Representability and formalization of (distributive quasi) relation algebras

Peter Jipsen

Chapman University

Cracow Logic Conference, CLoCk 2025

June 26 – 27, Cracow, Poland

Outline

1. Relation algebras (RAs)
2. Representable relation algebras (RRAs)
3. Proving (non)representability
4. Implementing Comer's 1-point extension method
5. Weakening relation algebras (with N. Galatos, J. Semrl)
6. Residuated lattices and (distributive) InFL-algebras
7. Distributive quasi Relation Algebras (with A. Craig, C. Robinson)
8. Minimal relation algebras and DqRAs
9. Theorem provers
10. Formalizing relation algebras in Lean (with P. Nelson)
11. An unrelated new result (with S. Santschi)

Definition of relation algebra

Alfred Tarski defined (abstract) relation algebras (RAs) in 1941.

A **relation algebra** $\mathbf{A} = \langle A, \sqcup, {}^c, ;, 1', {}^{-1} \rangle$ is a

Boolean algebra $\langle A, \sqcup, {}^c \rangle$ with operations $;$, $1'$, ${}^{-1}$ that satisfy

associativity: $\forall xyz,\ (x\,;y)\,;z = x\,;(y\,;z)$

right distributivity: $\forall xyz,\ (x \sqcup y)\,;z = x\,;z \sqcup y\,;z$

right identity: $\forall x,\ x\,;1' = x$

involution 1: $\forall x,\ x^{-1-1} = x$

involution 2: $\forall xy,\ (x\,;y)^{-1} = y^{-1}\,;x^{-1}$

converse distributivity: $\forall xy,\ (x \sqcup y)^{-1} = x^{-1} \sqcup y^{-1}$

Schröder inequality: $\forall xy,\ x^{-1}\,;(x\,;y)^c \sqcup y^c = y^c$

Shorter definition of relation algebras

The definition by **universal equational axioms** shows that

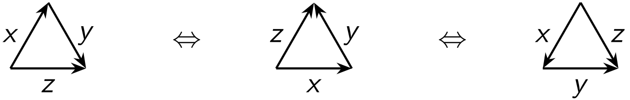
the class RA of relation algebras is a **variety**, i.e., closed under $\mathbb{HSP}$

The following shorter definition is **equivalent** (using $\bot = (1'^c \sqcup 1')^c$):

A **relation algebra** $\mathbf{A} = \langle A, \sqcup, {}^c, ;, 1', {}^{-1} \rangle$ is a monoid $\langle A, ;, 1' \rangle$

and a Boolean algebra $\langle A, \sqcup, {}^c \rangle$ with operation ${}^{-1}$ that satisfies

$$x;y \sqcap z = \bot \quad \Leftrightarrow \quad z;y^{-1} \sqcap x = \bot \quad \Leftrightarrow \quad x^{-1};z \sqcap y = \bot$$



Note ; has priority over the meet operation $x \sqcap y = (x^c \sqcup y^c)^c$

Representable relation algebras RRA

The **full algebra of binary relations on** a set X is

$$\mathbf{Rel}(X) = \langle \mathcal{P}(X^2), \cup, {}^c, ;, id_X, {}^{-1} \rangle \text{ where } R^c = X^2 \setminus R$$

composition $R\,;S = \{(x,y) \mid \exists z,\ (x,z) \in R \text{ and } (z,y) \in S\}$

converse $R^{-1} = \{(x,y) \mid (y,x) \in R\}$

Proposition: $\mathrm{Rel}(X)$ is a relation algebra

RRA $=$ **representable relation algebras** $= \mathbb{SP}\{\mathbf{Rel}(X) \mid X \text{ is a set}\}$

Tarski [1956] proved that **RRA** is a variety (i.e., closed under $\mathbb{H}$)

For more details see the books by Givant [2017] and Maddux [2006]

Representability by a group

For a group $\mathbf{G} = \langle G, \cdot, 1, ^{-1} \rangle$ we define the **group relation algebra**

$$Cm(\mathbf{G}) = \langle \mathcal{P}(G), \cup, ^c, \cdot, \{1\}, ^{-1} \rangle$$

where $X \cdot Y = \{xy \mid x \in X, y \in Y\}$ and $X^{-1} = \{x^{-1} \mid x \in X\}$

$Cm(\mathbf{G})$ is **representable** by **Cayley's theorem**: for $g \in G$,

each atom $\{g\}$ is represented by $R_g = \{(x, gx) \mid x \in G\}$

Representability

When is a RA **representable** as an algebra of binary relations?

Donald Monk (1964): the variety of representable RAs is **not axiomatized by finitely many formulas**.

Robin Hirsch and Ian Hodkinson (2001): it is **undecidable** whether a finite relation algebra is representable.

Roger Maddux (1983): *n*-dimensional bases to **prove nonrepresentability**.

Steve Comer (∼1980): one-point extension method to **prove representability** for some small RAs.

Finding and checking these proofs by hand is laborious.

Implementing Comer's one-point extension method

```
def ExtensionsList(A):
  # ext[i] is the list of atoms k,con[j] such that i <= j;k
    n = len(A)
    con = Converses(A)
    ext = [set([]) for i in range(n)]
    for j in range(1,n):
        for k in range(1,n):
            for i in A[j][k]:
                ext[i] |= set([(k, con[j])])
    return [list(x) for x in ext]

def FindOnePointExtension(A):
  """
  Returns rules for a one-point extension if possible,
  returns false otherwise.
  Uses a backtrack algorithm to search the space
  """
```

Atomic networks

A **consistent atomic network** $N : X^2 \to At(A)$ is a function s.t.

