Untyping Typed Algebraic Structures and Colouring Proof Nets of Cyclic Linear Logic

Damien Pous

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Untyping Typed Algebraic Structures and Colouring Proof Nets of Cyclic Linear Logic

Damien Pous

N° 7176

Janvier 2009

Domaine 1

ISSN 0249-6399 ISRN INRIA/RR--7176--FR+ENG

INSTITUT NATIONAL
DE RECHERCHE
EN INFORMATIQUE
ET EN AUTOMATIQUE

INRIA

centre de recherche
GRENOBLE - RHÔNE-ALPES

Untyping Typed Algebraic Structures and Colouring Proof Nets of Cyclic Linear Logic

Damien Pous * †

Domaine : Mathématiques appliquées, calcul et simulation
Équipe-Projet SARDES

Rapport de recherche n° 7176 — Janvier 2009 — 12 pages

Abstract: Algebraic structures sometimes need to be typed. For example, matrices over real numbers form a ring, but multiplication is a only a partial operation: dimensions have to agree. Therefore, a natural way to look at the ring of matrices algebraically is to consider "typed rings". We prove several "untyping" theorems: in some algebras (semirings, Kleene algebras, residuated lattices, involutive residuated lattices), typed equations can be derived from the underlying untyped equations. As a consequence, the corresponding untyped decision procedures can be extended for free to the typed settings. Some of these theorems are obtained via a detour through fragments of cyclic linear logic.

Key-words: residuated lattices, cyclic linear logic, kleene algebra, decision procedures, sequent calculus, typed algebra, partial algebra.

* CNRS, UMR 5217 (Laboratoire d'Informatique de Grenoble)
† Work partially funded by the French ANR projet blanc "Curry-Howard pour la Concurrence" CHOCO ANR-07-BLAN-0324

Centre de recherche INRIA Grenoble – Rhône-Alpes
655, avenue de l'Europe, 38334 Montbonnot Saint Ismier
Téléphone : +33 4 76 61 52 00 — Télécopie +33 4 76 61 52 52

Détyper des structures algébrique typées et colorier la logique linéaire cyclique

Résumé : Les structures algébriques doivent parfois être typées. Par exemple, les matrices sur des nombres réels forment un anneau, mais la multiplication est seulement une opération partielle: les dimensions doivent concorder. Une façon naturelle de considérer les matrices de façon algébrique consiste à travailler avec des "anneaux typés". Nous prouvons plusieurs théorèmes de "détypage": dans certaines algèbres (semi-anneaux, algèbres de Kleene, treillis résidués, treillis résidués involutifs), les équations typées peuvent être dérivées à partir des équations non typées sous-jacentes. Les procédures de décision non typées correspondantes peuvent ainsi être étendues directement aux structures typées. Certains de ces théorèmes sont obtenus par un détour à propos de fragments de la logique linéaire cyclique.

Mots-clés : treillis résidués, logique linéaire cyclique, algèbres de Kleene, procédures de décision, calcul de séquents, algèbres typées, algèbres partielles.

Untyping Typed Algebraic Structures and Colouring Proof Nets of Cyclic Linear Logic

Damien Pous
CNRS (LIG, UMR 5217)
Damien.Pous@inria.fr

lgebraic structures sometimes need to be typed. For example, matrices over real numbers form a ring, but multiplication is a only a partial operation: dimensions have to agree. Therefore, a natural way to look at the ring of matrices algebraically is to consider "typed rings". We prove several "untyping" theorems: in some algebras (semirings, Kleene algebras, residuated lattices, involutive residuated lattices), typed equations can be derived from the underlying untyped equations. As a consequence, the corresponding untyped decision procedures can be extended for free to the typed settings. Some of these theorems are obtained via a detour through fragments of cyclic linear logic.

1 Introduction

Motivations. Algebra can be quite useful to design decision procedures in interactive theorem provers. For example, the *ring* tactic [10] of the Coq proof assistant allows one to automatically prove equations like

$$x \cdot (y + f(y)) \cdot z \;=\; x \cdot y \cdot z + x \cdot f(y) \cdot z \qquad (1)$$

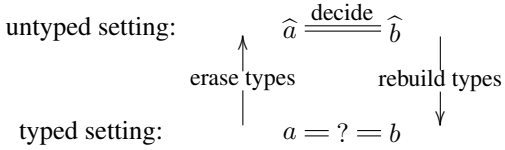
provided that operations $\cdot$ and $+$ have been declared to form a semiring. The underlying mechanism consists in moving to the initial (syntactic) model of semirings, proving the equality in that particular model, and using the initiality property to come back to the concrete model.

A limitation of this approach is that it does not directly scales to "typed" structures like matrices or heterogeneous binary relations. Considering equation (1), it might be the case that x, y, $f(y)$, z are rectangular matrices whose dimensions make the equation meaningful, or binary relations whose domain and co-domain agree. The equation holds in both cases: although they are partial operations, matrix multiplication and addition (or relation composition and union) satisfy semiring laws. Therefore, we should be able to extend the above mechanism to decide such equations.

†Work partially funded by the French ANR projet blanc "Curry-Howard pour la Concurrence" CHOCO ANR-07-BLAN-0324

A possibility would be to consider "typed initial models", and to extend standard decision procedures to work on these annotated syntactic models. However, while this seems fairly feasible for simple theories like rings, it happens to be untractable for more complex decision procedures (e.g., deciding the equational theory of Kleene algebras requires one to implement and prove a lot of finite automata algorithms; one does not want to introduce a new layer of complexity by keeping track of type annotations).

We propose another approach, which is depicted below: we prove "untyping" theorems that allow one to erase type informations, prove the equation using standard, untyped, decision procedures, and then derive a typed proof from the untyped one.



An important motivation behind this work actually comes from our Coq library ATBR [2], whose aim is to provide algebraic tools for working with binary relations. In particular, we developed efficient tactics for proving theorems of the underlying decidable partial axiomatisations (among which, semirings, residuated lattices, and Kleene algebras). The ideas presented in this paper were proved and integrated in this library, so as to extend our tactics to heterogeneous structures, for free.

Overview. We shall mainly focus on two algebraic structures, which raise different problems and illustrate several aspects of these untyping theorems: Kleene algebras [14] and residuated lattices [13].

The case of Kleene algebras is the simplest; the main difficulty comes from the annihilating element (0), whose really lose typing rule requires us to show that equational proofs in semirings can be factorised so as to use the annihilation laws at first, and then reason using the other axioms.

The case of residuated structures is more involved: due to the particular form of axioms about residuals, we cannot rely on standard equational axiomatisations of these structures. Instead, we need to exploit an equivalent cut-free sequent proof system (first proposed by Ono and Komori [20]), and to notice that this proof system corresponds to the intuitionistic fragment of cyclic linear logic [26]. The latter logic is much more concise and the corresponding proof nets are easier to reason about, so that we can obtain the untyping theorem in this setting, and port it back to residuated lattices by standard means.

The above sequent proof systems have the sub-formula property, so that they yield decision procedures, using a simple proof search algorithm. As an unexpected application, we show that the untyping theorem makes it possible to improve these algorithms by reducing the set of proofs that have to be explored.

