Commutative Residual Algebra motivation, decision, and applications

Vincent van Oostrom[0000−0002−4818−7383]

University of Sussex, School of Engineering and Informatics, Brighton, UK
vvo@sussex.ac.uk

Abstract. Commutative residual algebras (CRAs) are algebras having an axiomatised residuation operation. Key examples of CRAs are cut-off subtraction (monus) on the natural numbers, set and multiset difference, and cut-off division (dovision) on the positive natural numbers.
Here we revisit CRAs, showing the usual construction of a commutative group (the integers) out of a monoid (the natural numbers) as pairs, can be extended and generalised, by constructing the latter from CRAs (the bits) as sequences (up to order), and yielding a commutative lattice-ordered group. This then affords a decision procedure for CRAs, by employing results known from the literature.
We show CRAs arise from residual systems (going back to Stark and unpublished work of Plotkin) by imposing a commutativity condition, and we identify residuation as a Skolemised diamond property. Accordingly, we let it together with its associated rewrite technique of tiling take front and centre stage in our approach. We finally show that CRAs are at the natural level of abstraction to state and prove some examples from the literature, in particular the inclusion–exclusion principle, making the latter applicable to, among others, (measurable) multisets.

Keywords: residuation · diamond property · tiling · commutative residual algebras · commutative ℓ-groups · multisets · inclusion–exclusion.

1 Introduction

A standard way to prove sets A, B *the same* is to show both inclusions $A \subseteq B$ and $B \subseteq A$. This can be reformulated as both differences being empty $A - B = \emptyset = B - A$. Similarly, natural numbers n, m may be proven *distinct* by showing one of the cut-off subtractions (*monus*) $n \dot{-} m$ and $m \dot{-} n$ to be non-0. In this paper, we further develop the theory of CRAs (*commutative residual algebra* [21]) enabling this reasoning. CRAs are algebras $\langle A, 1, / \rangle$ having a *residuation* operation / and *unit* 1 and *residuation* laws that are so few that we can give them immediately:

$$a/1 = a \tag{1}$$

$$(a/b)/(c/b) = (a/c)/(b/c) \tag{4}$$

$$(a/b)/a = 1 \tag{5}$$

$$a/(a/b) = b/(b/a) \tag{6}$$

The above CRA laws are independent. We are interested in the equational theory of CRAs (Sec. 3). Examples of derivable laws are:

$$a/a = 1 \tag{2}$$
$$1/a = 1 \tag{3}$$

To show the equational theory is decidable we proceed in two steps. First, we show (Sec. 4) CRAs embed in CRACs, CRAs *with composition* $\cdot$ satisfying:

$$c/(a \cdot b) = (c/a)/b \tag{7}$$
$$(a \cdot b)/c = (a/c) \cdot (b/(c/a)) \tag{8}$$
$$1 \cdot 1 = 1 \tag{9}$$

Next, we show (Sec. 5) CRACs embed in commutative ℓ-groups (*lattice-ordered*) having a suitable *inverse* $^{-1}$. By decidability of the latter [18, 37] we conclude.

We proceed in such a way (as often done in analogous situations) since CRAs are hard to work with directly, due to the contravariance of residuation (in its second argument). Indeed, for many of the (equational) proofs in this paper we employed an ATP (Prover9 and Mace4 [22]) to obtain and check them. On the other hand, commutative ℓ-groups are well-behaved and well-studied, cf. [15].

The first embedding generalises how the bits $\mathbb{B} := \{0, 1\}$ can be embedded in the natural numbers $\mathbb{N}$ by viewing the latter as (non-empty) bit*strings* modulo 0-*contraction* (which yields bitstrings having at most one 0), i.e. natural numbers in unary, on which *addition* $+$ is represented by string-concatenation.

The second embedding generalises the standard embedding of $\mathbb{N}$ in the integers $\mathbb{Z}$, viewing the latter as *pairs* of natural numbers modulo *normalisation* (yielding pairs where at most one component is non-0), i.e. integers as signed natural numbers, on which *unary minus* $-$ is represented by pair-swapping.

Further examples are given below and include the usual embedding of the prime numbers first in the positive natural numbers (as multisets of prime numbers) and next in the (non-negative) rationals (as normalised fractions), and of the embedding of sets as multisets first and in signed multisets next. The laws of CRAs are sufficiently strong to enable both, generalising $\mathbb{B} \hookrightarrow \mathbb{N} \hookrightarrow \mathbb{Z}$.

We show CRAs are equivalent to cBCKrc's (*commutative BCK algebras with relative cancellation* [14]), which gives an alternative route to their embedding in commutative ℓ-groups via known results for cBCKrcs. Still, we present our embeddings as they often can be seen as the *untyped* versions of known constructions for *typed* systems as explained in Sec. 2. This will allow us not only to reuse, but also to present the embeddings in an intuitive diagrammatic way.

Sample problems we will tackle (Secs. 6) for the reader to ponder are:

Problem 1. Can we give a calculational proof of that for any two propositions, one entails the other, $(p \to q) \vee (q \to p) = \top$, using only its operations $\{\to, \vee, \top\}$?

Problem 2 (EWD1313). The note [10] asks for a *nice* calculational proof of that if $\gcd(n, m) = 1$, n and m are relatively prime, then $\gcd(n, m \cdot k) = \gcd(n, k)$.

Problem 3 (Mechanical Mathematicians). How to show [5] $(\gcd(n,m) = 1 \text{ and } \ell \mid n \cdot m \text{ and } n' = \gcd(\ell,n) \text{ and } m' = \gcd(\ell,m)) \implies (n' \cdot m' \mid \ell \text{ and } \ell \mid n' \cdot m')$?

Problem 4 (Inclusion–Exclusion). $\bigcup M_I = \left(\biguplus_{\emptyset \subset J \subseteq\!\!\!\subseteq I} \bigcap M_J\right) - \left(\biguplus_{\emptyset \subset J \subseteq\!\!\!\supseteq I} \bigcap M_J\right)$ for a finite family M_I of multisets? Here $\uplus, -, \cup, \cap$ denote multiset sum, difference, union, intersection and $\subseteq\!\!\!\subseteq$ / $\subseteq\!\!\!\supseteq$ taking subsets of odd / even cardinality. For instance, does it hold for $I := \{1,2,3\}$, $M_1 := [a,b]$, $M_2 := [b,c]$, $M_3 := [c,a]$?

Related work There is a vast literature on residuated lattices and on embedding algebras in groups. Our vantage point, of starting with algebras with residuation but without composition, we only found in the literature on cBCKrc's cited. Traditional accounts are biased in that they focus on the loop, composition and reverse operations with their (monoid and group) laws. We show that residuation comes *prior* to them. (Indeed CRAs need neither have composition nor reverse; the CRAs of bits and sets do not.) That then also explains the pervasiveness of tiling techniques in the literature, since we identify residuation as the Skolem function arising from the diamond property, i.e. from *having* tiles. We argue this perspective, coming from rewriting, is novel, unifying and fruitful.

2 Intuition for residual from removal / rewriting

Our first take on CRAs is that one may think of their objects as being *composite*, made up of *components*, such that the residual a/b of a after b is obtained by *removing* b's components from a's (whence their name residual algebra). To denote that nothing is left, $a/b = 1$, we write $a \leqslant b$ (Def. 1). In the case of sets and natural numbers this *natural* order $\leqslant$ instantiates to the usual subset $\subseteq$ and less-than-or-equal $\leq$ orders. The removing-idea immediately gives intuition for the unit laws (1)–(3) and for (5) expressing monotonicity of removal. A way to understand the two remaining laws (4) and (6) is as being designed to guarantee $\leqslant$ is a *partial order*, enabling the method of proving equality by two inequalities, central to our endeavour. Indeed, we check that if $a \leqslant b \leqslant c$, then $a \leqslant c$:

$$a/c \overset{(1)}{=} (a/c)/1 \overset{\text{hyp}}{=} (a/c)/(b/c) \overset{(4)}{=} (a/b)/(c/b) \overset{\text{hyp}}{=} 1/(c/b) \overset{(3)}{=} 1, \text{ and}$$
$$a \overset{(1)}{=} a/1 \overset{\text{hyp}}{=} a/(a/b) \overset{(6)}{=} b/(b/a) \overset{\text{hyp}}{=} b/1 \overset{(1)}{=} b, \text{ if } a \leqslant b \leqslant a.$$

Our second, key, take is based on interpreting the carrier as comprising *steps* of

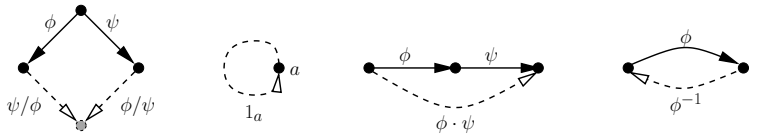

Fig. 1. Operations on steps: residuation, loop, composition, reverse

a *rewrite system* $\to$ [25][36, Sec. 8.2].[1] Operations *must* then be lifted to steps in a way respecting sources and targets. Fig. 1, where ϕ, ψ, χ range over steps, depicts *how*. Each operation can be seen as the Skolem function for the formula depicted, where ordinary arrows denote *universally* quantified steps, and dashed open-headed arrows *existentially* quantified steps. For instance, Skolemising the formula $\forall \phi, \psi$ if $\mathsf{tgt}(\phi) = \mathsf{src}(\psi)$ then $\exists \chi$ with $\mathsf{src}(\phi) = \mathsf{src}(\chi)$ and $\mathsf{tgt}(\psi) = \mathsf{tgt}(\chi)$ as depicted second-from-the-right, yields *composition* $\cdot$ mapping any pair of *consecutive* steps to a step from the source of the former to the target of the latter. Although maybe not usually perceived as Skolemisations, the operations $1, \cdot, ^{-1}$ are very well-known (in any case with additional laws making them *units* (unit-laws), *morphisms* (associativity), and *inverses* (left/right-inverse laws)). We focus on the odd one out, on residuation $/$, which comes *prior* to the others.

Residuation $/$ arises from Skolemising the so-called *diamond* property [25, 3, 36] expressing that for any *peak* ϕ, ψ (a pair of *co-initial* steps, $\mathsf{src}(\phi) = \mathsf{src}(\psi)$), there exists a *valley* ψ', ϕ' (a pair of *co-final* steps, $\mathsf{tgt}(\psi') = \mathsf{tgt}(\phi')$), that is *composable* to it, $\mathsf{tgt}(\phi) = \mathsf{src}(\psi')$ and $\mathsf{tgt}(\psi) = \mathsf{src}(\phi')$. Skolemising this formula *a priori* gives rise to *two* functions f, g, corresponding to mapping ϕ, ψ to the first component ψ' (by $f(\phi, \psi)$) respectively the second component ϕ' (by $g(\phi, \psi)$) of the pair ψ', ϕ'. But we may use a *single* one $/$ as depicted in Fig. 1, as follows from that we may set $\phi/\psi := g(\phi, \psi)$ if $\phi \preccurlyeq \psi$ and $f(\psi, \phi)$ otherwise, for *some*[2] total order $\preccurlyeq$ on steps.

Now thinking of *steps* as being *composite*, made up of (*parallel / independent!*) *tasks*, we may interpret ϕ/ψ as those tasks of ϕ that remain to be done *after* ψ. The intuition for ϕ/ψ and ψ/ϕ yielding a diamond, a *tile*, is that either way a *common reduct* is reached by having performed the tasks of *both* ϕ *and* ψ*, removing double ones*. This brings that we may reason (in)formally by means of *tiling*, by repeatedly filling a peak ϕ, ψ by a tile, as we do below (Figs. 2 and 5). Tiling is pervasive in the rewriting literature since its inception [8, 25, 19, 23, 27].

Remark 1. If a (confluent) rewrite system $\to$ doesn't have the diamond property, we may try to build its *diamond* closure, a least extension having the diamond property. To that end one may iteratively *adjoin joining-reductions as steps* [26, Sec. 3.1].[3] This often works, if only as intuition pump; in the λ-calculus, from β*-steps* iterative adjoining yields *parallel* β*-steps* as diamond closure, *in the limit*.

Remark 2. We will refer to the transitions between ordinary algebras and those having steps as carrier and operations subject to Fig. 1, as *typing / untyping*. For instance, we will refer to a category / groupoid as a typed monoid / group [32]. Note that though typed algebras could be handled at the ordinary algebraic level,

[1] The structure of *rewrite systems*, of sets of objects and steps equipped with maps $\mathsf{src}, \mathsf{tgt}$ from the latter to the former, has been invented many times over; e.g. programming language theorists might call them *abstract machines*, and mathematicians may call them *pre-categories*, *quivers*, or *multidigraphs* depending on their area.

[2] Existence may be assumed using the axiom of choice, even of a well-order.

[3] We learned this process from Hans Zantema and facetiously dubbed it *subcommufication* [26], but nowadays call it *faceting* as one cuts diagrams into diamonds [27].

it would involve explicitly working with *partial* operations; one would lose the *typing-constraints* implicit in Fig. 1, e.g. that composition requires *consecutive* steps. Also note that any algebra can be trivially typed by taking a rewrite system on a singleton object, having as steps the elements of the carrier.