$N(x,x) \leq 1',\ \ N(x,z) \leq N(x,y); N(y,z)$ and $N(x,y) = N(y,x)^{-1}$.

It is a **representation** if N is onto and for all atoms a, b,
$N(x,y) \leq a; b \implies \exists z$ s.t. $N(x,z) = a$ and $N(z,y) = b$.

The representation homomorphism $h : At(A) \to \mathcal{P}(X^2)$
is given by $h(a) = N^{-1}[\{a\}]$.

Then $c \leq a; b$ implies $h(c) \subseteq h(a; b)$ and $h(a^{-1}) = (h(a))^{-1}$.

The existence of a representation is equivalent to a winning strategy for the existential player in the representation game.

Extending atomic networks step-by-step

Given a consistent network, we want to extend it to a representation in a step-by-step way:

For all $a, b \in At(A)$, and $x, y \in X$ such that $N(x, y) \leq a; b$,

if there does not exists $z \in X$ such that $N(x, z) = a$ and $N(z, y) = b$ then choose z not in X, let $X' = X \cup \{z\}$ and define $N'(x, z) = a$ and $N'(z, y) = b$.

Need to define $N'(u, z)$ for all $u \in X \setminus \{x, y\}$ s.t. N' is still consistent.

So we need to ensure $N'(u, z) \leq N'(u, v); N'(v, z)$.

E.g. $N'(u, z) =$ "flexible atom" is a valid one-point extension.

If a solution exists, this algorithm finds one.

Representation does not have to be infinite

Implementing Comer's one-point extension method

```
def NextColor(A,i):
# choose the next color for extension  i  and recurse
# return false if no choice worked.
  if i >= len(exta): return True # found the last color
  if i == ei:        # skip the extension I'm working on
      return NextColor(A,i+1)
  for c in colset[i]:        # try each color
      col[i] = c; j = 0
      ok = True
      while ok and j <= i:
          if j != ei: # skip the extension I'm working on
              x = set(A[exta[j][0]][con[exta[i][0]]])
              x &= A[exta[j][1]][con[exta[i][1]]]
              ok = (x <= A[con[col[j]]][c]) #check subset
          j += 1
      if ok and NextColor(A,i+1): return True
  return False
```

Implementing Comer's one-point extension method

```
def ColorSets(A,ex,i):
# return sets of permissible colors between  ex[i]  and  ex[k]  for each  k.
  cs = [[] for x in ex]
  for k in range(len(ex)):
      if k != i:   # skip the extension I'm working on
          x = set(A[ex[i][0]][con[ex[k][0]]])
          x &= A[ex[i][1]][con[ex[k][1]]]
          cs[k] = list(x)
  return cs

con = Converses(A)
ext = ExtensionsList(A)
collist = []
for atm in range(1,len(A)):
      exta = ext[atm]
      for ei in range(len(exta)):
          colset = ColorSets(A,exta,ei)
          col = [0 for x in exta]
          if not NextColor(A,0):
              return False
          collist.append(col)
return collist
```

A database of finite integral relation algebras up to 5 atoms

Let a, b, c, d be symmetric atoms ($x^{-1} = x$) and r, s nonsymmetric

The number of RAs up to isomorphism is given below:

2	4	8	8	16	16	32	32	32
$1'$	$1'a$	$1'rr^{-1}$	$1'ab$	$1'arr^{-1}$	$1'abc$	$1'rr^{-1}ss^{-1}$	$1'abrr^{-1}$	$1'abcd$
1	2	3	7	37	65	83	1316	3013

Their (non)representability is known up to size 16 ($= 4$ atoms).

For the list of 83 there are 15 RAs that are not known to be (non)representable: 30,31,32,40,44,45,54,56,59,60,61,63,65,69,79 (see [Maddux 2006])

Unknown if representable: 235 out of 1316; 485 out of 3013

Representable **weakening** relation algebras RwkRA

The set of **weakening relations** on a poset $(X, \leq)$ is

$Wk(X, \leq) = \{R \subseteq X^2 \mid \leq; R; \leq \, = R\}$.

The **full algebra of weakening relations on a poset** $(X, \leq)$ is

$\mathbf{wk}(X, \leq) = (Wk(X, \leq), \cap, \cup, \emptyset, \top, ;, \leq, \sim)$ where $\sim R = X^2 \setminus R^{-1}$

The class of **representable weakening relation algebras** is

$\mathbf{RwkRA} = \mathbb{SP}\{\mathbf{wk}(X, \leq) \mid (X, \leq)$ is a poset$\}$.

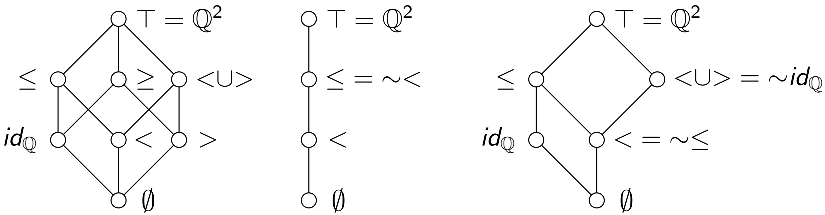
It is a quasivariety (defined by implications) but not a variety.

