Cardinality and Representation of Stone Relation Algebras

Walter Guttmann

September 26, 2023

Abstract

In relation algebras, which model unweighted graphs, the cardinality operation counts the number of edges of a graph. We generalise the cardinality axioms to Stone relation algebras, which model weighted graphs, and study the relationships between various axioms for cardinality. We also give a representation theorem for Stone relation algebras.

Contents

The theories formally verify results in [1]. See this papers for further details and related work. Stone relation algebras have been introduced in [2] and formalised in [3].

theory *Representation*

imports *Stone-Relation-Algebras.Matrix-Relation-Algebras*

begin

1 Representation of Stone Relation Algebras

We show that Stone relation algebras can be represented by matrices if we assume a point axiom. The matrix indices and entries and the point axiom are based on the concepts of ideals and ideal-points. We start with general results about sets and finite suprema.

lemma *finite-ne-subset-induct'* [*consumes 3*, *case-names singleton insert*]:
assumes *finite F*
and $F \neq \{\}$
and $F \subseteq S$
and *singleton*: $\bigwedge x\ .\ x \in S \Longrightarrow P\ \{x\}$
and *insert*: $\bigwedge x\ F\ .\ finite\ F \Longrightarrow F \neq \{\} \Longrightarrow F \subseteq S \Longrightarrow x \in S \Longrightarrow x \notin F \Longrightarrow P\ F \Longrightarrow P\ (insert\ x\ F)$
shows *P F*
⟨*proof*⟩

context *order-bot*
begin

abbreviation *atom* :: $'a \Rightarrow bool$
where $atom\ x \equiv x \neq bot \wedge (\forall y\ .\ y \neq bot \wedge y \leq x \longrightarrow y = x)$

end

context *semilattice-sup*
begin

lemma *nested-sup-fin*:
assumes *finite X*
and $X \neq \{\}$
and *finite Y*
and $Y \neq \{\}$
shows *Sup-fin* { *Sup-fin* { *f x y* | *x* . $x \in X$ } | *y* . $y \in Y$ } = *Sup-fin* { *f x y* | *x y* . $x \in X \wedge y \in Y$ }
⟨*proof*⟩

end

context *bounded-semilattice-sup-bot*
begin

lemma *one-point-sup-fin*:
assumes *finite X*
and $y \in X$
shows *Sup-fin* { (*if x* = *y then f x else bot*) | *x* . $x \in X$ } = *f y*
⟨*proof*⟩

end

1.1 Ideals and Ideal-Points

We study ideals in Stone relation algebras, which are elements that are both a vector and a covector. We include general results about Stone relation algebras.

context *times-top*
begin

abbreviation *ideal* :: *'a* ⇒ *bool* **where** *ideal x* ≡ *vector x* ∧ *covector x*

end

context *bounded-non-associative-left-semiring*
begin

lemma *ideal-fixpoint*:
ideal x ⟷ *top* ∗ *x* ∗ *top* = *x*
⟨*proof*⟩

lemma *ideal-top-closed*:
ideal top
⟨*proof*⟩

end

context *bounded-idempotent-left-semiring*
begin

lemma *ideal-mult-closed*:
ideal $x \Longrightarrow$ *ideal* $y \Longrightarrow$ *ideal* $(x * y)$
⟨*proof*⟩

end

context *bounded-idempotent-left-zero-semiring*
begin

lemma *ideal-sup-closed*:
ideal $x \Longrightarrow$ *ideal* $y \Longrightarrow$ *ideal* $(x \sqcup y)$
⟨*proof*⟩

end

context *idempotent-semiring*
begin

lemma *sup-fin-sum*:
fixes $f :: 'b{::}finite \Rightarrow 'a$
shows *Sup-fin* $\{ f\ x \mid x\ .\ x \in UNIV \} = (\bigsqcup_x f\ x)$
⟨*proof*⟩

end

context *stone-relation-algebra*
begin

lemma *dedekind-univalent*:
assumes *univalent* y
shows $x * y \sqcap z = (x \sqcap z * y^T) * y$
⟨*proof*⟩

lemma *dedekind-injective*:
assumes *injective* x
shows $x * y \sqcap z = x * (y \sqcap x^T * z)$
⟨*proof*⟩

lemma *domain-vector-conv*:
$1 \sqcap x * top = 1 \sqcap x * x^T$
⟨*proof*⟩

lemma *domain-vector-covector*:
$1 \sqcap x * top = 1 \sqcap top * x^T$
⟨*proof*⟩

lemma *domain-covector-conv*:
$1 \sqcap top * x^T = 1 \sqcap x * x^T$

⟨*proof*⟩

lemma *ideal-bot-closed*:
ideal bot
⟨*proof*⟩

lemma *ideal-inf-closed*:
ideal $x \Longrightarrow$ *ideal* $y \Longrightarrow$ *ideal* $(x \sqcap y)$
⟨*proof*⟩

lemma *ideal-conv-closed*:
ideal $x \Longrightarrow$ *ideal* (x^T)
⟨*proof*⟩

lemma *ideal-complement-closed*:
ideal $x \Longrightarrow$ *ideal* $(-x)$
⟨*proof*⟩

lemma *ideal-conv-id*:
ideal $x \Longrightarrow x = x^T$
⟨*proof*⟩

lemma *ideal-mult-inf*:
ideal $x \Longrightarrow$ *ideal* $y \Longrightarrow x * y = x \sqcap y$
⟨*proof*⟩

lemma *ideal-mult-import*:
ideal $x \Longrightarrow y * z \sqcap x = (y \sqcap x) * (z \sqcap x)$
⟨*proof*⟩

lemma *point-meet-one*:
point $x \Longrightarrow x * x^T = x \sqcap 1$
⟨*proof*⟩

lemma *below-point-eq-domain*:
point $x \Longrightarrow y \leq x \Longrightarrow y = x * x^T * y$
⟨*proof*⟩

lemma *covector-mult-vector-ideal*:
vector $x \Longrightarrow$ *vector* $z \Longrightarrow$ *ideal* $(x^T * y * z)$
⟨*proof*⟩

abbreviation *ideal-point* :: $'a \Rightarrow$ *bool* **where** *ideal-point* $x \equiv$ *point* $x \wedge (\forall y\ z\ .$ *point* $y \wedge$ *ideal* $z \wedge z \neq$ *bot* $\wedge\ y * z \leq x \longrightarrow y \leq x)$

lemma *different-ideal-points-disjoint*:
assumes *ideal-point p*
and *ideal-point q*
and $p \neq q$

shows $p \sqcap q = bot$
⟨*proof*⟩

lemma *points-disjoint-iff*:
assumes *vector x*
shows $x \sqcap y = bot \longleftrightarrow x^T * y = bot$
⟨*proof*⟩

lemma *different-ideal-points-disjoint-2*:
assumes *ideal-point p*
and *ideal-point q*
and $p \neq q$
shows $p^T * q = bot$
⟨*proof*⟩

lemma *mult-right-dist-sup-fin*:
assumes *finite X*
and $X \neq \{\}$
shows *Sup-fin* $\{ f\ x \mid x::'b\ .\ x \in X \} * y =$ *Sup-fin* $\{ f\ x * y \mid x\ .\ x \in X \}$
⟨*proof*⟩

lemma *mult-left-dist-sup-fin*:
assumes *finite X*
and $X \neq \{\}$
shows $y *$ *Sup-fin* $\{ f\ x \mid x::'b\ .\ x \in X \} =$ *Sup-fin* $\{ y * f\ x \mid x\ .\ x \in X \}$
⟨*proof*⟩

lemma *inf-left-dist-sup-fin*:
assumes *finite X*
and $X \neq \{\}$
shows $y \sqcap$ *Sup-fin* $\{ f\ x \mid x::'b\ .\ x \in X \} =$ *Sup-fin* $\{ y \sqcap f\ x \mid x\ .\ x \in X \}$
⟨*proof*⟩

lemma *top-one-sup-fin-iff*:
assumes *finite P*
and $P \neq \{\}$
and $\forall\, p{\in}P\ .$ *point p*
shows *top* = *Sup-fin P* $\longleftrightarrow$ *1* = *Sup-fin* $\{ p * p^T \mid p\ .\ p \in P \}$
⟨*proof*⟩

abbreviation *ideals* :: *'a set* **where** *ideals* $\equiv \{ x\ .$ *ideal x* $\}$
abbreviation *ideal-points* :: *'a set* **where** *ideal-points* $\equiv \{ x\ .$ *ideal-point x* $\}$

lemma *surjective-vector-top*:
surjective x $\Longrightarrow$ *vector x* $\Longrightarrow x^T * x =$ *top*
⟨*proof*⟩

lemma *point-mult-top*:
point x $\Longrightarrow x^T * x =$ *top*

⟨*proof*⟩

end

1.2 Point Axiom

The following class captures the point axiom for Stone relation algebras.

class *stone-relation-algebra-pa* = *stone-relation-algebra* +
  assumes *finite-ideal-points*: *finite ideal-points*
  assumes *ne-ideal-points*: *ideal-points* ≠ {}
  assumes *top-sup-ideal-points*: *top* = *Sup-fin ideal-points*
begin

lemma *one-sup-ideal-points*:
  1 = *Sup-fin* { *p* ∗ *p*^{T} | *p* . *ideal-point p* }
⟨*proof*⟩

lemma *ideal-point-rep-1*:
  x = *Sup-fin* { *p* ∗ *p*^{T} ∗ *x* ∗ *q* ∗ *q*^{T} | *p q* . *ideal-point p* ∧ *ideal-point q* }
⟨*proof*⟩

lemma *atom-below-ideal-point*:
  assumes *atom a*
    shows ∃ *p* . *ideal-point p* ∧ *a* ≤ *p*
⟨*proof*⟩

end

1.3 Ideals, Ideal-Points and Matrices as Types

Stone relation algebras will be represented by matrices with ideal-points as entries and ideals as indices. To define the type of such matrices, we first derive types for the set of ideals and ideal-points.

