-redundant [7]. Denote by CA^{op} the color algebra from which op is removed. Denote the classical relational algebra by RA. Assume, for the sake of contradiction, that op can be expressed by means of an expression e_{op} in CA^{op}. In particular, e_{op} expresses op on *monochromatic* databases, i.e., color databases that consists of color relations in which all tuples are assigned the same color. Given the following observations:

(1) Monochromatic database simulate classical relational databases;
(2) The simulation of the color algebra by RA given in Table 1;
(3) $\sigma_{\text{color}\geq k}$ on monochromatic databases is always empty;
(4) $\boxtimes$ on monochromatic databases amounts to $\bowtie$;

we can conclude that redundancy of op in CA would imply redundancy of op in RA, which we know is false.

Note that this reduction argument does not work for the operator $\bowtie$, because $\bowtie$ is used in the simulation of $\boxtimes$.

It remains to show the non-redundancy of $\mathsf{op} \in \{\boxtimes, \bowtie, \sigma_{\text{color} \geq k}\}$. For each of these three operators op, we establish a characteristic property P_{op} that is satisfied by every expression in CA^{op} but not by op itself. From this, the non-redundancy of these operators is immediate.

Tuple join ($\boxtimes$) We claim that the following property $\mathsf{P}_{\boxtimes}$ holds for any expression in $\text{CA}^{\boxtimes}$: Let R and S be the relation schemes $\{A\}$ and $\{B\}$, respectively. Let r' be the color relation over R that consists of the single tuple (a, red) and let s' be the color relation over S consisting of the single tuple (b, blue), such that $a \neq b$ and where 'red' and 'blue' denote two distinct colors. Then, for each $e \in \text{CA}^{\boxtimes}$, *each* tuple in the result of evaluating e on r' and s' is either (i) the empty tuple (possibly colored with both red and blue); (ii) a tuple consisting entirely of a's (and colored with red only); or (iii) a tuple consisting entirely of b's (and colored with blue only).

The validity of the claim follows by a straightforward induction on the structure of expressions in $\text{CA}^{\boxtimes}$. The property is readily seen to hold for $e = e_1 \cup e_2$, $e = e_1 - e_2$, $e = \rho_{A/B}(e_1)$, $e = \sigma_{A=B}(e_1)$ and $e = \pi^{\text{col}}_X(e_1)$. For $e = e_1 \bowtie e_2$, suppose that $(t_1 \cup t_2, c)$ is in the result of the evaluation of e on r' and s'. Clearly, if a (resp. b) appears in both t_1 and t_2, then by induction t_1 and t_2 entirely consist out of a's (resp. b's) and are solely colored with red (resp. blue). Hence, also $t_1 \cup t_2$ consists of a's (resp. b's) only and $c = \text{red}$ (resp. $c = \text{blue}$). If a (resp. b) only appears in t_1, then t_2 necessarily needs to be the empty tuple that is colored with (at least) red. Indeed, otherwise t_2 would consist entirely out of b's (resp. a's) and would be colored with blue (resp. red). As a result, t_1 and t_2 are distinctly colored and can therefore not be joined by $\bowtie$. The case that a (resp. b) only appears in t_2 can be dealt with similarly. Finally, if both t_1 and t_2 are empty tuples, then $t_1 \cup t_2$ is the empty tuple possibly colored with both red and blue. Hence, all cases lead to a joint tuple that satisfies property $\mathsf{P}_{\boxtimes}$.

In contrast, $r' \boxtimes s' = \{(a, b, \text{red}), (a, b, \text{blue})\}$ which does not satisfy satisfy $\mathsf{P}_{\boxtimes}$.

Full join ($\bowtie$) Consider the color relation $r' = \{(a, \text{red}), (a, \text{blue})\}$ over the relation scheme $\{A\}$, and $s' = \{(\text{blue})\}$ over the empty relation scheme, where a is some value and 'red' and 'blue' are distinct colors. We claim the following property $\mathsf{P}_{\bowtie}$ for any expression e of $\text{CA}^{\bowtie}$ and any nonempty tuple t in the relation underlying $e(r', s')$: *both* $(t, \textit{red})$ *and* $(t, \textit{blue})$ *belong to* $e(r', s')$. Note that the property clearly holds for r' and s'. We prove the claim by induction on the structure of e. The only case that is not immediately clear is $e = e_1 - e_2$. Let $r'_1 = e_1(r', s')$ and $r'_2 = e_2(r', s')$ and let t be a nonempty tuple in the relation underlying $r'_1 - r'_2$. Since $t \in r_1$, by induction both (t, red) and (t, blue) are in

r'_1. Also, t cannot be in r_2, since otherwise both (t, red) and (t, blue) would be in r'_2 and thus t would not be in the relation underlying $r'_1 - r'_2$. Hence, both (t, red) and (t, blue) are in $r'_1 - r'_2$, as desired.

In contrast, $r' \bowtie s' = \{(a, \text{blue})\}$ does not satisfy property $\mathsf{P}_{\bowtie}$.

Color selection ($\sigma_{\text{color}\geq k}$) We claim that the following property $\mathsf{P}_{\sigma_{\text{color}\geq k}}$ holds for any expression in $\text{CA}^{\sigma_{\text{color}\geq k}}$: Let R be the empty relation schema and let r'_k be the color relation over R consisting of the empty tuple colored with k distinct colors. Then for each $e \in \text{CA}^{\sigma_{\text{color}\geq k}}$ we have that (i) $e(r'_k) = \emptyset$ iff $e(r'_{k-1}) = \emptyset$; and (ii) $e(r'_k) = r'_k$ iff $e(r'_{k-1}) = r'_{k-1}$; and (iii) no other possible cases than (i) and (ii) exist.

We verify that any expression e in $\text{CA}^{\sigma_{\text{color}\geq k}}$ satisfies $\mathsf{P}_{\sigma_{\text{color}\geq k}}$ by induction on the structure of e. Suppose that $e = e_1 \cup e_2$. From the induction hypothesis (applied to e_1 and e_2) we obtain the following possible outcomes for $e(r'_k)$ and $e(r'_{k-1})$:

$e(r'_k)$	$e_1(r'_k) \cup e_2(r'_k)$		$e_1(r'_{k-1}) \cup e_2(r'_{k-1})$	$e(r'_{k-1})$
r'_k	$r'_k \cup r'_k$	$\Leftrightarrow$	$r'_{k-1} \cup r'_{k-1}$	r'_{k-1}
r'_k	$r'_k \cup \emptyset$	$\Leftrightarrow$	$r'_{k-1} \cup \emptyset$	r'_{k-1}
r'_k	$\emptyset \cup r'_k$	$\Leftrightarrow$	$\emptyset \cup r'_{k-1}$	r'_{k-1}
$\emptyset$	$\emptyset \cup \emptyset$	$\Leftrightarrow$	$\emptyset \cup \emptyset$	$\emptyset$

The case $e = e_1 - e_2$ is analogous. For $e = e_1 \boxtimes e_2$, we observe that e is equivalent to $e_1 \cup e_2$ when working on nullary color relations, i.e., on color relations over the empty relation schema. The case that $e = e_1 \bowtie e_2$ is equivalent to $e = e_1 \cap e_2$ which can be dealt with in an analogous way as $\cup$. We observe that renaming, selection and projection do not have any effect on nullary color relations and therefore can be omitted from this case analysis. Therefore, the last remaining case is $e = \sigma_{\text{color}\geq \ell}(e_1)$. We distinguish between the case that (i) $\ell < k$ and (ii) $\ell > k$. By the induction hypothesis (applied on e_1) we obtain the possible results for $e(r'_k)$ and $e(r'_{k-1})$ as shown in the table on the left for case (i) and on the right for case (ii).