The **point algebra** is a relation algebra with 3 atoms $id_{\mathbb{Q}}, <, >$ where $<$ is the strict order on the rational numbers $\mathbb{Q}$.

It has two **weakening subalgebras**: $\mathbf{S}_4 = \{\emptyset, <, \leq, \top\}$ and $\mathbf{A} = \{\emptyset, id_{\mathbb{Q}}, <, \leq, <\cup>, \top\}$.

Like the point algebra, both can only be represented on **infinite** sets.

Note that **A** is **diagonally representable**, while $\mathbf{S}_4$ is not.



The point algebra $\quad$ $\mathbf{S}_4$ $\quad$ $\mathbf{A}$

Residuated lattices

A **residuated lattice** (**RL**) is of the form $\mathbf{A} = (A, \sqcap, \sqcup, \cdot, 1, \backslash, /)$ where $(A, \sqcap, \sqcup)$ is a lattice, $(A, \cdot, 1)$ is a monoid and $\backslash, /$ are the **left** and **right residuals** of $\cdot$, i.e., for all $x, y, z \in A$

$$xy \leq z \iff y \leq x\backslash z \iff x \leq z/y.$$

The previous formula is equivalent to the following 4 identities:

$$\begin{array}{ll} x \leq y\backslash(yx \sqcup z) & \quad x((x\backslash y) \sqcap z) \leq y \\ x \leq (xy \sqcup z)/y & \quad ((x/y) \sqcap z)y \leq x \end{array}$$

so residuated lattices form a variety.

A **full Lambek** (**FL**-)**algebra** is a **RL** with a **constant** 0,

used to define the **linear negations** ${\sim}x = x\backslash 0$ and $-x = 0/x$.

An **involutive FL-algebra** (**InFL**) is an FL-algebra such that ${\sim}{-}x = x = -{\sim}x$. It is **cyclic** if ${\sim}x = -x$

RwkRAs are cyclic distributive involutive FL-algebras

Recall that relation algebras satisfy

$$x;y \sqcap z = \bot \quad \Leftrightarrow \quad z;y^{-1} \sqcap x = \bot \quad \Leftrightarrow \quad x^{-1};z \sqcap y = \bot$$

$$x; y \leq z^c \quad \Leftrightarrow \quad x \leq (z; y^{-1})^c \quad \Leftrightarrow \quad y \leq (x^{-1}; z)^c$$

$$x; y \leq z \quad \Leftrightarrow \quad x \leq (y; z^{-1c})^{-1c} \quad \Leftrightarrow \quad y \leq (z^{-1c}; x)^{-1c}$$

replacing z by z^c, $(x; y)^{-1} = y^{-1}; x^{-1}$ and $x^{-1-1} = x$.

So letting $\sim z = z^{-1c}$, we have $\mathbf{A} \in \mathbf{RA}$ implies

$(A, \sqcup, ;, 1, \sim)$ is a **cyclic InFL-algebra**

and it satisfies **distributivity**: $x \sqcap (y \sqcup z) = (x \sqcap y) \sqcup (x \sqcap z)$

Distributive quasi relation algebras

To obtain nonclassical relation algebras that have the same signature as RA, we add a unary De Morgan operation $'$ that satisfies $(x \sqcup y)' = x' \sqcap y'$.

[Galatos, J. 2013] A **quasi relation algebra** is an InFL-algebra with a De Morgan operation such that $(x; y)' = x' + y'$ where $x + y = -({\sim}y; {\sim}x)$.

Distributive quasi relation algebras (DqRAs) are a finitely based variety of nonclassical RAs.

Similar to the list of 1662 residuated lattices with up to 6 elements `https://math.chapman.edu/~jipsen/preprints/RLlist3.pdf` [Galatos, J. 2017]

we recently made a list of 395 DqRAs with up to 8 elements `https://github.com/jipsen/Distributive-quasi-relation-algebras-and-DInFL/blob/main/DInFL1.pdf` [Craig, J., Robinson 2025]

DISTRIBUTIVE INVOLUTIVE RESIDUATED LATTICES UP TO CARDINALITY 8

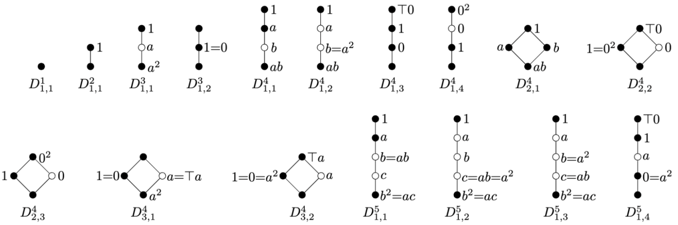
ANDREW CRAIG, PETER JIPSEN, AND CLAUDETTE ROBINSON

There are $1+1+2+9+8+43+49+282 = 395$ distributive involutive residuated lattices with ≤ 8 elements. In the list below, each algebra is named $D^n_{m,i}$ where n is the cardinality and m enumerates nonisomorphic involutive lattices of size n, in order of decreasing height. The index i enumerates nonisomorphic algebras with the same involutive lattice reduct. The linear negations $\sim, -$ are determined by the element labeled 0 (bottom for integral algebras). Algebras with more central elements (round circles) are listed earlier, hence commutative algebras precede noncommutative ones. Finally, algebras are listed in decreasing order of number of idempotents (black nodes).