typedef (**overloaded**) *'a ideal* = *ideals*::*'a*::*stone-relation-algebra-pa set*
  ⟨*proof*⟩

setup-lifting *type-definition-ideal*

instantiation *ideal* :: (*stone-relation-algebra-pa*) *stone-algebra*
begin

lift-definition *uminus-ideal* :: *'a ideal* ⇒ *'a ideal* **is** *uminus*
  ⟨*proof*⟩

lift-definition *inf-ideal* :: *'a ideal* ⇒ *'a ideal* ⇒ *'a ideal* **is** *inf*
  ⟨*proof*⟩

lift-definition *sup-ideal* :: *'a ideal* ⇒ *'a ideal* ⇒ *'a ideal* **is** *sup*

⟨*proof*⟩

lift-definition *bot-ideal* :: *'a ideal* **is** *bot*
⟨*proof*⟩

lift-definition *top-ideal* :: *'a ideal* **is** *top*
⟨*proof*⟩

lift-definition *less-eq-ideal* :: *'a ideal* ⇒ *'a ideal* ⇒ *bool* **is** *less-eq* ⟨*proof*⟩

lift-definition *less-ideal* :: *'a ideal* ⇒ *'a ideal* ⇒ *bool* **is** *less* ⟨*proof*⟩

instance
⟨*proof*⟩

end

instantiation *ideal* :: (*stone-relation-algebra-pa*) *stone-relation-algebra*
begin

lift-definition *conv-ideal* :: *'a ideal* ⇒ *'a ideal* **is** *id*
⟨*proof*⟩

lift-definition *times-ideal* :: *'a ideal* ⇒ *'a ideal* ⇒ *'a ideal* **is** *inf*
⟨*proof*⟩

lift-definition *one-ideal* :: *'a ideal* **is** *top*
⟨*proof*⟩

instance
⟨*proof*⟩

end

typedef (**overloaded**) *'a ideal-point* = *ideal-points*::*'a*::*stone-relation-algebra-pa set*
⟨*proof*⟩

instantiation *ideal-point* :: (*stone-relation-algebra-pa*) *finite*
begin

instance
⟨*proof*⟩

end

type-synonym *'a ideal-matrix* = (*'a ideal-point*,*'a ideal*) *square*

interpretation *ideal-matrix-stone-relation-algebra*: *stone-relation-algebra* **where**

sup = *sup-matrix* **and** *inf* = *inf-matrix* **and** *less-eq* = *less-eq-matrix* **and** *less* = *less-matrix* **and** *bot* = *bot-matrix*::*'a*::*stone-relation-algebra-pa ideal-matrix* **and** *top* = *top-matrix* **and** *uminus* = *uminus-matrix* **and** *one* = *one-matrix* **and** *times* = *times-matrix* **and** *conv* = *conv-matrix*
⟨*proof*⟩

lemma *ideal-point-rep-2*:
assumes x = *Sup-fin* { *Rep-ideal-point* p ∗ *Rep-ideal* (*f* p q) ∗ (*Rep-ideal-point* q)^T | p q . *True* }
shows *f* r s = *Abs-ideal* ((*Rep-ideal-point* r)^T ∗ x ∗ (*Rep-ideal-point* s))
⟨*proof*⟩

1.4 Isomorphism

The following two functions comprise the isomorphism between Stone relation algebras and matrices. We prove that they are inverses of each other and that the first one is a homomorphism.

definition *sra-to-mat* :: *'a*::*stone-relation-algebra-pa* ⇒ *'a ideal-matrix*
where *sra-to-mat* x ≡ λ(p,q) . *Abs-ideal* ((*Rep-ideal-point* p)^T ∗ x ∗ *Rep-ideal-point* q)

definition *mat-to-sra* :: *'a*::*stone-relation-algebra-pa ideal-matrix* ⇒ *'a*
where *mat-to-sra* f ≡ *Sup-fin* { *Rep-ideal-point* p ∗ *Rep-ideal* (f (p,q)) ∗ (*Rep-ideal-point* q)^T | p q . *True* }

lemma *sra-mat-sra*:
mat-to-sra (*sra-to-mat* x) = x
⟨*proof*⟩

lemma *mat-sra-mat*:
sra-to-mat (*mat-to-sra* f) = f
⟨*proof*⟩

lemma *sra-to-mat-sup-homomorphism*:
sra-to-mat (x ⊔ y) = *sra-to-mat* x ⊔ *sra-to-mat* y
⟨*proof*⟩

lemma *sra-to-mat-inf-homomorphism*:
sra-to-mat (x ⊓ y) = *sra-to-mat* x ⊓ *sra-to-mat* y
⟨*proof*⟩

lemma *sra-to-mat-conv-homomorphism*:
sra-to-mat (x^T) = (*sra-to-mat* x)^t
⟨*proof*⟩

lemma *sra-to-mat-complement-homomorphism*:
sra-to-mat (−x) = −(*sra-to-mat* x)
⟨*proof*⟩

lemma *sra-to-mat-bot-homomorphism*:
sra-to-mat bot = *bot*
⟨*proof*⟩

lemma *sra-to-mat-top-homomorphism*:
sra-to-mat top = *top*
⟨*proof*⟩

lemma *sra-to-mat-one-homomorphism*:
sra-to-mat 1 = *one-matrix*
⟨*proof*⟩

lemma *Abs-ideal-dist-sup-fin*:
assumes *finite X*
and $X \neq \{\}$
and $\forall x{\in}X$. *ideal* (*f x*)
shows *Abs-ideal* (*Sup-fin* { *f x* | *x* . *x* ∈ *X* }) = *Sup-fin* { *Abs-ideal* (*f x*) | *x* . *x* ∈ *X* }
⟨*proof*⟩

lemma *sra-to-mat-mult-homomorphism*:
sra-to-mat (*x* ∗ *y*) = *sra-to-mat x* ⊙ *sra-to-mat y*
⟨*proof*⟩

end
theory *Cardinality*

imports *List−Infinite.InfiniteSet2 Representation*

begin

context *uminus*
begin

no-notation *uminus* (− - [*81*] *80*)

end

2 Atoms Below an Element in Partial Orders

We define the set and the number of atoms below an element in a partial order. To handle infinitely many atoms we use *enat*, which are natural numbers with infinity, and *icard*, which modifies *card* by giving a separate option of being infinite. We include general results about *enat*, *icard*, sets functions and atoms.

lemma *enat-mult-strict-mono*:
assumes *a* < *b c* < *d* (*0*::*enat*) < *b 0* ≤ *c*
shows *a* ∗ *c* < *b* ∗ *d*

⟨*proof*⟩

lemma *enat-mult-strict-mono'*:
assumes $a < b$ **and** $c < d$ **and** $(0::enat) \leq a$ **and** $0 \leq c$
shows $a * c < b * d$
⟨*proof*⟩

lemma *finite-icard-card*:
$finite\ A \Longrightarrow icard\ A = icard\ B \Longrightarrow card\ A = card\ B$
⟨*proof*⟩

lemma *icard-eq-sum*:
$finite\ A \Longrightarrow icard\ A = sum\ (\lambda x.\ 1)\ A$
⟨*proof*⟩

lemma *icard-sum-constant-function*:
assumes $\forall\, x{\in}A\ .\ f\ x = c$
and *finite A*
shows $sum\ f\ A = (icard\ A) * c$
⟨*proof*⟩

lemma *icard-le-finite*:
assumes $icard\ A \leq icard\ B$
and *finite B*
shows *finite A*
⟨*proof*⟩

lemma *bij-betw-same-icard*:
$bij\text{-}betw\ f\ A\ B \Longrightarrow icard\ A = icard\ B$
⟨*proof*⟩

lemma *surj-icard-le*: $B \subseteq f\ `\ A \Longrightarrow icard\ B \leq icard\ A$
⟨*proof*⟩

lemma *icard-image-part-le*:
assumes $\forall\, x{\in}A\ .\ f\ x \subseteq B$
and $\forall\, x{\in}A\ .\ f\ x \neq \{\}$
and $\forall\, x{\in}A\ .\ \forall\, y{\in}A\ .\ x \neq y \longrightarrow f\ x \cap f\ y = \{\}$
shows $icard\ A \leq icard\ B$
⟨*proof*⟩

lemma *finite-image-part-le*:
assumes $\forall\, x{\in}A\ .\ f\ x \subseteq B$
and $\forall\, x{\in}A\ .\ f\ x \neq \{\}$
and $\forall\, x{\in}A\ .\ \forall\, y{\in}A\ .\ x \neq y \longrightarrow f\ x \cap f\ y = \{\}$
and *finite B*
shows *finite A*
⟨*proof*⟩

context *semiring-1*
begin

lemma *sum-constant-function*:
  assumes $\forall\, x{\in}A\ .\ f\ x = c$
    shows *sum* $f\ A =$ *of-nat* $(card\ A) * c$
⟨*proof*⟩

end

context *order*
begin

lemma *ne-finite-has-minimal*:
  assumes *finite* S
      and $S \neq \{\}$
    shows $\exists\, m{\in}S\ .\ \forall\, x{\in}S\ .\ x \leq m \longrightarrow x = m$
⟨*proof*⟩

end

context *order-bot*
begin

abbreviation *atoms-below* :: $'a \Rightarrow 'a\ set$ (AB)
  where *atoms-below* $x \equiv \{\ a\ .\ atom\ a \wedge a \leq x\ \}$

definition *num-atoms-below* :: $'a \Rightarrow enat$ (nAB)
  where *num-atoms-below* $x \equiv$ *icard* $(atoms\text{-}below\ x)$