Outline. We introduce our notations and make our notion of typed structure precise in Section 2. We study the cases of Kleene algebras and residuated lattices in Sections 3 and 4, respectively. We conclude with applications, related work, and directions for future work in Section 5.

2 Notations, typed structures

All our arguments being proof-theoretic, we shall work with syntactic models, by implicitly relying on initiality arguments to reach other models. Since we are mainly interested in structures equipped with a *composition* operation (relational composition or matrix multiplication), our "types" basically record the domain and co-domain informations: using category theory terminology, they delimit homsets.

Let $\mathcal{X}$ be an arbitrary set of *variables*, ranged over using letters x, y. Given a signature Σ, we let a, b, c range over the set $T(\Sigma + \mathcal{X})$ of *terms with variables*. Given a set $\mathcal{T}$ of *objects* (ranged over using letters n, m, p, q), a *type* is a pair (n, m) of objects (which we denote by $n \to m$, following categorical notations), a *type environment* $\Gamma : \mathcal{X} \to \mathcal{T}^2$ is a function from variables to types, and we will define *type judgements* of the form $\Gamma \vdash a : n \to m$, to be read "in environment Γ, term a has type $n \to m$, or, equivalently, a is a morphism from n to m". By $\Gamma \vdash a, b : n \to m$, we mean that both a and b have type $n \to m$; type judgements will include the following rule for variables:

$$\frac{\Gamma(x) = (n, m)}{\Gamma \vdash x : n \to m}\ \textsc{Tv}$$

Similarly, we will define *typed equalities* using judgements of the form $\Gamma \vdash a = b : n \to m$: "in environment Γ, terms a and b are equal, at type $n \to m$". Equality judgements will generally include the following rules, so as to obtain an equivalence relation at each type:

$$\frac{\Gamma(x) = (n, m)}{\Gamma \vdash x = x : n \to m}\ \textsc{v} \qquad \frac{\Gamma \vdash a = b : n \to m \quad \Gamma \vdash b = c : n \to m}{\Gamma \vdash a = c : n \to m}\ \textsc{t}$$

$$\frac{\Gamma \vdash a = b : n \to m}{\Gamma \vdash b = a : n \to m}\ \textsc{s}$$

Informally, a model is a multigraph with $\mathcal{T}$ as set of vertices, equipped with a set of operations on edges corresponding to the signature and satisfying both the typing and the equality inference rules.

By taking the singleton set as set of objects ($\mathcal{T} = \{\emptyset\}$), we recover standard, untyped structures: the only typing environment is $\widehat{\emptyset} : x \mapsto (\emptyset, \emptyset)$, and types become uninformative (this corresponds to working in a one-object category; all operations are total functions). To alleviate notations, since the typing environment will always be either $\widehat{\emptyset}$ or a an abstract constant value Γ, we shall leave it implicit in type and equality judgements, by relying on the absence or presence of types to indicate which one to use. For example, we shall write $\vdash a = b : n \to m$ for $\Gamma \vdash a = b : n \to m$, while $\vdash a = b$ will denote the judgement $\widehat{\emptyset} \vdash a = b : \emptyset \to \emptyset$.

The question we study in this paper is the following one: given a signature and a set of inference rules defining a type judgement and an equality judgement, does the implication below holds, for all a, b, n, m such that $\vdash a, b : n \to m$?

$$\vdash a = b \quad \textit{entails} \quad \vdash a = b : n \to m\ .$$

In other words, in order to prove an equality in a typed model, is it safe to remove all type annotations, so as to work in the untyped underlying free structure?

3 Kleene algebras

We study the case of residuated lattices in Section 4; here we focus on Kleene algebras. In order to illustrate our methodology, we actually give the proof in three steps, by considering two intermediate algebraic structures: monoids and (non-commutative) semirings. The former admit a rather simple and direct proof, while the latter are sufficient to expose concisely the main difficulty in handling Kleene algebras.

3.1 Monoids

Definition 1. *Typed monoids* are defined by the signature $\{\cdot_2, 1_0\}$, together with the following type and equality in-

ference rules (in addition to the rules from Section 2).

$$\frac{}{\vdash 1 : n \to n}\ \textsc{To} \qquad \frac{\vdash a : n \to m \qquad \vdash b : m \to p}{\vdash a \cdot b : n \to p}\ \textsc{Td}$$

$$\frac{}{\vdash 1 = 1 : n \to n}\ \textsc{o}$$

$$\frac{\vdash a = a' : n \to m \qquad \vdash b = b' : m \to p}{\vdash a \cdot b = a' \cdot b' : n \to p}\ \textsc{d}$$

$$\frac{\vdash a : n \to m}{\vdash 1 \cdot a = a : n \to m}\ \textsc{od} \qquad \frac{\vdash a : n \to m}{\vdash a \cdot 1 = a : n \to m}\ \textsc{do}$$

$$\frac{\vdash a : n \to m \qquad \vdash b : m \to p \qquad \vdash c : p \to q}{\vdash (a \cdot b) \cdot c = a \cdot (b \cdot c) : n \to q}\ \textsc{da}$$

In other words, models of typed monoids are just categories: typing rules (To) and (Td) assert that 1 and $\cdot$ correspond to identities and composition; the last three rules assert that identities are left and right units, and that composition is associative. Rules (o) and (d) ensure that equality is reflexive (point (i) below) and preserved by composition. As expected, equality relates correctly typed terms only (ii):

Lemma 2.

(i) For all a, n, m such that $\vdash a : n \to m$, we have $\vdash a = a : n \to m$.

(ii) For all a, b, n, m such that $\vdash a = b : n \to m$, we have $\vdash a, b : n \to m$.

Moreover, in this setting, type judgements enjoy some form of injectivity (types are not uniquely determined due to the unit element, which is typed in a polymorphic way).

Lemma 3. *Let a, n, m, n', m' such that $\vdash a : n \to m$ and $\vdash a : n' \to m'$; we have $n = n'$ iff $m = m'$.*

We need another lemma to obtain the untyping theorem: all terms related by the untyped equality admit the same types.

Lemma 4. *Let a, b such that $\vdash a = b$; for all n, m, we have $\vdash a : n \to m$ iff $\vdash b : n \to m$.*

Theorem 5. *Let a, b, n, m such that $\vdash a, b : n \to m$. If $\vdash a = b$ then $\vdash a = b : n \to m$.*

Proof. We reason by induction on the derivation $\vdash a = b$; the interesting cases are the following ones:

- the last rule used is the transitivity rule (t): we have $\vdash a = b$, $\vdash b = c$, $\vdash a, c : n \to m$, and we need to show that $\vdash a = c : n \to m$. By Lemma 4, we have $\vdash b : n \to m$, so that, by the induction hypotheses, we get $\vdash a = b : n \to m$ and $\vdash b = c : n \to m$, which allow us to apply rule (t) in the typed setting.
- the last rule used is the compatibility of $\cdot$ (d): we have $\vdash a = a'$, $\vdash b = b'$, $\vdash a \cdot b, a' \cdot b' : n \to m$, and we need to show that $\vdash a \cdot b = a' \cdot b' : n \to m$. By case analysis on the typing judgements, we deduce that $\vdash a : n \to p$, $\vdash b : p \to m$, $\vdash a' : n \to q$, $\vdash b' : q \to m$, for some p, q. Thanks to Lemmas 3 and 4, we have $p = q$, so that we can conclude using the induction hypotheses ($\vdash a = a' : n \to p$ and $\vdash b = b' : p \to m$), and rule (d). ■

Note that the converse of Theorem 5 ($\vdash a = b : n \to m$ entails $\vdash a = b$) is straightforward, so that we actually have an equivalence.