$e(r'_k)$	$e_1(r'_k)$		$e_1(r'_{k-1})$	$e(r'_{k-1})$
r'_k	r'_k	$\Leftrightarrow$	r'_{k-1}	r'_{k-1}
$\emptyset$	$\emptyset$	$\Leftrightarrow$	$\emptyset$	$\emptyset$

$e(r'_k)$	$e_1(r'_k)$		$e_1(r'_{k-1})$	$e(r'_{k-1})$
$\emptyset$	r'_k	$\Leftrightarrow$	r'_{k-1}	$\emptyset$
$\emptyset$	$\emptyset$	$\Leftrightarrow$	$\emptyset$	$\emptyset$

In contrast, $\sigma_{\text{color}\geq k}(r'_k) = r'_k$ while $\sigma_{\text{color}\geq k}(r'_{k-1}) = \emptyset$. Hence, $\sigma_{\text{color}\geq k}$ does not satisfy property $\mathsf{P}_{\sigma_{\text{color}\geq k}}$. $\square$

Note the use of the empty relation scheme in the proof. This is a feature of the proof, in that non-redundancy involving the empty relation scheme implies non-redundancy involving non-empty relation schemes. An interesting question that we leave open is whether the color algebra is also non-redundant if we are only interested in the expression of yes/no queries.

6 Simulation of the relational algebra by the color algebra

In this section, we prove Theorem 5. It is actually sufficient to do this for a restricted fragment of the relational algebra, which we call the color-typed relational algebra, denoted by RA^{c}. In order to define this fragment, we must first go from our one special color attribute *col* to an infinite set $\mathcal{C}$ of color attributes, and agree that these are, like *col*, never used in relation schemes of color relations. Of course we put $col \in \mathcal{C}$. The color-typed restriction now only lies in a condition imposed on selections and renamings. Specifically, if e is an expression, then $\sigma_{A=B}(e)$ and $\rho_{A/B}(e)$ are only allowed if either A and B are both color attributes, or are both not color attributes. Expressions of the form $e_1 \cup e_2$, $e_1 - e_2$, $e_1 \bowtie e_2$, or $\pi_X(e)$ can be constructed just like in the classical relational algebra.

Of course, every RA^{c} expression is a finite expression and uses only finitely many of the color attributes, but there is no fixed bound on this number over all possible expressions. Note that something similar happens in the general relational algebra when used to query color relations. Indeed, such expressions can perform arbitrary renamings on the *col* attribute.

A result on the first-order completeness of many-sorted logic [17] implies that every relational algebra expression over a database schema $\bar{\mathcal{S}}$ with result relation scheme of the form $\bar{R}$ can be expressed in RA^{c}. (We point out that this depends crucially on the disjointness of the universes $\mathbb{D}$ of data values and $\mathbb{C}$ of colors.) So, we indeed only have to prove Theorem 5 for RA^{c}.

Our proof uses the following technical notions:

Let R be a relation scheme.

(1) An *R-parameterized monadic database schema* is a relational database schema where every relation name has the same relation scheme $\bar{R}$. (Equivalently, it can be viewed as a color database schema where every relation name has the same relation scheme R.)

(2) An RA^{c}-expression f over an R-parameterized monadic database schema is called *R-uniform* if it satisfies the following:
   - f uses only renamings $\rho_{A/B}$ and selections $\sigma_{A=B}$ where A and B are

color attributes;
- all projections π_X appearing in f satisfy $R \subseteq X$.

The intuition is that an R-uniform expression does not explicitly work with the attributes in R; these attributes are merely dragged along as parameters.

We now show that CA can simulate R-uniform RA^{c}, in the following sense:

Lemma 7 *Let f be an R-uniform RA^{c}-expression over the R-parameterized monadic database schema $\mathcal{S}$. Let S be the result relation scheme of f.*

- *If $S \cap \mathcal{C} = \emptyset$, i.e., $S = R$, then there exists a CA-expression $\mathrm{sim}(f)$ such that $f(D)$ equals the relation underlying $\mathrm{sim}(f)(D)$, for each color database D over $\mathcal{S}$.*
- *If $S \cap \mathcal{C} \neq \emptyset$, then for each equivalence relation E on $S \cap \mathcal{C}$, there exists a finite set $\mathrm{sim}_E(f)$ of mappings from $S \cap \mathcal{C}$ to CA, such that $f(D)$ equals*

$$\bigcup_E \bigcup_{\tau \in \mathrm{sim}_E(f)} \sigma_{\bigwedge_{(col', col'') \in E} col' = col''} \sigma_{\bigwedge_{(col', col'') \notin E} col' \neq col''} \bowtie_{col' \in S \cap \mathcal{C}} \rho_{col/col'}(\tau(col')(D))$$

PROOF. Assume that $\mathcal{S}$ consists of the relation names $z_1, \ldots, z_n$. We begin by refining the classical correspondence between the relational algebra and the relational calculus (first-order logic, FO) to R-uniform RA^{c}. The corresponding fragment of FO, which we denote by $\mathrm{FO}^{\mathrm{c}}_R$, is obtained as follows. Let $R = \{A_1, \ldots, A_m\}$. We use the A_j's, plus all color attributes, as first-order variables. The allowed atomic formulas are of two forms:

(1) $z_i(A_1, \ldots, A_m, col')$ with $col' \in \mathcal{C}$. We abbreviate such formulas by $z_i(R, col')$.
(2) $col' = col''$ with $col', col'' \in \mathcal{C}$.

The only variables that can be quantified are color attributes. It is then readily seen that R-uniform RA^{c} corresponds to $\mathrm{FO}^{\mathrm{c}}_R$ under the active-domain semantics, with the understanding that, when evaluating a formula in a database D, the tuple of free variables $A_1, \ldots, A_m$ is only instantiated by R-tuples that actually appear in D.