lemma *AB-iso*:
  $x \leq y \Longrightarrow AB\ x \subseteq AB\ y$
  ⟨*proof*⟩

lemma *AB-bot*:
  $AB\ bot = \{\}$
  ⟨*proof*⟩

lemma *nAB-bot*:
  $nAB\ bot = 0$
⟨*proof*⟩

lemma *AB-atom*:
  $atom\ a \longleftrightarrow AB\ a = \{a\}$
  ⟨*proof*⟩

lemma *nAB-atom*:
  $atom\ a \Longrightarrow nAB\ a = 1$
⟨*proof*⟩

lemma *nAB-iso*:
$x \leq y \Longrightarrow nAB\ x \leq nAB\ y$
⟨*proof*⟩

end

context *bounded-semilattice-sup-bot*
begin

lemma *nAB-iso-sup*:
$nAB\ x \leq nAB\ (x \sqcup y)$
⟨*proof*⟩

end

context *bounded-lattice*
begin

lemma *different-atoms-disjoint*:
$atom\ x \Longrightarrow atom\ y \Longrightarrow x \neq y \Longrightarrow x \sqcap y = bot$
⟨*proof*⟩

lemma *AB-dist-inf*:
$AB\ (x \sqcap y) = AB\ x \cap AB\ y$
⟨*proof*⟩

lemma *AB-iso-inf*:
$AB\ (x \sqcap y) \subseteq AB\ x$
⟨*proof*⟩

lemma *AB-iso-sup*:
$AB\ x \subseteq AB\ (x \sqcup y)$
⟨*proof*⟩

lemma *AB-disjoint*:
assumes $x \sqcap y = bot$
shows $AB\ x \cap AB\ y = \{\}$
⟨*proof*⟩

lemma *nAB-iso-inf*:
$nAB\ (x \sqcap y) \leq nAB\ x$
⟨*proof*⟩

end

context *distrib-lattice-bot*
begin

lemma *atom-in-sup*:
assumes *atom a*
and $a \leq x \sqcup y$
shows $a \leq x \vee a \leq y$
⟨*proof*⟩

lemma *atom-in-sup-iff*:
assumes *atom a*
shows $a \leq x \sqcup y \longleftrightarrow a \leq x \vee a \leq y$
⟨*proof*⟩

lemma *atom-in-sup-xor*:
$atom\ a \Longrightarrow a \leq x \sqcup y \Longrightarrow x \sqcap y = bot \Longrightarrow (a \leq x \wedge \neg\ a \leq y) \vee (\neg\ a \leq x \wedge a \leq y)$
⟨*proof*⟩

lemma *atom-in-sup-xor-iff*:
assumes *atom a*
and $x \sqcap y = bot$
shows $a \leq x \sqcup y \longleftrightarrow (a \leq x \wedge \neg\ a \leq y) \vee (\neg\ a \leq x \wedge a \leq y)$
⟨*proof*⟩

lemma *AB-dist-sup*:
$AB\ (x \sqcup y) = AB\ x \cup AB\ y$
⟨*proof*⟩

end

context *bounded-distrib-lattice*
begin

lemma *nAB-add*:
$nAB\ x + nAB\ y = nAB\ (x \sqcup y) + nAB\ (x \sqcap y)$
⟨*proof*⟩

lemma *nAB-split-disjoint*:
assumes $x \sqcap y = bot$
shows $nAB\ (x \sqcup y) = nAB\ x + nAB\ y$
⟨*proof*⟩

end

context *p-algebra*
begin

lemma *atom-in-p*:
$atom\ a \Longrightarrow a \leq x \vee a \leq -x$
⟨*proof*⟩

lemma *atom-in-p-xor*:
$atom\ a \Longrightarrow (a \leq x \wedge \neg\ a \leq -x) \vee (\neg\ a \leq x \wedge a \leq -x)$
⟨*proof*⟩

The following two lemmas also hold in distributive lattices with a least element (see above). However, p-algebras are not necessarily distributive, so the following results are indepenent.

lemma *atom-in-sup'*:
$atom\ a \Longrightarrow a \leq x \sqcup y \Longrightarrow a \leq x \vee a \leq y$
⟨*proof*⟩

lemma *AB-dist-sup'*:
$AB\ (x \sqcup y) = AB\ x \cup AB\ y$
⟨*proof*⟩

lemma *AB-split-1*:
$AB\ x = AB\ ((x \sqcap y) \sqcup (x \sqcap -y))$
⟨*proof*⟩

lemma *AB-split-2*:
$AB\ x = AB\ (x \sqcap y) \cup AB\ (x \sqcap -y)$
⟨*proof*⟩

lemma *AB-split-2-disjoint*:
$AB\ (x \sqcap y) \cap AB\ (x \sqcap -y) = \{\}$
⟨*proof*⟩

lemma *AB-pp*:
$AB\ (--x) = AB\ x$
⟨*proof*⟩

lemma *nAB-pp*:
$nAB\ (--x) = nAB\ x$
⟨*proof*⟩

lemma *nAB-split-1*:
$nAB\ x = nAB\ ((x \sqcap y) \sqcup (x \sqcap -\ y))$
⟨*proof*⟩

lemma *nAB-split-2*:
$nAB\ x = nAB\ (x \sqcap y) + nAB\ (x \sqcap -y)$
⟨*proof*⟩

end

3 Atoms Below an Element in Stone Relation Algebras

We extend our study of atoms below an element to Stone relation algebras. We consider combinations of the following five assumptions: the Stone relation algebra is atomic, atom-rectangular, atom-simple, a relation algebra, or has finitely many atoms. We include general properties of atoms, rectangles and simple elements.

context *stone-relation-algebra*
begin

abbreviation *rectangle* :: $'a \Rightarrow bool$ **where** $rectangle\ x \equiv x * top * x \leq x$
abbreviation *simple* :: $'a \Rightarrow bool$ **where** $simple\ x \equiv top * x * top = top$

lemma *rectangle-eq*:
$rectangle\ x \longleftrightarrow x * top * x = x$
⟨*proof*⟩

lemma *arc-univalent-injective-rectangle-simple*:
$arc\ a \longleftrightarrow univalent\ a \wedge injective\ a \wedge rectangle\ a \wedge simple\ a$
⟨*proof*⟩

lemma *conv-atom*:
$atom\ x \Longrightarrow atom\ (x^T)$
⟨*proof*⟩

lemma *conv-atom-iff*:
$atom\ x \longleftrightarrow atom\ (x^T)$
⟨*proof*⟩

lemma *counterexample-different-atoms-top-disjoint*:
$atom\ x \Longrightarrow atom\ y \Longrightarrow x \neq y \Longrightarrow x * top \sqcap y = bot$
nitpick[*expect=genuine*,*card=4*]
⟨*proof*⟩

lemma *counterexample-different-univalent-atoms-top-disjoint*:
$atom\ x \Longrightarrow univalent\ x \Longrightarrow atom\ y \Longrightarrow univalent\ y \Longrightarrow x \neq y \Longrightarrow x * top \sqcap y = bot$
nitpick[*expect=genuine*,*card=4*]
⟨*proof*⟩

lemma *AB-card-4-1*:
$a \leq x \wedge a \leq y \longleftrightarrow a \leq x \sqcup y \wedge a \leq x \sqcap y$
⟨*proof*⟩

lemma *AB-card-4-2*:
assumes *atom a*
shows $(a \leq x \wedge \neg\ a \leq y) \vee (\neg\ a \leq x \wedge a \leq y) \longleftrightarrow a \leq x \sqcup y \wedge \neg\ a \leq x \sqcap y$
⟨*proof*⟩

lemma *AB-card-4-3*:

assumes *atom a*
shows $\neg\ a \leq x \wedge \neg\ a \leq y \longleftrightarrow \neg\ a \leq x \sqcup y \wedge \neg\ a \leq x \sqcap y$
⟨*proof*⟩

lemma *AB-card-5-1*:
assumes *atom a*
and $a \leq x^T * y \sqcap z$
shows $x * a \sqcap y \leq x * z \sqcap y$
and $x * a \sqcap y \neq bot$
⟨*proof*⟩

lemma *AB-card-5-2*:
assumes *univalent x*
and *atom a*
and *atom b*
and $b \leq x^T * y \sqcap z$
and $a \neq b$
shows $(x * a \sqcap y) \sqcap (x * b \sqcap y) = bot$
and $x * a \sqcap y \neq x * b \sqcap y$
⟨*proof*⟩

lemma *AB-card-6-0*:
assumes *univalent x*
and *atom a*
and $a \leq x$
and *atom b*
and $b \leq x$
and $a \neq b$
shows $a * top \sqcap b * top = bot$
⟨*proof*⟩

lemma *AB-card-6-1*:
assumes *atom a*
and $a \leq x \sqcap y * z^T$
shows $a * z \sqcap y \leq x * z \sqcap y$
and $a * z \sqcap y \neq bot$
⟨*proof*⟩

lemma *AB-card-6-2*:
assumes *univalent x*
and *atom a*
and $a \leq x \sqcap y * z^T$
and *atom b*
and $b \leq x \sqcap y * z^T$
and $a \neq b$
shows $(a * z \sqcap y) \sqcap (b * z \sqcap y) = bot$
and $a * z \sqcap y \neq b * z \sqcap y$
⟨*proof*⟩

lemma *nAB-conv*:
$nAB\ x = nAB\ (x^T)$
⟨*proof*⟩

lemma *domain-atom*:
assumes *atom a*
shows *atom* $(a * top \sqcap 1)$
⟨*proof*⟩

lemma *codomain-atom*:
assumes *atom a*
shows *atom* $(top * a \sqcap 1)$
⟨*proof*⟩