3.2 Non-commutative semirings

Definition 6. *(Non-commutative) typed semirings* are defined by the signature $\{\cdot_2, +_2, 1_0, 0_0\}$, together with the following type and equality inference rules (in addition to the rules from Definition 1 and Section 2).

$$\frac{}{\vdash 0 : n \to m}\ \textsc{Tz} \qquad \frac{\vdash a, b : n \to m}{\vdash a + b : n \to m}\ \textsc{Tp}$$

$$\frac{}{\vdash 0 = 0 : n \to m}\ \textsc{z}$$

$$\frac{\vdash a = a' : n \to m \qquad \vdash b = b' : n \to m}{\vdash a + b = a' + b' : n \to m}\ \textsc{p}$$

$$\frac{\vdash a : n \to m}{\vdash a + 0 = a : n \to m}\ \textsc{pz} \qquad \frac{\vdash a, b : n \to m}{\vdash a + b = b + a : n \to m}\ \textsc{pc}$$

$$\frac{\vdash a, b, c : n \to m}{\vdash (a + b) + c = a + (b + c) : n \to m}\ \textsc{pa}$$

$$\frac{\vdash a : n \to m \qquad \vdash b, c : m \to p}{\vdash a \cdot (b + c) = a \cdot b + a \cdot c : n \to p}\ \textsc{dp}$$

$$\frac{\vdash a : n \to m}{\vdash a \cdot 0 = 0 : n \to p}\ \textsc{dz}$$

$$\frac{\vdash a : n \to m \qquad \vdash b, c : p \to n}{\vdash (b + c) \cdot a = b \cdot a + c \cdot a : p \to m}\ \textsc{pd}$$

$$\frac{\vdash a : n \to m}{\vdash 0 \cdot a = 0 : p \to m}\ \textsc{zd}$$

In other words, typed semiring are categories enriched over a commutative monoid: each homset is equipped with a commutative monoid structure (typing rules (Tz,Tp) and rules (p,pz,pc,pa)), and composition distributes over these monoid structures (rules (dp,dz,pd,zd)).

Lemma 2 is also valid in this setting: equality is reflexive and relates correctly typed terms only. However, due to the presence of the annihilator element (0), Lemmas 3 and 4 no longer hold: 0 has any type, and we have $\vdash x \cdot 0 \cdot x = 0$ while $x \cdot 0 \cdot x$ only admits $\Gamma(x)$ as a valid type. Moreover, some valid proofs cannot be typed just by adding decorations: for example, $0 = 0 * a = 0$ is a valid untyped proof of $0 = 0$, for any a; however, this proof cannot be typed if a is ill-typed. Therefore, we have to adopt another strategy: we reduce the problem to the annihilator-free structure, by showing that equality proofs can be factorised so as to use rules (PZ), (DZ), and (ZD) at first, as oriented rewriting rules.

Definition 7. Let a be a term; we denote by $a_\downarrow$ the *normal form of* a, obtained with the following convergent rewriting system:

$$a + 0 \to a \qquad 0 + a \to a \qquad 0 \cdot a \to 0 \qquad a \cdot 0 \to 0$$

We say that a is *strict* if $a_\downarrow \neq 0$. We let $\_ \vdash^+ \_ = \_ : \_ \to \_$ denote the *strict equality* judgement obtained by removing rules (DZ) and (ZD), and replacing rules (DP) and (PD) with the following variants, where the factor has to be strict.

$$\frac{\vdash a : n \to m \qquad a_\downarrow \neq 0 \qquad \vdash b, c : m \to p}{\vdash^+ a \cdot (b + c) = a \cdot b + a \cdot c : n \to p}\ \text{DP}^+$$

$$\frac{\vdash a : n \to m \qquad a_\downarrow \neq 0 \qquad \vdash b, c : p \to n}{\vdash^+ (b + c) \cdot a = b \cdot a + c \cdot a : p \to m}\ \text{PD}^+$$

Type derivations about strict terms enjoy the kind of injectivity we had for monoids:

Lemma 8. *For all strict term a such that $\vdash a : n \to m$ and $\vdash a : n' \to m'$, we have $n = n'$ iff $m = m'$.*

Then, using the same methodology as previously, one easily obtain the untyping theorem for strict equality judgements:

Proposition 9. *Let a, b such that $\vdash^+ a = b$; for all n, m such that $\vdash a, b : n \to m$, we have $\vdash^+ a = b : n \to m$.*

Note that the patched rules for distributivity, (DP^+) and (PD^+) are required in order to obtain the counterpart of Lemma 4: if a was not required to be strict, then we would have $\vdash^+ 0 \cdot (x + y) = 0 \cdot x + 0 \cdot y$, and the right-hand side can be typed in environment $\Gamma = \{x \mapsto (3, 2),\ y \mapsto (4, 2)\}$ while the left-hand side cannot.

We now have to show that any equality proof can be factorised, so as to obtain a strict equality proof relating the corresponding normal forms:

Proposition 10. *For all a, b such that $\vdash a = b$, we have $\vdash^+ a_\downarrow = b_\downarrow$.*

Proof. We first show by induction that whenever $\vdash a = b$, a is strict iff b is strict $(\dagger)$. Then we proceed by induction on the derivation $\vdash a = b$, we detail only some cases:

(D) we have $\vdash^+ a_\downarrow = a'_\downarrow$ and $\vdash^+ b_\downarrow = b'_\downarrow$ by induction; we need to show that $\vdash^+ (a \cdot b)_\downarrow = (a' \cdot b')_\downarrow$. If one of a, a', b, b' is not strict, then $(a \cdot b)_\downarrow = (a' \cdot b')_\downarrow = 0$, thanks to $(\dagger)$, so that we are done; otherwise, $(a \cdot b)_\downarrow = a_\downarrow \cdot b_\downarrow$, and $(a' \cdot b')_\downarrow = a'_\downarrow \cdot b'_\downarrow$, so that we can apply rule (D).

(DZ) trivial, since $(a \cdot 0)_\downarrow = 0$.

(DP) we need to show that $\vdash^+ (a \cdot (b + c))_\downarrow = (a \cdot b + a \cdot c)_\downarrow$; if one of a, b, c is not strict, both sides reduce to the same term, so that we can apply Lemma 2(i) (which holds in this setting); otherwise we have $(a \cdot (b + c))_\downarrow = a_\downarrow \cdot (b_\downarrow + c_\downarrow)$ and $(a \cdot b + a \cdot c)_\downarrow = a_\downarrow \cdot b_\downarrow + a_\downarrow \cdot c_\downarrow$, so that we can apply rule (DP^+) ($a_\downarrow$ is obviously strict). ■

Since the normalisation procedure preserves types and respects equalities, we finally obtain the untyping theorem.