The monoid operation is indicated by labels. If a nonobvious product xy is not listed, then it can be deduced from the given information: either it follows from idempotence ($x^2 = x$) indicated by a black node or from commutativity or there are products $uv = wz$ such that $u \leq x \leq w$ and $v \leq y \leq z$ (possibly $uv = \bot\bot$ or $wz = \top\top$).

If you have comments or notice any issues in this list, please email `jipsen.AT.chapman.edu`.

● = central idempotent
○ = central nonidempotent
■ = noncentral idempotent
□ = noncentral nonidempotent



Minimal relation algebras

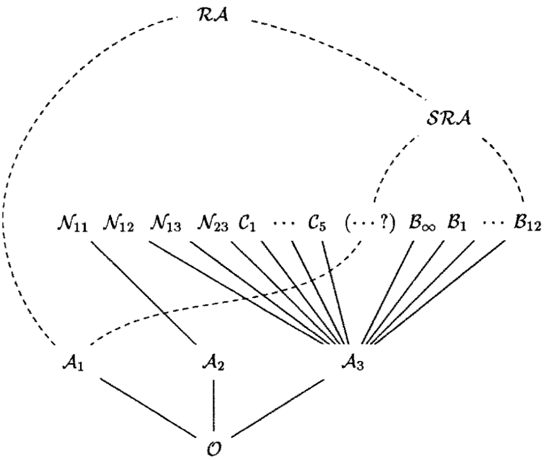


Figure 1. Join irreducibles on the bottom levels of Λ.

$$\mathcal{A}_1 = \mathbf{BA}, \quad \mathcal{A}_2 = Cm(\mathbb{Z}_2), \quad \mathcal{A}_3 < Cm(\mathbb{Z}_3), \quad \mathcal{N}_{11} = \mathbf{Rel}(2)$$

New minimal varieties from subreducts

A relation algebra $\mathbf{A}$ has a $\sim$-reduct $\tilde{\mathbf{A}}$ where ${}^c, {}^{-1}$ are removed and $\sim$ is added.

A subalgebra of a $\sim$-reduct is distributive, but not necessarily Boolean, and is called a $\sim$-subreduct.

If $\mathbf{A} \in$ RRA then every $\sim$-subreduct of $\mathbf{A}$ is in RwkRA and satisfies $1' \sqcap \sim 1' = \bot$, which means $1'$ is the identity relation.

The other nonsymmetric minimal RAs also have proper $\sim$-subreducts.

These algebras are called diagonal RwkRAs, and they are discriminator algebras [J., Semrl 2023].

$\mathcal{C}_1 = \langle\{1, 2, -3\}\rangle^{Cm(\mathbb{Z}_7)}$ has 8 elements but $R = \{1, 2, -3\}$ generates a minimal variety of RwkRA with 6 elements.

Brief background on proof assistants

Automated theorem provers have been developed since the 1960s, see McCune and Wos [1997] for a brief history.

Mostly restricted to first-order logic: Otter, Prover9/Mace4, SPASS, E-prover, Vampire, ...

Satisfiability Modulo Theories (SMT) solvers: Z3, CVC5, ...

Interactive theorem provers: Mizar, PVS, HOL, HOL-light, Isabelle, Rocq, Agda, Lean, ...

Based on higher-order logics, (dependent) type theories

Large libraries of formal proofs, but no common language

A Lean class for relation algebras

```
class RelationAlgebra (A : Type u) extends
  BooleanAlgebra A, Comp A, One A, Inv A where
  assoc : ∀ x y z : A, (x ; y) ; z = x ; (y ; z)
  rdist : ∀ x y z : A, (x ⊔ y) ; z = x ; z ⊔ y ; z
  comp_one  : ∀ x : A, x ; 1 = x
  conv_conv : ∀ x : A, x⁻¹⁻¹ = x
  conv_dist : ∀ x y : A, (x ⊔ y)⁻¹ = x⁻¹ ⊔ y⁻¹
  conv_comp : ∀ x y : A, (x ; y)⁻¹ = y⁻¹ ; x⁻¹
  schroeder : ∀ x y : A, x⁻¹ ; (x ; y)ᶜ ≤ yᶜ
```

This definition is based on Lean's mathlib4

Relation algebra in proof assistants

Rocq: Damien Pous, **Relation Algebra and KAT in Coq**, 2012,
`https://perso.ens-lyon.fr/damien.pous/ra/`