The intuition conveyed is then that steps of a rewrite system having a residuation can be thought of as being *parallel*, extended in space, and that composition is *sequential*, extends them in time. This reinforces our point that residuation in itself is interesting, parallel steps are / space is, and meshes well with the idea underlying rewriting, to lift properties from steps to compositions thereof, i.e. to reductions.

The idea to exploit typed algebras is that our embeddings, from CRAs in CRACs in commutative ℓ-groups, are 'just commutative untyped' versions of known typed embeddings, say of *simple* braids in *positive* braids in (all) braids [9], or of a *reversible* rewrite system $\to$ (both $\to$ and $\leftarrow$ are deterministic) in reversible *reductions* $\twoheadrightarrow$ first and in reversible *expansions* and reductions $\twoheadleftarrow \cup \twoheadrightarrow$ next (due to *affluence* [30]), in each case enabled by having a residuation satisfying the *typable* laws (1)–(4) (but *not* (5),(6), which are *un*typable in that they make $\mathsf{src} = \mathsf{tgt}$ forcing a collapse to a discrete / singleton carrier, to algebra), enabling proof-by-tiling. (E.g. it is *because* sets have a residuation (difference) that they embed in the CRAC of multisets, which then embeds in the commutative ℓ-group of signed multisets.)

3 Commutative Residual Algebras

We further intuitions by expanding our stock of example CRAs, deriving some simple but interesting laws, defining a number of derived operations to see how they instantiate on the example CRAs, recapitulating the multiset representation theorem for well-founded CRAs, relating CRAs to cBCKrc's, and seeing how CRAs arise by assuming commutativity and untyping residual *systems* [31, 35, 36], rewrite systems satisfying the laws (1)–(4) depicted in Fig. 2.[4] Unless stated otherwise we work with a CRA $\langle A, 1, / \rangle$.

Example 1 (Some CRAs). The natural numbers $\langle \mathbb{N}, 0, \dot{-} \rangle$ with monus seen before; the positive natural numbers $\langle \mathsf{Pos}, 1, \cdot/ \rangle$ where $\cdot/$ is cut-off division (*dovision*) $n \cdot/ m := n/\gcd(n, m)$; the multisets over A (maps from A to $\mathbb{N}$) $\langle \mathsf{Mst}(A), \emptyset, - \rangle$ with multiset difference; the non-negative reals $\langle \mathbb{R}_{\geq 0}, 0, \dot{-} \rangle$ with monus; the real numbers ≥ 1 $\langle \mathbb{R}_{\geq 1}, 1, \div \rangle$, where $\div$ is *truncated* division $x \div y := x/\min(x, y)$; the subsets of A $\langle \mathsf{Set}(A), \emptyset, - \rangle$ with set-difference. One checks laws (1)–(6) hold.

Example 2 (sub-CRAs). Taking a dc-subset (*downward-closed* w.r.t. $\leqslant$) of A induces a CRA again. For instance, $\mathbb{B}$, or any initial segment of $\mathbb{N}$, is a CRA. Restricting Pos to the prime numbers yields a CRA Prm, but restricting for any prime number p to its prime powers does so too, $p^{\mathbb{N}}$. From $\mathsf{Mst}(A)$ CRAs are

[4] In Fig. 2 we use a, b, c where ϕ, ψ, χ would be more precise, but distracting. Colours are there only to easily keep track of residuals (they have the same colour).

obtained by restricting multiplicities to ≤ 1 ($\mathsf{Set}(A)$), to be bounded ($\mathsf{Mst}_{\mathsf{bnd}}(A)$), and/or requiring supports to be finite ($\mathsf{Mst}_{\mathsf{fin}}(A)$), where for a multiset M and $a \in A$, $M(a)$ is the *multiplicity* of a and $\{a \in A \mid M(a) > 0\}$ the *support* of M.

To give a flavour of CRA reasoning we show two interesting laws (used in Thm. 2, but also interesting to check on the example CRAs) whose proofs being easy enough but not quite trivial, illustrates that CRA proofs are best left to ATPs (for equational proofs that's easy), as we will predominantly do below. The first one says that the order of removing is irrelevant, and the second one captures that the two parts a/b and b/a of the *symmetric difference* of a and b are disjoint.[5]

Proposition 1. *$(a/b)/c = (a/c)/b$ and $(a/b)/(b/a) = a/b$.*

Proof. Abbreviating $a/(a/b)$ to $a \wedge b$ (cf. Def. 1), the former is seen to hold by

$$(a/b)/c \overset{(1,5)}{=} ((a/b)/c)/((c/b)/c) \overset{\text{(i)}}{=} ((a/c)/(b/c))/(c \wedge b) \overset{\text{(ii)}}{=} (a/c)/b$$

where (i) and (ii) are derived as (instances of) respectively:

$$((a/b)/c)/((c/b)/c) \overset{(4)}{=} ((a/b)/(c/b))/(c/(c/b)) \overset{(4),\text{def}}{=} ((a/c)/(b/c))/(c \wedge b)$$

$$(a'/(b/c))/(c \wedge b) \overset{(6),\text{def}}{=} (a'/(b/c))/(b \wedge c) \overset{\text{def},(4)}{=} (a'/b)/((b/c)/b) \overset{(5,1)}{=} a'/b$$

For the latter, saying parts a/b and b/a of the symmetric difference are disjoint:

$$(a/b)/(b/a) \overset{(1),(5)}{=} ((a/b)\wedge a)/((b/a)\wedge b) \overset{\text{def},(6)}{=} (a/(b\wedge a))/(b/(b\wedge a)) \overset{(4),(5),(1)}{=} a/b.$$

Residuation only yields smaller elements making that composition is not a total operation on CRAs in general. Still within the confines of a given CRA we can determine whether or not an element would be a composition. For instance, for $\mathbb{B}$ and $\mathsf{Set}(\{x, y, z\})$ the compositions of 1 and 1 respectively of $\{x, y\}$ and $\{y, z\}$ do not exist, but those of 0 and 1 and of $\{x\}$ and $\{z\}$ do (1 resp. $\{x, z\}$).

Definition 1 (Derived operations $\leqslant$, $\wedge$, $\cdot$, $\vee$).

natural order	$a \leqslant b := a/b = 1$		
meet	$a \wedge b := a/(a/b)$		
composition	$a \cdot b := c$	*if* $a/c = 1$ *and* $c/a = b$	(partial)
join	$a \vee b := a \cdot (b/a)$		(partial)

By *partial* we mean that such an element need not exist, i.e. the expression may not denote, but that if it does (as we may express by $\downarrow$), then uniquely so. (For f a partial function and expressions $e_1, \ldots, e_n$, the expression $e := f(e_1, \ldots, e_n)$ *denotes* v, if e_i denotes v_i and $(v_1, \ldots, v_n)$ is in the domain of f, and f applied to it has value v.[6]) *Kleene* equality $e \simeq e'$ asserts that if either of e, e' denotes then so does the other and then their denotations are equal.

See Tab. 1 for the derived operations and properties for some CRAs from Ex. 1. Note that if $a \cdot b\downarrow$ then $a \vee b\downarrow$ but not necessarily the other way around. The names of derived operations are justified by the next lemma, used in Sec. 6.

[5] Cf. [33] for reasoning about *distance* based on the symmetric difference.

[6] Denoting is *strict*; e.g. $0 \cdot \frac{1}{0}$ does not denote as its sub-expression $\frac{1}{0}$ doesn't.

	CRA	$\mathbb{N}$	$\mathbb{R}_{\geq 0}$	$\mathsf{Mst}(A)$	$\mathsf{Set}(A)$	Pos
unit	1	0	0	$\emptyset$	$\emptyset$	1
residuation	$/$	$\dot{-}$	$\dot{-}$	$-$	$-$	$\cdot/$
natural order	$\leqslant$	$\leq$	$\leq$	$\subseteq$	$\subseteq$	$\mid$
total order?		✓	✓	✗	✗	✗
well-founded?		✓	✗	✓ (fin)	✓ (fin)	✓
meet	$\wedge$	min	min	$\cap$	$\cap$	gcd
composition	$\cdot$	$+$	$+$	$\uplus$	$\cup$ (if $\downarrow$)	$\cdot$
join	$\vee$	max	max	$\cup$	$\cup$	lcm

Table 1. Derived operations for some CRAs from Ex. 1

Lemma 1 (Algebraic structure of CRAs).

- $\langle A, \leqslant \rangle$ *is a partial order; enables proving* $a = b$ *by inclusions* $a \leqslant b$ *and* $b \leqslant a$*;*
- $\langle A, \wedge \rangle$ *is a meet-semilattice; and* $a \leqslant b$ *iff* $a \wedge b = a$*; and* $1 \wedge a = 1$*;*
- $\langle A, 1, \cdot \rangle$ *is a* partial *commutative monoid; and* $a \leqslant b$ *iff* $a \cdot c \simeq b$ *for some* c*;*
- $\langle A, \vee \rangle$ *is a* partial *join-semilattice; and* $a \leqslant b$ *iff* $a \vee b \simeq b$*; and* $1 \vee a = a$*;*
- $\langle A, \leqslant \rangle$ *is a* partial *lattice; and* $a \vee (a \wedge b) \simeq a$ *and* $a \wedge (a \vee b) \simeq a$ *if* $(a \vee b)\!\downarrow$*;*
- $\langle A, \wedge, \vee \rangle$ *is a partial distributive lattice;* $(a \vee b) \wedge c \simeq (a \wedge c) \vee (b \wedge c)$ *if* $(a \vee b)\!\downarrow$*.*

Proof. All proofs were done by ATP. See App. B for the Prover9 representation we used, and its instantiation to a proof of the last item.

If composition is total CRAs are distributive lattices, not necessarily bounded ($\mathbb{N}$). We recall the representation theorem for well-founded CRAs [21, Sec. 5], expressing that it's not wrong to think of elements of such simply *as* multisets.

Definition 2 (Decomposition). *Given a partial commutative monoid* $\langle A, 1, \cdot \rangle$*. Call* a indecomposable[7] *if* $a \neq 1$ *and* $a = b \cdot c$ *implies* $b = 1$ *or* $c = 1$*, and say a multiset* $[a_1, \ldots, a_n]$ *is a* decomposition *of* a *if each* a_i *is indecomposable and* $a \simeq a_1 \cdot \ldots \cdot a_n$*.* Divisibility *is defined by* $a \leqslant b$ *if* $b \simeq a \cdot c$ *for some* c*.*

These notions apply to CRAs via the partial commutative monoid of their composition, and the natural order of the CRA then *is* the divisibility order. Having *unique decompositions* means that decompositions exist uniquely.

Theorem 1 (Multiset representation [21]). *Well-founded CRAs have unique decomposition, and any well-founded CRA* $\langle A, 1, / \rangle$ *is isomorphic to the CRA* $\langle A', \emptyset, - \rangle$*, with* A' *the initial segment of finite multisets of indecomposables.*

For the CRA $\mathbb{N}$ the first item boils down to the triviality $n = \overbrace{1 + \ldots + 1}^{n}$, but for Pos it corresponds to the Fundamental Theorem of Arithmetic (FTA) saying that every positive natural number has a unique decomposition into prime numbers. Though the CRA need not be finite ($\mathbb{N}$ is not), well-foundedness is essential (unique decomposition fails for $\mathbb{R}_{\geq 0}$ in the absence of indecomposables).

[7] For rings this is known as being *irreducible*.

Next, we show CRAs have the same equational theory as cBCKrc's (*commutative BCK algebras with relative cancellation* as introduced by Dvurečenskij and Graziano [14]).[8] BCK and BCI[9] algebras are algebraic structures introduced in [17, 16, 2] unifying set difference and implication in propositional logic. Many variations have been studied [14, 13, 11, 12], but we focus on cBCKrc's.

Definition 3. $\langle A, 1, / \rangle$ *is a cBCKrc if for all* a, b, c

$$(a/b)/(a/c) \leqslant c/b \tag{10}$$
$$a/(a/b) \leqslant b \tag{11}$$
$$a \leqslant a \tag{12}$$
$$a = b \quad \text{if } a \leqslant b \text{ and } b \leqslant a \tag{13}$$
$$1 \leqslant a \tag{14}$$
$$a \wedge b = b \wedge a \tag{15}$$
$$b = c \quad \text{if } a \leqslant b, c \text{ and } b/a = c/a \tag{16}$$

where, as for CRAs, $a \leqslant b$ *if* $a/b = 1$ *and* $a \wedge b$ *abbreviates* $a/(a/b)$.

Theorem 2. $\langle A, 1, / \rangle$ *is a CRA iff it is a cBCKrc.*

Proof. We factor our proof through the alternative equational specification of cBCKrc's as given in [11] comprising five laws: (2),(1),(6) and the two laws of Prop. 1.[10] That these laws hold for CRAs is then immediate. For the other direction we employed Prover9. Only showing that the second law of Prop. 1 holds for cBCKrc's took substantial time, 1.5 hours; see App. B.