Lemma 11. *For all a, n, m such that $\vdash a : n \to m$, we have $\vdash a_\downarrow : n \to m$ and $\vdash a = a_\downarrow : n \to m$.*

Theorem 12. *In semirings, for all a, b, n, m such that we have $\vdash a, b : n \to m$, $\vdash a = b$ iff $\vdash a = b : n \to m$.*

Proof. The reverse implication is straightforward; we prove the direct one. By Lemma 11, using the transitivity and symmetry rules, it suffices to show $\vdash a_\downarrow = b_\downarrow : n \to m$. This is clearly the case whenever $\vdash^+ a_\downarrow = b_\downarrow : n \to m$, which follows from Props. 10 and 9. ■

3.3 Kleene algebras

Kleene algebras are idempotent semirings equipped with a Kleene star operation [14]; they admit several important models, among which binary relations and *regular languages* (the latter is the initial model [17, 15]; since equality of regular languages is decidable, so is the equational theory of Kleene algebras). Like previously, this structure can be typed in a rather natural way, where star operates on "square" types: types of the form $n \to n$, i.e., square matrices or homogeneous binary relations.

Definition 13. *Typed Kleene algebras* are defined by the signature $\{\cdot_2, +_2, \star_1, 1_0, 0_0\}$, together with the following inference rules (in addition to the rules from Definitions 1 and 6, and Section 2), where $\vdash a \leq b : n \to m$ is an

abbreviation for $\vdash a + b = b : n \to m$.

$$\frac{\vdash a : n \to n}{\vdash a^\star : n \to n}\ \textsc{Ts} \qquad \frac{\vdash a = b : n \to n}{\vdash a^\star = b^\star : n \to n}\ \textsc{s}$$

$$\frac{\vdash a : n \to m}{\vdash a + a = a : n \to m}\ \textsc{pi} \qquad \frac{\vdash a : n \to n}{\vdash 1 + a \cdot a^\star = a^\star : n \to n}\ \textsc{sp}$$

$$\frac{\vdash a \cdot b \leq b : n \to m}{\vdash a^\star \cdot b \leq b : n \to m}\ \textsc{sl} \qquad \frac{\vdash b \cdot a \leq b : n \to m}{\vdash b \cdot a^\star \leq b : n \to m}\ \textsc{sr}$$

The untyped version of this axiomatisation is that from Kozen [15]: axiom (PI) corresponds to idempotence of $+$, the three other rules define the star operation. The last two rules have premises, so that we are no longer in a purely equational setting (the algebra of regular events is not finitely based [23]). This is not problematic for our purpose: one can extend the proofs from the previous section without unexpected difficulties: we add the rule $0^\star \to 1$ to the rewriting system used for normalising terms, and we require b to be strict in the strict versions of rules (SL) and (SR). We do not give more details here: complete proofs are available as Coq scripts [21].

Theorem 14. *In Kleene algebras, for all a, b, n, m with $\vdash a, b : n \to m$, we have $\vdash a = b$ iff $\vdash a = b : n \to m$.*

4 Residuated lattices

We now move to our second main example, *residuated lattices*. These structures also admit binary relations as models; they are of special interest since they make it possible to reason algebraically about well-founded relations. For example, one can use residuation to prove Newman's Lemma in relation algebras [7]. We start with a simpler structure.

A *residuated monoid* is a tuple $(X, \leq, \cdot, 1, /, \backslash)$, such that $(X, \leq)$ is a partial order, $(X, \cdot, 1)$ is a monoid whose product is monotonic ($a \leq a'$ and $b \leq b'$ entail $a \cdot b \leq a' \cdot b'$), and $\backslash, /$ are binary operations, respectively called *left* and *right divisions*, characterised by the following equivalences:

$$a \cdot b \leq c \quad \Leftrightarrow \quad b \leq a \backslash c \quad \Leftrightarrow \quad a \leq c / b$$

Such a structure can be typed in a natural way, by using the following rules for left and right divisions:

$$\frac{\vdash c : n \to m \qquad \vdash a : n \to p}{\vdash a \backslash c : p \to m}\ \textsc{TL}$$

$$\frac{\vdash c : n \to m \qquad \vdash b : p \to m}{\vdash c / b : n \to p}\ \textsc{TR}$$

Although we can easily define a set of axioms to capture equalities provable in residuated monoids [13], the transitivity rule (T) becomes problematic in this setting. Instead, we exploit a characterisation due to Ono and Komori [20], based on a Gentzen proof system. Indeed, the "cut" rule corresponding to this system, which plays the role of the transitivity rule, can be eliminated. Therefore, this characterisation allows us to avoid the problems we encounter with a standard equational proof system.

4.1 Gentzen proof system for residuated monoids

Let letters l, k, h range over lists of terms, let $l; k$ denote the concatenation of lists l and k, and let ϵ be the empty list. The Gentzen proof system is presented below; it relates lists of terms to terms. Its presentation is quite standard [13]: there is an axiom rule (V), and, for each operator, and introduction and an elimination rule.

$$\frac{}{x \vdash x}\ \textsc{v} \qquad \frac{}{\epsilon \vdash 1}\ \textsc{Io} \qquad \frac{l; l' \vdash a}{l; 1; l' \vdash a}\ \textsc{Eo}$$

$$\frac{l \vdash a \qquad l' \vdash a'}{l; l' \vdash a \cdot a'}\ \textsc{Id} \qquad \frac{l; b; c; l' \vdash a}{l; b \cdot c; l' \vdash a}\ \textsc{Ed}$$

$$\frac{l; b \vdash a}{l \vdash a / b}\ \textsc{Ir} \qquad \frac{k \vdash b \qquad l; c; l' \vdash a}{l; c/b; k; l' \vdash a}\ \textsc{Er}$$

$$\frac{b; l \vdash a}{l \vdash b \backslash a}\ \textsc{Il} \qquad \frac{k \vdash b \qquad l; c; l' \vdash a}{l; k; b \backslash c; l' \vdash a}\ \textsc{El}$$

The axiom rule can be generalised to terms (i), the cut rule is admissible (ii), and the proof system is correct and complete w.r.t. residuated monoids (iii):

Proposition 15.

(i) For all a, we have $a \vdash a$.

(ii) For all l, k, k', a, b such that $l \vdash a$ and $k; a; k' \vdash b$, we have $k; l; k' \vdash b$.

(iii) For all a, b, we have $a \vdash b$ iff $a \leq b$ holds in all residuated monoids.

Proof. The first point is easy; see [20, 19, 13] for cut admissibility and completeness. ■

Since the proof system has the sub-formula property, The third point (iii) leads to decidability of the equational theory of residuated monoids. (The fact that the sequent system actually characterises the partial order is not problematic: to decide an equality $a = b$, it suffices to decide whether both $a \vdash b$ and $b \vdash a$ hold.)

