Cardinalities of Finite Relations in Coq (Rough Diamond)

Paul Brunet, Damien Pous, Insa Stucke

Cardinalities of Finite Relations in Coq (Rough Diamond) ⋆

Paul Brunet[1], Damien Pous[1] ⋆⋆, and Insa Stucke[2]

[1] CNRS - LIP, ENS Lyon, UMR 5668
[2] Institut für Informatik, Christian-Albrechts-Universität zu Kiel, Germany

Abstract. We present an extension of a Coq library for relation algebras, where we provide support for cardinals in a point-free way. This makes it possible to reason purely algebraically, which is well-suited for mechanisation. We discuss several applications in the area of graph theory and program verification.

1 Introduction

Binary relations have a rich algebraic structure: rather than considering relations as objects relating points, one can see them as abstract objects that can be combined using various operations (e.g., union, intersection, composition, transposition). Those operations are subject to many laws (e.g., associativity, distributivity). One can thus use equational reasoning to prove results about binary relations, graphs, or programs manipulating such structures. This is the so-called relation-algebraic method [12, 14, 15].

Lately, the second author developed a library for the Coq proof assistant [9, 10], allowing one to formalise proofs using the relation algebraic approach. This library contains powerful automation tactics for some decidable fragments of relation algebra (Kleene algebra and Kleene algebra with tests), normalisation tactics, and tools for rewriting modulo associativity of relational composition.

The third author recently relied on this library to formalise algebraic correctness proofs for several standard algorithms from graph theory: computing vertex colourings [1] and bipartitions [2].

Here we show how to extend this library to deal with cardinals of relations, thus allowing one to reason about quantitative aspects. We study several applications in [3]; in this extended abstract we focus on a basic result about the size of a linear order and an intermediate result from graph theory.

2 Preliminaries

Given two sets X, Y, a *binary relation* is a subset $R \in \mathcal{P}(X \times Y)$. The set X (resp. Y) is called the *domain* (resp. *codomain*) of the relation.

⋆ Author's version of the article published by Springer in Proc. ITP'16 and available at http://doi.dx.org/10.1007/978-3-319-43144-4_29.

⋆⋆ This work was supported by the project ANR 12IS02001 PACE and the European Research Council (H2020 programme, CoVeCe, grant agreement No 678157).

With the usual set-theoretic operations of inclusion ($\subseteq$), union ($\cup$), intersection ($\cap$), complement ($\bar{\cdot}$), the empty relation (O_{XY}) and the universal relation (L_{XY}), binary relations between two sets X and Y form a Boolean lattice. Given three sets X, Y, Z and relations $R \in \mathcal{P}(X \times Y)$ and $S \in \mathcal{P}(Y \times Z)$ we also consider the operations of *composition* ($RS \in \mathcal{P}(X \times Z)$) and *transposition* ($R^{\mathsf{T}} \in \mathcal{P}(Y \times X)$), as well as the *identity relation* ($\mathsf{I}_X \triangleq \{(x, x) \mid x \in X\} \in \mathcal{P}(X \times X)$). These operations can be abstracted through the axiomatic notion of *relation algebra*. Binary relations being the standard model of such an algebra, we use the same notations.

Definition 2.1 (Relation Algebra). *A* relation algebra *is a category whose homsets are Boolean lattices, together with an operation of* transposition *($\cdot^{\mathsf{T}}$) such that:*

($P1$) *composition is monotone in its two arguments, distributes over unions and is absorbed by the bottom elements;*
($P2$) *transposition is monotone, involutive ($R^{\mathsf{T}^{\mathsf{T}}} = R$), and reverses compositions: for all morphisms R, S of appropriate types, we have $(RS)^{\mathsf{T}} = S^{\mathsf{T}} R^{\mathsf{T}}$;*
($P3$) *for all morphisms Q, R, S of appropriate types, $QR \subseteq S$ iff $Q^{\mathsf{T}} \overline{S} \subseteq \overline{R}$ iff $\overline{S}\, R^{\mathsf{T}} \subseteq \overline{Q}$;*
($P4$) *for all morphism $R : X \to Y$, $R \neq \mathsf{O}$ iff for all objects X', Y', $\mathsf{L} R \mathsf{L} = \mathsf{L}_{X'Y'}$.*