We next apply the well-known quantifier elimination method for monadic first-order logic to $\mathrm{FO}^{\mathrm{c}}_R$ [2,4]. Concretely, this gives us that every $\mathrm{FO}^{\mathrm{c}}_R$ formula can be written without quantifiers if we additionally allow predicates of the form $|z_\alpha(R)| \geq \ell$ in formulas, where $\ell \geq 1$ is a natural number, and α is a nonempty subset of $\{1, \ldots, n\}$. The meaning of such a predicate, for a given tuple t over

R, is that $|z_\alpha(t)| \geq \ell$, where $z_\alpha(t)$ equals

$$\{t' \in \bigcup_i z_i \mid t'|_R = t \text{ and } \bigwedge_{i\in\alpha} t' \in z_i \text{ and } \bigwedge_{i\in\hat{\alpha}} t' \notin z_i\},$$

where $\hat{\alpha}$ abbreviates $\{1,\ldots,n\} - \alpha$.

Putting the quantifier-free formula in disjunctive normal form, and simplifying each conjunction, we obtain a disjunction of conjunctions of factors of the following possible forms:

- If $S \cap \mathcal{C} = \emptyset$, then each factor of the conjunction is of one of the following three forms: (i) $|z_\alpha(R)| \geq 1$: this can be expressed in CA by $\bowtie_{i\in\alpha} z_i - \bigcup_{i\in\hat{\alpha}} z_i$; (ii) $|z_\alpha(R)| \geq \ell$ with $\ell \geq 2$: this can be expressed in CA by $\sigma_{\text{color}\geq\ell}(|z_\alpha(R)| \geq 1)$; and (iii) $\neg(|z_\alpha(R)| \geq \ell)$: this can be expressed in CA by $\bigcup_i z_i \boxminus (|z_\alpha(R)| \geq \ell)$. Recall that $\boxminus$ is the tuple difference introduced in Example 4.
- If $S \cap \mathcal{C} \neq \emptyset$, then factors may additionally be of the following possible forms: (iv) $z_i(R, col')$ for some color attribute col': this can be expressed in CA by z_i; (v) $\neg z_i(R, col')$: this can be expressed in CA by $\bigcup_j z_j - z_i$; and (vi) equalities and inequalities among color attributes.

Without loss of generality, we may assume that in each conjunction γ, the set of equalities and inequalities among color attributes is maximally consistent, involving all color attributes in $S \cap \mathcal{C}$. Such a maximally consistent set gives rise to an equivalence relation E_γ on the color attributes.

We now construct, for each conjunction γ, the following mapping τ from $S \cap \mathcal{C}$ to CA and put it in $\text{sim}_{E_\gamma}(f)$. For each color attribute col', we take the CA-expressions for all factors of types (i)–(iii) above, together with the expression $\bigcup_{z\in\mathcal{S}} \pi_\emptyset(z)$, and conjoin them all using $\boxtimes$. Observe that the tuple join with $\bigcup_{z\in\mathcal{S}} \pi_\emptyset(z)$ assigns all tuples the same set of colors, i.e., all colors that appear in any of the relations in $\mathcal{S}$. In order to obtain the correct set of colors, we further take the CA-expressions for all factors of types (iv)–(v) that concern the particular color attribute col', conjoining these with each other and with the previous part using $\bowtie$. The resulting CA-expression then equals $\tau(col')$. $\square$

We illustrate Lemma 7 with the following example:

Example 8 *Let $R = \{A\}$ and $\mathcal{S} = \{z_1, z_2\}$ (hence $n = 2$). Furthermore, let f be the R-uniform* RA^c*-expression over $\mathcal{S}$:*

$$\pi_{A,col}\left(\sigma_{col\neq col'}(z_1 \bowtie \rho_{col/col'}(z_1)\right) \bowtie \rho_{col/col''}(z_2),$$

with result relation schema $S = \{A, col, col''\}$. We now closely follow the proof of Lemma 7 to obtain a simulation of f by the color algebra. We first translate

f into the calculus FO^c_R *resulting in*

$$\exists col'\,(z_1(A, col) \wedge z_1(A, col') \wedge col \neq col') \wedge z_2(A, col'').$$

This expression, after the quantifier is eliminated, is equivalent to the disjunction $\gamma_1 \vee \gamma_2 \vee \gamma_3$*, where:*

$$\begin{aligned}
\gamma_1 &:= z_1(A, col) \wedge \Big(|z_{\{1\}}(A)| \geqslant 2\Big) \wedge z_2(A, col'')\\
\gamma_2 &:= z_1(A, col) \wedge \Big(|z_{\{1\}}(A)| \geqslant 1 \wedge |z_{\{1,2\}}(A)| \geqslant 1\Big) \wedge z_2(A, col'')\\
\gamma_3 &:= z_1(A, col) \wedge \Big(|z_{\{1,2\}}(A)| \geqslant 2\Big) \wedge z_2(A, col'').
\end{aligned}$$

In the proof of Lemma 7, we assume that in each conjunction γ_i*, the set of equalities and inequalities among color attributes in* $S \cap \mathcal{C} = \{col, col''\}$ *is maximally consistent. Therefore, we complete the conjunctions* γ_i*, for* $i = 1, 2, 3$*, as follows. Observe that* $\gamma_i = (\gamma_i \wedge col = col'') \vee (\gamma_i \wedge col \neq col'')$*. Hence, we obtain an equivalent set of expressions* $\gamma'_1 = \gamma_1 \wedge col = col''$*,* $\gamma'_2 = \gamma_1 \wedge col \neq col''$*,* $\gamma'_3 = \gamma_2 \wedge col = col''$*,* $\gamma'_4 = \gamma_2 \wedge col \neq col''$*,* $\gamma'_5 = \gamma_3 \wedge col = col''$*, and finally,* $\gamma'_6 = \gamma_3 \wedge col \neq col''$*.*

In the current example, the equalities and inequalities on $S \cap \mathcal{C}$ *induce two equivalence relations* E_1 *and* E_2*, corresponding to* $col = col''$ *and* $col \neq col''$*, respectively. The conjunctions* γ'_i *are partitioned accordingly.*

Before we instantiate $\mathrm{sim}_{E_1}(f)$ *and* $\mathrm{sim}_{E_2}(f)$*, observe that the factors in the conjunctions* γ'_i*, for* $i \in \{1, \ldots, 6\}$*, are translated into the color algebra according to following translation rules:*

$$\begin{aligned}
z_1(A, col) &\mapsto z_1 & |z_{\{1\}}(A)| \geqslant 2 &\mapsto \sigma_{\mathrm{color}\geq 2}(z_1 - z_2)\\
z_2(A, col'') &\mapsto z_2 & |z_{\{1,2\}}(A)| \geqslant 1 &\mapsto z_1 \bowtie z_2\\
|z_{\{1\}}(A)| \geqslant 1 &\mapsto z_1 - z_2 & |z_{\{1,2\}}(A)| \geqslant 2 &\mapsto \sigma_{\mathrm{color}\geq 2}(z_1 \bowtie z_2).
\end{aligned}$$