By the theorem, results for cBCKrc's can be transferred to CRAs and vice versa, in particular that the former embed in commutative ℓ-groups. We follow a different route here, by untyping typed *tiling* constructions for residual systems (not possible for cBCKrc's' laws; untyping them makes $\mathsf{src} = \mathsf{tgt}$). Toward that goal observe that CRAs arise from residual *systems* by assuming commutativity.

Remark 3. They do, since that is how I constructed them: At the time (around MM) for a Coq formalisation I needed properties of *multisets* that were lacking from its libraries. Just having developed residual systems, I noted that since those could be used for reasoning about *lists* [36, Intro of Sec. 8.7], adjoining *commutativity* laws should be sufficient to reason about *multisets*. To obtain the residuation-laws for commutativity, I took a *peak* spanned by *both* possible orders of two consecutive steps, and took the laws arising from that *tiling* should yield a valley comprising 1s only. Proceeding like this, as in the diagram depicted at the bottom of Fig. 2, yields four laws along the top–right of the *valley*: each of $(b/a)/b$, $(a/(a/b))/(b/(b/a))$, $(b/(b/a))/(a/(a/b))$, and $(a/b)/a$ *should* be 1 (to

[8] CRAs and cBCKrc's were introduced independently, around the turn of the century.
[9] *BCI* has the law $a = 1$ if $a \leqslant 1$ instead of (14).
[10] On page 5 of [13] and also in the proof of Thm. 5.2.29 of [11], law (3) is given instead of law (1); that is incorrect, as a 2-point model with / the constant-1-function shows.

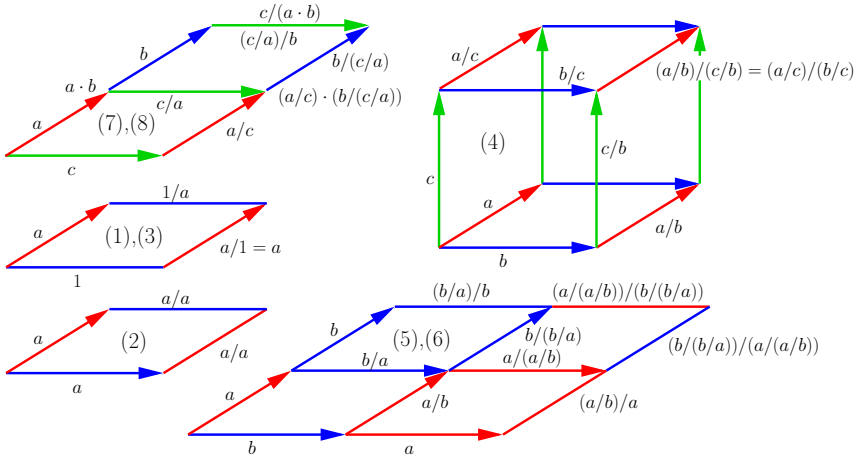

Fig. 2. Visualisation of CRA laws (1)–(6) and CRAC laws (7),(8)

make both possible orders of the two steps the same). This gave rise to laws (5) and (with anti-symmetry of $\leqslant$) (6).

The law (4) is known as (Lévy's) *cube* law [20], for reasons clear from Fig. 2. It captures *causal independence* of the trident (3-*peak*) a, b, c hence frequently plays a pivotal rôle in fields where causality does [31], e.g. in the λ-calculus [20], concurrency [35], in Garside theory [9], and in Wolfram's physics project, cf. [29].

We developed residual systems in [36] off Stark's CTSs (*concurrent transition systems* [35]), but they were introduced (without a name) by Plotkin already in [31]. Concrete residual systems are omnipresent: e.g. parallel β-steps in the λ-calculus [8, 20], simple braids in algebra [9], left-convex sets of positions in self-distributivity [34]; if one looks, one finds residual systems everywhere: From the fact that $A \mid B$ is read in probability as A *after* B, one may already suspect residuation is at play, as indeed it is. Taking the *event* $A \mid B$ as notation for a

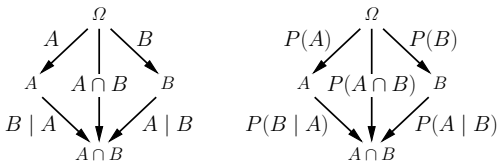

Fig. 3. Bayes' Theorem as map P on residual systems: from events to fractions

step from B to $A \cap B$, and leaving B implicit if it's the whole sample space Ω, the diamond property holds (Fig. 3 left; laws (1)–(4) hold by being a semilattice). Bayes' Theorem $P(A) \cdot P(B \mid A) = P(A \cap B) = P(B \cap A) = P(B) \cdot P(A \mid B)$ is then nothing but a map P from it to fractions (of the cardinalities; Fig. 3 right).

4 Embedding CRAs in CRACs

Recall CRACs are CRAs satisfying the residuation laws for *composition* (7)–(9). Laws (7) and (8) feature prominently in rewriting and concurrency theory [4, 31, 35, 36] and are obtained by *tiling* as visualised in Fig. 2 (top–left): the laws simply decree that tiling of $a \cdot b, c$, so of the *composition* $a \cdot b$ and c, is the same thing as tiling for its first component a, c followed by tiling with its second $b, c/a$.

Remark 4. In CRAs with *derived* composition $\cdot$ *partial* versions of (7)–(8) hold: $c/(a \cdot b) = (c/a)/b$ and $(a \cdot b)/c \simeq (a/c) \cdot (b/(c/a))$ if $(a \cdot b)\!\downarrow$, and $1 \cdot 1 = 1$.

Vice versa, composition $\cdot$ in CRACs *satisfies* the laws of *derived* composition in CRAs: $a/(a \cdot b) = (a/a)/b = 1$ and $(a \cdot b)/a = 1 \cdot b = (1 \cdot b)/((1 \cdot b)/b) = b$.

As known [31, 35, 36] residual systems, rewrite systems satisfying (typed) laws (1)–(4), embed in residual systems *with composition* satisfying (typed) laws (7)–(9), or in the terminology of [35]: concurrent transition systems embed in computation categories. This is shown in two phases: first residuation / is extended from the steps of a rewrite system $\to$ to a residuation $/\!\!/$ on compositions thereof, i.e. to *reductions* $\twoheadrightarrow$ [35, Lem. 2.3.1][36, Lem. 8.7.47] by means of *tiling* [35, Fig. 3][36, Fig. 8.50]. Next, to regain that the natural order is a partial order, that it is anti-symmetric, one quotients out projection equivalence $\equiv$, where two reductions are *projection* equivalent if the result of *tiling* their peak yields a valley of only 1s [35, Thm. 2.5][36, Prop. 8.7.48]. To embed a CRA in a CRAC it now suffices to untype that construction, giving *lists* (of objects) instead of *reductions* (of steps), and then to impose commutativity yielding *multisets* instead of *lists*:

Theorem 3. *$\mathcal{C} = \langle A, 1, / \rangle$ embeds in CRAC $\mathcal{C}^* = \langle \mathsf{Mst}_{\mathsf{fin}}(A)/\!\equiv, \emptyset, /\!\!/, \uplus \rangle$.*

Proof. For a family a_I with I finite we write $[a_I]$ to denote its multiset. Let $I := \{0, \ldots, n \dot{-} 1\}$ and $J := \{0, \ldots, m \dot{-} 1\}$. Define residuation $[a_I] /\!\!/ [b_J] := [a'_I]$ by *tiling* as in Fig. 4 putting family members in *some* order. (Residuation may

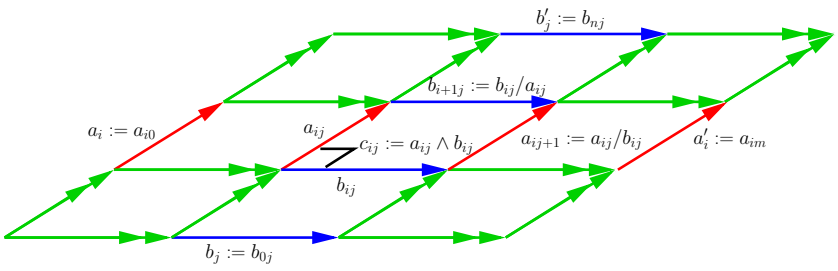

Fig. 4. Projection equivalence of multisets by tiling with CRA diamonds

be *computed* by viewing (7)–(8) as rules and eliding units 1 of $\mathcal{C}$.) By tiling with the cube law (4) and commutativity, as at the bottom of Fig. 2, the resulting multiset is seen to be independent of the chosen order, making $/\!\!/$ well-defined. On

such multisets *projection* equivalence is $\equiv := \sqsubseteq \cap \sqsupseteq$ with $\sqsubseteq$ defined by $[a_I] \sqsubseteq [b_J]$ if $[a_I] /\!\!/ [b_J] = \emptyset$; $\sqsubseteq$ is a quasi-order since laws (1)–(4) hold as inherited from the same for residual systems, and laws (5)–(6) hold by the reasoning for them in Fig. 2. Setting the unit to $\emptyset$ and composition to multiset sum $\uplus$, laws (7),(8) are inherited from residual systems with composition, and (9) holds since $\emptyset \uplus \emptyset = \emptyset$. $\mathcal{C}$ is embedded in $\mathcal{C}^*$ by mapping objects to singletons, $a \mapsto [a]$.

Projection equivalence is *needed* for the natural order to be a partial order:

Example 3. Let $\mathcal{C} := \langle \{0, \ldots, 9\}, 0, \dot{-} \rangle$ be the CRA of *digits*, a sub-ARS of $\mathbb{N}$. It has some compositions, e.g. $7 = 3 + 4$ (since $3 \dot{-} 7 = 0$ and $7 \dot{-} 3 = 4$) but others not, e.g. $7 + 6$ and $9 + 4$, are *not* defined in $\mathcal{C}$. These are represented as $[7, 6]$ and $[9, 4]$ in $\mathcal{C}^*$, which therefore *should* be projection equivalent, and they are: $[7,6] /\!\!/ [9,4] \overset{(7)}{=} ([7,6] /\!\!/ [9]) /\!\!/ [4] \overset{(8)}{=} [7 \dot{-} 9, 6 \dot{-} (9 \dot{-} 7)] /\!\!/ [4] = [0,4] /\!\!/ [4] = \emptyset$ resp. $[9,4] /\!\!/ [7,6] \overset{(7)}{=} ([9,4] /\!\!/ [7]) /\!\!/ [6] \overset{(8)}{=} [9 \dot{-} 7, 4 \dot{-} (7 \dot{-} 9)] /\!\!/ [6] = [2,4] /\!\!/ [6] = \emptyset$.

Each of $\mathcal{C}^*$ and $\langle \mathbb{N}, 0, \dot{-} \rangle^*$ and $\langle \mathbb{B}, 0, \dot{-} \rangle^*$ is isomorphic to the CRAC $\langle \mathbb{N}, 0, \dot{-}, + \rangle$, both $\langle \mathsf{Pos}, 1, \cdot / \rangle^*$ and $\langle \mathsf{Prm}, 1, \cdot / \rangle^*$ are isomorphic to the CRAC $\langle \mathsf{Pos}, 0, \cdot /, \cdot \rangle$, and $\langle \mathsf{Set}(A), \emptyset, - \rangle^*$ to the CRAC $\langle \mathsf{Mst}_{\mathsf{bnd}}(A), \emptyset, -, \uplus \rangle$.

Remark 5. P in Fig. 3 maps a residual system with composition to a CRAC.

Remark 6. A multiset can typically be seen as a multiset sum of sets in many ways. Greedily decomposing / topologically multisorting [9, 29], repeatedly selecting a *maximal* set, yields a unique representation (Fig. 6 left) analogous to the Foata normal form in trace monoids; cf. Gross–Knuth reduction in $\lambda\beta$-calculus.

By refining the proof of Thm. 3, the embedding is seen to be downward closed, in that the only objects in $\mathcal{C}^*$ below an embedded object of $\mathcal{C}$ are other such.

Lemma 2. *$\mathcal{C}$ embeds downward-closedly in $\mathcal{C}^*$, i.e. $M /\!\!/ [b] \equiv \emptyset \implies \exists\, a. M \equiv [a]$.*

Corollary 1. *For CRA-expressions t, s, the universal statement $\forall \boldsymbol{\alpha}. t = s$ is valid in CRAs iff it is valid in CRACs.*

We show Lem. 3, entailing any CRAC $\mathcal{C}$ has left- *and* right-*cancellation* (as $\cdot$ is commutative; Lem. 1) and the diamond property (by *push-outs*), used in Sec. 5.