We add types to this proof system in Fig. 1, where the typing judgement has been extended to lists of terms using the following rules: concatenation is typed like a product.

$$\frac{}{\vdash \epsilon : n \to n}\ \textsc{TE} \qquad \frac{\vdash a : n \to m \qquad \vdash l : m \to p}{\vdash a; l : n \to p}\ \textsc{TC}$$

$$\frac{\Gamma(x) = (n,m)}{x \vdash x : n \to m}\ \textsc{v} \qquad \frac{}{\epsilon \vdash 1 : n \to n}\ \textsc{Io} \qquad \frac{l; l' \vdash a : n \to m}{l; 1; l' \vdash a : n \to m}\ \textsc{Eo}$$

$$\frac{l \vdash a : n \to m \qquad l' \vdash a' : m \to p}{l; l' \vdash a \cdot a' : n \to p}\ \textsc{Id} \qquad \frac{l; b; c; l' \vdash a : n \to m}{l; b \cdot c; l' \vdash a : n \to m}\ \textsc{Ed}$$

$$\frac{\vdash b : m \to p \qquad l; b \vdash a : n \to p}{l \vdash a/b : n \to m}\ \textsc{Ir} \qquad \frac{\vdash l' : m \to q \qquad k \vdash b : n \to m \qquad l; c; l' \vdash a : p \to q}{l; c/b; k; l' \vdash a : p \to q}\ \textsc{Er}$$

$$\frac{\vdash b : p \to m \qquad b; l \vdash a : p \to n}{l \vdash b\backslash a : m \to n}\ \textsc{Il} \qquad \frac{\vdash l : p \to m \qquad k \vdash b : m \to n \qquad l; c; l' \vdash a : p \to q}{l; k; b\backslash c; l' \vdash a : p \to q}\ \textsc{El}$$

Figure 1. Typed Sequents for Residuated Monoids.

(Be careful not to confuse $\vdash a, b : n \to m$, which indicates that both a and b have type $n \to m$, with $\vdash a; b : n \to m$, which indicates that the list $a; b$ has type $n \to m$.) Like previously, the untyped proof system can be recovered by considering trivial types and environments. Note that there are several ways to add type decorations to the proof system; we chose to add a minimal set of decorations, with the constraint that the following sanity requirement should be satisfied (counterpart of Lemma $2(ii)$):

Lemma 16. *For all l, a, n, m, if $l \vdash a : n \to m$ then we have $\vdash l, a : n \to m$.*

We were able to prove the untyping theorem using this proof system, but for the unit-free fragment only: we needed to assume that terms have at most one type, which is not true in presence of 1. This proof was rather involved, so that we did not manage to circumvent this difficulty in a direct way. Instead, as hinted in the introduction, we have to move to the following more symmetrical setting.

4.2 Cyclic MLL

The sequent system for residuated monoids actually corresponds to a non-commutative version of intuitionistic multiplicative linear logic (IMLL) [9]: the product $(\cdot)$ is a non-commutative tensor $(\otimes)$, and left and right divisions $(\backslash, /)$ are the corresponding left and right linear implications $(\multimap, \circ\!\!-)$. Moreover, it happens that this system is just the intuitionistic fragment of cyclic multiplicative linear logic (MLL) [26]. The untyping theorem revealed easier to prove in this setting, which we describe below.

We assume a copy $\mathcal{X}^\perp$ of the set of variables $(\mathcal{X})$, and we denote by $x^\perp$ the corresponding elements which we call *dual variables*. From now on, we shall consider terms with both kinds of variables: $T(\Sigma + X + X^\perp)$. We keep an algebraic terminology to remain consistent with the previous sections; notice that using logic terminology, a term is a formula and a variable is an atomic formula.

Definition 17. *Typed MLL terms* are defined by the signature $\{\otimes_2, \parr_2, 1_0, \bot_0\}$, together with the following typing rules:

$$\frac{\vdash a : n \to m \qquad \vdash b : m \to p}{\vdash a \otimes b : n \to p}\ \textsc{T}_\otimes$$

$$\frac{\vdash a : n \to m \qquad \vdash b : m \to p}{\vdash a \parr b : n \to p}\ \textsc{T}_\parr$$

$$\frac{}{\vdash 1 : n \to n}\ \textsc{T}_1 \qquad \frac{}{\vdash \bot : n \to n}\ \textsc{T}_\bot$$

$$\frac{\Gamma(x) = (n,m)}{\vdash x : n \to m}\ \textsc{Tv} \qquad \frac{\Gamma(x) = (n,m)}{\vdash x^\perp : m \to n}\ \textsc{Tv}^\perp$$

Tensor $(\otimes)$ and par $(\parr)$ are typed like the previous dot operation; bottom $(\bot)$ is typed like the unit (1); dual variables are typed by mirroring the types of the corresponding variables. *Linear negation* is defined recursively over terms and list of terms, as follows:

$$\begin{array}{lll} (a \otimes b)^\perp \triangleq b^\perp \parr a^\perp & 1^\perp \triangleq \bot & (x)^\perp \triangleq x^\perp \\ (a \parr b)^\perp \triangleq b^\perp \otimes a^\perp & \bot^\perp \triangleq 1 & (x^\perp)^\perp \triangleq x \\ (a; l)^\perp \triangleq l^\perp; a^\perp & \epsilon^\perp \triangleq \epsilon & \end{array}$$

Note that since we are in a non-commutative setting, negation has to reverse the arguments of tensors and pars, as well as lists. Negation is involutive and mirrors types judgements:

Lemma 18. *For all l we have $l^{\perp\perp} = l$, and for all n, m, $\vdash l : n \to m$ iff $\vdash l^\perp : m \to n$.*

If we were using a two-sided presentation of MLL, judgements would be of the form $l \vdash k : m \to n$, intuitively meaning "$l \vdash k$ is derivable in cyclic MLL, and lists l and k have type $m \to n$". Instead, we work with one-sided sequents to benefit from the symmetrical nature of MLL. At

$$\frac{\Gamma(x)=(n,m)}{\vdash x^\perp; x : m}\ \mathrm{A} \qquad \frac{}{\vdash 1 : n}\ 1 \qquad \frac{\vdash l : n}{\vdash \bot; l : n}\ \bot$$

$$\frac{\vdash l; a : n \qquad \vdash b; k : n}{\vdash l; a \otimes b; k : n}\ \otimes \qquad \frac{\vdash a; b; l : n}{\vdash a \parr b; l : n}\ \parr$$

$$\frac{\vdash a : n \to m \qquad \vdash l; a : m}{\vdash a; l : n}\ \mathrm{E}$$

Figure 2. Typed Sequents for Cyclic MLL.

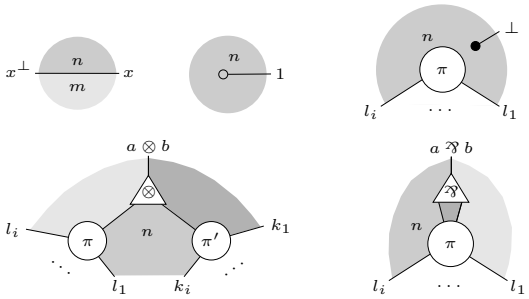


Figure 3. Proof nets for Cyclic MLL.

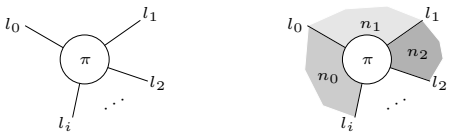
the untyped level, this means that we replace $l \vdash k$ with $\vdash l^\perp; k$. According to the previous intuitions, the list $l^\perp; k$ has a square type $n \to n$: object m is hidden in the concatenation, so that it suffices to record the outer object (n). Judgements finally take the form $\vdash l : n$, meaning "the one-sided MLL sequent $\vdash l$ is derivable at type $n \to n$".