lemma *atom-rectangle-atom-one-rep*:
$(\forall a\ .\ atom\ a \longrightarrow a * top * a \leq a) \longleftrightarrow (\forall a\ .\ atom\ a \wedge a \leq 1 \longrightarrow a * top * a \leq 1)$
⟨*proof*⟩

lemma *AB-card-2-1*:
assumes $a * top * a \leq a$
shows $(a * top \sqcap 1) * top * (top * a \sqcap 1) = a$
⟨*proof*⟩

lemma *atomsimple-atom1simple*:
$(\forall a\ .\ atom\ a \longrightarrow top * a * top = top) \longleftrightarrow (\forall a\ .\ atom\ a \wedge a \leq 1 \longrightarrow top * a * top = top)$
⟨*proof*⟩

lemma *AB-card-2-2*:
assumes *atom a*
and $a \leq 1$
and *atom b*
and $b \leq 1$
and $\forall a\ .\ atom\ a \longrightarrow top * a * top = top$
shows $a * top * b * top \sqcap 1 = a$ **and** $top * a * top * b \sqcap 1 = b$
⟨*proof*⟩

abbreviation *dom-cod* :: $'a \Rightarrow 'a \times 'a$
where $dom\text{-}cod\ a \equiv (a * top \sqcap 1,\ top * a \sqcap 1)$

lemma *dom-cod-atoms-1*:
$dom\text{-}cod\ `\ AB\ top \subseteq AB\ 1 \times AB\ 1$
⟨*proof*⟩

end

3.1 Atomic

class *stone-relation-algebra-atomic* = *stone-relation-algebra* +
  assumes *atomic*: $x \neq bot \longrightarrow (\exists\, a\ .\ atom\ a \wedge a \leq x)$
begin

lemma *AB-nonempty*:
  $x \neq bot \Longrightarrow AB\ x \neq \{\}$
  ⟨*proof*⟩

lemma *AB-nonempty-iff*:
  $x \neq bot \longleftrightarrow AB\ x \neq \{\}$
  ⟨*proof*⟩

lemma *atomsimple-simple*:
  $(\forall\, a\ .\ a \neq bot \longrightarrow top * a * top = top) \longleftrightarrow (\forall\, a\ .\ atom\ a \longrightarrow top * a * top = top)$
⟨*proof*⟩

lemma *AB-card-2-3*:
  assumes $a \neq bot$
    and $a \leq 1$
    and $b \neq bot$
    and $b \leq 1$
    and $\forall\, a\ .\ a \neq bot \longrightarrow top * a * top = top$
   shows $a * top * b * top \sqcap 1 = a$ **and** $top * a * top * b \sqcap 1 = b$
⟨*proof*⟩

lemma *injective-down-closed*:
  $x \leq y \Longrightarrow injective\ y \Longrightarrow injective\ x$
  ⟨*proof*⟩

lemma *univalent-down-closed*:
  $x \leq y \Longrightarrow univalent\ y \Longrightarrow univalent\ x$
  ⟨*proof*⟩

lemma *nAB-bot-iff*:
  $x = bot \longleftrightarrow nAB\ x = 0$
  ⟨*proof*⟩

It is unclear if *atomic* is necessary for the following two results, but it seems likely.

lemma *nAB-univ-comp-meet*:
  assumes *univalent x*
   shows $nAB\ (x^T * y \sqcap z) \leq nAB\ (x * z \sqcap y)$
⟨*proof*⟩

lemma *nAB-univ-meet-comp*:
  assumes *univalent x*
   shows $nAB\ (x \sqcap y * z^T) \leq nAB\ (x * z \sqcap y)$

⟨*proof*⟩

end

3.2 Atom-rectangular

class *stone-relation-algebra-atomrect* = *stone-relation-algebra* +
  assumes *atomrect*: *atom a* ⟶ *rectangle a*
begin

lemma *atomrect-eq*:
  atom a ⟹ *a* ∗ *top* ∗ *a* = *a*
  ⟨*proof*⟩

lemma *AB-card-2-4*:
  assumes *atom a*
    shows (*a* ∗ *top* ⊓ *1*) ∗ *top* ∗ (*top* ∗ *a* ⊓ *1*) = *a*
  ⟨*proof*⟩

lemma *simple-atom-2*:
  assumes *atom a*
      and *a* ≤ *1*
      and *atom b*
      and *b* ≤ *1*
      and *x* ≠ *bot*
      and *x* ≤ *a* ∗ *top* ∗ *b*
    shows *x* = *a* ∗ *top* ∗ *b*
⟨*proof*⟩

lemma *dom-cod-inj-atoms*:
  inj-on dom-cod (*AB top*)
⟨*proof*⟩

lemma *finite-AB-iff*:
  finite (*AB top*) ⟷ *finite* (*AB 1*)
⟨*proof*⟩

lemma *nAB-top-1*:
  nAB top ≤ *nAB 1* ∗ *nAB 1*
⟨*proof*⟩

lemma *atom-vector-injective*:
  assumes *atom x*
    shows *injective* (*x* ∗ *top*)
⟨*proof*⟩

lemma *atom-injective*:
  atom x ⟹ *injective x*
  ⟨*proof*⟩

lemma *atom-covector-univalent*:
atom x $\Longrightarrow$ *univalent* (*top* $*$ *x*)
⟨*proof*⟩

lemma *atom-univalent*:
atom x $\Longrightarrow$ *univalent x*
⟨*proof*⟩

lemma *counterexample-atom-simple*:
atom x $\Longrightarrow$ *simple x*
nitpick[*expect=genuine,card=3*]
⟨*proof*⟩

lemma *symmetric-atom-below-1*:
assumes *atom x*
and $x = x^T$
shows $x \leq 1$
⟨*proof*⟩

end

3.3 Atomic and Atom-Rectangular

class *stone-relation-algebra-atomic-atomrect* = *stone-relation-algebra-atomic* + *stone-relation-algebra-atomrect*
begin

lemma *point-dense*:
assumes $x \neq bot$
and $x \leq 1$
shows $\exists a\ .\ a \neq bot \wedge a * top * a \leq 1 \wedge a \leq x$
⟨*proof*⟩

end

3.4 Atom-simple

class *stone-relation-algebra-atomsimple* = *stone-relation-algebra* +
assumes *atomsimple*: *atom a* $\longrightarrow$ *simple a*
begin

lemma *AB-card-2-5*:
assumes *atom a*
and $a \leq 1$
and *atom b*
and $b \leq 1$
shows $a * top * b * top \sqcap 1 = a$ **and** $top * a * top * b \sqcap 1 = b$
⟨*proof*⟩

lemma *simple-atom-1*:
atom a $\Longrightarrow$ *atom b* $\Longrightarrow$ $a * top * b \neq bot$
⟨*proof*⟩

end

3.5 Atomic and Atom-simple

class *stone-relation-algebra-atomic-atomsimple* = *stone-relation-algebra-atomic* + *stone-relation-algebra-atomsimple*
begin

lemma *simple*:
$a \neq bot \Longrightarrow top * a * top = top$
⟨*proof*⟩

lemma *AB-card-2-6*:
assumes $a \neq bot$
and $a \leq 1$
and $b \neq bot$
and $b \leq 1$
shows $a * top * b * top \sqcap 1 = a$ **and** $top * a * top * b \sqcap 1 = b$
⟨*proof*⟩

lemma *dom-cod-atoms-2*:
$AB\ 1 \times AB\ 1 \subseteq dom\text{-}cod\ `\ AB\ top$
⟨*proof*⟩

lemma *dom-cod-atoms*:
$AB\ 1 \times AB\ 1 = dom\text{-}cod\ `\ AB\ top$
⟨*proof*⟩

end

3.6 Atom-rectangular and Atom-simple

class *stone-relation-algebra-atomrect-atomsimple* = *stone-relation-algebra-atomrect* + *stone-relation-algebra-atomsimple*
begin

lemma *simple-atom*:
assumes *atom a*
and $a \leq 1$
and *atom b*
and $b \leq 1$
shows *atom* $(a * top * b)$
⟨*proof*⟩

lemma *nAB-top-2*:
$nAB\ 1 * nAB\ 1 \leq nAB\ top$

⟨*proof*⟩

lemma *nAB-top*:
nAB 1 ∗ nAB 1 = nAB top
⟨*proof*⟩

lemma *atom-covector-mapping*:
atom a ⟹ mapping (top ∗ a)
⟨*proof*⟩

lemma *atom-covector-regular*:
atom a ⟹ regular (top ∗ a)
⟨*proof*⟩

lemma *atom-vector-bijective*:
atom a ⟹ bijective (a ∗ top)
⟨*proof*⟩

lemma *atom-vector-regular*:
atom a ⟹ regular (a ∗ top)
⟨*proof*⟩

lemma *atom-rectangle-regular*:
atom a ⟹ regular (a ∗ top ∗ a)
⟨*proof*⟩

lemma *atom-regular*:
atom a ⟹ regular a
⟨*proof*⟩

end

3.7 Atomic, Atom-rectangular and Atom-simple

class *stone-relation-algebra-atomic-atomrect-atomsimple =*
stone-relation-algebra-atomic + stone-relation-algebra-atomrect +
stone-relation-algebra-atomsimple
begin

subclass *stone-relation-algebra-atomic-atomrect* ⟨*proof*⟩
subclass *stone-relation-algebra-atomic-atomsimple* ⟨*proof*⟩
subclass *stone-relation-algebra-atomrect-atomsimple* ⟨*proof*⟩

lemma *nAB-atom-iff*:
atom a ⟷ nAB a = 1
⟨*proof*⟩

end

3.8 Finitely Many Atoms

class *stone-relation-algebra-finiteatoms* = *stone-relation-algebra* +
  assumes *finiteatoms*: *finite* { *a* . *atom a* }
begin

lemma *finite-AB*:
  finite (*AB x*)
  ⟨*proof*⟩

lemma *nAB-top-finite*:
  nAB top ≠ ∞
  ⟨*proof*⟩

end

3.9 Atomic and Finitely Many Atoms

class *stone-relation-algebra-atomic-finiteatoms* = *stone-relation-algebra-atomic* +
stone-relation-algebra-finiteatoms
begin

lemma *finite-ideal-points*:
  finite { *p* . *ideal-point p* }
⟨*proof*⟩

end

3.10 Atom-rectangular and Finitely Many Atoms

class *stone-relation-algebra-atomrect-finiteatoms* =
stone-relation-algebra-atomrect + *stone-relation-algebra-finiteatoms*