Isabelle: A. Armstrong, S. Foster, G. Struth, T. Weber, 2014,
Archive of Formal Proofs, Relation Algebra
`https://www.isa-afp.org/entries/Relation_Algebra.html`

```
lemma top_conv : (⊤ : A)⁻¹ = ⊤ := by
  have : (⊤ : A)⁻¹ = (⊤ ⊔ ⊤⁻¹)⁻¹ := by simp
  have : (⊤ : A)⁻¹ = ⊤⁻¹ ⊔ ⊤ := by rw [conv_dist,
    conv_conv] at this; exact this
  have : (⊤ : A) ≤ ⊤⁻¹ := by rw [left_eq_sup] at this;
    exact this
  exact top_unique this

lemma ldist (x y z : A) : x ; (y ⊔ z) = x ; y ⊔ x ; z :=
    by
  calc
    x ; (y ⊔ z) = (x ; (y ⊔ z))⁻¹⁻¹ := by rw [conv_conv]
    _ = ((y ⊔ z)⁻¹ ; x⁻¹)⁻¹ := by rw [conv_comp]
    _ = ((y⁻¹ ⊔ z⁻¹) ; x⁻¹)⁻¹ := by rw [conv_dist]
    _ = (y⁻¹ ; x⁻¹ ⊔ z⁻¹ ; x⁻¹)⁻¹ := by rw [rdist]
    _ = ((x ; y)⁻¹ ⊔ (x ; z)⁻¹)⁻¹ := by rw [←conv_comp,
    ←conv_comp]
    _ = (x ; y) ⊔ (x ; z) := by rw [←conv_dist,
    conv_conv]
```

```
lemma comp_le_comp_right (z : A) {x y : A} (h : x ≤ y) :
    x ; z ≤ y ; z := by
  calc
    x ; z ≤ x ; z ⊔ y ; z := by simp
    _ = (x ⊔ y) ; z := by rw [←rdist]
    _ = y ; z := by simp [h]

lemma comp_le_comp_left (z : A) {x y : A} (h : x ≤ y) : z
    ; x ≤ z ; y := by
  calc
    z ; x ≤ z ; x ⊔ z ; y := by simp
    _ = z ; (x ⊔ y) := by rw [←ldist]
    _ = z ; y := by simp [h]

lemma conv_le_conv {x y : A} (h : x ≤ y) : x⁻¹ ≤ y⁻¹ :=
    by
  calc
    x⁻¹ ≤ x⁻¹ ⊔ y⁻¹ := by simp
    _ = (x ⊔ y)⁻¹ := by rw [←conv_dist]
    _ = y⁻¹ := by simp [h]
```

lemma conv_compl_le_compl_conv (x : A) : $x^{-1c} \le x^{c-1}$:= **by**
  have : $x \sqcup x^{c} = \top$:= **by** simp
  have : $(x \sqcup x^{c})^{-1} = \top^{-1}$:= **by** simp
  have : $x^{-1} \sqcup x^{c-1} = \top$:= **by** rw [conv_dist, top_conv]
    at this; exact **this**
  rw[join_eq_top_iff_compl_le] **at this**; exact **this**

lemma conv_compl_eq_compl_conv (x : A) : $x^{c-1} = x^{-1c}$:= **by**
  have : $x^{-1-1c} \le x^{-1c-1}$:= conv_compl_le_compl_conv x^{-1}
  have : $x^{c} \le x^{-1c-1}$:= **by** rw [conv_conv] **at this**; exact
    this
  have : $x^{c-1} \le x^{-1c-1-1}$:= conv_le_conv **this**
  rw [conv_conv] **at this**; exact le_antisymm **this**
    (conv_compl_le_compl_conv x)

```lean
lemma one_conv_eq_one : (1 : A)⁻¹ = 1 := by
  calc
    (1 : A)⁻¹ = 1⁻¹ ; 1 := by rw [comp_one]
    _ = (1⁻¹ ; 1)⁻¹⁻¹ := by rw [conv_conv]
    _ = (1⁻¹ ; 1⁻¹⁻¹)⁻¹ := by rw [conv_comp]
    _ = (1⁻¹ ; 1)⁻¹ := by rw [conv_conv]
    _ = 1 := by rw [comp_one, conv_conv]

lemma one_comp (x : A) : 1 ; x = x := by
  calc
    1 ; x = (1 ; x)⁻¹⁻¹ := by rw [conv_conv]
    _ = (x⁻¹ ; 1⁻¹)⁻¹ := by rw [conv_comp]
    _ = (x⁻¹ ; 1)⁻¹ := by rw [one_conv_eq_one]
    _ = x⁻¹⁻¹ := by rw [comp_one]
    _ = x := by rw [conv_conv]
```

```
lemma peirce_law1 (x y z : A) :
  x ; y ⊓ z = ⊥ ↔ x⁻¹ ; z ⊓ y = ⊥ := by
  constructor
  · intro h
    have : x ; y ≤ zᶜ := by rw [meet_eq_bot_iff_le_compl]
    at h; exact h
    have : z ≤ (x ; y)ᶜ := by rw [←compl_le_compl_iff_le,
     compl_compl] at this; exact this
    have : x⁻¹ ; z ≤ x⁻¹ ; (x ; y)ᶜ := comp_le_comp_left
    x⁻¹ this
    have : x⁻¹ ; z ⊓ y ≤ ⊥ := by calc
        x⁻¹ ; z ⊓ y ≤ x⁻¹ ; (x ; y)ᶜ ⊓ y :=
    inf_le_inf_right y this
        _ ≤ yᶜ ⊓ y := inf_le_inf_right y (schroeder x y)
        _ = ⊥ := by simp
    exact bot_unique this
```

```
· intro h
  have : x⁻¹ ; z ≤ yᶜ := by rw
  [meet_eq_bot_iff_le_compl] at h; exact h
  have : y ≤ (x⁻¹ ; z)ᶜ := by
    rw [←compl_le_compl_iff_le, compl_compl] at this;
  exact this
  have : x⁻¹⁻¹ ; y ≤ x⁻¹⁻¹ ; (x⁻¹ ; z)ᶜ :=
  comp_le_comp_left x⁻¹⁻¹ this
  have : x⁻¹⁻¹ ; y ⊓ z ≤ ⊥ := by calc
    x⁻¹⁻¹ ; y ⊓ z ≤ x⁻¹⁻¹ ; (x⁻¹ ; z)ᶜ ⊓ z :=
  inf_le_inf_right z this
    _ ≤ zᶜ ⊓ z := inf_le_inf_right z (schroeder x⁻¹ z)
    _ = ⊥ := by simp
  have : x ; y ⊓ z ≤ ⊥ := by rw [conv_conv] at this;
  exact this
  exact bot_unique this

lemma peirce_law2 (x y z : A) :
  x ; y ⊓ z = ⊥ ↔ z ; y⁻¹ ⊓ x = ⊥ := by
  ...
```