Satisfying laws (1)–(4) and (7)–(9) makes a CRAC $\mathcal{C}$ a special case of a residual system with composition [36, Def. 8.7.38] having a natural order that is a partial order.[11] These have many good properties [31, 35][36, Tab. 8.5]. In particular, for any such system $\langle \rightarrow, 1, /, \cdot \rangle$, we have $\langle \rightarrow, 1, \cdot \rangle$ is a typed *monoid* (a category) that is *left-cancellative* (each χ is *epi*: for all ϕ, ψ, if $\chi \cdot \phi = \chi \cdot \psi$ then $\phi = \psi$), *gaunt* (isomorphisms are 1) and has *push-outs* (in the standard categorical sense). Calling these typed *residuation* monoids, we have [31, 35]:

Lemma 3. *$\langle \rightarrow, 1, \cdot \rangle$ is a typed residuation monoid iff $\langle \rightarrow, 1, /, \cdot \rangle$ is a residual system with composition having a natural order that is a partial order and with $\phi/\psi := \phi'$ for every peak ϕ, ψ and its push-out ψ', ϕ'.*

[11] Absent law (6), it only need be a *quasi*-order for residual systems (with composition).

5 Embedding CRACs in commutative ℓ-groups

In turn, CRACs can be embedded in *commutative ℓ-groups* (ℓ = *lattice-ordered*).

Definition 4. $\langle A, 1, {}^{-1}, \cdot, \wedge, \vee\rangle$ *is a commutative ℓ-group if* $\langle A, \wedge, \vee\rangle$ *is a* lattice *and* $\langle A, 1, {}^{-1}, \cdot\rangle$ *a* commutative group *and* $\cdot$ preserves *order*, $a \leqslant b \implies a{\cdot}c \leqslant b{\cdot}c$.

Then the lattice $\langle A, \wedge, \vee\rangle$ is *distributive*. Generalising $\mathbb{N} \hookrightarrow \mathbb{Z}$ we embed a CRAC $\mathcal{C}$ in a commutative ℓ-group $\widehat{\mathcal{C}}$ by means of pairs called *fractions* here, written $\frac{a}{b}$, we proceed in two phases. The first works for residual systems with composition:

Lemma 4. $\langle \frac{A}{A}, \frac{1}{1}, \cdot, {}^{-1}\rangle$ *is an involutive monoid if* $\frac{a}{b} \cdot \frac{c}{d} := \frac{a\cdot(c/b)}{d\cdot(b/c)}$, $(\frac{a}{b})^{-1} := \frac{b}{a}$.

Proof. Reciprocal ${}^{-1}$ is clearly an involution and anti-automorphic by $(\frac{a}{b}\cdot\frac{a'}{b'})^{-1} = \left(\frac{a\cdot(a'/b)}{b'\cdot(b/a')}\right)^{-1} = \frac{b'\cdot(b/a')}{a\cdot(a'/b)} = \frac{b'}{a'}\cdot\frac{b}{a} = (\frac{a'}{b'})^{-1}\cdot(\frac{a}{b})^{-1}$. Associativity is Fig. 5.

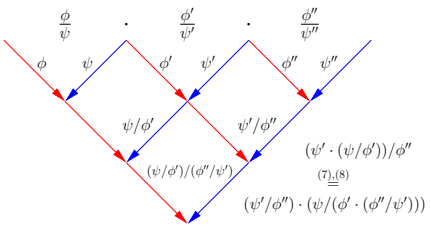

Fig. 5. Associativity of composition of fractions by *tiling*

Though this works, numerators and denominators of fractions often contain common factors that should be taken into account to obtain a group. By Ore's Theorem [9, Prop. II.3.11] the *diamond property* and *left- and right-cancellation* must hold for that. Though only the former two are guaranteed for residual systems with composition by Lem. 3,[12] for CRACs all three hold by commutativity of $\cdot$:

Theorem 4. *Any CRAC* $\mathcal{C}{=}\langle A, 1, /, \cdot\rangle$ *embeds in a commutative ℓ-group* $\widehat{\mathcal{C}}$.

Proof. As *carrier* of $\widehat{\mathcal{C}}$ we take (formal) *fractions* $\frac{a}{b}$ with $a, b \in A$ that are *normalised*: $a \wedge b = 1$. *Unit* and *reciprocal* are as for the involutive monoid, both preserve being normalised. But composition does not, so must be normalised, where the *normalisation*[13] of a fraction $\frac{a}{b}$ is $\frac{a/b}{b/a}$. Normalising the above $\frac{a\cdot(c/b)}{d\cdot(b/c)}$, assuming both $\frac{a}{b}$ and $\frac{c}{d}$ are normalised and using Prop. 1, now yields $\frac{(a/d)\cdot(c/b)}{(d/a)\cdot(b/c)}$ as definition of the *composition* $\frac{a}{b}\cdot\frac{c}{d}$. The lattice operations are *meet* $\frac{a}{b}\wedge\frac{c}{d} := \frac{a\wedge c}{b\vee d}$ and *join* $\frac{a}{b}\vee\frac{c}{d} := \frac{a\vee c}{b\wedge d}$. The *embedding* $\widehat{\ }$ of $\mathcal{C}$ in $\widehat{\mathcal{C}}$ proceeds by $a \mapsto \frac{a}{1}$ and mapping operations to 'themselves' except $a/b \mapsto (\widehat{a}\cdot(\widehat{b})^{-1})\vee 1$. This works.

[12] Right-cancellation fails for the residual system with composition for the $\lambda\beta$-calculus.
[13] This normalisation operation cannot be typed; ϕ, ψ form a valley not a peak in $\frac{\phi}{\psi}$.

Example 4. Applying this construction to $\langle \mathbb{N}, 0, \dot{-}, + \rangle$ gives the (signed) integers with the standard order on them with lattice operations minimum and maximum. $\langle \mathsf{Pos}, 1, \cdot/, \cdot \rangle$ gives (normalised) fractions $\frac{n}{m}$. For instance, $\frac{6}{5} \cdot \frac{5}{2} = \frac{3}{1}$, $\frac{6}{5} \wedge \frac{5}{2} = \frac{1}{10}$, and $\frac{6}{5} \vee \frac{5}{2} = \frac{30}{1}$. Applied to $\langle \mathsf{Mst}(A), \emptyset, -, \uplus \rangle$ we obtain *signed* multisets[14] ordered via the pointwise less–than–or–equal of integer multiplicities.

Lemma 5. *$\mathcal{C}$ embeds in the* positive cone $\widehat{\mathcal{C}}_{\geqslant 1}$ *(elements $\geqslant \frac{1}{1}$) of $\widehat{\mathcal{C}}$.*

Proof. By definition $\frac{1}{1} \leqslant \frac{a}{b}$ iff $\frac{1}{1} = \frac{1}{1} \wedge \frac{a}{b} = \frac{1 \wedge a}{1 \vee b} = \frac{1}{b}$, hence iff $1 = b$, but then the element is in the image of the embedding.

Corollary 2. *The universal statement $\forall \boldsymbol{a}.t = s$ for CRA-expressions t, s, is valid in CRACs iff $\forall \boldsymbol{\alpha} \in \mathcal{G}_{\geqslant 1}.\widehat{t} = \widehat{s}$ is valid in commutative ℓ-group $\mathcal{G}$, for $\widehat{\ }$ such that $\widehat{r/u} := (\widehat{r} \cdot (\widehat{u})^{-1}) \vee \frac{1}{1}$. This problem is decidable and in co-NP.*

Proof. By Cor. 1, $\forall \boldsymbol{\alpha}.t = s$ is valid in CRAs iff it is so in CRACs. Since $\frac{a}{b} \vee \frac{1}{1} = \frac{a}{1}$ and $\frac{a}{1} \cdot (\frac{b}{1}^{-1}) = \frac{a/b}{\cdots}$, evaluating (the $\widehat{\ }$-image of) residuation on the embedding of elements in the positive cone, is the same as the embedding of their residuation. By Lem. 5 quantifying over the elements of the CRAC, embeds as quantifying over the elements of the positive cone. It remains to show the formula is of shape $\forall \boldsymbol{\alpha}. \left(\bigwedge_{i=1}^{m} t_i = \frac{1}{1} \implies t = \frac{1}{1}\right)$ as required for the decidability / complexity result of [37]. Writing the domain-constraints and equation as $\frac{1}{1} \wedge \alpha = \frac{1}{1}$, $\frac{\widehat{t}}{\widehat{s}} = \frac{1}{1}$, it is.

6 Solutions

Solution 1 (of Problem 1). The set of operations $\{\vee, \rightarrow, \top\}$ is not functionally complete as negation can't be expressed. However, the operations do give rise to the CRA $\langle \{\top, \bot\}, 1, / \rangle$ when defining $1 := \top$ and $p/q := q \rightarrow p$; it is isomorphic to the CRA $\langle \{0, 1\}, 0, \dot{-} \rangle$. Note that $\top$ is the *least* element in the $\leqslant$-order, and that boolean *or* therefore corresponds to the *meet* $\wedge$ in the CRA. Thus the problem is naturally stated for CRAs as $(a/b) \wedge (b/a) = 1$, which is the purport of Prop. 1.

Solution 2 (of Problem 2). The problem is an instance of that $a \wedge b = 1$ entails $a \wedge (b \cdot c) = a \wedge c$ for CRACs. Setting $d := b \cdot c$. We conclude by $a \overset{\text{def},(5)}{=} (a/b) \cdot (a/(a/b)) \overset{\text{def}}{=} (a/b) \cdot (a \wedge b) \overset{\text{hyp}}{=} a/b$, hence $a \wedge d \overset{\text{def}}{=} a/(a/d) \overset{(1)}{=} a/((a/d)/1) \overset{\text{hyp}}{=} a/((a/d)/(b/d)) \overset{(4)}{=} a/((a/b)/(d/b)) = a/(a/(d/b)) \overset{\text{hyp,def}}{=} a \wedge c$. Whether this is *nice* depends on what laws one accepts, but calculational it is. Note the analysis in [10] was inconclusive, suggesting a way forward via FTA not used here.

Solution 3 (of Problem 3). We restate it for CRAs: if $a \wedge b = 1$ and $(a \cdot b)\!\downarrow$ and $d \leqslant a \cdot b$, then $(d \wedge a) \cdot (d \wedge b) \simeq d$. It is left to readers to check it can be proven.

[14] Those of [6, Sec. 7] arise by restricting to having finite support.

We devote the remainder of this section to discussing the solution of Problem 4 and its ramifications, since we think it gives interesting novel results. We are interested in both stating and proving in CRAs / CRACs / commutative ℓ-groups versions of the Inclusion–Exclusion principle (IE) for a finite family $A_I := (A_i)_{i\in I}$ of finite sets. Since CRAs don't have 'inverses', in the CRA version of IE we separate the positive (O odd-sized index-sets) from the negative (E even-sized index-sets) contributions maintaining the invariant that O is at least as large as E, we take the residuation of the former after the latter. Taking into account that composition and join are partial operations we obtain, where the o and e inscribed on the $\subseteq$ express restriction to odd- and even-sized subsets respectively:

Theorem 5 (CRA version of Inclusion–Exclusion for finite family a_I).

$$O := \left(\prod_{\emptyset\subset J\underline{\subseteq}_o I} \bigwedge a_J\right)\downarrow \text{ and } E := \left(\prod_{\emptyset\subset J\underline{\subseteq}_e I} \bigwedge a_J\right)\downarrow \Longrightarrow \bigvee a_I \simeq O/E \text{ and } E \leqslant O$$

Proof. Lem. 1 and the derived CRA laws (easily shown by ATP / Prover9):

$$\begin{array}{rll}(b/a)\wedge(c/a) = & (c/a)/(c/b) & = (b\wedge c)/(a\wedge c)\\ (a\cdot b)/(c\cdot d) = & (a/c)/(d/b) & \text{if } c\leqslant a,\ b\leqslant d \text{ and } (a\cdot b)\downarrow,\ (c\cdot d)\downarrow\\ (a\cdot b)\wedge c \simeq & (a\wedge c)\cdot(b\wedge(c/a)) & \text{if } (a\cdot b)\downarrow\end{array}$$

allow to mimic every step of the standard proof by induction on $|I|$, splitting off 1 element at the time from I, by reasoning within CRAs only.

This IE applies to all CRAs encountered, natural numbers, multisets, etc.. For instance, for $a_1 := 6$, $a_2 := 15$, and $a_3 := 10$ in $\langle\mathbb{N}, 0, \dot{-}\rangle$:

$$\max(6,15,10) = 6{+}15{+}10{+}\min(6,15,10)\dot{-}\min(6,15)\dot{-}\min(15,10)\dot{-}\min(10,6)$$

Note that it is simpler than the usual IE for sets, by doing away with the cardinalities, but that the latter can be regained from the instantiation for multisets, which we illustrate now for *measurable* multisets and sets, a novel result as far as we know. Consider the following *simple* case of the notion of algebra in measure theory (usually the stronger closure under *countable* unions is assumed).