3.11 Atomic, Atom-rectangular and Finitely Many Atoms

class *stone-relation-algebra-atomic-atomrect-finiteatoms* =
stone-relation-algebra-atomic + *stone-relation-algebra-atomrect* +
stone-relation-algebra-finiteatoms
begin

subclass *stone-relation-algebra-atomic-atomrect* ⟨*proof*⟩
subclass *stone-relation-algebra-atomic-finiteatoms* ⟨*proof*⟩
subclass *stone-relation-algebra-atomrect-finiteatoms* ⟨*proof*⟩

lemma *counterexample-nAB-atom-iff*:
  atom x ⟷ *nAB x* = *1*
  nitpick[*expect*=*genuine*,*card*=*3*]
  ⟨*proof*⟩

lemma *counterexample-nAB-top-iff-eq*:

nAB x = nAB top ⟷ x = top
nitpick[*expect=genuine,card=3*]
⟨*proof*⟩

lemma *counterexample-nAB-top-iff-leq*:
nAB top ≤ nAB x ⟷ x = top
nitpick[*expect=genuine,card=3*]
⟨*proof*⟩

end

3.12 Atom-simple and Finitely Many Atoms

class *stone-relation-algebra-atomsimple-finiteatoms =*
stone-relation-algebra-atomsimple + stone-relation-algebra-finiteatoms

3.13 Atomic, Atom-simple and Finitely Many Atoms

class *stone-relation-algebra-atomic-atomsimple-finiteatoms =*
stone-relation-algebra-atomic + stone-relation-algebra-atomsimple +
stone-relation-algebra-finiteatoms
begin

subclass *stone-relation-algebra-atomic-atomsimple* ⟨*proof*⟩
subclass *stone-relation-algebra-atomic-finiteatoms* ⟨*proof*⟩
subclass *stone-relation-algebra-atomsimple-finiteatoms* ⟨*proof*⟩

lemma *nAB-top-2*:
nAB 1 ∗ nAB 1 ≤ nAB top
⟨*proof*⟩

lemma *counterexample-nAB-atom-iff-2*:
atom x ⟷ nAB x = 1
nitpick[*expect=genuine,card=6*]
⟨*proof*⟩

lemma *counterexample-nAB-top-iff-eq-2*:
nAB x = nAB top ⟷ x = top
nitpick[*expect=genuine,card=6*]
⟨*proof*⟩

lemma *counterexample-nAB-top-iff-leq-2*:
nAB top ≤ nAB x ⟷ x = top
nitpick[*expect=genuine,card=6*]
⟨*proof*⟩

end

3.14 Atom-rectangular, Atom-simple and Finitely Many Atoms

class *stone-relation-algebra-atomrect-atomsimple-finiteatoms* = *stone-relation-algebra-atomrect* + *stone-relation-algebra-atomsimple* + *stone-relation-algebra-finiteatoms*
begin

subclass *stone-relation-algebra-atomrect-atomsimple* ⟨*proof*⟩
subclass *stone-relation-algebra-atomrect-finiteatoms* ⟨*proof*⟩
subclass *stone-relation-algebra-atomsimple-finiteatoms* ⟨*proof*⟩

end

3.15 Atomic, Atom-rectangular, Atom-simple and Finitely Many Atoms

class *stone-relation-algebra-atomic-atomrect-atomsimple-finiteatoms* = *stone-relation-algebra-atomic* + *stone-relation-algebra-atomrect* + *stone-relation-algebra-atomsimple* + *stone-relation-algebra-finiteatoms*
begin

subclass *stone-relation-algebra-atomic-atomrect-atomsimple* ⟨*proof*⟩
subclass *stone-relation-algebra-atomic-atomrect-finiteatoms* ⟨*proof*⟩
subclass *stone-relation-algebra-atomic-atomsimple-finiteatoms* ⟨*proof*⟩
subclass *stone-relation-algebra-atomrect-atomsimple-finiteatoms* ⟨*proof*⟩

lemma *all-regular*:
regular x
⟨*proof*⟩

sublocale *ra*: *relation-algebra* **where** *minus* = $\lambda x\ y\ .\ x \sqcap -\ y$
⟨*proof*⟩

end

class *stone-relation-algebra-finite* = *stone-relation-algebra* + *finite*
begin

subclass *stone-relation-algebra-atomic-finiteatoms*
⟨*proof*⟩

end

3.16 Relation Algebra and Atomic

class *relation-algebra-atomic* = *relation-algebra* + *stone-relation-algebra-atomic*
begin

lemma *nAB-atom-iff*:

atom a $\longleftrightarrow$ *nAB a* = *1*
⟨*proof*⟩

end

3.17 Relation Algebra, Atomic and Finitely Many Atoms

class *relation-algebra-atomic-finiteatoms* = *relation-algebra-atomic* + *stone-relation-algebra-atomic-finiteatoms*
begin

Sup-fin only works for non-empty finite sets.

lemma *atomistic*:
assumes *x* ≠ *bot*
shows *x* = *Sup-fin* (*AB x*)
⟨*proof*⟩

lemma *counterexample-nAB-top*:
1 ≠ *top* ⟹ *nAB top* = *nAB 1* ∗ *nAB 1*
nitpick[*expect*=*genuine*,*card*=*4*]
⟨*proof*⟩

end

class *relation-algebra-atomic-atomsimple-finiteatoms* = *relation-algebra-atomic-finiteatoms* + *stone-relation-algebra-atomic-atomsimple-finiteatoms*
begin

lemma *counterexample-atom-rectangle*:
atom x ⟶ *rectangle x*
nitpick[*expect*=*genuine*,*card*=*4*]
⟨*proof*⟩

lemma *counterexample-atom-univalent*:
atom x ⟶ *univalent x*
nitpick[*expect*=*genuine*,*card*=*4*]
⟨*proof*⟩

lemma *counterexample-point-dense*:
assumes *x* ≠ *bot*
and *x* ≤ *1*
shows ∃ *a* . *a* ≠ *bot* ∧ *a* ∗ *top* ∗ *a* ≤ *1* ∧ *a* ≤ *x*
nitpick[*expect*=*genuine*,*card*=*4*]
⟨*proof*⟩

end

class *relation-algebra-atomic-atomrect-atomsimple-finiteatoms* = *relation-algebra-atomic-atomsimple-finiteatoms* +

4 Cardinality in Stone Relation Algebras

We study various axioms for a cardinality operation in Stone relation algebras.

class *card* =
  fixes *cardinality* :: *'a* ⇒ *enat* (#- [*100*] *100*)

class *sra-card* = *stone-relation-algebra* + *card*
begin

abbreviation *card-bot* :: *'a* ⇒ *bool* **where** *card-bot* - ≡ *#bot* = *0*
abbreviation *card-bot-iff* :: *'a* ⇒ *bool* **where** *card-bot-iff* - ≡ ∀ *x*::*'a* . *#x* = *0* ⟷ *x* = *bot*
abbreviation *card-top* :: *'a* ⇒ *bool* **where** *card-top* - ≡ *#top* = *#1* ∗ *#1*
abbreviation *card-conv* :: *'a* ⇒ *bool* **where** *card-conv* - ≡ ∀ *x*::*'a* . *#*(x^T) = *#x*
abbreviation *card-add* :: *'a* ⇒ *bool* **where** *card-add* - ≡ ∀ *x* *y*::*'a* . *#x* + *#y* = *#*(*x* ⊔ *y*) + *#*(*x* ⊓ *y*)
abbreviation *card-iso* :: *'a* ⇒ *bool* **where** *card-iso* - ≡ ∀ *x* *y*::*'a* . *x* ≤ *y* ⟶ *#x* ≤ *#y*
abbreviation *card-univ-comp-meet* :: *'a* ⇒ *bool* **where** *card-univ-comp-meet* - ≡ ∀ *x* *y* *z*::*'a* . *univalent x* ⟶ *#*(x^T ∗ *y* ⊓ *z*) ≤ *#*(*x* ∗ *z* ⊓ *y*)
abbreviation *card-univ-meet-comp* :: *'a* ⇒ *bool* **where** *card-univ-meet-comp* - ≡ ∀ *x* *y* *z*::*'a* . *univalent x* ⟶ *#*(*x* ⊓ *y* ∗ z^T) ≤ *#*(*x* ∗ *z* ⊓ *y*)
abbreviation *card-comp-univ* :: *'a* ⇒ *bool* **where** *card-comp-univ* - ≡ ∀ *x* *y*::*'a* . *univalent x* ⟶ *#*(*y* ∗ *x*) ≤ *#y*
abbreviation *card-univ-meet-vector* :: *'a* ⇒ *bool* **where** *card-univ-meet-vector* - ≡ ∀ *x* *y*::*'a* . *univalent x* ⟶ *#*(*x* ⊓ *y* ∗ *top*) ≤ *#y*
abbreviation *card-univ-meet-conv* :: *'a* ⇒ *bool* **where** *card-univ-meet-conv* - ≡ ∀ *x* *y*::*'a* . *univalent x* ⟶ *#*(*x* ⊓ *y* ∗ y^T) ≤ *#y*
abbreviation *card-domain-sym* :: *'a* ⇒ *bool* **where** *card-domain-sym* - ≡ ∀ *x*::*'a* . *#*(*1* ⊓ *x* ∗ x^T) ≤ *#x*
abbreviation *card-domain-sym-conv* :: *'a* ⇒ *bool* **where** *card-domain-sym-conv* - ≡ ∀ *x*::*'a* . *#*(*1* ⊓ x^T ∗ *x*) ≤ *#x*
abbreviation *card-domain* :: *'a* ⇒ *bool* **where** *card-domain* - ≡ ∀ *x*::*'a* . *#*(*1* ⊓ *x* ∗ *top*) ≤ *#x*
abbreviation *card-domain-conv* :: *'a* ⇒ *bool* **where** *card-domain-conv* - ≡ ∀ *x*::*'a* . *#*(*1* ⊓ x^T ∗ *top*) ≤ *#x*
abbreviation *card-codomain* :: *'a* ⇒ *bool* **where** *card-codomain* - ≡ ∀ *x*::*'a* . *#*(*1* ⊓ *top* ∗ *x*) ≤ *#x*
abbreviation *card-codomain-conv* :: *'a* ⇒ *bool* **where** *card-codomain-conv* - ≡ ∀ *x*::*'a* . *#*(*1* ⊓ *top* ∗ x^T) ≤ *#x*
abbreviation *card-univ* :: *'a* ⇒ *bool* **where** *card-univ* - ≡ ∀ *x*::*'a* . *univalent x* ⟶ *#x* ≤ *#*(*x* ∗ *top*)