Definition 19. *Typed cyclic MLL* is defined by the sequent calculus from Fig. 2.

Except for type decorations, the system is standard: the five first rules are the logical rules of MLL [9]. Rule (E) is the only structural rule, this is a restricted form of the exchange rule, yielding cyclic permutations: sequents have to be thought as rings [26]. As before, we added type decorations in a minimal way, so as to ensure that derivable sequents have square types, as explained above:

Lemma 20. *For all l, n, if $\vdash l : n$ then $\vdash l : n \to n$.*

We now give a graphical interpretation of the untyping theorem, using proof nets. Since provability is preserved by cyclic permutations, one can draw proof structures by putting the terms of a sequent on a circle [26]. For example, a proof π of a sequent $\vdash l_0, \ldots, l_i$ will be represented by a proof net whose interface is given by the left drawing below.

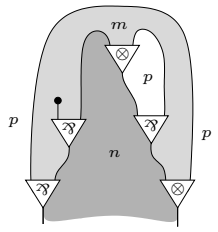


Suppose now that the corresponding list admits a square type: $\vdash l : n \to n$, i.e., $\forall j \leq i$, $\vdash l_j : n_j \to n_{j+1}$, for some $n_0, \ldots, n_{i+1}$ with $n = n_0 = n_{i+1}$. One can add these type decorations as background colours, in the areas delimited by the terms, as we did on the right-hand side.

The logical rules of the proof system (Fig. 2) can then be represented by the proof net constructions from Fig. 3. (Thanks to this sequent representation, the exchange rule (E) is implicit.) Since these constructions preserve planarity, all proof nets are planar, so that the idea of background colours makes sense. Moreover, they can be coloured in a consistent way, so that typed derivations correspond to proof nets that can be entirely and consistently coloured. Therefore, one way to prove the untyping theorem consists in showing that any proof net whose outer interface can be coloured can be coloured entirely.

As an example, we give an untyped derivation below, together with the corresponding proof net. Since the conclusion of this proof has type $p \to p$ whenever $\Gamma(x) = n \to m$ and $\Gamma(y) = m \to p$, the outer interface of the proof net can be coloured (here, with colours p and n). Our untyping theorem will ensure that there exists a typed proof; indeed, the whole proof net can be coloured in a consistent way.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash x^\perp, x}\ \mathrm{A} \quad \dfrac{\dfrac{}{\vdash y^\perp, y}\ \mathrm{A}}{\vdash y, y^\perp}\ \mathrm{E}}{\vdash x^\perp, (x \otimes y), y^\perp}\ \otimes}{\vdash x^\perp, (x \otimes y) \parr y^\perp}\ \parr \quad \dfrac{\dfrac{}{\vdash y^\perp, y}\ \mathrm{A}}{\vdash y, y^\perp}\ \mathrm{E}}{\vdash x^\perp, ((x \otimes y) \parr y^\perp) \otimes y, y^\perp}\ \otimes}{\vdash \bot, x^\perp, ((x \otimes y) \parr y^\perp) \otimes y, y^\perp}\ \bot}{\vdash y^\perp, \bot, x^\perp, ((x \otimes y) \parr y^\perp) \otimes y}\ \mathrm{E}}{\vdash y^\perp \parr \bot \parr x^\perp, ((x \otimes y) \parr y^\perp) \otimes y}\ \parr$$



We do not give a formal account of this graphical interpretation here, since we merely use it to give some intuitions. We now embark in the proof of the untyping theorem for cyclic MLL; the key property is that the types of derivable sequents are all squares:

Proposition 21. *For all l such that $\vdash l$, for all n, m such that $\vdash l : n \to m$, we have $n = m$.*

Proof. We proceed by induction on the untyped derivation $\vdash l$, but we prove a stronger property: "the potential types of all cyclic permutations of l are squares", i.e., for all h,k such that $l = h; k$, for all n, m such that $\vdash k; h : n \to m$, $n = m$. The most involved case is that of the tensor rule. Using symmetry arguments, we can assume that the cutting point belongs to the left premise: the conclusion of the tensor rule is $\vdash l; l'; a \otimes b; k$, we suppose that the induction hypothesis holds for $l; l'; a$ and $b; k$, and knowing that $\vdash l'; a \otimes b; k; l : n \to m$, we have to show $n = m$. Clearly, we have $\vdash l'; a : n \to p$, $\vdash b; k : p \to q$, and $\vdash l : q \to m$

for some p, q. By induction on the second premise, we have $p = q$, so that $\vdash l'; a; l : n \to m$. Since the latter list is a cyclic permutation of $l; l'; a$, we can conclude with the induction hypothesis on the first premise. ■

Theorem 22. *In cyclic MLL, for all l, n with $\vdash l : n \to n$, we have $\vdash l$ iff $\vdash l : n$.*

Proof. The right-to-left implication is straightforward; for the direct implication, we proceed by induction on the untyped derivation. The previous proposition is required in the case of the tensor rule: we know that $\vdash l; a$, $\vdash b; k$, and $\vdash l; a \otimes b; k : n \to n$, and we have to show that $\vdash l; a \otimes b; k : n$. Necessarily, there is some m such that $\vdash l; a : n \to m$ and $\vdash b; k : m \to n$; moreover, by Prop. 21, $n = m$. Therefore, we can apply the induction hypotheses (so that $\vdash l; a : n$ and $\vdash b; k : n$) and we conclude with the typed tensor rule. ■

4.3 Intuitionistic fragment

To deduce that the untyping theorem holds in residuated monoids, it suffices to show that the typed proof system from Fig. 1 corresponds to the intuitionistic fragment of that from Fig. 2. This is well-known for the untyped, commutative, case, and type decorations or cyclicity do not add particular difficulties. Therefore, we just give a brief overview of the corresponding proof.

The idea is to define the following families of *input* and *output* terms (Danos-Regnier polarities [24]), and to work with sequents composed of exactly one output term and an arbitrary number of input terms: the rules from Fig. 2 (or the proof nets from Fig. 3) preserve this property.

$$\begin{array}{rcl} i & ::= & x^\perp \mid \bot \mid i \parr i \mid i \otimes o \mid o \otimes i \\ o & ::= & x \mid 1 \mid o \otimes o \mid i \parr o \mid o \parr i \end{array}$$

Negation $(-^\perp)$ establishes a bijection between input and output terms. Terms of residuated monoids (i.e., IMLL formulae) are encoded into output terms as follows.

$$\begin{array}{ll} \lfloor a \cdot b \rfloor \triangleq \lfloor a \rfloor \otimes \lfloor b \rfloor & \lfloor x \rfloor \triangleq x \\ \lfloor a/b \rfloor \triangleq \lfloor a \rfloor \parr \lfloor b \rfloor^\perp & \lfloor 1 \rfloor \triangleq 1 \\ \lfloor a\backslash b \rfloor \triangleq \lfloor a \rfloor^\perp \parr \lfloor b \rfloor & \end{array}$$

This encoding is a bijection between IMLL terms and output MLL terms; it preserves typing judgements:

Lemma 23. *For all a, n, m, we have $\vdash a : n \to m$ iff $\vdash \lfloor a \rfloor : n \to m$.*

(Note that we heavily rely on overloading to keep notations simple.) The following proposition shows that we actually obtained a fragment of typed cyclic MLL; it yields to the untyping theorem for residuated monoids.