From properties ($P2$), we deduce that transposition commutes with all Boolean connectives, and that $\mathsf{I}^{\mathsf{T}} = \mathsf{I}$. Equivalences ($P3$) are called *Schröder equivalences* in [12]; they correspond to the fact that the structure is residuated [5]. The last property ($P4$) is known as *Tarski's rule*; it makes it possible to reason algebraically about non-emptiness.

Important classes of morphisms can be defined algebraically. For instance, we say in the sequel that a morphism $R : X \to Y$ is:

- *injective* if $RR^{\mathsf{T}} \subseteq \mathsf{I}$,
- *surjective* if $\mathsf{I} \subseteq R^{\mathsf{T}} R$,
- *univalent* if its transpose is injective (i.e., $R^{\mathsf{T}} R \subseteq \mathsf{I}$),
- *total* if its transpose is surjective (i.e., $\mathsf{I} \subseteq RR^{\mathsf{T}}$),
- a *mapping* if R is total and univalent.

One can easily check that these definitions correspond to the standard definitions in the model of binary relations.

Before introducing cardinals, we need a way to abstract over the singleton sets from the model of binary relations; we use the following definition:

Definition 2.2 (Unit in a Relation Algebra). *A* unit *in a relation algebra is an object 1 such that $\mathsf{O}_{11} \neq \mathsf{L}_{11}$ and $\mathsf{I}_1 = \mathsf{L}_{11}$.*

In other words, there are only two morphisms from a unit to itself. In the model of binary relations, every singleton set is a unit. Using units, we can axiomatise the notion of cardinal in a relation algebra; we mainly follow Kawahara [8]:

Definition 2.3 (Cardinal). *A relation algebra with* cardinal *is a relation algebra with a unit* 1 *and a monotone function* $|\cdot|$ *from morphisms to natural numbers such that for all morphisms* Q, R, S *of appropriate types:*

($C1$) $|\mathsf{O}| = 0$,
($C2$) $|\mathsf{I}_1| = 1$,
($C3$) $|R^\mathsf{T}| = |R|$,
($C4$) $|R \cup S| + |R \cap S| = |R| + |S|$,
($C5$) *if* Q *is univalent, then* $|R \cap Q^\mathsf{T} S| \leq |QR \cap S|$ *and* $|Q \cap SR^\mathsf{T}| \leq |QR \cap S|$.

Note that these requirements for a cardinal rule out infinite binary relations: we have to restrict to binary relations between finite sets, i.e., graphs. Typically, in this model, the cardinal of a relation is the number of pairs it contains. This restriction is harmless in practice: we only work with finite sets when we study, for example, algorithms.

Many natural facts of cardinal can be derived just from conditions ($C1$) to ($C4$), e.g., monotonicity. The last condition ($C5$) is less intuitive; it is called the *Dedekind inequality* in [8]. It allows one to compare cardinalities of morphisms of different types. Kawahara uses it to obtain, e.g., the following result:

Lemma 2.4. *Assume a relation algebra with cardinal. For all morphisms* Q, R, S *of appropriate type, we have:*

1. *If* R *and* S *are univalent, then* $|RS \cap Q| = |R \cap QS^\mathsf{T}|$.
2. *If* R *is univalent and* S *is a mapping, then* $|RS| = |R|$.