abbreviation *card-atom* :: *'a* ⇒ *bool* **where** *card-atom* - ≡
∀ *x*::*'a* . *atom x* ⟶ #*x* = *1*
abbreviation *card-atom-iff* :: *'a* ⇒ *bool* **where** *card-atom-iff* - ≡
∀ *x*::*'a* . *atom x* ⟷ #*x* = *1*
abbreviation *card-top-iff-eq* :: *'a* ⇒ *bool* **where** *card-top-iff-eq* - ≡
∀ *x*::*'a* . #*x* = #*top* ⟷ *x* = *top*
abbreviation *card-top-iff-leq* :: *'a* ⇒ *bool* **where** *card-top-iff-leq* - ≡
∀ *x*::*'a* . #*top* ≤ #*x* ⟷ *x* = *top*
abbreviation *card-top-finite* :: *'a* ⇒ *bool* **where** *card-top-finite* - ≡
#*top* ≠ ∞

lemma *card-domain-iff*:
card-domain - ⟷ *card-domain-sym* -
⟨*proof*⟩

lemma *card-codomain-conv-iff*:
card-codomain-conv - ⟷ *card-domain* -
⟨*proof*⟩

lemma *card-codomain-iff*:
assumes *card-conv*: *card-conv* -
shows *card-codomain* - ⟷ *card-codomain-conv* -
⟨*proof*⟩

lemma *card-domain-conv-iff*:
card-codomain - ⟷ *card-domain-conv* -
⟨*proof*⟩

lemma *card-domain-sym-conv-iff*:
card-domain-conv - ⟷ *card-domain-sym-conv* -
⟨*proof*⟩

lemma *card-bot*:
assumes *card-bot-iff*: *card-bot-iff* -
shows *card-bot* -
⟨*proof*⟩

lemma *card-comp-univ-implies-card-univ-comp-meet*:
assumes *card-conv*: *card-conv* -
and *card-comp-univ*: *card-comp-univ* -
shows *card-univ-comp-meet* -
⟨*proof*⟩

lemma *card-univ-meet-conv-implies-card-domain-sym*:
assumes *card-univ-meet-conv*: *card-univ-meet-conv* -
shows *card-domain-sym* -
⟨*proof*⟩

lemma *card-add-disjoint*:

assumes *card-bot*: *card-bot* -
and *card-add*: *card-add* -
and $x \sqcap y = bot$
shows $\#(x \sqcup y) = \#x + \#y$
⟨*proof*⟩

lemma *card-dist-sup-disjoint*:
assumes *card-bot*: *card-bot* -
and *card-add*: *card-add* -
and $A \neq \{\}$
and *finite* A
and $\forall x{\in}A\ .\ \forall y{\in}A\ .\ x \neq y \longrightarrow x \sqcap y = bot$
shows $\#Sup\text{-}fin\ A = sum\ cardinality\ A$
⟨*proof*⟩

lemma *card-dist-sup-atoms*:
assumes *card-bot*: *card-bot* -
and *card-add*: *card-add* -
and $A \neq \{\}$
and *finite* A
and $A \subseteq AB\ top$
shows $\#Sup\text{-}fin\ A = sum\ cardinality\ A$
⟨*proof*⟩

lemma *card-univ-meet-comp-implies-card-domain-sym*:
assumes *card-univ-meet-comp*: *card-univ-meet-comp* -
shows *card-domain-sym* -
⟨*proof*⟩

lemma *card-top-greatest*:
assumes *card-iso*: *card-iso* -
shows $\#x \leq \#top$
⟨*proof*⟩

lemma *card-pp-increasing*:
assumes *card-iso*: *card-iso* -
shows $\#x \leq \#(--x)$
⟨*proof*⟩

lemma *card-top-iff-eq-leq*:
assumes *card-iso*: *card-iso* -
shows *card-top-iff-eq* - $\longleftrightarrow$ *card-top-iff-leq* -
⟨*proof*⟩

lemma *card-univ-comp-meet-implies-card-comp-univ*:
assumes *card-iso*: *card-iso* -
and *card-conv*: *card-conv* -
and *card-univ-comp-meet*: *card-univ-comp-meet* -
shows *card-comp-univ* -

⟨*proof*⟩

lemma *card-comp-univ-iff-card-univ-comp-meet*:
assumes *card-iso*: *card-iso* -
and *card-conv*: *card-conv* -
shows *card-comp-univ* - ⟷ *card-univ-comp-meet* -
⟨*proof*⟩

lemma *card-univ-meet-vector-implies-card-univ-meet-comp*:
assumes *card-iso*: *card-iso* -
and *card-univ-meet-vector*: *card-univ-meet-vector* -
shows *card-univ-meet-comp* -
⟨*proof*⟩

lemma *card-univ-meet-comp-implies-card-univ-meet-vector*:
assumes *card-iso*: *card-iso* -
and *card-univ-meet-comp*: *card-univ-meet-comp* -
shows *card-univ-meet-vector* -
⟨*proof*⟩

lemma *card-univ-meet-vector-iff-card-univ-meet-comp*:
assumes *card-iso*: *card-iso* -
shows *card-univ-meet-vector* - ⟷ *card-univ-meet-comp* -
⟨*proof*⟩

lemma *card-univ-meet-vector-implies-card-univ-meet-conv*:
assumes *card-iso*: *card-iso* -
and *card-univ-meet-vector*: *card-univ-meet-vector* -
shows *card-univ-meet-conv* -
⟨*proof*⟩

lemma *card-domain-sym-implies-card-univ-meet-vector*:
assumes *card-comp-univ*: *card-comp-univ* -
and *card-domain-sym*: *card-domain-sym* -
shows *card-univ-meet-vector* -
⟨*proof*⟩

lemma *card-domain-sym-iff-card-univ-meet-vector*:
assumes *card-iso*: *card-iso* -
and *card-comp-univ*: *card-comp-univ* -
shows *card-domain-sym* - ⟷ *card-univ-meet-vector* -
⟨*proof*⟩

lemma *card-univ-meet-conv-iff-card-univ-meet-comp*:
assumes *card-iso*: *card-iso* -
and *card-comp-univ*: *card-comp-univ* -
shows *card-univ-meet-conv* - ⟷ *card-univ-meet-comp* -
⟨*proof*⟩

lemma *card-domain-sym-iff-card-univ-meet-comp*:
assumes *card-iso*: *card-iso* -
and *card-comp-univ*: *card-comp-univ* -
shows *card-domain-sym* - $\longleftrightarrow$ *card-univ-meet-comp* -
⟨*proof*⟩

lemma *card-univ-comp-mapping*:
assumes *card-comp-univ*: *card-comp-univ* -
and *card-univ-meet-comp*: *card-univ-meet-comp* -
and *univalent x*
and *mapping y*
shows $\#(x * y) = \#x$
⟨*proof*⟩

lemma *card-point-one*:
assumes *card-comp-univ*: *card-comp-univ* -
and *card-univ-meet-comp*: *card-univ-meet-comp* -
and *card-conv*: *card-conv* -
and *point x*
shows $\#x = \#1$
⟨*proof*⟩

end

4.1 Cardinality in Relation Algebras

class *ra-card* = *sra-card* + *relation-algebra*
begin

lemma *card-iso*:
assumes *card-bot*: *card-bot* -
and *card-add*: *card-add* -
shows *card-iso* -
⟨*proof*⟩

lemma *card-top-iff-eq*:
assumes *card-bot-iff*: *card-bot-iff* -
and *card-add*: *card-add* -
and *card-top-finite*: *card-top-finite* -
shows *card-top-iff-eq* -
⟨*proof*⟩

end

class *ra-card-atomic-finiteatoms* = *ra-card* + *relation-algebra-atomic-finiteatoms*
begin

lemma *card-nAB*:
assumes *card-bot*: *card-bot* -

and *card-add*: *card-add* -
and *card-atom*: *card-atom* -
shows $\#x = nAB\ x$
⟨*proof*⟩

end

class *card-ab* = *sra-card* +
assumes *card-nAB'*: $\#x = nAB\ x$

class *sra-card-ab-atomsimple-finiteatoms* = *sra-card* + *card-ab* +
stone-relation-algebra-atomsimple-finiteatoms +
assumes *card-bot-iff*: *card-bot-iff* -
assumes *card-top*: *card-top* -
begin