Definitions for binary relations: Math vs. Lean

Let X be a set and $R, S, T \in \mathcal{P}(X \times X)$ **binary relations** on X

```
import Mathlib.Data.Set.Basic
variable {X : Type u} (R S T : Set (X × X))
```

Define **composition** $R \,;S = \{(x, y) \mid \exists z,\ (x, z) \in R \wedge (z, y) \in S\}$.

```
def composition (R S : Set (X × X)) : Set (X × X) :=
  { (x, y) | ∃ z, (x, z) ∈ R ∧ (z, y) ∈ S }
```

Define the **inverse** of R by $R^{-1} = \{(y, x) \mid (x, y) \in R\}$

```
infixl:90 " ; "  => composition
postfix:100 "⁻¹" => inverse
```

```
theorem comp_assoc : (R ; S) ; T = R ; (S ; T) := by
  rw [Set.ext_iff]
  intro (a,b)
  constructor
  intro h
  rcases h with ⟨z, h₁, _⟩
  rcases h₁ with ⟨x,_,_⟩
  use x
  constructor
  trivial
  use z
  intro h₂
  rcases h₂ with ⟨x, h₃, h₄⟩
  rcases h₄ with ⟨y,_,_⟩
  use y
  constructor
  use x
  trivial
```

Algebras of binary relations

An **algebra of binary relations** is a set of relations closed under the operations $\cup, \cap, ^c, ;, ^{-1}, 1'$.

Can prove the axioms of RAs hold for algebras of binary relations.

A relation algebra is **representable** if it is isomorphic to an algebra of binary relations.

Roger Lyndon [1956] found axioms that hold in all algebras of relations but not in all relation algebras.

Shortest axioms of Roger Lyndon

J: $t \le u;v \sqcap w;x$ and $u^{-1};w \sqcap v;x^{-1} \le y;z$
$$\implies t \le (u;y \sqcap w;z^{-1});(y^{-1};v \sqcap x;z)$$

L: $x;y \sqcap z;w \sqcap u;v \le$
$x;(x^{-1};u \sqcap y;v^{-1} \sqcap (x^{-1};z \sqcap y;w^{-1});(z^{-1};u \sqcap w;v^{-1}));v$

M: $t \sqcap (u \sqcap v;w);(x \sqcap y;z) \le$
$v;((v^{-1};t \sqcap w;x);z^{-1} \sqcap w;y \sqcap v^{-1};(u;y \sqcap t;z^{-1}));z$

```
theorem Jtrue : t ⊆ u;v ∩ w;x  ∧  u⁻¹;w ∩ v;x⁻¹ ⊆ y;z
    → t ⊆ (u;y ∩ w;z⁻¹);(y⁻¹;v ∩ z;x) := by
  intro h
  intro (a,b)
  intro h₁
  rcases h with ⟨h₂,h₃⟩
  have h₄ : (a, b) ∈ u ; v ∩ w ; x :=
    Set.mem_of_mem_of_subset h₁ h₂
  rcases h₄ with ⟨h₅, h₆⟩
  rcases h₅ with ⟨c, h₇, h₈⟩
  rcases h₆ with ⟨d, h₉, H₁⟩
  have H₂ : (c, a) ∈ u⁻¹ := by rw [inv]; dsimp; trivial
  have H₃ : (c, d) ∈ u⁻¹ ; w := by use a
  have H₄ : (b, d) ∈ x⁻¹ := by rw [inv]; dsimp; trivial
  have H₅ : (c, d) ∈ v ; x⁻¹ := by use b
  have H₆ : (c, d) ∈ u⁻¹ ; w ∩ v ; x⁻¹ := by constructor;
    trivial; trivial
  have H₇ : (c, d) ∈ y ; z := Set.mem_of_mem_of_subset H₆
    h₃
  rcases H₇ with ⟨e, H₈, H₉⟩
  ...
```

```
theorem Ltrue :
  x;y ∩ z;w ∩ u;v ⊆ x;((x⁻¹;z ∩ y;w⁻¹);(z⁻¹;u ∩ w;v⁻¹) ∩
     x⁻¹;u ∩ y;v⁻¹);v := by
  intro (a,b)
  intro h
  rcases h with ⟨h1, h2⟩
  rcases h1 with ⟨h3,h4⟩
  rcases h3 with ⟨e, h3, h5⟩
  rcases h4 with ⟨d, h3, h4⟩
  rcases h2 with ⟨c, h6, h7⟩
  use c
  constructor
  use e
  constructor
  trivial
  constructor
  constructor
  use d
  constructor
  constructor
  ...
```

```
theorem Mtrue :
  t ∩ (u ∩ v ; w) ; (x ∩ y;z) ⊆ v;((v⁻¹;t ∩ w;x);z⁻¹ ∩
    w;y ∩ v⁻¹;(u;y ∩ t;z⁻¹));z := by
  intro (a,b)
  intro h
  rcases h with ⟨h1,h2⟩
  rcases h2 with ⟨c,h1,h2⟩
  rcases h1 with ⟨h3,h4⟩
  rcases h4 with ⟨d,h5,h6⟩
  rcases h2 with ⟨h7,h8⟩
  rcases h8 with ⟨e,h9,h10⟩
  use e
  constructor
  use d
  constructor
  trivial
  constructor
  constructor
  use b
  constructor
  ...
```