Leaving cardinals aside, two important classes of morphisms are that of vectors and points, as introduced in [11], for providing a way to model subsets and single elements of sets, respectively:

- *vectors*, denoted with lower case letters v, w in the sequel, are morphisms $v : X \to Y$ such that $v = v\mathsf{L}$. In the standard model, this condition precisely amounts to being of the special shape $V \times Y$ for a subset $V \subseteq X$.
- *points*, denoted with lower case letters p, q in the sequel, are injective and nonempty vectors. In the standard model, this condition precisely amounts to being of the special shape $\{x\} \times Y$ for an element $x \in X$.

In the binary relations model, one can characterise vectors and points from their Boolean-matrix representation of binary relations: a vector is a matrix whose rows are either zero everywhere or one everywhere, and a point is a matrix with a single row of ones and zeros everywhere else. Every morphism with unit as its codomain is a vector; points with unit as their codomain have cardinal one:

Lemma 2.5. *Let* $p : X \to 1$ *be a point in a relation algebra with a cardinal (and unit). We have* $|p| = 1$.

We conclude this preliminary section with the notion of pointed relation algebra. Indeed, in the model of binary relations, the universal relation between X and Y is the least upper bound of all points between X and Y. This property is called the *point axiom* in [4]. Since we restrict to finite relations, we give a finitary presentation of this law.

Definition 2.6 (Pointed Relation Algebra). *A relation algebra is* pointed *if for all X, Y there exists a (finite) set P_{XY} of points such that $\mathsf{L}_{XY} = \bigcup_{p \in P_{XY}} p$.*

As a consequence, in pointed relation algebras it holds $\mathsf{I}_X = \bigcup_{p \in P_{XX}} pp^\mathsf{T}$. When working in pointed relation algebras with cardinal, we also have results like the following, where we use $|X|$ as a shorthand notation for $|\mathsf{L}_{X1}|$:

Lemma 2.7. *For all objects X and Y we have $|\mathsf{L}_{XY}| = |X| \cdot |Y|$ and $|\mathsf{I}_X| = |X|$.*

Any pointed relation algebra with cardinal is in fact isomorphic to an algebra of relations on finite sets; therefore, the above list of axioms can be seen as a convenient list of facts about binary relations which make it possible to reason algebraically. Still, our modular presentation of the theory makes it possible to work in fragments of it where this representation theorem breaks, i.e., for which other models exist than that of binary relations.

3 Relation Algebra in Coq

The Coq library RelationAlgebra [9, 10] provides axiomatisations and tools for various fragments of the calculus of relations: from ordered monoids to Kleene algebra, residuated structures, and Dedekind Categories. It is structured in a modular way: one can easily decide which operations and axioms to include.

In the present case, these are Boolean operations and constants, composition, identities, transposition. We extended the library by a module `relalg` containing definitions and facts about this particular fragment. For instance, this module defines many classes of relations, some of which we already mentioned in Section 2. For those properties we use classes in Coq:

```
Class is_vector (C: ops) X Y (v: C X Y) := vector: v*top == v.
```

Here we assume an ambient relation algebra `C`, `ops` being the corresponding notion, as exported by the RelationAlgebra library. Variables `X`,`Y` are objects of the category, and `v: C X Y` is a morphism from `X` to `Y`. The symbols `*` and `==` respectively denote composition and equality; `top` is the top morphism of appropriate type: its source and target (`Y` twice) are inferred automatically.

The RelationAlgebra library provides several automation tactics to ease equational reasoning [9, 10]. The most important ones are:

- `ra_normalise` for normalising the current goal w.r.t. the simplest laws (mostly about idempotent semirings, units and transposition),
- `ra` for solving goals by normalisation and comparison,
- `lattice` for solving lattice-theoretic goals,
- `mrewrite` for rewriting modulo associativity of categorical composition.

The library also contains a decision procedure for Kleene algebra with tests, which we do not discuss here for lack of space. Those tactics are defined either by reflection, where a decision procedure is certified within Coq (`ra_normalise`, `ra`); by exhaustive proof search (`lattice`); or as ad hoc technical solutions (`mrewrite`,

which is a plugin in OCaml that applies appropriate lemmas to reorder parentheses and generalise the considered (in)equation).