subclass *stone-relation-algebra-atomic-atomsimple-finiteatoms*
⟨*proof*⟩

lemma *dom-cod-inj-atoms*:
inj-on dom-cod (*AB top*)
⟨*proof*⟩

subclass *stone-relation-algebra-atomic-atomrect-atomsimple-finiteatoms*
⟨*proof*⟩

lemma *atom-rectangle-card*:
assumes *atom a*
shows $\#(a * top * a) = 1$
⟨*proof*⟩

lemma *atom-regular-rectangle*:
assumes *atom a*
shows $--a = a * top * a$
⟨*proof*⟩

sublocale *ra-atom*: *relation-algebra-atomic* **where** *minus* = $\lambda x\ y\ .\ x \sqcap - y$
⟨*proof*⟩

end

class *ra-card-atomic-atomsimple-finiteatoms* = *ra-card* +
relation-algebra-atomic-atomsimple-finiteatoms +
assumes *card-bot*: *card-bot* -
assumes *card-add*: *card-add* -
assumes *card-atom*: *card-atom* -
assumes *card-top*: *card-top* -
begin

subclass *ra-card-atomic-finiteatoms*
⟨*proof*⟩

subclass *sra-card-ab-atomsimple-finiteatoms*
⟨*proof*⟩

subclass *relation-algebra-atomic-atomrect-atomsimple-finiteatoms*
⟨*proof*⟩

end

4.2 Counterexamples

class *ra-card-notop* = *ra-card* +
assumes *card-bot-iff*: *card-bot-iff* -
assumes *card-conv*: *card-conv* -
assumes *card-add*: *card-add* -
assumes *card-atom-iff*: *card-atom-iff* -
assumes *card-univ-comp-meet*: *card-univ-comp-meet* -
assumes *card-univ-meet-comp*: *card-univ-meet-comp* -

class *ra-card-all* = *ra-card-notop* +
assumes *card-top*: *card-top* -
assumes *card-top-finite*: *card-top-finite* -

class *ra-card-notop-atomic-finiteatoms* = *ra-card-atomic-finiteatoms* + *ra-card-notop*

class *ra-card-all-atomic-finiteatoms* = *ra-card-notop-atomic-finiteatoms* + *ra-card-all*

abbreviation *r0000* :: *bool* ⇒ *bool* ⇒ *bool* **where** *r0000 x y* ≡ *False*
abbreviation *r1000* :: *bool* ⇒ *bool* ⇒ *bool* **where** *r1000 x y* ≡ ¬*x* ∧ ¬*y*
abbreviation *r0001* :: *bool* ⇒ *bool* ⇒ *bool* **where** *r0001 x y* ≡ *x* ∧ *y*
abbreviation *r1001* :: *bool* ⇒ *bool* ⇒ *bool* **where** *r1001 x y* ≡ *x* = *y*
abbreviation *r0110* :: *bool* ⇒ *bool* ⇒ *bool* **where** *r0110 x y* ≡ *x* ≠ *y*
abbreviation *r1111* :: *bool* ⇒ *bool* ⇒ *bool* **where** *r1111 x y* ≡ *True*

lemma *r-all-different*:
r0000 ≠ *r1000 r0000* ≠ *r0001 r0000* ≠ *r1001 r0000* ≠ *r0110 r0000* ≠ *r1111*
r1000 ≠ *r0000 r1000* ≠ *r0001 r1000* ≠ *r1001 r1000* ≠ *r0110 r1000* ≠ *r1111*
r0001 ≠ *r0000 r0001* ≠ *r1000 r0001* ≠ *r1001 r0001* ≠ *r0110 r0001* ≠ *r1111*
r1001 ≠ *r0000 r1001* ≠ *r1000 r1001* ≠ *r0001 r1001* ≠ *r0110 r1001* ≠ *r1111*
r0110 ≠ *r0000 r0110* ≠ *r1000 r0110* ≠ *r0001 r0110* ≠ *r1001 r0110* ≠ *r1111*

r1111 ≠ *r0000 r1111* ≠ *r1000 r1111* ≠ *r0001 r1111* ≠ *r1001 r1111* ≠ *r0110*
⟨*proof*⟩

typedef (**overloaded**) *ra1* = {*r0000*,*r1001*,*r0110*,*r1111*}
⟨*proof*⟩

typedef (**overloaded**) *ra2* = {*r0000*,*r1000*,*r0001*,*r1001*}
⟨*proof*⟩

setup-lifting *type-definition-ra1*
setup-lifting *type-definition-ra2*
setup-lifting *type-definition-prod*

instantiation *Enum.finite-4* :: *ra-card-atomic-finiteatoms*
begin

definition *one-finite-4* :: *Enum.finite-4* **where** *one-finite-4* = *finite-4*.a_2
definition *conv-finite-4* :: *Enum.finite-4* ⇒ *Enum.finite-4* **where** *conv-finite-4 x* = *x*
definition *times-finite-4* :: *Enum.finite-4* ⇒ *Enum.finite-4* ⇒ *Enum.finite-4*
where *times-finite-4 x y* = (*case* (*x*,*y*) *of* (*finite-4*.a_1,-) ⇒ *finite-4*.a_1 | (-,*finite-4*.a_1) ⇒ *finite-4*.a_1 | (*finite-4*.a_2,*y*) ⇒ *y* | (*x*,*finite-4*.a_2) ⇒ *x* | - ⇒ *finite-4*.a_4)
definition *cardinality-finite-4* :: *Enum.finite-4* ⇒ *enat* **where** *cardinality-finite-4 x* = (*case x of finite-4*.a_1 ⇒ *0* | *finite-4*.a_4 ⇒ *2* | - ⇒ *1*)

instance
⟨*proof*⟩

end

instantiation *Enum.finite-4* :: *ra-card-notop-atomic-finiteatoms*
begin

instance
⟨*proof*⟩

end

instantiation *ra1* :: *ra-card-atomic-finiteatoms*
begin

lift-definition *bot-ra1* :: *ra1* **is** *r0000* ⟨*proof*⟩
lift-definition *one-ra1* :: *ra1* **is** *r1001* ⟨*proof*⟩
lift-definition *top-ra1* :: *ra1* **is** *r1111* ⟨*proof*⟩
lift-definition *conv-ra1* :: *ra1* ⇒ *ra1* **is** *id* ⟨*proof*⟩
lift-definition *uminus-ra1* :: *ra1* ⇒ *ra1* **is** λ*r x y* . ¬ *r x y* ⟨*proof*⟩
lift-definition *sup-ra1* :: *ra1* ⇒ *ra1* ⇒ *ra1* **is** λ*q r x y* . *q x y* ∨ *r x y* ⟨*proof*⟩
lift-definition *inf-ra1* :: *ra1* ⇒ *ra1* ⇒ *ra1* **is** λ*q r x y* . *q x y* ∧ *r x y* ⟨*proof*⟩

lift-definition *times-ra1* :: *ra1* ⇒ *ra1* ⇒ *ra1* **is** λ*q r x y* . ∃ *z* . *q x z* ∧ *r z y*
⟨*proof*⟩
lift-definition *minus-ra1* :: *ra1* ⇒ *ra1* ⇒ *ra1* **is** λ*q r x y* . *q x y* ∧ ¬ *r x y*
⟨*proof*⟩
lift-definition *less-eq-ra1* :: *ra1* ⇒ *ra1* ⇒ *bool* **is** λ*q r* . ∀ *x y* . *q x y* ⟶ *r x y*
⟨*proof*⟩
lift-definition *less-ra1* :: *ra1* ⇒ *ra1* ⇒ *bool* **is** λ*q r* . (∀ *x y* . *q x y* ⟶ *r x y*) ∧
q ≠ *r* ⟨*proof*⟩
lift-definition *cardinality-ra1* :: *ra1* ⇒ *enat* **is** λ*q* . *if q* = *r0000 then 0 else if q*
= *r1111 then 2 else 1* ⟨*proof*⟩

instance
⟨*proof*⟩

end

lemma *four-cases*:
assumes *P x1 P x2 P x3 P x4*
shows ∀ *y* ∈ { *x* . *x* ∈ {*x1*, *x2*, *x3*, *x4*} } . *P y*
⟨*proof*⟩

lemma *r-aux*:
(λ*x y*. *r1001 x y* ∨ *r0110 x y*) = *r1111* (λ*x y*. *r1001 x y* ∧ *r0110 x y*) = *r0000*
(λ*x y*. *r0110 x y* ∨ *r1001 x y*) = *r1111* (λ*x y*. *r0110 x y* ∧ *r1001 x y*) = *r0000*
(λ*x y*. *r1000 x y* ∨ *r0001 x y*) = *r1001* (λ*x y*. *r1000 x y* ∧ *r0001 x y*) = *r0000*
(λ*x y*. *r1000 x y* ∨ *r1001 x y*) = *r1001* (λ*x y*. *r1000 x y* ∧ *r1001 x y*) = *r1000*
(λ*x y*. *r0001 x y* ∨ *r1000 x y*) = *r1001* (λ*x y*. *r0001 x y* ∧ *r1000 x y*) = *r0000*
(λ*x y*. *r0001 x y* ∨ *r1001 x y*) = *r1001* (λ*x y*. *r0001 x y* ∧ *r1001 x y*) = *r0001*
(λ*x y*. *r1001 x y* ∨ *r1000 x y*) = *r1001* (λ*x y*. *r1001 x y* ∧ *r1000 x y*) = *r1000*
(λ*x y*. *r1001 x y* ∨ *r0001 x y*) = *r1001* (λ*x y*. *r1001 x y* ∧ *r0001 x y*) = *r0001*
⟨*proof*⟩