Ralph McKenzie's 16-element relation algebra

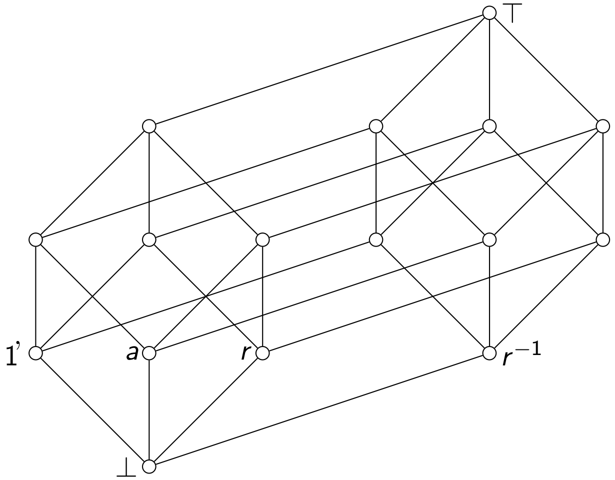
This algebra is named 14_{37} in Roger Maddux's book [5]

It is a **nonrepresentable** RA of smallest cardinality

with four atoms: $1', a, r, r^{-1}$ and top element $\top = 1' \sqcup a \sqcup r \sqcup r^{-1}$

;	$1'$	a	r	r^{-1}
$1'$	a	a	r	r^{-1}
a	a	$1' \sqcup r \sqcup r^{-1}$	$a \sqcup r$	$a \sqcup r^{-1}$
r	r	$a \sqcup r$	r	$\top$
r^{-1}	r^{-1}	$a \sqcup r^{-1}$	$\top$	r^{-1}

All 16 elements of McKenzie's algebra



```lean
inductive M : Type | e : M | a : M | r : M | r₁ : M
open M
def M.ternary : M → M → M → Prop := fun
| e, e, e => True  | e, a, a => True | e, r, r => True
| e, r₁, r₁ => True | a, e, a => True | a, a, e => True
| a, a, r => True  | a, a, r₁ => True | a, r, a => True
| a, r, r => True  | a, r₁, a => True | a, r₁, r₁ => True
| r, e, r => True  | r, a, a => True  | r, a, r => True
| r, r, r => True  | r, r₁, e => True | r, r₁, a => True
| r, r₁, r => True | r, r₁, r₁ => True | r₁, e, r₁ => True
| r₁, a, a => True | r₁, a, r₁ => True | r₁, r, e => True
| r₁, r, a => True | r₁, r, r => True  | r₁, r, r₁ => True
| r₁, r₁, r₁ => True | _, _, _ => False
def M.inv : M → M := fun | e => e | a => a | r=>r₁ | r₁=>r
def M.unary : M → Prop := fun | e => True | _ => False
```

Theorem [McKenzie 1966] *McKenzie's algebra* 14_{37} *is not representable.*

Proof. The formula **M** fails in this algebra:

Let $t = a, u = r, v = a, w = a, x = r^{-1}, y = a, z = a$.

From the table we see $u \sqcap v; w = r \sqcap a; a = r \sqcap (1' \sqcup r \sqcup r^{-1}) = r$

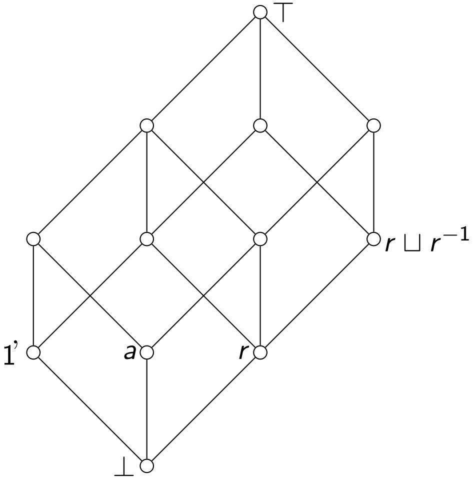
and $x \sqcap y; z = r^{-1} \sqcap a; a = r^{-1} \sqcap (1' \sqcup r \sqcup r^{-1}) = r^{-1}$.