We are now ready to define $\mathrm{sim}_{E_1}(f)$*. As noted above, the conjunctions* γ'_1*,* γ'_3 *and* γ'_5 *correspond to the equivalence relation* E_1*. Consider first* $\gamma'_1 = z_1(A, col) \wedge \Big(|z_{\{1\}}(A)| \geqslant 2\Big) \wedge z_2(A, col'') \wedge col = col''$*. We need to add the following mapping* τ_1 *(from* $\{col, col''\}$ *to CA) to* $\mathrm{sim}_{col=col''}(f)$*, defined by*

$$\tau_1 : \begin{cases} col \mapsto \sigma_{\mathrm{color}\geq 2}(z_1 - z_2) \boxtimes (\pi_\emptyset(z_1) \cup \pi_\emptyset(z_2)) \bowtie z_1\\ col'' \mapsto \sigma_{\mathrm{color}\geq 2}(z_1 - z_2) \boxtimes (\pi_\emptyset(z_1) \cup \pi_\emptyset(z_2)) \bowtie z_2 \end{cases}$$

Similarly, the following mappings are added to $\mathrm{sim}_{E_1}(f)$*: For* γ_3'*:*

$$\tau_3 : \begin{cases} col \;\; \mapsto (z_1 - z_2) \boxtimes (z_1 \bowtie z_2) \boxtimes (\pi_\emptyset(z_1) \cup \pi_\emptyset(z_2)) \bowtie z_1 \\ col'' \mapsto (z_1 - z_2) \boxtimes (z_1 \bowtie z_2) \boxtimes (\pi_\emptyset(z_1) \cup \pi_\emptyset(z_2)) \bowtie z_2, \end{cases}$$

and finally, for γ_5'*:*

$$\tau_5 : \begin{cases} col \;\; \mapsto \sigma_{\mathrm{color} \geq 2}(z_1 \bowtie z_2) \boxtimes (\pi_\emptyset(z_1) \cup \pi_\emptyset(z_2)) \bowtie z_1 \\ col'' \mapsto \sigma_{\mathrm{color} \geq 2}(z_1 \bowtie z_2) \boxtimes (\pi_\emptyset(z_1) \cup \pi_\emptyset(z_2)) \bowtie z_2 \end{cases}$$

The mappings τ_2*,* τ_4 *and* τ_6 *inserted in* $\mathrm{sim}_{E_2}(f)$ *corresponding to* γ_2'*,* γ_4' *and* γ_6'*, respectively, are similar. Hence, the final expression simulating* f *then consists of*

$$\bigcup_{i=1,3,5} \sigma_{col=col''}\tau_i(col) \bowtie \rho_{col/col''}(\tau_i(col'')) \cup \bigcup_{i=2,4,6} \sigma_{col\neq col''}\tau_i(col) \bowtie \rho_{col/col''}(\tau_i(col''))$$

□

Our second lemma connects R-uniform expressions to general RA^{c}-expressions.

Lemma 9 *Let* h *be an* RA^{c}*-expression over* $\bar{\mathcal{S}}$ *with result relation scheme* S*, and let* $R = S - \mathcal{C}$*. Then there exist a natural number* n*; CA-expressions* e_1*, . . . ,* e_n*, all with result relation scheme* R*; and an* R*-uniform* RA^{c}*-expression* $f(z_1, \ldots, z_n)$*, such that the composition* $f(e_1, \ldots, e_n)$ *is equivalent to* h*.*

PROOF. By induction on the structure of h. If h is a relation name x, then $n = 1$; e_1 is x; and f is z_1. If h is $h_1 \cup h_2$, by induction we have, for $j = 1, 2$, the natural number n_j, the sequence of CA-expressions $\boldsymbol{e^j} = e_1^j, \ldots, e_{n_j}^j$, and the RA^{c}-expression f_j. Then we put

$$\begin{aligned} n &:= n_1 + n_2 \\ e_1, \ldots, e_n &:= \boldsymbol{e^1}, \boldsymbol{e^2} \\ f &:= f_1(z_1, \ldots, z_{n_1}) \cup f_2(z_{n_1+1}, \ldots, z_n). \end{aligned}$$

The case where h is $h_1 - h_2$ is similar, but now f is $f_1 - f_2$.

If h is $h_1 \bowtie h_2$, we again begin by obtaining the ingredients for h_1 and h_2 by induction, as above. By Lemma 7, we can simulate f_1 and f_2 in CA. We now perform a case analysis based on how the result relation schemes S_1 and S_2 of h_1 and h_2 intersect with $\mathcal{C}$. There are four cases.

First, $S_1 \cap \mathcal{C} = \emptyset = S_2 \cap \mathcal{C}$. We put

$$\begin{aligned} n &:= 1 \\ e_1 &:= \mathrm{sim}(f_1)(\boldsymbol{e^1}) \boxtimes \mathrm{sim}(f_2)(\boldsymbol{e^2}) \\ f &:= \pi_R(z_1). \end{aligned}$$

Second, $S_1 \cap \mathcal{C} = \emptyset$ and $S_2 \cap \mathcal{C} \neq \emptyset$. Now in this case we take n to be the total number of expressions occurring in all sets $\mathrm{sim}_{E_2}(f_2)$, for all equivalence relations E_2 on $S_2 \cap \mathcal{C}$. For each of those expressions g, we form $g' := g(\boldsymbol{e^2}) \rhd \mathrm{sim}(f_1)(\boldsymbol{e^1})$, and all these expressions g' constitute the e_i's. (Recall the definition of the derived CA operator $\rhd$ in Example 2). Denoting the relation name corresponding to g' by z_g, we can then use the following expression for f:

$$\bigcup_{E_2} \bigcup_{\tau \in \mathrm{sim}_{E_2}(f_2)} \sigma_{\bigwedge_{(col', col'') \in E_2} col' = col''} \sigma_{\bigwedge_{(col', col'') \notin E_2} col' \neq col''} \bowtie_{col' \in S_2 \cap \mathcal{C}} \rho_{col/col'}\big(z_{\tau(col')}\big).$$

Third, $S_1 \cap \mathcal{C} = \emptyset$ and $S_2 \cap \mathcal{C} \neq \emptyset$. This case is symmetric to the previous case.