Proposition 24. *For all l, a, n, m with $\vdash l, a : n \to m$, we have $l \vdash a : n \to m$ iff $\vdash \lfloor l \rfloor^\perp; \lfloor a \rfloor : m$.*

Proof. The forward implication is proved by an induction on the sequent derivation. For the reverse direction, we actually prove the following stronger property, by induction on the untyped MLL derivation: "for all h, a, k such that $\vdash \lfloor h \rfloor^\perp; \lfloor a \rfloor; \lfloor k \rfloor^\perp$, for all n, m such that $\vdash h; k : n \to m$ and $\vdash a : n \to m$, we have $h; k \vdash a : n \to m$". Prop. 21 is required in the case of the tensor rule. (The generalisation of the statement with two lists h, k is required to handle the exchange rule; the fact that we use an induction on an untyped derivation is just for commodity, it does not matter thanks to Thm. 22.) ■

Corollary 25. *In residuated monoids, for all l, a, n, m such that $\vdash l, a : n \to m$, we have $l \vdash a$ iff $l \vdash a : n \to m$.*

4.4 Residuated lattices: additives.

The Gentzen proof system we presented for residuated monoids was actually designed for residuated *lattices* [20], obtained by further requiring the partial order $(X, \leq)$ to be a lattice $(X, \vee, \wedge)$. Binary relations fall into this family, by considering set-theoretic unions and intersections. On the logical side, this amounts to considering the additive binary connectives $(\oplus, \&)$, i.e., working in the intuitionistic fragment of multiplicative additive linear logic (IMALL), without additive constants.

The previous proofs scale without major difficulty: by working in cyclic MALL without additive constants, we get an untyping theorem for *involutive residuated lattices* [25]; we deduce the untyping theorem for residuated lattices by considering the corresponding intuitionistic fragment.

5 Conclusions and directions for future work

We proved untyping theorems for several standard structures, allowing us to extend decidability results to the typed settings. All results have been checked [21] in the Coq proof assistant. We conclude by discussing applications, related work, and directions for future work.

5.1 Applications

Improving proof search for residuated structures. All sequent proof systems we mentioned have the sub-formula property, so that provability is decidable in each case, using a simple proof search algorithm [19]. The untyping theorem can be used to cut off useless branches. Indeed, recall the typed rule for the tensor (Fig. 2), which is equivalent to the following one, since sequents are circular lists:

$$\frac{\vdash a : m \to n \qquad \vdash l; a : n \qquad \vdash b; k : n}{\vdash a \otimes b; k; l : m}\ \otimes$$

When using this rule during proof search, one has to choose where to split the sequent (where to put the semicolon between k and l). While all possibilities have to be tried in the untyped setting, this is not the case in the typed setting: the splitting point has to respect types.

Now, starting from an untyped list of terms l, one can easily compute an abstract 'most general type and environment' (n, Γ), such that $\Gamma \vdash l : n \to n$ holds (taking $\mathbb{N}$ as the set of types, for example). Thanks to the untyping theorem, if $\vdash l$ holds, then we also have $\Gamma \vdash l : n$, so that we have at least one proof that respects the most general type, and we can restrict proof search to the corresponding well-behaved branches. Moreover, this trick can be refined during proof search, by generalising the most general type whenever possible: moving to the premises of the tensor rule may release some type constraints, half of the terms being thrown away.

Some experiments on a prototype tend to show that this idea is promising; we still need to understand how it interacts with focusing [1].

Decision of typed Kleene algebras in Coq. The untyping theorem for typed Kleene algebras is quite important in the ATBR Coq library [2]: it allows one to use our tactic for Kleene algebras [3] in the typed setting, and, in particular, with heterogeneous binary relations. The underlying decision procedure being quite involved, we can hardly imagine to prove its soundness in the typed setting; even writing a type-preserving version of the algorithm seems challenging.

At another level, we used the untyping theorem for semirings in order to formalise Kozen's completeness proof [15] for Kleene algebras (the fact that regular languages are the initial model, which we had to prove to reach all models). Indeed, this proof heavily relies on matrix constructions, so that having adequate lemmas and tactics for working with possibly rectangular matrices was a big plus: this allowed us to avoid the ad-hoc constructions Kozen used to inject rectangular matrices into square ones.

5.2 References and related work

The relationship between residuated lattices and substructural logics is due to Ono and Komori [20]; see [13] for a detailed survey about these structures. Cyclic linear logic was suggested by Girard and studied by Yetter [26]. To the best of our knowledge, the idea of adding of types to the above structures is new. The axiomatisation of Kleene algebras is due to Kozen [15].

Our typed structures can be seen as very special cases of *partial* algebras [4], where the domain of partial operations is defined by typing judgements. Similarly, one could use *many-sorted* algebras [11] to mimic our types using sorts. Several encodings from partial algebras to total ones were proposed in the literature [18, 5]. Although they are much more general than our case-by-case study, these results do not apply here: these encodings do not preserve the considered theory since they need to introduce new symbols and equations. As a consequence, ordinary untyped decision procedures can no longer be used after the translation. Dojer has also shown that under some conditions, convergent term rewriting systems for total algebras can be used to prove existence equations in partial algebras [6]. Although this seems applicable for semirings; this approach does not scale to Kleene algebras or residuated lattices, for which decidability does not arise from a term rewriting system.

Closer to our work is that from Kozen, who first proposed the idea of untyping typed Kleene algebras, in order to avoid the aforementioned matrix constructions [16]. He provided a different answer, however: using model-theoretic arguments, he proved an untyping theorem for the Horn theory of "1-free Kleene algebras". The restriction to 1-free expressions is required, as shown by the following counter-example: $\vdash 0 = 1 \Rightarrow a = b$ is a theorem of semirings, although there are non trivial typed semirings where $0 = 1$ holds at some types (e.g., empty matrices), while $a = b$ is not universally true at other types.

5.3 Handling other structures

Residuated pointed lattices: additive constants. We do not know how to handle residuated *pointed* lattices, i.e., residuated lattices with lower and upper bounds $(0, \top)$. These bounds correspond to additive constants in MALL; like for semirings (Sect. 3.2), the problem comes from their polymorphic typing. The typing rules are given below, together with the rule for top (there is no rule for zero).

$$\frac{}{\vdash 0 : n \to m}\ \mathrm{T}_0 \qquad \frac{}{\vdash \top : n \to m}\ \mathrm{T}_\top \qquad \frac{\vdash l : m \to n}{\vdash \top; l : n}\ \top$$

First, our proof breaks because Prop. 21 no longer holds: we have $\vdash \top$, while this sequent admits non square types. More importantly, there are valid untyped derivations that cannot be typed directly, by adding decorations: if $\Gamma(x) = n \to m$ with $n \neq m$, the left derivation below is not valid, while the underlying untyped derivation is.

$$\frac{\dfrac{}{\vdash \top; x : m}\ \top \quad \dfrac{}{\vdash \top; x^\perp : n}\ \top}{\vdash \top; x \otimes \top; x^\perp : n}\ \otimes \qquad \frac{}{\vdash \top; x \otimes \top; x^\perp : n}\ \top$$

As shown on the right-hand side, this typed sequent is nonetheless derivable: the untyping theorem is not refuted by this example. However this means that any potential proof has to rearrange derivations, as we did in Section 3.2. Unlike for semirings or Kleene algebras, we were not able to obtain a factorisation property here (Prop. 10): expressions like $x \otimes \top$ and $x \parr 0$ (or $0/x$ and $x/\top$ in the intuitionistic fragment) cannot be simplified, so that we cannot reduce the problem to the annihilator-free case.