A crucial aspect for this work is the interplay between the definitions from this library and Coq's support for setoid rewriting [13], which makes it possible to rewrite using both equations and inequations in a streamlined way, once the monotonicity or anti-monotonicity of all operations has been proved.

This is why we use a class to define the above predicate `is_vector`: in this case, the tactic `rewrite vector` will look for a subterm of a shape `v*top` where `v` is provably a vector using typeclass resolution, and replace it with `v`. Similar classes are set-up for all notions discussed in the sequel (injective, surjective, univalent, total, mapping, points, and many more).

We also define classes to represent relation algebra with unit, relation algebra with cardinal, and pointed relation algebra. Units are introduced as follows:

```
Class united (C: ops) := {
 unit: ob C;
 top_unit: top' unit unit == 1;
 nonempty_unit:> is_nonempty (top' unit unit) }.
```

The field `unit` is the unit object; the two subsequent fields correspond to the requirements from Definition 2.2. The symbol `1` is our notation for identity morphisms. Assuming units, one can then define cardinals:

```
Class cardinal (C: ops) (U: united C) := {
 card: forall X Y, C X Y → nat;
 card0: forall X Y, @card X Y 0 = 0;
 card1: @card unit unit 1 = 1;
 cardcnv: forall X Y (R: C X Y), card R^T = card R;
 cardcup: forall X Y (R S: C X Y), card (R ∪ S) + card (R ∩ S) = card R + card S;
 cardded: forall X Y Z (R: C X Y) (S: C Y Z) (T: C X Z),
     is_injective R → card (T ∩ (R*S)) ≤ card (R^T * T ∩ S);
 cardded': forall X Y Z (R: C Y X) (S: C Y Z) (T: C Z X),
     is_univalent R → card (R ∩ (S*T)) ≤ card (R * T^T ∩ S) }.
```

The first field is the cardinal operation itself. The remaining ones correspond to the conditions from Definition 2.3.

Next we give two Coq proofs about cardinals, to show the ease with which it is possible to reason about them. The first one correspond to Lemma 2.4(2).

```
Lemma card_unimap X Y Z (R: C X Y) (S: C Y Z):
      is_univalent R → is_mapping S → card (R*S) = card R.
Proof. rewrite ←capxt, card_uniuni, surjective_tx. apply card_weq. ra. Qed.
```

Here, Lemma `uniuni` corresponds to Lemma 2.4(1); `capxt` states that top is a unit for meet; `surjective_tx` that every surjective morphism R satisfies $\mathsf{L}R = \mathsf{L}$; and `card_weq` that cardinals are preserved by equality.

The second illustrative proof is that of Lemma 2.5, which becomes a oneliner:

```
Lemma card_point X (R: C X unit): is_point R → card R = 1.
Proof. rewrite ←cardcnv, ←dot1x. rewrite card_unimap. apply card1. Qed.
```

(Lemma `dot1x` states that `I` is a left unit for composition.)

4 Applications

We first detail an easy example where we link the cardinality of morphisms representing linear orders to the cardinality of their carrier sets. The second example is based on a graph theoretic result giving a lower bound for the cardinality of an independent set.

4.1 Linear orders

A morphism $R : X \to X$ is a *partial order* on X if R is *reflexive*, *antisymmetric* and *transitive* (i.e., $\mathsf{I} \subseteq R$, $R \cap R^\mathsf{T} \subseteq \mathsf{I}$ and $RR \subseteq R$). If R is additionally *linear* (i.e., $R \cup R^\mathsf{T} = \mathsf{L}$) we call R a *linear order*. Recall that for an object X, $|X|$ is a shorthand for $|\mathsf{L}_{X1}|$. We have

Theorem 4.1. *If $R : X \to X$ is a linear order, then $|R| = \frac{|X|^2+|X|}{2}$.*