Fourth, $S_1 \cap \mathcal{C} \neq \emptyset \neq S_2 \cap \mathcal{C}$. In this case we use three kinds of CA-expressions:

(1) $\tau_1(col')(\boldsymbol{e^1}) \bowtie \tau_2(col')(\boldsymbol{e^2})$, with $col' \in S_1 \cap S_2 \cap \mathcal{C}$ and $\tau_j \in \mathrm{sim}_{E_j}(f_j)$, for an equivalence relation E_j of $S_j \cap \mathcal{C}$, for $j = 1, 2$;
(2) $\tau_1(col')(\boldsymbol{e^1}) \rhd \tau_2(col'')(\boldsymbol{e^2})$, with $col' \in (S_1 \cap \mathcal{C}) - (S_2 \cap \mathcal{C})$ and $col'' \in S_2 \cap \mathcal{C}$, and τ_j as above;
(3) $\tau_2(col'')(\boldsymbol{e^2}) \rhd \tau_1(col')(\boldsymbol{e^1})$, with $col'' \in (S_2 \cap \mathcal{C}) - (S_1 \cap \mathcal{C})$ and $col' \in S_1 \cap \mathcal{C}$, and again τ_j as above.

So, n equals the total number of all possible CA-expressions of those three kinds. For all these expressions, which are all of the form $i \bowtie j$ or $i \rhd j$, the underlying R-parameterized monadic database schema has corresponding relation names $z_{i,j}$. The expression f then becomes:

$$\begin{aligned} \bigcup_{E_1} \bigcup_{E_2} \bigcup_{\tau_1} \bigcup_{\tau_2} & \sigma_{\bigwedge_{(col', col'') \in E_1} col' = col''} \sigma_{\bigwedge_{(col', col'') \notin E_1} col' \neq col''} \\ & \sigma_{\bigwedge_{(col', col'') \in E_2} col' = col''} \sigma_{\bigwedge_{(col', col'') \notin E_2} col' \neq col''} \\ & \bowtie_{col' \in S_1 \cap S_2 \cap \mathcal{C}} \rho_{col/col'}\big(z_{\tau_1(col'), \tau_2(col')}\big) \\ & \bowtie_{\substack{col' \in (S_1 \cap \mathcal{C}) - (S_2 \cap \mathcal{C}) \\ col'' \in S_2 \cap \mathcal{C}}} \rho_{col/col'}\big(z_{\tau_1(col'), \tau_2(col'')}\big) \\ & \bowtie_{\substack{col'' \in (S_2 \cap \mathcal{C}) - (S_1 \cap \mathcal{C}) \\ col' \in S_1 \cap \mathcal{C}}} \rho_{col/col''}\big(z_{\tau_2(col''), \tau_1(col')}\big). \end{aligned}$$

If h is $\rho_{A/B}(h_1)$ with A and B not in $\mathcal{C}$, then we put $n := n_1$; $e_i := \rho_{A/B}(e_i^1)$; and $f := f_1$.

If h is $\rho_{col'/col''}(h_1)$ with $col', col'' \in \mathcal{C}$, then $n := n_1$; $e_i := e_i^1$; and $f := \rho_{col'/col''}(f_1)$.

If h is $\sigma_{A=B}(h_1)$ with A and B not in $\mathcal{C}$, then we put $n := n_1$; $e_i := \sigma_{A=B}(e_i^1)$; and $f := f_1$.

If h is $\sigma_{col'=col''}(h_1)$ with $col', col'' \in \mathcal{C}$, then $n := n_1$; $e_i := e_i^1$; and $f := \sigma_{col'=col''}(f_1)$.

Finally, if h is $\pi_X(h_1)$, then we simulate f_1 in CA according to Lemma 7. Now if the intersection of the result relation scheme S_1 of h_1 with $\mathcal{C}$ is empty, then we put $n := 1$; $e_1 := \pi_X^{\mathrm{col}}(\mathrm{sim}(f_1))(\boldsymbol{e^1})$; and $f := z_1$. If $S_1 \cap \mathcal{C} \neq \emptyset$, then we take n to be the total number of expressions occurring in all sets $\mathrm{sim}_E(f_1)$, for all equivalence relations E on $S_1 \cap \mathcal{C}$. For each of those expressions g, we form $g' := \pi_{X-\mathcal{C}}^{\mathrm{col}}(g)(\boldsymbol{e^1})$, and all these expressions g' constitute the e_i's. Denoting the relation name corresponding to g' by z_g, we can then use the following expression for f:

$$\pi_X \bigcup_E \bigcup_{\tau \in \mathrm{sim}_E(f_1)} \sigma_{\bigwedge_{(col',col'') \in E} col'=col''} \sigma_{\bigwedge_{(col',col'') \notin E} col' \neq col''} \bowtie_{col' \in S_1 \cap \mathcal{C}} \rho_{col/col'}(z_{\tau(col')}).$$

□

We illustrate Lemma 9 with the following example.

Example 10 *Consider the following* RA$^{\mathrm{c}}$*-expression* h

$$\pi_{A,col}(\overbrace{\sigma_{col \neq col'}\, \sigma_{A \neq B \wedge B \neq C}\, (\overbrace{\underbrace{R_1(A,col)}_{h_1} \bowtie \underbrace{\rho_{col/col'} R_2(B,col)}_{h_2}}^{h_4} \bowtie \underbrace{\rho_{col/col'} \rho_{B/C} R_2(B,col)}_{h_3})}^{h_6}),$$

h_5 h_7

in which various subexpressions are indicated with h_i*, for* $i \in \{1, \ldots, 7\}$*. It is readily verified that when applied to* h_1*,* h_2 *and* h_3*, Lemma 9 results in*

$$h_1 : \begin{cases} n_1 = 1 \\ e_1 = R_1 \\ f_1 = z_1 \end{cases} \qquad h_2 : \begin{cases} n_2 = 1 \\ e_2 = R_2 \\ f_2 = \rho_{col/col'}(z_2) \end{cases} \qquad h_3 : \begin{cases} n_3 = 1 \\ e_3 = \rho_{B/C}(R_2) \\ f_3 = \rho_{col/col'}(z_3) \end{cases}$$

Consider next $h_4 = h_1 \bowtie h_2$. *The simulation of* f_1 *and* f_2 *given by Lemma 7 simply consists* $\tau_1(col) = z_1$ *and* $\rho_{col/col'}(\tau_2(col')) $ *with* $\tau_2(col') = z_2$, *respectively. Hence,* $\tau_1(col)(e_1) = e_1$ *and* $\tau_2(col')(e_2) = e_2$. *Given that the result schemes of* h_1 *and* h_2 *have no color attributes in common, we obtain that*

$$h_4 : \begin{cases} n_4 = 2 \\ e_4^1 = e_1 \rhd e_2 & \textit{(with corresponding relation name } z_{\tau_1(col),\tau_2(col')}\textit{)} \\ e_4^2 = e_2 \rhd e_1 & \textit{(with corresponding relation name } z_{\tau_2(col'),\tau_1(col)}\textit{)} \\ f_4 = z_{\tau_1(col),\tau_2(col')} \bowtie \rho_{col/col'}(z_{\tau_2(col'),\tau_1(col)}). \end{cases}$$