Definition 5. *A collection of sets $\mathcal{A}$ is an* algebra *if $\mathcal{A}\subseteq\wp(A)$, $A\in\mathcal{A}$ and $\mathcal{A}$ is closed under union and complement (sub-algebra of the Boolean algebra $\wp(A)$).*

Definition 6. *A multiset M is $\mathcal{A}$-measurable if:*

- *$M^i\in\mathcal{A}$ for each i, with $M^i := \{a \mid M(a) = i\}$ (*set *at* height *i of M)*
- *$M^{>i} = \emptyset$ for some i, with $M^{>i} := \bigcup_{j>i} M^j = \{a \mid M(a) > i\}$ (with the least such i the* height *of M; so multisets are assumed bounded (Ex. 2))*

Lemma 6.
- *the sets M^i at height i* partition *A;*
- *$M^{>0}$ is the support (Ex. 2) of M (may be* in*finite!); M empty iff height 0;*
- *$\langle\mathsf{Mst}(\mathcal{A}),\emptyset,-\rangle$ of $\mathcal{A}$-measurable multisets is a CRA (is closed under $-$).*

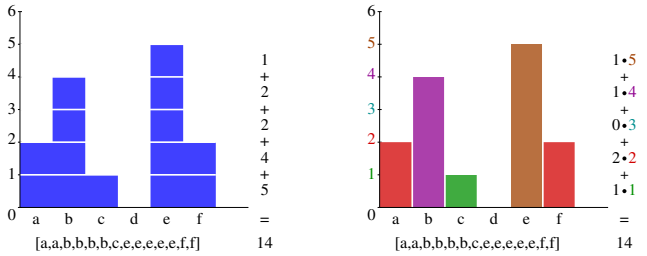

Fig. 6. Measuring horizontally / set-wise = measuring vertically / element-wise

Definition 7. *A function μ from algebra $\mathcal{A}$ to non-negative reals is a* measure *if $\mu(\emptyset) = 0$ and $\mu(A \cup B) = \mu(A) + \mu(B)$ for $A, B \in \mathcal{A}$ and disjoint. μ is extended to* measurable *multisets by $\mu(M) := \sum_i \mu(M^{>i}) = \sum_j j \cdot \mu(L^j)$ (see Fig. 6).*

Corollary 3 (IE for finite family of measurable multisets / sets).

$$\bigcup M_I = \left(\biguplus_{\emptyset \subset J \subseteq I} \bigcap M_J\right) - \left(\biguplus_{\emptyset \subset J \subseteq I} \bigcap M_J\right)$$

$$\mu\left(\bigcup A_I\right) = \left(\sum_{\emptyset \subset J \subseteq I} \mu\left(\bigcap A_J\right)\right) \dot{-} \left(\sum_{\emptyset \subset J \subseteq I} \mu\left(\bigcap A_J\right)\right)$$

Proof. For measurable multisets: *instance* of Theorem 5. For sets: *via* the multiset result, viewing sets *as multisets* and using $\mu(M \uplus N) = \mu(M) + \mu(N)$.

7 Conclusion

It should be interesting to build in support for the main result, Cor. 2, in proof assistants, in some user-friendly (qua proofs produced) way, cf. [15]. In particular, support for algebraic reasoning about multisets seems desirable.

We have extended the usual embedding $\mathbb{N} \hookrightarrow \mathbb{Z}$ by supplying the extra layer of bits, into $\mathbb{B} \hookrightarrow \mathbb{N} \hookrightarrow \mathbb{Z}$, on which we have based its generalisation CRA $\hookrightarrow$ CRAC $\hookrightarrow$ commutative ℓ-group. Having a residuation at the lower level of CRAs enabled constructing the CRAC and commutative ℓ-group at the higher level via *embeddings*.

We have identified residuation as the Skolemised diamond property at the basis of tiling, coming *prior* to other operations such as composition and inverse. Since the rewriting technique of (in)formal proving by tiling is pervasive both in the rewriting literature and in other fields such as higher categories [1] and algebra [9], we also expect residuation to become more prominent in those areas (where it is currently seen mainly as a tool, not as key notion as in [31, 35, 36]).

As an illustration of the interest of the algebras considered, CRAs, CRACs, and commutative ℓ-groups, we discussed some examples that could be both stated and proven using them (and their laws). They arguably constitute the natural level of abstraction to state and prove the Inclusion–Exclusion principle, and as a testimony to that we gave a novel instance for measurable multisets.

Acknowledgments. This work was done in small steps. Thanks to Albert Visser for early collaboration on CRAs while at Utrecht University. Research on the Inclusion–Exclusion principle was performed while at the University of Innsbruck, on decidability via the embedding while at the University of Bath (supported by EPSRC Project EP/R029121/1 Typed lambda-calculi with sharing and unsharing) and on residuation as the Skolemised diamond property while self-supported. Integrating them was performed at the University of Sussex.

Disclosure of Interests. The author has no competing interests to declare that are relevant to the content of this article.

References

1. Ara, D., Burroni, A., Guiraud, Y., Malbos, P., Métayer, F., Mimram, S.: Polygraphs: From rewriting to higher categories (2023). https://doi.org/10.48550/arXiv.2312.00429
2. Arai, Y., Iséki, K., Tanaka, S.: Characterizations of BCI, BCK-algebras. Proc. Japan Acad. **42**(2), 105–107 (1966). https://doi.org/10.3792/pja/1195522126
3. Baader, F., Nipkow, T.: Term Rewriting and All That. Cambridge University Press (1998)
4. Barendregt, H.: The Lambda Calculus: Its Syntax and Semantics, Studies in Logic and the Foundations of Mathematics, vol. 103. North-Holland, Amsterdam, 2nd revised edn. (1984)
5. Bentkamp, A., Blanchette, J., Nummelin, V., Tourret, S., Vukmirović, P., Waldmann, U.: Mechanical mathematicians. Communications of the ACM **66**(4), 80–90 (mar 2023). https://doi.org/10.1145/3557998
6. Blanchette, J., Fleury, M., Traytel, D.: Nested multisets, hereditary multisets, and syntactic ordinals in Isabelle/HOL. In: Miller, D. (ed.) 2nd International Conference on Formal Structures for Computation and Deduction, FSCD 2017, September 3-9, 2017, Oxford, UK. LIPIcs, vol. 84, pp. 11:1–11:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2017). https://doi.org/10.4230/LIPIcs.FSCD.2017.11, https://doi.org/10.4230/LIPIcs.FSCD.2017.11
7. Chen, C.H., Sabry, A.: A computational interpretation of compact closed categories: reversible programming with negative and fractional types. Proc. ACM Program. Lang. **5**(POPL) (Jan 2021). https://doi.org/10.1145/3434290
8. Church, A., Rosser, J.: Some properties of conversion. Transactions of the American Mathematical Society **39**, 472–482 (1936)
9. Dehornoy, P., alii: Foundations of Garside Theory. European Mathematical Society (2015). https://doi.org/10.4171/139
10. Dijkstra, E.: The gcd and the minimum. Tech. Rep. 1313, Department of Computer Sciences, The University of Texas at Austin (Nov 2001), https://www.cs.utexas.edu/users/EWD/transcriptions/EWD13xx/EWD1313.html
11. Dvurečenskij, A., Pulmannová, S.: BCK-algebras, pp. 293–377. Springer Netherlands, Dordrecht (2000). https://doi.org/10.1007/978-94-017-2422-7_6
12. Dvurečenskij, A., Pulmannová, S.: BCK-algebras in Applications, pp. 379–446. Springer Netherlands, Dordrecht (2000). https://doi.org/10.1007/978-94-017-2422-7_7
13. Dvurečenskij, A.: On categorical equivalences of commutative BCK-algebras (Jun 1998), preprint 16/1998

14. Dvurečenskij, A., Graziano, M.: Commutative BCK-algebras and lattice ordered groups. Mathematica japonicae **49**(2), 159–174 (Mar 1999), https://ci.nii.ac.jp/naid/10010236889/en/
15. Galatos, N., Metcalfe, G.: Proof theory for lattice-ordered groups. Annals of Pure and Applied Logic **167**(8), 707–724 (2016). https://doi.org/10.1016/j.apal.2016.04.004
16. Imai, Y., Iséki, K.: On axiom systems of propositional calculi, xiv. Proc. Japan Acad. **42**(1), 19–22 (1966). https://doi.org/10.3792/pja/1195522169
17. Iséki, K.: An algebra related with a propositional calculus. Proc. Japan Acad. **42**(1), 26–29 (1966). https://doi.org/10.3792/pja/1195522171
18. Khisamiev, N.: Universal theory of lattice-ordered abelian groups. Algebra i Logika **5**(3), 71–76 (1966)
19. Klop, J.: Combinatory Reduction Systems. Ph.D. thesis, Rijksuniversiteit Utrecht (1980)
20. Lévy, J.J.: Réductions correctes et optimales dans le λ-calcul. Thèse de doctorat d'état, Université Paris VII (1978)
21. Luttik, S., van Oostrom, V.: Decomposition orders—another generalisation of the fundamental theorem of arithmetic. Theoretical Computer Science **335**(2), 147–186 (2005). https://doi.org/https://doi.org/10.1016/j.tcs.2004.11.019
22. McCune, W.: Prover9 and mace4 (2005–2010), http://www.cs.unm.edu/ mccune/prover9, `http://www.cs.unm.edu/~mccune/prover9/`
23. Melliès, P.A.: Description Abstraite des Systèmes de Réécriture. Thèse de doctorat, Université Paris VII (Dec 1996), http://www.irif.fr/ mellies/phd-mellies.pdf
24. Melliès, P.: Axiomatic rewriting theory VI residual theory revisited. In: Tison, S. (ed.) Rewriting Techniques and Applications, 13th International Conference, RTA 2002, Copenhagen, Denmark, July 22–24, 2002, Proceedings. Lecture Notes in Computer Science, vol. 2378, pp. 24–50. Springer (2002). https://doi.org/10.1007/3-540-45610-4_4
25. Newman, M.: On theories with a combinatorial definition of "equivalence". Annals of Mathematics **43**, 223–243 (1942). https://doi.org/10.2307/2269299
26. van Oostrom, V.: Course notes on braids (1998), http://www.javakade.nl/research/pdf/braids.pdf
27. van Oostrom, V.: Some symmetries of commutation diamonds. In: 9th IWC. pp. 1–7 (2020), http://www.javakade.nl/research/talk/iwc300620.pdf
28. van Oostrom, V.: Multi-redexes and multi-treks induce residual systems; least upper bounds and left-cancellation up to homotopy. In: IWC 2021. pp. 1–7 (Jul 2021), http://www.javakade.nl/research/pdf/axrs-iwc-2021.pdf
29. van Oostrom, V.: On causal equivalence by tracing in string rewriting. In: Grabmayer, C. (ed.) Proceedings Twelfth International Workshop on Computing with Terms and Graphs, Technion, Haifa, Israel, 1st August 2022. Electronic Proceedings in Theoretical Computer Science, vol. 377, pp. 27–43. Open Publishing Association (2023). https://doi.org/10.4204/EPTCS.377.2
30. van Oostrom, V., Zantema, H.: Triangulation in rewriting. In: RTA. LIPIcs, vol. 15, pp. 240–255. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2012). https://doi.org/10.4230/LIPIcs.RTA.2012.240
31. Plotkin, G.: (1980?), handwritten unpublished notes (communicated to me by J.W. Klop in September 2022)
32. Pous, D.: Untyping typed algebraic structures and colouring proof nets of cyclic linear logic. In: Dawar, A., Veith, H. (eds.) Computer Science Logic, 24th International Workshop, CSL 2010, 19th Annual Conference of the EACSL, Brno, Czech

Republic, August 23-27, 2010. Proceedings. Lecture Notes in Computer Science, vol. 6247, pp. 484–498. Springer (2010). https://doi.org/10.1007/978-3-642-15205-4_37
33. Połacik, T., Ruitenburg, W.: Simple axioms that are obviously true in $\mathbb{N}$. Review of Modern Logic **9**(1-2), 67–79 (2003). https://doi.org/rml/1081173835
34. Schikora, R.: On Orthogonality of Self-Distributivity. Master's thesis, University of Innsbruck (2022), https://diglib.uibk.ac.at/download/pdf/8160517.pdf
35. Stark, E.: Concurrent transition systems. Theoretical Computer Science **64**, 221–269 (1989)
36. Terese: Term Rewriting Systems. Cambridge University Press (2003)
37. Weispfenning, V.: The complexity of the word problem for abelian l-groups. Theoretical Computer Science **48**, 127–132 (1986). https://doi.org/https://doi.org/10.1016/0304-3975(86)90089-7

A Selected proofs and remarks omitted from main text

Proof (of Lem. 2). Note that $b = (a \wedge b) \cdot (b/a)$, for all a, b. In particular, $b_{ij} = (a_{ij} \wedge b_{ij}) \cdot b_{(i+1)j}$, hence $b_j = (\prod c_{Ij}) \cdot b'_j$ for column j in Fig. 4.