Action algebras. *Action algebras* [22, 12] are a natural extension of the structures we studied in this paper: they are also called *residuated Kleene algebras*; they combine the ingredients from residuated lattices and Kleene algebras. Although we do not know whether the untyping theorem holds in this case; we can think of two strategies to tackle this problem: 1) find a cut-free extension of the Gentzen proof system for residuated lattices (this is left as an open question in [12]; it would entail decidability of the equational theory of action algebras), and adapt our current proof; 2) find a 'direct' proof of the untyping theorem for residuated monoids, without going through the Gentzen proof system, so that the methodology we used to obtain the untyping theorem for Kleene algebras can be extended.

Allegories. Our proofs about semirings can easily be adapted to handle the cases of *allegories* and *distributive allegories* [8]; however, the case of *division allegories*, where left and right divisions are added, remains open. The first approach we mentioned above is not applicable here: the equational theory of allegories is undecidable, so that there is no hope to find a useful extension of the proof system we used for residuated monoids.

5.4 Towards a generic theory

The typed structures we focused on can be described in terms of enriched categories, and the untyping theorems can be rephrased as asserting the existence of a faithful functor to a one-object category. It would therefore be interesting to find out whether our typed structures can be given a generic definition in category theory, and whether one can give a reasonable characterisation of the class of structures for which the untyping theorem holds.

For structures that are varieties, another approach would consist in using term rewriting theory to obtain generic factorisation theorems (Lemma 10, which we used to handle the annihilating element in semirings would become a particular case). This seems rather difficult, however, since this kind of properties are quite sensitive to the whole set of operations and axioms that are considered: adding left and right divisions prevents us from removing annihilators, although axioms about divisions do not mention annihilators.

Last, and rather surprisingly, we have no counter-example yet, i.e., an algebraic structure which can be typed in an interesting way, and for which we could prove that the untyping theorem does not hold. An ad-hoc system could certainly be designed to this end; finding a standard structure would be much more interesting: it would help us in delimiting the scope of a potential generic theory.

References

[1] J.-M. Andreoli. Logic programming with focusing proofs in linear logic. *J. of Logic and Comput.*, 2(3):297–347, 1992.

[2] T. Braibant and D. Pous. Coq library: ATBR, algebraic tools for binary relations. `http://sardes.inrialpes.fr/~braibant/atbr/`, May 2009.

[3] T. Braibant and D. Pous. A tactic for deciding Kleene algebras. In *1st Coq Workshop*. Tech. Univ. München, 2009.

[4] P. Burmeister. *Algebras and Orders*, chapter Partial Algebra—An Introductory Survey. Kluwer Ac. Pub., 1993.

[5] R. Diaconescu. An encoding of partial algebras as total algebras. *Inf. Process. Lett.*, 109(23-24):1245–1251, 2009.

[6] N. Dojer. Applying term rewriting to partial algebra theory. *Fundamenta Informaticae*, 63(4):375–384, 2004.

[7] H. Doornbos, R. Backhouse, and J. van der Woude. A calculational approach to mathematical induction. *Theoretical Computer Science*, 179(1-2):103–135, 1997.

[8] P. Freyd and A. Scedrov. *Categories, Allegories*. North Holland, 1990.

[9] J.-Y. Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.

[10] B. Grégoire and A. Mahboubi. Proving equalities in a commutative ring done right in Coq. In *Proc. TPHOL '05*, volume 3603 of *LNCS*, pages 98–113. Springer Verlag, 2005.

[11] P. J. Higgins. Algebras with a scheme of operators. *Mathematische Nachrichten*, 27:115–132, 1963.

[12] P. Jipsen. From semirings to residuated Kleene lattices. *Studia Logica*, 76(2):291–303, 2004.

[13] P. Jipsen and C. Tsinakis. A survey of residuated lattices. *Ordered Algebraic Structures*, 2002.

[14] S. C. Kleene. Representation of events in nerve nets and finite automata. In *Automata Studies*, pages 3–41. Princeton University Press, 1956.

[15] D. Kozen. A completeness theorem for Kleene algebras and the algebra of regular events. *Information and Computation*, 110(2):366–390, 1994.

[16] D. Kozen. Typed Kleene algebra. Technical Report TR98-1669, C.S. Department, Cornell University, March 1998.

[17] D. Krob. Complete systems of B-rational identities. *Theoretical Computer Science*, 89(2):207–343, 1991.

[18] T. Mossakowski. Relating CASL with other specification languages: the institution level. *Theoretical Computer Science*, 286(2):367–475, 2002.

[19] M. Okada and K. Terui. The finite model property for various fragments of intuitionistic linear logic. *Journal of Symbolic Logic*, 64(2):790–802, 1999.

[20] H. Ono and Y. Komori. Logics without the contraction rule. *Journal of Symbolic Logic*, 50(1):169–201, 1985.

[21] D. Pous. Coq proofs of this paper, 2010. `http://sardes.inrialpes.fr/~pous/utas/`.

[22] V. R. Pratt. Action logic and pure induction. In *JELIA*, volume 478 of *LNCS*, pages 97–120. Springer Verlag, 1990.

[23] V. Redko. On defining relations for the algebra of regular events (russian). *Ukrain. Mat. Z.*, 16:120–126, 1964.

[24] L. Regnier. *Lambda-calcul et réseaux*. Thèse de doctorat, Université Paris VII, 1992.

[25] A. M. Wille. A Gentzen system for involutive residuated lattices. *Algebra Universalis*, 54:449–463, 2005.

[26] D. N. Yetter. Quantales and (noncommutative) linear logic. *The Journal of Symbolic Logic*, 55(1):41–64, 1990.

Centre de recherche INRIA Grenoble – Rhône-Alpes
655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)

Centre de recherche INRIA Bordeaux – Sud Ouest : Domaine Universitaire - 351, cours de la Libération - 33405 Talence Cedex
Centre de recherche INRIA Lille – Nord Europe : Parc Scientifique de la Haute Borne - 40, avenue Halley - 59650 Villeneuve d'Ascq
Centre de recherche INRIA Nancy – Grand Est : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex
Centre de recherche INRIA Paris – Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex
Centre de recherche INRIA Rennes – Bretagne Atlantique : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex
Centre de recherche INRIA Saclay – Île-de-France : Parc Orsay Université - ZAC des Vignes : 4, rue Jacques Monod - 91893 Orsay Cedex
Centre de recherche INRIA Sophia Antipolis – Méditerranée : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399