The expression $h_5 = h_4 \bowtie h_3$ *is treated similarly. It is easily verified that the simulation of* f_3 *and* f_4 *given by Lemma 7 leads to* $\tau_3(col')(e_3) = e_3$, $\tau_4(col)(e_4^1) = e_4^1$ *and* $\tau_4(col')(e_4^2) = e_4^2$. *Since the color attributes of* h_3 *and* h_4 *overlap, we obtain:*

$$h_5 : \begin{cases} n_5 = 2 \\ e_5^1 = e_4^1 \rhd e_3 & \textit{(with corresponding relation name } z_{\tau_4(col),\tau_3(col')}\textit{)} \\ e_5^2 = e_4^2 \bowtie e_3 & \textit{(with corresponding relation name } z_{\tau_4(col'),\tau_3(col')}\textit{)} \\ f_5 = z_{\tau_4(col),\tau_3(col')} \bowtie \rho_{col/col'}(z_{\tau_4(col'),\tau_3(col')}) \end{cases}$$

The expressions h_6 *and* h_7 *are dealt with in a straightforward way:*

$$h_6 : \begin{cases} n_6 = 2 \\ e_6^1 = \sigma_{A\neq B \wedge B \neq C}(e_5^1) \\ e_6^2 = \sigma_{A\neq B \wedge B \neq C}(e_5^2) \\ f_6 = f_5 \end{cases} \qquad h_7 : \begin{cases} n_7 = 2 \\ e_7^1 = e_6^1 \\ e_7^2 = e_6^2 \\ f_7 = \sigma_{col\neq col'}(f_6) \end{cases}$$

Finally, we consider $h = \pi_{A,col}(h_7)$. *Note that* f_7 *is simulated by* $\sigma_{col\neq col'}\tau_7(col) \bowtie \rho_{col/col'}(\tau_7(col'))$ *with* $\tau_7(col) = e_7^1$ *and* $\tau_7(col') = e_7^2$. *Since the projection involves color attributes we therefore obtain*

$$h : \begin{cases} n = 2 \\ e^1 = \pi_A^{\text{col}}(e_7^1) & \textit{(with corresponding relation name } z_{\tau_7(col)}\textit{)} \\ e^2 = \pi_A^{\text{col}}(e_7^2) & \textit{(with corresponding relation name } z_{\tau_7(col')}\textit{)} \\ f = \pi_{A,col}(\sigma_{col\neq col'}(z_{\tau_7(col)} \bowtie \rho_{col/col'}(z_{\tau_7(col')}))) \end{cases}$$

Note that f *is indeed an* R*-uniform expression with* $R = \{A\}$. $\square$

We conclude this section by showing how Lemma 7 together with Lemma 9 establish Theorem 5. Let e be an RA^{c}-expression over $\bar{\mathcal{S}}$ with result relation schema $S = R \cup \{col\}$, i.e., e returns a color relation when evaluated on

color databases. By Lemma 9, the expression e can be equivalently written as the composition of an R-uniform RA^{c}-expression $f(z_1, \ldots, z_n)$ and CA-expressions $e_1, \ldots, e_n$. Lemma 7 shows that in case that S only contains a single color attribute (as is the case here since e returns a color relation over R), then f collapses to a union of CA-expressions of the form $\tau(col)$. Hence, e is indeed equivalent to a CA-expression, as desired.

Example 11 *Continuing with Example 10, we need to simulate the R-uniform expression $f = \pi_{A,col}(\sigma_{col \neq col'}(z_{\tau_7(col)} \bowtie \rho_{col/col'}(z_{\tau_7(col')})))$ over $\mathcal{S} = \{z_{\tau_7(col)}, z_{\tau_7(col')}\}$ following Lemma 7. The result, in this case, will be a CA expression over $\mathcal{S}$. To obtain the CA expression equivalent to h, it remains to substitute $z_{\tau_7(col)}$ and $z_{\tau_7(col')}$ in f as follows:*

$$z_{\tau_7(col)} \mapsto \pi^{\mathrm{col}}_A(\sigma_{A \neq B \wedge B \neq C}((R_1 \rhd R_2) \rhd \rho_{B/C}(R_2)))$$
$$z_{\tau_7(col')} \mapsto \pi^{\mathrm{col}}_A(\sigma_{A \neq B \wedge B \neq C}((R_2 \rhd R_1) \bowtie \rho_{B/C}(R_2))). \qquad \square$$

7 Completeness in the presence of built-in predicates on colors

So far, we did not assume any additional structure on the set of colors $\mathbb{C}$, except that $\mathbb{C}$ is a set equipped with the default equality predicate. In this section, we consider the case that we have general *built-in predicates* on $\mathbb{C}$. For example, in many situations it is natural to assume that there is a linear order $<$ on $\mathbb{C}$, e.g., when colors are of numerical type. Other examples of built-in predicates include arithmetic operations, string comparisons and the like. In general, let $\Pi = \{P_1, \ldots, P_k\}$ be a set of built-in predicates on $\mathbb{C}$, each of a fixed arity n_i.

We first recall (see Section 6) that every expression in RA can be expressed in the color-typed relational algebra RA^{c}. As before, we assume an infinite set $\mathcal{C}$ of color attributes col. We expand RA^{c} with the built-in predicates in Π in the standard way and denote the resulting algebra by $\mathrm{RA}^{\mathrm{c}}_{\Pi}$. More specifically, $\mathrm{RA}^{\mathrm{c}}_{\Pi}$ is the same as RA^{c} except that the selection operator $\sigma_{col=col'}$ for $col, col' \in \mathcal{C}$ is replaced with a *generalized selection predicate* $\sigma_{\theta(col_1,\ldots,col_m)}$ where $col_1, \ldots, col_m \in \mathcal{C}$ and θ is one of the predicates P_i of Π and $m = n_i$. (The standard selection operator $\sigma_{A=B}$ for non-color attributes remains unchanged.)

We next define a generalization of the color algebra, denoted by CA_{Π}, and show that (i) every expression in CA_{Π} on color relations can be simulated in RA_{Π}; and (ii) every operation on color relations that is definable in RA_{Π} is already definable in CA_{Π}. In other words, CA_{Π} is relationally complete in the presence of the built-in predicates Π on $\mathbb{C}$.