Embedding being trivial ($a = b$ iff $a/b = 1$ and $b/a = 1$), to show downward-closedness assume $M /\!\!/ [b] \equiv \emptyset$ for some $M = [a_I]$. Setting $b = b_0$ the before gives $b_0 = (\prod c_{I0}) \cdot b'_0$ and $a_i = c_{i0}$ for each $i \in I$. Hence $b_0 = (\prod a_I) \cdot b'_0$, showing that $(\prod a_I)\!\downarrow$ from which $M \equiv [\prod a_I]$, i.e. M is indeed a singleton.

Proof (of Lem. 3).

- *Only–if-direction*: In general, push-outs are only defined up to isomorphism, but here they are unique by isos being units, making / well-defined as a function, which per construction witnesses the diamond property. As is well-known (and easy to show) push-out-diagrams compose. Here, since push-out valleys are unique, we observe that the push-out valley $(\psi \cdot \chi)/\phi, \phi/(\psi \cdot \chi)$ for the peak $\phi, \psi \cdot \chi$ is (componentwise) the same as the valley $(\psi/\phi) \cdot (\chi/(\phi/\psi)), (\phi/\psi)/\chi$ constructed from the push-out valleys $\psi/\phi, \phi/\psi$ for the peak ϕ, ψ first, and $\chi/(\phi/\psi), (\phi/\psi)/\chi$ for the peak $\phi/\psi, \chi$ second. That is, we then have $(\psi \cdot \chi)/\phi = (\psi/\phi) \cdot (\chi/(\phi/\psi))$ and $\phi/(\psi \cdot \chi) = (\phi/\psi)/\chi$. We now check the laws of residual systems with composition (1)– (4) and (7)–(9) hold:
  (1) For a peak $\phi, 1$ and valley $1, \phi$, we have $\phi \cdot 1 = 1 \cdot \phi$ by 1 being the unit. That $1, \phi$ is universal among all valleys ψ'', ϕ'' such that $\phi \cdot \psi'' = 1 \cdot \phi''$ is witnessed by the step ψ'' which is unique by 1 being the unit. This shows law (1) holds;
  (2) For a peak ϕ, ϕ and valley $1, 1$, we have $\phi \cdot 1 = \phi \cdot 1$. That $1, 1$ is universal among all valleys ψ'', ϕ'' such that $\phi \cdot \psi'' = \phi \cdot \phi''$, follows since for such valleys $\psi'' = \phi''$ by left-cancellation, hence we may take that as witness, which is unique by 1 being the unit, showing laws (2) holds;
  (3) That law (3) holds follows immediately from the reasoning in item (1);

(4) By the above observation that push-out valleys compose, we have that if for three co-initial steps ϕ, ψ, χ we consider the peaks between ϕ and the lhs respectively rhs of $\psi \cdot (\chi/\psi) = \chi \cdot (\psi/\chi)$ obtained by pushing-out ψ, χ, we obtain that $(\phi/\psi)/(\chi/\psi) = \phi/(\psi \cdot (\chi/\psi)) = \phi/(\chi \cdot (\psi/\chi)) = (\phi/\chi)/(\psi/\chi)$ showing law (4) holds;
(7) That law (7) holds follows immediately by the observation;
(8) That law (8) holds follows immediately by the observation;
(9) Law (9) is an instance of the assumption that 1 is a unit for composition.

Finally, since $\leqslant$ is a quasi-order as follows from $\langle \to, 1, / \rangle$ being a residual system, it suffices to show $\leqslant$ is anti-symmetric. But $\phi \leqslant \psi$ and $\psi \leqslant \phi$ is equivalent to saying that the push-out valley $\psi/\phi, \phi/\psi$ for the peak ϕ, ψ is the valley $1, 1$, entailing that $\phi = \phi \cdot 1 = \psi \cdot 1 = \psi$.

– *If-direction*: Since by assumption $\leqslant$ is a partial order, we may and often will show equality of steps ϕ and ψ, by proving both $\phi \leqslant \psi$ and $\psi \leqslant \phi$. We first check $\langle \to, 1, \cdot \rangle$ is a category:
  - $\phi \equiv \phi \cdot 1$ as $\phi/(\phi \cdot 1) = (\phi/\phi)/1 = 1/1 = 1$ and $(\phi \cdot 1)/\phi = (\phi/\phi) \cdot (1/(\phi/\phi)) = 1 \cdot 1 = 1$;
  - $\phi \equiv 1 \cdot \phi$ since $\phi/(1 \cdot \phi) = (\phi/1)/\phi = \phi/\phi = 1$ and $(1 \cdot \phi)/\phi = (1/\phi) \cdot (\phi/(\phi/1)) = 1 \cdot (\phi/\phi) = 1 \cdot 1 = 1$; and
  - $(\phi \cdot \psi) \cdot \chi \equiv \phi \cdot (\psi \cdot \chi)$ since $((\phi \cdot \psi) \cdot \chi)/(\phi \cdot (\psi \cdot \chi)) = (((\phi \cdot \psi) \cdot \chi)/\phi)/(\psi \cdot \chi) = (((\phi \cdot \psi)/\phi) \cdot (\chi/(\phi/(\phi \cdot \psi))))/(\psi \cdot \chi) = (((\phi/\phi) \cdot (\psi/(\phi/\phi))) \cdot (\chi/((\phi/\phi)/\psi)))/(\psi \cdot \chi) = ((1 \cdot (\psi/1)) \cdot (\chi/(1/\psi)))/(\psi \cdot \chi) = (\psi \cdot (\chi/1))/(\psi \cdot \chi) = (\psi \cdot \chi)/(\psi \cdot \chi) = 1$ and $(\phi \cdot (\psi \cdot \chi))/((\phi \cdot \psi) \cdot \chi) = ((\phi \cdot (\psi \cdot \chi))/(\phi \cdot \psi))/\chi = (((\phi \cdot (\psi \cdot \chi))/\phi)/\psi)/\chi = (((\phi/\phi) \cdot ((\psi \cdot \chi)/(\phi/\phi)))/\psi)/\chi = ((1 \cdot ((\psi \cdot \chi)/1))/\psi)/\chi = ((\psi \cdot \chi)/\psi)/\chi = ((\psi/\psi) \cdot (\chi/(\psi/\psi)))/\chi = (1 \cdot (\chi/1))/\chi = \chi/\chi = 1$, using the second item repeatedly.

Next, we check the category is left-cancellative, gaunt, and has push-outs.

1. To see that composition is left-cancellative, i.e. that every step χ is epi, suppose $\chi \cdot \phi = \chi \cdot \psi$. for some ϕ, ψ. Then $\phi \leqslant \psi$ follows from $1 \overset{(2),\text{hyp}}{=} (\chi \cdot \phi)/(\chi \cdot \psi) \overset{(7)}{=} ((\chi \cdot \phi)/\chi)/\psi \overset{(8),(2)}{=} (1 \cdot (\phi/1))/\psi \overset{(1)}{=} \phi/\psi$ also using we have a category in the last equation. Since symmetrically $\psi \leqslant \phi$, we conclude to $\phi = \psi$ as desired;
2. To prove the category is gaunt, it suffices by item 1 and Remark **??** to show $\phi \cdot \psi = 1$ implies $\phi = 1$, which follows from $\phi \overset{(1)}{=} \phi/1 \overset{\text{hyp}}{=} \phi/(\phi \cdot \psi) \overset{(7)}{=} (\phi/\phi)/\psi \overset{(2)}{=} 1/\psi \overset{(3)}{=} 1$; and
3. We claim $\psi/\phi, \phi/\psi$ is a push-out-valley for a peak ϕ, ψ. To verify that $\phi \cdot (\psi/\phi) = \psi \cdot (\phi/\psi)$ it suffices by symmetry and by $\leqslant$ being a partial order, to show the lhs to be $\leqslant$-related to the rhs. This follows from $(\phi \cdot (\psi/\phi))/(\psi \cdot (\phi/\psi)) \overset{(7)}{=} ((\phi \cdot (\psi/\phi))/\psi)/(\phi/\psi) \overset{(8)}{=} ((\phi/\psi) \cdot ((\psi/\phi)/(\psi/\phi)))/(\phi/\psi) \overset{(2)}{=} ((\phi/\psi) \cdot 1)/(\phi/\psi) \overset{(2)}{=} 1$ also using we have a category in the last equation.

   Having shown the valley completes the peak into a commuting diagram, it remains to show that it is least among such. To that end, assume to have $\phi \cdot \chi = \psi \cdot \omega$. By reasoning as above, the peaks $\chi, \psi/\phi$

and ϕ/ψ are seen to be completed into commuting diagrams by valleys $(\psi/\phi)/\chi, \chi/(\psi/\phi)$ respectively $\omega/(\phi/\psi), (\phi/\psi)/\omega$. Since $(\psi/\phi)/\chi \overset{(7)}{=} \psi/(\phi\cdot\chi) \overset{\text{hyp}}{=} \psi/(\psi\cdot\omega) \overset{(7)}{=} (\psi/\psi)/\omega \overset{(2),(3)}{=} 1$ and symmetrically $(\phi/\psi)/\omega = 1$, the commuting diagrams give $\chi = (\psi/\phi)\cdot(\chi/(\psi/\phi))$ and $(\phi/\psi)\cdot(\omega/(\phi/\psi)) = \omega$. (Such mediating steps must in fact be unique, e.g., if $\chi = (\psi/\phi)\cdot\chi'$, then $\chi' = \chi/(\psi/\phi)$ by left-cancellation in item **??**.) We conclude by computing that both mediating steps are the same: $\chi/(\psi/\phi) \overset{(7),(8),(2),(1)}{=} (\phi\cdot\chi)/(\phi\cdot(\psi/\phi)) \overset{\text{hyp}}{=} (\psi\cdot\omega)/(\psi\cdot(\phi/\psi)) \overset{(7),(8),(2),(1)}{=} \omega/(\phi/\psi)$.

Remark 7 (On Lem. 3). In [28] we gave direct constructions of residual systems with composition from the (two sets of) *axioms* on residuals put forward by Melliès in [24]. Since in [24] it was shown that his axioms gave rise to a category having push-outs and left-cancellation, Lem. 3 gives a much quicker route to the same: it suffices to note that the category constructed in [24] is gaunt, to obtain a residual system with composition having a natural order that is a partial order.

One may specialise Lem. 3 to CRACs, by untyping the notions to algebraic ones:

Definition 8. *Call a monoid* $\langle A, 1, \cdot\rangle$ *a* residuation *monoid if it*

- *is* left-cancellative*, if* $c\cdot a = c\cdot b$ *then* $a = b$*;*
- *is* invertible-free*, if* $a\cdot b = 1$ *then* $a = 1 = b$*;*
- *has lcm's (*least common multiples*), where a pair* c, d *is a cm (*common multiple*) of the pair* a, b *if* $a\cdot c = b\cdot d$*, and the cm* b', a' *is least if* $b' \leqslant c$ *holds for all cm's* c, d *of* a, b*.*

where $a \leqslant b$ *if* $a\cdot e = b$ *for some* e*.*

That b', a' is an lcm of a, b in the above also entails $a' \leqslant d$ since if $b'\cdot e = c$, then $b\cdot a'\cdot e = a\cdot b'\cdot e = a\cdot c = b\cdot d$ hence $a'\cdot e = d$ by left-cancellation. Being gaunt untypes to being invertible-free, and having push-outs to having lcm's. If the monoid is commutative, then in the above it is sufficient to have $a = 1$ for being invertible-free, to also entail $b = 1$, and being left-cancellative coincides with being right-cancellative.

Lemma 7 (Lem. 3 untyped). $\langle A, 1, \cdot\rangle$ *is a commutative residuation monoid iff* $\langle A, 1, /, \cdot\rangle$ *is a CRAC with* $a/b := a'$ *for every pair* a, b *and its least cm* b', a'*.*

Proof. Everything is an immediate consequence of Lem. 3, except that for the if-direction we need to verify that the monoid is commutative, which we already know from Lem. 1 and Rem. 4, and for the only–if-direction we need to verify that laws (5) and (6) hold. That follows by *tiling* as depicted at the bottom in Fig. 2 and by the reasoning in Rem. 3: By commutativity the cm of $a\cdot b, b\cdot a$ is $1, 1$. Hence by *tiling* we obtain both $(b/a)/b\cdot(a/(a/b))/(b/(b/a)) = 1$ and $(a/b)/a\cdot(b/(b/a))/(a/(a/b)) = 1$, as depicted in Fig. 2. From the assumption that the monoid is invertible-free, we thus get that each of $(b/a)/b$, $(a/(a/b))/(b/(b/a))$, $(b/(b/a))/(a/(a/b))$, and $(a/b)/a$ is 1 as in Rem. 3. From the first (or fourth), the

law (5) follows immediately. From the second and third we get by definition of /that the pair $1, 1$ is an lcm of the pair $b/(b/a), a/(a/b)$, from which the law (6) follows.

Proof (details (some) of Thm. 4). As before, we checked properties using Prover9, proceeding as follows.