Hence the LHS $= a \sqcap r; r^{-1} = a \sqcap (1' \sqcup a \sqcup r \sqcup r^{-1}) = a$.

However the RHS $= a; ((a; a \sqcap a; r^{-1}); a \sqcap a; a \sqcap a; (r; a \sqcap a; a)); a$

$= a; (r^{-1}; a \sqcap a; a \sqcap a; r); a = a; \bot; a = \bot$ $\square$

A 12-element subreduct of McKenzie's algebra



Using the network game in [J., Semrl 2023] one can check that this $\sim$-subreduct is not representable.

A class K of algebras has the **amalgamation property**

if for all $\mathbf{A}, \mathbf{B}, \mathbf{C} \in \mathsf{K}$ and embeddings $f\colon \mathbf{A} \to \mathbf{B}$, $g\colon \mathbf{A} \to \mathbf{C}$

there exists $\mathbf{D} \in \mathsf{K}$ and embeddings $f'\colon \mathbf{B} \to \mathbf{D}$, $g'\colon \mathbf{C} \to \mathbf{D}$ such that

$f' \circ f = g' \circ g$.

$$\begin{tikzcd} & \mathbf{B} \arrow[dr, hook, dashed, "f'"] & \\ \mathbf{A} \arrow[ur, hook, "f"] \arrow[dr, hook, "g"'] & & \mathbf{D} \\ & \mathbf{C} \arrow[ur, hook, dashed, "g'"'] & \end{tikzcd}$$

The pair $\langle f, g\rangle$ is called a **span** and $\langle \mathbf{D}, f', g'\rangle$ is an **amalgam**.

Amalgamation for residuated lattices?

Does **AP** hold for **all residuated lattices**? (**open since** < 2002)

Commutative residuated lattices satisfy $x \cdot y = y \cdot x$

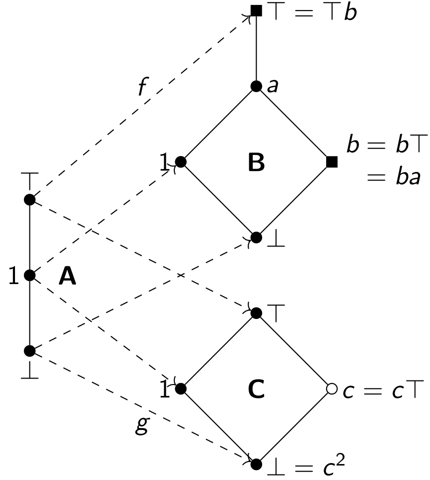
Kowalski, Takamura [2004]: **AP** **holds** for commutative RLs

Many other results are know for various subvarieties, e.g.,

Heyting algebras are integral ($x \leq 1$) idempotent ($xx = x$) RLs

Maksimova [1977]: Exactly 8 varieties of Heyting algebras have **AP**

Theorem: **AP** fails for RL



black = idempotent, round = central

Proof: Straightforward to check **A**, **B**, **C** are RLs and f, g are embeddings.

Assume by contradiction $\exists$ amalgam **D**.

$1 \vee c = \top$ and $1 \vee b = 1 \vee a = a < \top$

hence $g'(c) \neq f'(a)$ and $g'(c) \neq f'(b)$.

So f', g' are inclusions and $\mathbf{B}, \mathbf{C} \leq \mathbf{D}$

Now, since $c = c\top$ and $\top b = \top$,

in **D** we have $cb = c\top b = c\top = c$.

Moreover $\top = 1 \vee c$ and $c^2 = \bot$,

show $c = \top c = \top bc = (1 \vee c)bc$

$= bc \vee cbc = bc \vee c^2 = bc \vee \bot = bc$

(using $\bot \leq c$ implies $\bot = b\bot \leq bc$).

But also $b = b\top = b(1 \vee c) = b \vee bc$

gives $c = bc \leq b \leq a$. Hence

$\top = 1 \vee c \leq a \vee c = a$; contradiction!

Some remarks

The proof on the previous slide also shows that the **AP** already fails for the variety of **distributive residuated lattices**,

as well as for the $\{\backslash, /\}$-free subreducts of residuated lattices, i.e., for **lattice-ordered monoids**.

Also the proof does not depend on meet or on the constant 1 being in the signature, so the following varieties do not have **AP**:

- **residuated lattice-ordered semigroups**,
- **lattice-ordered semigroups**,
- **residuated join-semilattice-ordered semigroups** and
- **join-semilattice-ordered semigroups**.

Similar examples show that **AP** fails in idempotent RLs and in involutive FL-algebras.

References

- S. Givant, Relation Algebras, Vol 1, 572pp, Vol 2, 605pp, Springer 2017.
- L∃∀N Programming Language and Theorem Prover, `https://lean-lang.org/`
- L∃∀N Community and MathLib, `https://leanprover-community.github.io/`
- R. Hirsch, I. Hodkinson: *Relation Algebras by Games.* North Holland/Elsevier Vol 147 (2002)
- R. Maddux, *Relation Algebras.* Elsevier Vol 150 (2006), 731pp.
- R. McKenzie, *Representations of integral relation algebras.* Michigan Math. J., 17, (1970), 279–287.
- W. McCune, L. Wos, *Otter*, Journal of Automated Reasoning, 18, (1997), 211–220.

THANKS!