The color algebra CA_Π is obtained from CA by removing $\boxtimes$, $\bowtie$ and $\sigma_{\mathrm{color}\geq k}$ and by adding the *generalized color join operator*, defined as follows. Let r'_1, $\ldots$, r'_n be color relations over relation schemas R_1, $\ldots$, R_n, respectively. Let $\varphi(X_1, \ldots, X_n, col)$ be a first-order formula over the predicates in Π and the unary relation symbols X_1, $\ldots$, X_n, such that col is the only free individual variable of φ. Then the generalized color join $\bowtie_\varphi\{r'_1, \ldots, r'_n\}$ equals the following color relation over the relation schema $R_1 \cup \cdots \cup R_n$:

$$\{(t_1 \cup \cdots \cup\ t_n, c) \mid t_1 \cup \cdots \cup t_n \in r_1 \bowtie \cdots \bowtie r_n \\ \text{and } \varphi(r'_1(t_1), \ldots, r'_n(t_n), c) \text{ is true}\}.$$

Hence, when considering the joint tuple $t_1 \cup \cdots \cup t_n$, the unary relations X_i in φ are instantiated by the sets of colors associated with the tuples in their respective relations, i.e., $r'_i(t_i)$. The joint tuple is colored with all colors c satisfying the formula φ evaluated on these sets. Note that we use active-domain semantics for φ, i.e., c can only be a color that appears in at least one of these sets.

Example 12 *The CA-operators $\bowtie$, $\boxtimes$ and $\sigma_{\mathrm{color}\geq k}$ are all special cases of the generalized color join:*

$$\begin{aligned} x \boxtimes y &= \bowtie_{X_1(col)\vee X_2(col)}\{x, y\} \\ x \bowtie y &= \bowtie_{X_1(col)\wedge X_2(col)}\{x, y\} \\ \sigma_{\mathrm{color}\geq k}(x) &= \bowtie_{\varphi}\{x\} \end{aligned}$$

where φ in the last equation is

$$\exists col_1, \ldots, \exists col_k \bigwedge_{i=1}^{k} X_1(col_i) \wedge \bigwedge_{1\leq i<j\leq k} col_j \neq col_k \wedge X_1(col). \qquad \square$$

Example 13 *Using a predicate $<$ on $\mathbb{C}$, consider the operation*

$$\bowtie_{\neg\exists col_1(X_1(col_1)\wedge X_2(col)\wedge col_1<col)}\{x, y\}$$

When applied to color relations r' and s', it returns the joint tuples $t_1 \cup t_2$ from the natural join of the underlying relations r and s (with $t_1 \in r$ and $t_2 \in s$); these joint tuples will inherit the color of t_2 in s' provided that t_1 does not appear in r' with a smaller color. $\square$

Since the definition of the generalized color join is readily formalized in the relational calculus, the generalized color join is expressible in RA_Π, due to the equivalence of relational calculus and relational algebra, which also holds in the presence of built-in predicates [16].

Example 14 *The generalized color join from Example 13 can be expressed in* $\mathrm{RA}^{\mathrm{c}}_{\Pi}$ *as follows: (R and S refer to the relation schemas of x and y respectively)*

$$\pi_{R\cup\bar{S}}(\rho_{col/col_1}(x) \bowtie y) - \pi_{R\cup\bar{S}}\sigma_{col_1<col}(\rho_{col/col_1}(x) \bowtie y). \qquad \square$$

We now show that CA_{Π} is again relationally complete, i.e., can express every operation on color relations explicitly expressible in the relational algebra with built-in predicates on colors:

Theorem 15 *For every relational algebra expression over* $\bar{\mathcal{S}}$ *in* RA_{Π} *whose result schema is of the form* $\bar{R}$ *for some relation scheme* R*, there exists an equivalent* CA_{Π}*-expression over* $\mathcal{S}$*.*

This theorem can be proven in the same general way as Theorem 5: we first show that CA_{Π} can simulate R-uniform $\mathrm{RA}^{\mathrm{c}}_{\Pi}$ (suitably defined) and then show that every $\mathrm{RA}^{\mathrm{c}}_{\Pi}$-expression can be simulated by means of R-uniform $\mathrm{RA}^{\mathrm{c}}_{\Pi}$-expressions.

In order to state our first lemma, the generalized color join must be slightly generalized (sic) to allow for multiple color attributes. This is necessary since intermediate relations can have multiple color attributes.

Recall the definition of the generalized color join operator $\bowtie_{\varphi}\{r'_1, \ldots, r'_n\}$, which returns a color relation over the relation schema $R_1 \cup \cdots \cup R_n$. When we allow φ to have multiple, say m, free individual variables col_1, $\ldots$, col_m, we obtain an m-ary generalized color join. The result of this operation is the following relation over the relation scheme $R_1 \cup \cdots \cup R_n \cup \{col_1, \ldots, col_n\}$:

$$\{(t_1 \cup \cdots \cup t_n, c_1, \ldots, c_m) \mid t_1 \cup \cdots \cup t_n \in r_1 \bowtie \cdots \bowtie r_n$$
$$\text{and } c_1, \ldots, c_m \in \bigcup_{i=1}^{n} r'_i(t_i) \text{ and } \varphi(r'_1(t_1), \ldots, r'_n(t_n), c_1, \ldots, c_m) \text{ is true}\}$$

Note that this relation is not a color relation, unless $m = 1$, in which case we revert to the ordinary generalized color join defined before and the result is indeed a color relation over the relation schema $R_1 \cup \cdots \cup R_n$. In particular, the m-ary generalized color join is *not* part of CA_{Π} for $m \neq 1$. Its only purpose is to formalize the lemma below.

The notion of R-uniformity extends in the natural way to expressions in $\mathrm{RA}^{\mathrm{c}}_{\Pi}$.

Thus, an $\mathrm{RA}^{\mathrm{c}}_{\Pi}$-expression f over an R-parameterized monadic database schema is called R-uniform if it satisfies the following:

- f uses only renamings $\rho_{A/B}$ where A and B are color attributes in $\mathcal{C}$;

- f uses only selections $\sigma_{\theta(col_1,\ldots,col_m)}$ on color attributes, i.e., f uses no selections $\sigma_{A=B}$ on non-color attributes;
- all projections π_X appearing in f satisfy $R \subseteq X$.

We are now ready to state the analogue of Lemma 7 in the presence of built-in predicates. Since the m-ary generalized color join is quite powerful, the generalized lemma is actually easier to state and prove than the original.

Lemma 16 *Let f be an R-uniform RA^c_Π-expression over the R-parameterized monadic database schema $\mathcal{S}$. Let S be the result relation schema of f. If $S \cap \mathcal{C} = \{col_1, \ldots, col_m\}$, then $f(D)$ is equivalent to a finite union of m-ary generalized joins.*

PROOF. Assume that $\mathcal{S}$ consists of the relation names $z_1, \ldots, z_n$. Analogously to the proof of Lemma 7, we define $\mathrm{FO}^c_{\Pi,R}$ as the first-order logic defined as follows. Let $R = \{A_1, \ldots, A_k\}$. We use the A_j's, plus all color attributes, as first-order variables. The allowed atomic formulas are of two forms:

(1) $z_i(A_1, \ldots, A_k, col')$ with $col' \in \mathcal{C}$. We abbreviate such formulas with $z_i(R, col')$.
(2) $P_i(col_1, \ldots, col_{n_i})$ for $P_i \in \Pi$.