Commutativity of composition holds exploiting the symmetry in its definition, by the same for the composition of $\mathcal{C}$.

By Lem. 4 it suffices to show that having the same normalisation $\equiv$, is a congruence to obtain an involutive monoid again. Next, one checks the inverse law $f^{-1} \cdot f \equiv 1$ holds using that all and only fractions of shape $\frac{a}{a}$ normalise to the unit, so we have a group.

For the lattice operations, note that we may work exclusively with normalised fractions since these are preserved by joins and meets, hence all sub-expressions of the lattice laws yield normalised fractions as well. Next note that these laws, commutativity, associativity, idempotence, and absorption, for fractions, follow from the same laws for their numerators and denominators separately, above.

Since composition is commutative to verify the group is $\leqslant$-ordered it suffices to show $\frac{a}{b} \cdot \frac{e}{f} \leqslant \frac{c}{d} \cdot \frac{e}{f}$ if $\frac{a}{b} \leqslant \frac{c}{d}$. This can be reduced to checking CRAC properties of the numerators and denominators separately.

Qua embeding, one computes that $\frac{a \cdot b}{1} = \frac{a}{1} \cdot \frac{b}{1}$ and that a/b embeds as $\frac{a/b}{1}$.

The positive cone of a commutative ℓ-group constitutes a CRAC, via Lem. 7.

Lemma 8. *If $\mathcal{G} := \langle A, 1, {}^{-1}, \cdot, \wedge, \vee \rangle$ is a commutative ℓ-group, then its positive cone $\mathcal{G}_{\geqslant 1} := \langle A_{\geqslant 1}, 1, \cdot \rangle$ is a commutative residuation monoid.*

Proof. First note that trivially $1 \in A_{\geqslant 1}$ and $A_{\geqslant 1}$ is closed under $\cdot$ by orderedness and 1 being the unit. Also the lattice-structure is preserved on $A_{\geqslant 1}$ since if $1 \leqslant a, b$ then $1 \leqslant a \wedge b$ and if $a, b \leqslant c$ then $1 \leqslant c$. That $\mathcal{G}_{\geqslant 1}$ is (left-)cancellative follows from $\mathcal{G}$ being cancellative, by being a group. By definition $a \leqslant b$ in the commutative residuation monoid $\mathcal{G}_{\geqslant 1}$ if $a \cdot c = b$ from some c in $A_{\geqslant 1}$, but this is the same as in A, since we may define for a, b in the positive cone such that $a \cdot c = b$ we must have $1 \leqslant c$ since $c := (b \cdot a^{-1}) \vee 1$ in $\mathcal{G}$ works: $a \cdot c = (a \cdot b \cdot a^{-1}) \vee a = b$ and c is in the positive cone. It follows from the same that $\mathcal{G}_{\geqslant 1}$ is invertible-free since if $a \cdot b = 1$ for $a, b \geq 1$ then $b = a^{-1} \vee 1 = 1$. Finally, since the natural order on $\mathcal{G}_{\geqslant 1}$ coincides with that on $\mathcal{G}$, the lcm of $a, b \geq 1$ is just $a \vee b$ in $\mathcal{G}$.

At the end of Sec. 2 it was stated that our embeddings are 'just commutative untyped' versions of known typed embeddings, and we gave two examples of the latters, of braids and of reversible rewrite systems, cf. e.g. [9] resp. [7]. We add a bit more detail to that, showing that in fact both give rise to the commutative ℓ-group $\mathbb{Z}$.

Example 5. Consider braids as at the end of Sec. 2.

Positive braids over 3 strands are presented by generators $\{a, b\}$ with equation $aba = bab$. Forcing commutativity on these $ab = ba$ makes $aba = abb$, hence in the group $a = b$ by cancellation making it collapse to positive and negative

exponents of a, isomorphic to $\mathbb{Z}$. The same reasoning pertains to any number of strands, confirming the analogy suggested for it at the end of Sec. 2.

Example 6. Consider reversible rewrite systems as at the end of Sec. 2.

Instantiating the notion of affluence of [30] for a pair of rewrite systems, by taking the rewrite system $\to$ for both components of the pair, yields $\to$ is *one-step affluent* if $\leftarrow \cdot \to \subseteq \leftarrow \cup \to$ and *affluent* if its reductions $\twoheadrightarrow$ are one-step affluent. Now $\to$ being deterministic entails one-step affluence of its reflexive closure $\to^=$, hence by [30, Lem. 2.4] affluence of $\to$. By *tiling* we even have that ${}^n\!\leftarrow \cdot \to^m \subseteq {}^{n\dot{-}m}\!\leftarrow$ if $m \leq n$ and ${}^n\!\leftarrow \cdot \to^m \subseteq \to^{m\dot{-}n}$ if $n \leq m$ (Lemma `star_step_diamond` in CompCert[15]). If $\to$ is reversible also $\leftarrow$ is deterministic (i.e. $\to$ is *co*-deterministic in the sense of [30]), so $\to^n \cdot {}^m\!\leftarrow \subseteq \to^{n\dot{-}m}$ if $m \leq n$ and $\to^n \cdot {}^m\!\leftarrow \subseteq {}^{m\dot{-}n}\!\leftarrow$ if $n \leq m$. This makes the set of expansions,reductions $\{{}^n\!\leftarrow, \to^n \mid n \in \mathbb{N}\}$ into (the carrier of) a *typed* group; untyping it gives $\mathbb{Z}$. The group is lattice-ordered in the same way $\mathbb{Z}$ is, via $\min, \max$ on exponents when thinking of ${}^n\!\leftarrow$ as $\to^{-n}$.

Remark 8. As noted in [30], each component of the graph of a codeterministic rewrite system $\to$ consists of a number (possibly 1) of trees branching off (if at all) from pairwise distinct objects lying at a (possibly empty) cycle; cf. [30, Fig. 6]. (The trees may be infinite: both infinitely branching and non-rooted trees are allowed.) Hence if $\to$ is reversible, so $\to$ is also deterministic, then components classify as being either *straight lines* (infinite or finite to either side) or *cycles*. Thinking of the latter as straight lines too via their infinite unfolding, justifies picturing computation-trees in reversible rewrite systems as straight lines all *parallel* to each other; neither forward nor backward branching; of course.

We fix the Inclusion–Exclusion principle we based ourselves on, by giving a standard version of it and standard slick / inductive proofs of it, for a finite family $A_I := (A_i)_{i\in I}$ of finite sets:

Theorem 6 (de Moivre, da Silva, Sylvester $\mathbb{C}_{17/18\text{th}}$).

$$\left|\bigcup A_I\right| = \sum\nolimits_{\emptyset\subset J\subseteq I} (-1)^{|J|\dot{-}1} \cdot \left|\bigcap A_J\right|$$

Proof (slick, of Thm. 6). Count for each individual $x \in \bigcup A_I$ depending on $\#(x) := |\{i \mid x \in A_i\}|$:

$$\begin{array}{ll} 1 = 1 & \text{if } \#(x) = 1 \\ 1 = 2 - 1 & \text{if } \#(x) = 2 \\ 1 = 3 - 3 + 1 & \text{if } \#(x) = 3 \\ 1 = \sum_{1\leq j\leq n} (-1)^{j\dot{-}1} \binom{n}{j} & \text{if } \#(x) = n \end{array}$$

by double counting: $\sum_{0\leq j\leq n} (-1)^j \binom{n}{j} \Leftarrow (1-1)^n \Rightarrow 0$ ('critical peak')

[15] https://compcert.org/doc/html/compcert.common.Determinism.html.

Proof (by induction, of Thm. 6). we only show the step case $I \cup \{k\}$ of the standard proof by induction #sets $|I|$ of Inclusion–Exclusion for finite sets

$$\begin{aligned}
\left|\bigcup A_{I\cup\{k\}}\right| &=^{\mathsf{IE}_2,\cup\mathsf{semi}\ell} && \left|\bigcup A_I\right| + |A_k| - \left|\left(\bigcup A_I\right) \cap A_k\right| \\
&=^{\cup\cap\mathsf{distr}\ell} && \left|\bigcup A_I\right| + |A_k| - \left|\bigcup_{i\in I}(A_i \cap A_k)\right| \\
&=^{2\times\mathsf{IH}} && \left(\sum_{\emptyset\subset J\subseteq I} (-1)^{|J|\dot{-}1} \cdot \left|\bigcap A_J\right|\right) + |A_k| - \\
& && \sum_{\emptyset\subset J\subseteq I} (-1)^{|J|\dot{-}1} \cdot \left|\bigcap_{j\in J}(A_j \cap A_k)\right| \\
&=^{\cap\mathsf{semi}\ell,\mathsf{cgroup}} && \left(\sum_{\emptyset\subset J\subseteq I} (-1)^{|J|\dot{-}1} \cdot \left|\bigcap A_J\right|\right) + |A_k| + \\
& && \sum_{\{k\}\subset J\subseteq I\cup\{k\}} (-1)^{|J|\dot{-}1} \cdot \left|\bigcap A_J\right| \\
&=^{\mathsf{cgroup}} && \sum_{\emptyset\subset J\subseteq I\cup\{k\}} (-1)^{|J|\dot{-}1} \cdot \left|\bigcap A_J\right|
\end{aligned}$$

Proof (of Cor. 3 for sets via multiset result). We use $\mu(M \uplus N) = \mu(M) + \mu(N)$ and $\mathrm{O} \ni \mathrm{E}$ and

$$\begin{aligned}
\mu(\bigcup A_I) &= \mu(\left(\biguplus_{\emptyset\subset J\subseteq I} \bigcap A_J\right) - \left(\biguplus_{\emptyset\subset J\subseteq I} \bigcap A_J\right)) \\
&= \left(\sum_{\emptyset\subseteq J\subseteq I} \mu(\bigcap A_J)\right) \dot{-} \left(\sum_{\emptyset\subset J\subseteq I} \mu(\bigcap A_J)\right)
\end{aligned}$$

where the first equality is by viewing the measured set in the lhs as a multiset, which we replace by the IE for measurable multisets, after which we can distribute the measure over the multiset sum using $\mu(M \uplus N) = \mu(M) + \mu(N)$, to yield the desired result noting that the intersections inside the measures are sets.

Lemma 9. *For μ a measure and multisets M, N, $\mu(M \uplus N) = \mu(M) + \mu(M)$.*

Proof. Based on that $\sum_i \mu(M^{>i}) = \sum_j j \cdot \mu(L^j)$, see Fig. 6, we conclude by

$$\mu(M \uplus N) = \sum_{j,k} (j+k) \cdot \mu(M^j \cap N^k) = \mu(M) + \mu(M)$$

Remark 9. It is easy to state and prove a version of IE for commutative ℓ-groups.

Theorem 7 (Ordered ℓ-group version of Inclusion–Exclusion for finite family a_I). *Let $\mathcal{G} := \langle A, 1, {}^{-1}, \cdot, \wedge, \vee\rangle$ be a commutative ℓ-ordered group. Then*

$$\bigvee a_I = \prod_{\emptyset\subset J\subseteq I} (\bigwedge a_J)^{(-1)^{|J|\dot{-}1}}$$

B Selected Prover9 proofs

In this appendix we provide a few Prover9 [22] proofs of results from the main text, indicative of how we proceeded.[16] All our Prover 9 proofs were generated

[16] To be precise, we used Prover9 version LADR-2009-11A compiled and run on a 2018 MacBook Pro with macOS Catalina 10.15.4 with a 2.2 GHz 6-core Intel Core i7

without further guidance. The proofs provided here should allow interested readers to reconstruct the other proofs omitted from the main text by means of ATP themselves. To that end, we provide the input-file used as an example for the first, trivial, proposition below. For the two others, similar representations of the statements were used, and only the resulting proofs are given. In each case the initial part of the output allows to reconstruct (the assumptions used of) the input. To keep proofs, relatively, short and proving fast we typically added already derived (useful) equations to the assumptions.

To illustrate the Prover9 input and output we make use the following proposition that was omitted from the main text, but has a short and easy to understand proof.

Proposition 2. $\leqslant$ *is transitive in BCI algebras.*

Proof. To prove the statement we supplied Prover9 a file with contents:

```
formulas(sos).

((x / y) / (x / z)) / (z / y) = 1.
(x / (x / y)) / y = 1.
x / x = 1.
-(x / y = 1) | -(y / x = 1) | x = y.
-(x / 1 = 1) | x = 1.
-P(x,y) | x / y = 1.
-(x / y = 1) | P(x,y).

end_of_list.

formulas(goals).

-P(x,y) | -P(y,z) | P(x,z).

end_of_list.
```

upon which Prover9 provided the following proof:[17]

```
============================== PROOF =================================

% Proof 1 at 0.01 (+ 0.00) seconds.
% Length of proof is 22.
% Level of proof is 6.
% Maximum clause weight is 13.000.
% Given clauses 32.

1 -P(x,y) | -P(y,z) | P(x,z) # label(non_clause) # label(goal).  [goal].
2 ((x / y) / (x / z)) / (z / y) = 1.  [assumption].
3 (x / (x / y)) / y = 1.  [assumption].
4 x / x = 1.  [assumption].
5 x / y != 1 | y / x != 1 | x = y.  [assumption].
6 x / 1 != 1 | x = 1.  [assumption].
7 x / 1 != 1 | 1 = x.  [copy(6),flip(b)].
8 -P(x,y) | x / y = 1.  [assumption].
9 x / y != 1 | P(x,y).  [assumption].
10 P(c1,c2).  [deny(1)].
11 P(c2,c3).  [deny(1)].
12 -P(c1,c3).  [deny(1)].
24 (x / 1) / x = 1.  [para(4(a,1),3(a,1,1,2))].
27 x / (x / 1) = 1.  [hyper(7,a,3,a),flip(a)].
```

processor and 32GB of memory (but Prover9 only used 1 core and memory was not an issue).