Proof. Since R is antisymmetric we have $R \cap R^\mathsf{T} \subseteq \mathsf{I}$. Furthermore, we have $\mathsf{I} \subseteq R$ since R is reflexive so that $R \cap R^\mathsf{T} = \mathsf{I}$. Now we can calculate as follows:

$$\begin{aligned}
|X|^2 + |X| &= |\mathsf{L}_{XX}| + |\mathsf{I}_X| && \text{(by Lemma 2.7)}\\
&= |R \cup R^\mathsf{T}| + |\mathsf{I}_X| && (R \text{ linear})\\
&= |R \cup R^\mathsf{T}| + |R \cap R^\mathsf{T}| && (R \text{ reflexive and antisymmetric})\\
&= |R| + |R^\mathsf{T}| && \text{(by } (C4)\text{)}\\
&= |R| + |R| && \text{(by } (C3)\text{)} \quad \square
\end{aligned}$$

With the presented tools, this lemma can be proved in Coq in a very same way. First we need to define a notation for the cardinal of an object:

```
Notation card' X := card (top' X unit).
Lemma card_linear_order X (R: C X X): is_order R → is_linear R →
  2*card R = card' X * card' X  + card' X.
Proof.
  intros Ho Hli.
  rewrite ←card_top, ←card_one.
  rewrite ←Hli.
  rewrite ←kernel_refl_antisym.
  rewrite capC, cardcup.
  rewrite cardcnv. lia.
Qed.
```

The standard Coq tactic `lia` solves linear integer arithmetic. The lemmas `card_top` and `card_one` correspond to the statements of Lemma 2.7, i.e.,

```
Lemma card_top X Y: card (top' X Y) = card' X * card' Y.
Lemma card_one X: card (one X) = card' X.
```

Lemma `kernel_refl_antisym` states that the kernel of a reflexive and antisymmetric morphism is just the identity.

4.2 Independence number of a graph

In this section we prove bounds for the *independence number* of an undirected graph [16]. An *undirected (loopfree) graph* $g = (X, E)$ has a symmetric and irreflexive adjacency relation. It can thus be represented by a morphism $R : X \to X$ that is *symmetric* (i.e., $R^{\mathsf{T}} \subseteq R$) and *irreflexive* (i.e., $R \cap \mathsf{I} = \mathsf{O}$).

An *independent set* (or *stable set*) of g is a set of vertices S such that any two vertices in S are not connected by an edge, i.e., $\{x, y\} \notin E$, for all $x, y \in S$. Independent sets can be modelled abstractly using vectors: a vector $s : X \to 1$ models an independent set of a morphism R if $Rs \subseteq \overline{s}$. Furthermore, we say that an independent set S of g is *maximum* if for every independent set T of g we have $|T| \leq |S|$. The maximum size of an independent set is defined as:

$$\alpha_R \triangleq \max \{|s| \mid s \text{ is an independent set of } R\} \quad .$$

One easily obtain the lower bound $\alpha_R \leq \sqrt{|\overline{R}|}$. In fact, we have $|s| \leq \sqrt{|\overline{R}|}$ for every independent set s, which we can prove in two lines using our library.

The upper bound is harder to obtain. We have $\frac{|R|}{k+1} \leq \alpha_R$, where k is the maximum degree of R. Call *maximal* an independent set which cannot be enlarged w.r.t. the preorder $\subseteq$:

```
Definition maximal (v: C X unit) := forall w, v <== w → R ∗ w <== !w → w <== v.
```

As expected, maximum independent sets are maximal:

```
Lemma maximum_maximal (v: C X unit):
  R∗v <== !v → card v = independent_number R → maximal v.
```

(Note that the converse is not necessarily true.) Then we prove the following algebraic characterisation of maximal independent sets: while independent sets are characterised by an inequality ($Rv \subseteq \overline{v}$), maximal are characterised by an equality ($Rv = \overline{v}$).

```
Lemma maximal_independent_iff (v: C X unit):
  R∗v <== !v → (maximal v ↔ R∗v == !v).
```

Finally, obtaining the lower bound for the independence number consists in proving that maximal independent sets, defined algebraically, satisfy this bound:

```
Lemma maximal_lower_bound (v: C X unit):
  R∗v == !v → card' X ≤ (maximum_degree R + 1) ∗ card v.
Theorem independent_lower_bound:
  card' X <== (maximum_degree R + 1) ∗ independent_number R.
```

Including the proofs of the three key lemmas, the final theorem is eventually proved in 41 lines of Coq. We consider this a success as this is comparable to what is required for a detailed paper proof.