instantiation *ra1* :: *ra-card-notop-atomic-finiteatoms*
begin

instance
⟨*proof*⟩

end

instantiation *ra2* :: *ra-card-atomic-finiteatoms*
begin

lift-definition *bot-ra2* :: *ra2* **is** *r0000* ⟨*proof*⟩
lift-definition *one-ra2* :: *ra2* **is** *r1001* ⟨*proof*⟩
lift-definition *top-ra2* :: *ra2* **is** *r1001* ⟨*proof*⟩
lift-definition *conv-ra2* :: *ra2* ⇒ *ra2* **is** *id* ⟨*proof*⟩
lift-definition *uminus-ra2* :: *ra2* ⇒ *ra2* **is** λ*r x y* . *x* = *y* ∧ ¬ *r x y* ⟨*proof*⟩
lift-definition *sup-ra2* :: *ra2* ⇒ *ra2* ⇒ *ra2* **is** λ*q r x y* . *q x y* ∨ *r x y* ⟨*proof*⟩

lift-definition *inf-ra2* :: *ra2* ⇒ *ra2* ⇒ *ra2* **is** λ*q r x y* . *q x y* ∧ *r x y* ⟨*proof*⟩
lift-definition *times-ra2* :: *ra2* ⇒ *ra2* ⇒ *ra2* **is** λ*q r x y* . ∃ *z* . *q x z* ∧ *r z y* ⟨*proof*⟩
lift-definition *minus-ra2* :: *ra2* ⇒ *ra2* ⇒ *ra2* **is** λ*q r x y* . *q x y* ∧ ¬ *r x y* ⟨*proof*⟩
lift-definition *less-eq-ra2* :: *ra2* ⇒ *ra2* ⇒ *bool* **is** λ*q r* . ∀ *x y* . *q x y* ⟶ *r x y* ⟨*proof*⟩
lift-definition *less-ra2* :: *ra2* ⇒ *ra2* ⇒ *bool* **is** λ*q r* . (∀ *x y* . *q x y* ⟶ *r x y*) ∧ *q* ≠ *r* ⟨*proof*⟩
lift-definition *cardinality-ra2* :: *ra2* ⇒ *enat* **is** λ*q* . *if q* = *r0000 then 0 else if q* = *r1001 then 2 else 1* ⟨*proof*⟩

instance
⟨*proof*⟩

end

instantiation *ra2* :: *ra-card-notop-atomic-finiteatoms*
begin

instance
⟨*proof*⟩

end

instantiation *prod* :: (*stone-relation-algebra*,*stone-relation-algebra*) *stone-relation-algebra*
begin

lift-definition *bot-prod* :: *'a* × *'b* **is** (*bot*::*'a*,*bot*::*'b*) ⟨*proof*⟩
lift-definition *one-prod* :: *'a* × *'b* **is** (*1*::*'a*,*1*::*'b*) ⟨*proof*⟩
lift-definition *top-prod* :: *'a* × *'b* **is** (*top*::*'a*,*top*::*'b*) ⟨*proof*⟩
lift-definition *conv-prod* :: *'a* × *'b* ⇒ *'a* × *'b* **is** λ(*u*,*v*) . (*conv u*,*conv v*) ⟨*proof*⟩
lift-definition *uminus-prod* :: *'a* × *'b* ⇒ *'a* × *'b* **is** λ(*u*,*v*) . (*uminus u*,*uminus v*) ⟨*proof*⟩
lift-definition *sup-prod* :: *'a* × *'b* ⇒ *'a* × *'b* ⇒ *'a* × *'b* **is** λ(*u*,*v*) (*w*,*x*) . (*u* ⊔ *w*,*v* ⊔ *x*) ⟨*proof*⟩
lift-definition *inf-prod* :: *'a* × *'b* ⇒ *'a* × *'b* ⇒ *'a* × *'b* **is** λ(*u*,*v*) (*w*,*x*) . (*u* ⊓ *w*,*v* ⊓ *x*) ⟨*proof*⟩
lift-definition *times-prod* :: *'a* × *'b* ⇒ *'a* × *'b* ⇒ *'a* × *'b* **is** λ(*u*,*v*) (*w*,*x*) . (*u* ∗ *w*,*v* ∗ *x*) ⟨*proof*⟩
lift-definition *less-eq-prod* :: *'a* × *'b* ⇒ *'a* × *'b* ⇒ *bool* **is** λ(*u*,*v*) (*w*,*x*) . *u* ≤ *w* ∧ *v* ≤ *x* ⟨*proof*⟩
lift-definition *less-prod* :: *'a* × *'b* ⇒ *'a* × *'b* ⇒ *bool* **is** λ(*u*,*v*) (*w*,*x*) . *u* ≤ *w* ∧ *v* ≤ *x* ∧ ¬(*u* = *w* ∧ *v* = *x*) ⟨*proof*⟩

instance
⟨*proof*⟩

end

instantiation *prod* :: (*relation-algebra*,*relation-algebra*) *relation-algebra*
begin

lift-definition *minus-prod* :: $'a \times 'b \Rightarrow 'a \times 'b \Rightarrow 'a \times 'b$ **is** $\lambda(u,v)\ (w,x)\ .\ (u - w, v - x)$ ⟨*proof*⟩

instance
⟨*proof*⟩

end

instantiation *prod* ::
(*relation-algebra-atomic-finiteatoms*,*relation-algebra-atomic-finiteatoms*)
relation-algebra-atomic-finiteatoms
begin

instance
⟨*proof*⟩

end

instantiation *prod* ::
(*ra-card-notop-atomic-finiteatoms*,*ra-card-notop-atomic-finiteatoms*)
ra-card-notop-atomic-finiteatoms
begin

lift-definition *cardinality-prod* :: $'a \times 'b \Rightarrow$ *enat* **is** $\lambda(u,v)\ .\ \#u + \#v$ ⟨*proof*⟩

instance
⟨*proof*⟩

end

type-synonym *finite-4-square* = *Enum.finite-4* × *Enum.finite-4*

interpretation *finite-4-square*: *ra-card-atomic-finiteatoms* **where** *cardinality* = *cardinality* **and** *inf* = (⊓) **and** *less-eq* = (≤) **and** *less* = (<) **and** *sup* = (⊔) **and** *bot* = *bot*::*finite-4-square* **and** *top* = *top* **and** *uminus* = *uminus* **and** *one* = *1* **and** *times* = (∗) **and** *conv* = *conv* **and** *minus* = (−) ⟨*proof*⟩

interpretation *finite-4-square*: *ra-card-all-atomic-finiteatoms* **where** *cardinality* = *cardinality* **and** *inf* = (⊓) **and** *less-eq* = (≤) **and** *less* = (<) **and** *sup* = (⊔) **and** *bot* = *bot*::*finite-4-square* **and** *top* = *top* **and** *uminus* = *uminus* **and** *one* = *1* **and** *times* = (∗) **and** *conv* = *conv* **and** *minus* = (−)
⟨*proof*⟩

lemma *counterexample-atom-rectangle-2*:

atom a $\longrightarrow$ *a* $*$ *top* $*$ *a* $\leq$ (*a*::*finite-4-square*)
nitpick[*expect*=*genuine*]
⟨*proof*⟩

lemma *counterexample-atom-univalent-2*:
atom a $\longrightarrow$ *univalent* (*a*::*finite-4-square*)
nitpick[*expect*=*genuine*]
⟨*proof*⟩

lemma *counterexample-point-dense-2*:
assumes $x \neq bot$
and $x \leq 1$
shows $\exists\, a$::*finite-4-square* . $a \neq bot \wedge a * top * a \leq 1 \wedge a \leq x$
nitpick[*expect*=*genuine*]
⟨*proof*⟩

type-synonym *ra11* = *ra1* × *ra1*

interpretation *ra11*: *ra-card-atomic-finiteatoms* **where** *cardinality* = *cardinality* **and** *inf* = (⊓) **and** *less-eq* = (≤) **and** *less* = (<) **and** *sup* = (⊔) **and** *bot* = *bot*::*ra11* **and** *top* = *top* **and** *uminus* = *uminus* **and** *one* = *1* **and** *times* = (∗) **and** *conv* = *conv* **and** *minus* = (−) ⟨*proof*⟩

interpretation *ra11*: *ra-card-all-atomic-finiteatoms* **where** *cardinality* = *cardinality* **and** *inf* = (⊓) **and** *less-eq* = (≤) **and** *less* = (<) **and** *sup* = (⊔) **and** *bot* = *bot*::*ra11* **and** *top* = *top* **and** *uminus* = *uminus* **and** *one* = *1* **and** *times* = (∗) **and** *conv* = *conv* **and** *minus* = (−)
⟨*proof*⟩

interpretation *ra11*: *stone-relation-algebra-atomrect* **where** *inf* = (⊓) **and** *less-eq* = (≤) **and** *less* = (<) **and** *sup* = (⊔) **and** *bot* = *bot*::*ra11* **and** *top* = *top* **and** *uminus* = *uminus* **and** *one* = *1* **and** *times* = (∗) **and** *conv* = *conv*
⟨*proof*⟩

lemma ¬ (∀ *a*::*ra1*×*ra1* . *atom a* $\longrightarrow$ *a* $*$ *top* $*$ *a* $\leq$ *a*)
⟨*proof*⟩

end

References

[1] H. Furusawa and W. Guttmann. Cardinality and representation of Stone relation algebras. *arXiv*, 2309.11676, 2023. https://arxiv.org/abs/2309.11676.

[2] W. Guttmann. Stone relation algebras. In P. Höfner, D. Pous, and G. Struth, editors, *Relational and Algebraic Methods in Computer*

Science, volume 10226 of *Lecture Notes in Computer Science*, pages 127–143. Springer, 2017.

[3] W. Guttmann. Stone relation algebras. *Archive of Formal Proofs*, 2017.