[17] The main operations applied in the proofs here are paramodulation, hyperresolution, and rewriting. See the literature on Prover9 for more on these. Positions in expressions are represented as lists of positive natural numbers; as equality (=) is taken as a binary function symbol, positions in paramodulation of two equations start with 1 (usually; the lhs) or 2 (the rhs). E.g., in this proof the identity $(x/1)/x = 1$ on the line numbered 24 is obtained by unifying the lhs of that at line numbered 4 with the subterm at position 1.2, i.e. the subterm x/y, in the lhs of the identity at line numbered 3.

```
31 c1 / c2 = 1.  [hyper(8,a,10,a)].
32 c2 / c3 = 1.  [hyper(8,a,11,a)].
33 c1 / c3 != 1.  [ur(9,b,12,a)].
71 ((x / c3) / (x / c2)) / 1 = 1.  [para(32(a,1),2(a,1,2))].
82 x / 1 = x.  [para(24(a,1),5(a,1)),rewrite([27(6)]),xx(a),xx(b)].
85 (x / c3) / (x / c2) = 1.  [back_rewrite(71),rewrite([82(7)])].
173 c1 / c3 = 1.  [para(31(a,1),85(a,1,2)),rewrite([82(5)])].
174 $F.  [resolve(173,a,33,a)].

============================== end of proof ==========================
```

Proof (of last item of Lem. 1). Meet distributes over join in CRAs.

```
============================== PROOF =================================

% -------- Comments from original proof --------
% Proof 1 at 0.09 (+ 0.00) seconds.
% Length of proof is 38.
% Level of proof is 7.
% Maximum clause weight is 27.
% Given clauses 43.

1 x ^ (y v z) = (x ^ y) v (x ^ z) # label(non_clause) # label(goal).  [goal].
2 x / 1 = x.  [assumption].
4 x / x = 1.  [assumption].
5 (x / y) / (z / y) = (x / z) / (y / z).  [assumption].
6 (x / y) / x = 1.  [assumption].
7 x ^ y = x / (x / y).  [assumption].
9 x ^ y = y ^ x.  [assumption].
10 x / (x / y) = y / (y / x).  [copy(9),rewrite([7(1),7(3)])].
13 (x ^ y) / z = (x / z) ^ (y / z).  [assumption].
14 (x / (x / y)) / z = (x / z) / ((x / z) / (y / z)).  [copy(13),rewrite([7(1),7(6)])].
15 x v y = x * (y / x).  [assumption].
16 x v x = x.  [assumption].
17 x * 1 = x.  [copy(16),rewrite([15(1),4(1)])].
18 x v y = y v x.  [assumption].
19 x * (y / x) = y * (x / y).  [copy(18),rewrite([15(1),15(3)])].
22 (x * y) / z = (x / z) * (y / (z / x)).  [assumption].
23 (x / y) * (z / (y / x)) = (x * z) / y.  [copy(22),flip(a)].
24 x / (y * z) = (x / y) / z.  [assumption].
25 (x / y) / z = x / (y * z).  [copy(24),flip(a)].
26 (x / y) / z = (x / z) / y.  [assumption].
27 (c1 ^ c2) v (c1 ^ c3) != c1 ^ (c2 v c3).  [deny(1)].
28 (c1 / (c1 / c2)) * ((c1 / (c1 / c3)) / (c1 / (c1 / c2))) != c1 / ((c1 / c2) / (c3 / c2)).  [copy(27),rewrite([7(3),7(8),15(11),15(21),7(24),25(25,R)])].
32 (x / (y / z)) / (y / (y / z)) = x / y.  [para(6(a,1),5(a,1,2)),rewrite([2(3)]),flip(a)].
33 ((x / y) / (z / y)) / (x / z) = 1.  [para(5(a,1),6(a,1,1))].
34 (x / y) / ((x / z) / (y / z)) = (x / y) / ((x / y) / (z / y)).  [para(5(a,1),7(a,2,2)),rewrite([7(3)]),flip(a)].
37 (x / (y / z)) / (z / (z / y)) = x / y.  [para(10(a,1),5(a,1,2)),rewrite([6(8),2(8)])].
39 x / (x / (x / y)) = x / y.  [para(6(a,1),10(a,1,2)),rewrite([2(3)]),flip(a)].
93 (x * y) / y = x.  [para(10(a,1),15(a,2,2)),rewrite([15(2),19(4),6(2),17(2),23(4)]),flip(a)].
121 (x / (x / y)) * z = (x * z) / (x / y).  [para(6(a,1),23(a,1,2,2)),rewrite([2(4)])].
136 (c1 * ((c1 / (c1 / c3)) / (c1 / (c1 / c2)))) / (c1 / c2) != c1 / ((c1 / c2) / (c3 / c2)).  [back_rewrite(28),rewrite([121(17)])].
203 (x / y) / ((x / y) / (z / y)) = (x / y) / (x / z).  [para(26(a,1),14(a,1)),flip(a)].
247 (x / y) / ((x / z) / (y / z)) = (x / y) / (x / z).  [back_rewrite(34),rewrite([203(10)])].
352 (c1 * ((c1 / c2) / (c1 / c3))) / (c1 / c2) != c1 / ((c1 / c2) / (c3 / c2)).  [para(26(a,1),136(a,1,1,2)),rewrite([39(8)])].
353 (c1 * ((c1 / c2) / (c1 / c3))) / (c1 / c2) != c1 / ((c1 / c3) / (c2 / c3)).  [para(5(a,1),352(a,2,2))].
661 (x * (y / z)) / y = x / (y / (y / z)).  [para(93(a,1),32(a,1,1)),flip(a)].
675 c1 / ((c1 / c3) / (c2 / c3)) != c1 / ((c1 / c2) / ((c1 / c2) / (c1 / c3))).  [back_rewrite(353),rewrite([661(13)]),flip(a)].
1150 x / ((y / z) / ((y / z) / (y / u))) = x / ((y / u) / (z / u)).  [para(33(a,1),37(a,1,1,2)),rewrite([2(2),247(6)])].
1151 $F.  [resolve(1150,a,675,a(flip))].

============================== end of proof ==========================
```

Proof (that the second law of Prop. 1 holds for cBCKrc.). This took Prover9 a bit more than one and a half hour to conclude:

```
============================== PROOF =================================

% Proof 1 at 5810.83 (+ 33.71) seconds.
% Length of proof is 43.
% Level of proof is 10.
% Maximum clause weight is 36.000.
% Given clauses 2350.

1 (x / y) / (y / x) = x / y # label(non_clause) # label(goal).  [goal].
2 x / x = 1.  [assumption].
3 1 / x = 1.  [assumption].
4 x ^ y = x / (x / y).  [assumption].
5 x ^ y = y ^ x.  [assumption].
6 x / (x / y) = y / (y / x).  [copy(5),rewrite([4(1),4(3)])].
7 (x / y) / z = (x / z) / y.  [assumption].
8 x / y != 1 | x / z != 1 | y / x != z / x | y = z.  [assumption].
9 x / 1 = x.  [assumption].
10 (c1 / c2) / (c2 / c1) != c1 / c2.  [deny(1)].
11 x / (y / (y / x)) = x / (x / (x / y)).  [para(6(a,1),4(a,2,2)),rewrite([4(2)]),flip(a)].
12 (x / y) / x = 1.  [para(2(a,1),7(a,1,1)),rewrite([3(2)]),flip(a)].
14 (x / y) / ((x / z) / y) = z / (z / (x / y)).  [para(7(a,1),6(a,1,2))].
15 (x / (x / y)) / z = (y / z) / (y / x).  [para(6(a,1),7(a,1,1))].
```

```
16 ((x / y) / z) / u = ((x / u) / y) / z.  [para(7(a,1),7(a,1,1)),flip(a)].
24 x / (y / z) != 1 | x / u != 1 | (y / x) / z != u / x | y / z = u.  [para(7(a,1),8(c,1))].
28 x / (y / (y / x)) = x / y.  [para(11(a,2),6(a,1)),rewrite([12(6),9(6)])].
32 x / (x / (x / (y / z))) = x / (y / z).  [para(7(a,1),11(a,1,2)),rewrite([7(4),28(5)]),flip(a)].
40 x / (x / (x / y)) = x / y.  [para(11(a,1),11(a,2,2,2)),rewrite([7(5),2(5),9(4),28(3),32(5)]),flip(a)].
46 ((x / y) / z) / x = 1.  [para(12(a,1),7(a,1,1)),rewrite([3(2)]),flip(a)].
47 ((x / y) / z) / (x / z) = 1.  [para(7(a,1),12(a,1,1))].
52 x / (x / ((x / y) / z)) = (x / y) / z.  [para(46(a,1),6(a,1,2)),rewrite([9(4)]),flip(a)].
53 ((x / (x / y)) / z) / y = 1.  [para(6(a,1),46(a,1,1,1))].
54 (x / y) / (x / (y / z)) = 1.  [para(6(a,1),46(a,1,1)),rewrite([7(4)])].
90 ((x / (x / y)) / z) / (y / z) = 1.  [para(6(a,1),47(a,1,1,1))].
113 (x / y) / (x / (z / (z / y))) = 1.  [para(6(a,1),53(a,1,1)),rewrite([7(5)])].
124 (x / (x / y)) / (y / ((y / x) / z)) = 1.  [para(6(a,1),54(a,1,1))].
213 (x / (x / (y / z))) / (y / (z / u)) = 1.  [para(54(a,1),15(a,2,1)),rewrite([3(10)])].
531 ((x / (x / (y / z))) / u) / ((y / u) / z) = 1.  [para(7(a,1),90(a,1,2))].
551 ((x / (x / y)) / (y / z)) / (z / (y / x)) = 1.  [para(15(a,2),90(a,1,1))].
557 ((x / y) / (z / y)) / (x / z) = 1.  [para(90(a,1),16(a,2))].
583 (x / (x / (y / z))) / (y / (u / (u / z))) = 1.  [para(113(a,1),15(a,2,1)),rewrite([3(11)])].
2967 x / (y / ((y / x) / z)) != 1 | x / u != 1 | z / (z / (y / x)) != u / x | y / ((y / x) / z) = u.  [para(6(a,1),24(c,1))].
3237 x / (y / ((y / x) / z)) = x / y.  [para(124(a,1),14(a,1,2)),rewrite([9(6),54(11),9(7)])].
3280 x / y != 1 | x / z != 1 | u / (u / (y / x)) != z / x | y / ((y / x) / u) = z.  [back_rewrite(2967),rewrite([3237(4)])].
5206 ((x / y) / (z / y)) / (x / (z / u)) = 1.  [para(557(a,1),213(a,1,1,2)),rewrite([9(5)])].
21101 (x / (y / (z / u))) / (x / ((y / w) / (z / w))) = 1.  [para(5206(a,1),113(a,1,2,2,2)),rewrite([9(8)])].
27486 (x / (y / z)) / (x / ((u / (u / y)) / (z / (y / u)))) = 1.  [para(551(a,1),531(a,1,2)),rewrite([7(9),9(11)])].
28777 (x / (x / (y / (z / u)))) / (y / ((z / w) / (u / w))) = 1.  [para(557(a,1),583(a,1,2,2,2)),rewrite([9(9)])].
75243 (x / (x / ((y / (y / z)) / (u / z)))) / (z / (u / y)) = 1.  [para(28777(a,1),27486(a,1,2)),rewrite([9(11)])].
81865 x / ((x / y) / (y / x)) = y / ((y / x) / (x / y). [hyper(3280,a,75243,a,b,21101,a,c,7,a),rewrite([2(1),9(2),2(1),9(2),
  2(1),9(2),2(1),9(2),28(3),2(2),9(3),2(2),9(3),40(4),2(5),9(6),2(5),9(6),40(7),2(6),9(7),2(6),9(7),2(6),9(7),28(8)])].
81872 (x / y) / (y / x) = x / y.  [para(81865(a,1),4(a,2,2)),rewrite([4(4),52(5),3237(8)])].
81873 $F.  [resolve(81872,a,10,a)].

============================ end of proof ==========================
```