5 Conclusion

We presented an extension of the Coq RelationAlgebra library [3], that makes it possible to reason algebraically about cardinalities of binary relations. A key feature of the Coq proof assistant for this work is *dependent types*: they allow us to define relation algebras as categories in a straightforward way, so that we can talk about vectors or units as one would do on paper. While our approach to cardinals would certainly work when starting from Kahl's implementation of allegories in Agda [7], it remains unclear to us whether it could be adapted to his formalisation of relation algebra in Isabelle/Isar [6].

References

1. R. Berghammer, P. Höfner, and I. Stucke. Tool-based verification of a relational vertex coloring program. In *Proc. RAMiCS*, volume 9348 of *LNCS*, pages 275–292. Springer, 2015.
2. R. Berghammer, I. Stucke, and M. Winter. Investigating and computing bipartitions with algebraic means. In *Proc. RAMiCS*, volume 9348 of *LNCS*, pages 257–274. Springer, 2015.
3. P. Brunet, D. Pous, and I. Stucke. Cardinalities of relations in Coq. Coq Development and full version of this extended abstract; available from `http://media.informatik.uni-kiel.de/cardinal/`, 2016.
4. H. Furusawa. *Algebraic Formalisations of Fuzzy Relations and Their Representation Theorems*. PhD thesis, Department of Informatics, Kyushu University, 1998.
5. N. Galatos, P. Jipsen, T. Kowalski, and H. Ono. *Residuated Lattices: An Algebraic Glimpse at Substructural Logics*. Elsevier, 2007.
6. W. Kahl. Calculational relation-algebraic proofs in Isabelle/Isar. In *Proc. RelMiCS*, volume 3051 of *LNCS*, pages 178–190. Springer, 2003.
7. W. Kahl. Dependently-typed formalisation of relation-algebraic abstractions. In *Proc. RAMiCS*, volume 6663 of *LNCS*, pages 230–247. Springer, 2011.
8. Y. Kawahara. On the Cardinality of Relations. In *Proc. RelMiCS/AKA*, volume 4136 of *LNCS*, pages 251–265. Springer, 2006.
9. D. Pous. Relation Algebra and KAT in Coq. Website. `http://perso.ens-lyon.fr/damien.pous/ra/`.
10. D. Pous. Kleene Algebra with Tests and Coq tools for while programs. In *Proc. ITP*, volume 7998 of *LNCS*, pages 180–196. Springer, 2013.
11. G. Schmidt and T. Ströhlein. Relation Algebras: Concept of Points and Representability. *Discrete Mathematics*, 54(1):83–92, 1985.
12. G. Schmidt and T. Ströhlein. *Relations and Graphs - Discrete Mathematics for Computer Scientists*. EATCS Monographs on Th. Comp. Sci. Springer, 1993.
13. M. Sozeau. A new look at generalized rewriting in type theory. *J. Formalized Reasoning*, 2(1):41–62, 2009.
14. A. Tarski. On the calculus of relations. *J. of Symbolic Logic*, 6(3):73–89, 1941.
15. A. Tarski and S. Givant. *A Formalization of Set Theory without Variables*, volume 41 of *Colloquium Publications*. AMS, Providence, Rhode Island, 1987.
16. V. Wei. A Lower Bound for the Stability Number of a Simple Graph. *Bell Laboratories Technical Memorandum 81-11217-9*, 1981.