Like in the proof of Lemma 7, R-uniform RA^c_Π corresponds to $\mathrm{FO}^c_{\Pi,R}$.

For any non-empty subset α of $\{1, \ldots, n\}$ let param_α be the CA-expression

$$\boxtimes_{i \in \alpha} z_i \boxminus \bigcup_{i \notin \alpha} z_i.$$

It computes the tuples over R that are in all relations with indexes in α, but that are in none of the remaining relations. Moreover, the tuples inherit the colors from their corresponding relations.

Now let Ψ be the $\mathrm{FO}^c_{\Pi,R}$-formula equivalent to the R-uniform RA^c_Π-expression f from the statement of the Lemma. It is then clear that Ψ is equivalent to the following finite union of m-ary generalized color joins:

$$\bigcup_{\substack{\alpha \subseteq \{1,\ldots,n\} \\ \alpha \neq \emptyset}} \bowtie_{\Psi\alpha(X_1,\ldots,X_n,col_1,\ldots,col_m)} \{z_i \rhd \mathsf{param}_\alpha \mid i \in \alpha\},$$

where Ψ_α is the formula Ψ in which $z_i(R, col')$ is replaced by $X_i(col')$ in case that $i \in \alpha$ and by false otherwise. $\square$

The literal analog of Lemma 9 is the following. The proof proceeds entirely analogously to the proof of Lemma 9, so we omit it.

Lemma 17 *Let h be an $\mathrm{RA}^{\mathrm{c}}_{\Pi}$-expression over $\bar{\mathcal{S}}$ with result relation scheme S, and let $R = S - \mathcal{C}$. Then there exist a natural number n; CA_{Π}-expressions e_1, ..., e_n, all with result relation scheme R; and an R-uniform $\mathrm{RA}^{\mathrm{c}}_{\Pi}$-expression $f(z_1, \ldots, z_n)$, such that the composition $f(e_1, \ldots, e_n)$ is equivalent to h.*

8 Conclusion

We conclude the paper by listing some interesting research directions: First, in Proposition 6 we established the non-redundancy of the color algebra. It is open, however, whether the color algebra is also non-redundant when considering the stronger notion of non-redundancy that allows the use of yes/no queries only. Second, although it is readily verified that the translation described in Section 6 leads to an exponential blow-up in the size of the expressions (both the quantifier elimination and the introduction of the equivalence relations on color attributes in Lemma 7 might incur an exponential blow-up), exact lower and upper bounds on the translation of relational algebra expressions into the color algebra (possibly with a different method) are, however, unknown. Finally, it is worth exploring the extension of the color algebra with built-in predicates on data (and not solely on colors as in Section 7).

Acknowledgments

We would like to thank the referees for their comments which led to significant improvements in the presentation of the paper.

References

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

[2] W. Ackermann. *Solvable Cases of the Decision Problem*. Studies in Logic and the Foundations of Mathematics. North-Holland, 1968.

[3] D. Bhagwat, L. Chiticariu, W.-C. Tan, and G. Vijayvargiya. An annotation management system for relational databases. *The VLDB Journal*, 14(4):373–396, 2005.

[4] E. Börger, E. Grädel, and Y. Gurevich. *The Classical Decision Problem*. Springer, 1997.

[5] P. Buneman, J. Cheney, and S. Vansummeren. On the expressiveness of implicit provenance in query and update languages. In T. Schwentick and D. Suciu, editors, *Database Theory—ICDT 2007*, volume 4353 of *Lecture Notes in Computer Science*, pages 209–223. Springer, 2007.

[6] P. Buneman, S. Khanna, and W.-C. Tan. On propagation of deletions and annotations through views. In *Proceedings 21st ACM Symposium on Principles of Database Systems*, pages 150–158. ACM Press, 2002.

[7] A. Chandra and D. Harel. Computable queries for relational data bases. *Journal of Computer and System Sciences*, 21(2):156–178, 1980.

[8] L. Chiticariu, W.-C. Tan, and G. Vijayvargiya. DBNotes: A post-it system for relational databases based on provenance. In *Proceedings 2005 ACM SIGMOD International Conference on Management of Data*, pages 942–944. ACM Press, 2005.

[9] G. Cong, W. Fan, and F. Geerts. Annotation propagation revisited for key preserving views. In *Proceedings 15th ACM International Conference on Information and Knowledge Management*, pages 632–641. ACM Press, 2006.

[10] Y. Cui, J. Widom, and J.L. Wiener. Tracing the lineage of view data in a warehousing environment. *ACM Trans. Database Syst.*, 25(2):179–227, 2000.

[11] F. Geerts, A. Kementsietsidis, and D. Milano. *i*MONDRIAN: A visual tool to annotate and query scientific databases. In Y.E. Ioannidis et al., editors, *Advances in Database Technology—EDBT 2006*, volume 3896 of *Lecture Notes in Computer Science*, pages 1168–1171. Springer, 2006.

[12] F. Geerts, A. Kementsietsidis, and D. Milano. MONDRIAN: Annotating and querying databases through colors and blocks. In *Proceedings 22th International Conference on Data Engineering*, page 82. IEEE Computer Society, 2006. 10 pages.

[13] T.J. Green, G. Karvounarakis, and V.Tannen. Provenance semirings. In *Proceedings 26th ACM Symposium on Principles of Database Systems*, pages 31–40. ACM Press, 2007.

[14] D. Srivastava and Y. Velegrakis. Intensional associations between data and metadata. In *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data*, pages 401–412. ACM Press, 2007.

[15] D. Srivastava and Y. Velegrakis. Using queries to associate metadata with data. In *Proceedings 23rd International Conference on Data Engineering*, pages 1451–1453. IEEE Computer Society, 2007.

[16] J.D. Ullman. *Principles of Database and Knowledge-Base Systems*, volume I. Computer Science Press, 1988.

[17] J. Van den Bussche and L. Cabibbo. Converting untyped formulas into typed ones. *Acta Informatica*, 35(8):637–643, 1998.

[18] Y.R. Wang and S.E. Madnick. A polygen model for heterogeneous database systems: the source taging perspective. In D. McLeod, R. Sacks-Davis, and H. Schek, editors, *Proceedings of the 16th International Conference on Very Large Data Bases*, pages 518–538. Morgan Kaufmann, 1990.