Isomorphisms of Generic Recursive Polynomial Types

Marcelo Fiore*
Computer Laboratory
University of Cambridge

Abstract

This paper gives the first decidability results on type isomorphism for recursive types, establishing the explicit decidability of type isomorphism for the type theory of sums and products over an inhabited generic recursive polynomial type. The technical development provides connections between themes in programming-language theory (type isomorphism) and computational algebra (Gröbner bases).

Categories and Subject Descriptors: F.3.3 [Logics and Meanings of Programs]: Studies of Program Constructs—*type structure*; F.4.2 [Mathematical Logic and Formal Languages]: Grammars and Other Rewriting Systems—*Decision problems*; F.4.3 [Mathematical Logic and Formal Languages]: Formal Languages—*Decision problems*; I.1.0 [Symbolic and Algebraic Manipulation]: General; F.4.1 [Mathematical Logic and Formal Languages]: Mathematical Logic—*Lambda calculus and related systems*.

General Terms: Theory, Languages, Algorithms.

Keywords: Data structure, Type theory, Type isomorphism, Recursive types, Semigroups, Rigs, Word problem, Gröbner bases.

1 Introduction

Data structure is a central theme is computer science in general and in programming languages in particular. In programming language theory there is a well-established body of work on type systems for describing, analysing, organising, classifying, modularising, *etc.* data and its structure [14]. Within this framework, we are concerned here with the study of the intrinsic structural properties of data as manipulated within a programming language or type theory. Specifically, this paper follows the line of investigation that identifies two types as structurally equivalent if they are isomorphic in the sense that there are mutually inverse procedures for transforming data between them; see, *e.g.*, [4]. This notion should not be confused with the weaker one of type equality, the reflexive closure of subtyping, which is typically given independently of the programming language under consideration, and either neglects to incorporate data transformation in its definition altogether, or is justified by the *ad-hoc* notion of mutual coercion between types. For instance, given any two types T_1 and T_2 the product types $T_1 \times T_2$ and $T_2 \times T_1$ are clearly isomorphic; whilst, according to traditional subtyping, they are not subtypes of each other unless so are T_1 and T_2. Recent investigations of subtyping, however, have relaxed the view by considering it modulo associativity and commutativity of product types [13, 5]; broadening it towards the notion of isomorphism. But isomorphism is strictly more general. Indeed, given types T, T_1, T_2 the types

$$T \times (T_1 + T_2) \quad \text{and} \quad (T \times T_1) + (T \times T_2) \tag{1}$$

where $+$ and $\times$ respectively denote the sum and product type constructors, are isomorphic but they are generally not considered subtypes of each other.

This work studies type isomorphism in the context of recursive types. As explained above, the conceptual difference between type isomorphism and type equality is such that the relationship of this work with that of subtyping recursive types is essentially disjoint (see [2] and the line of work that arose from it and also [9]). The focus of this paper is on generic recursive polynomial types, by which we mean recursively defined types built out of sum and product type constructors that are generic in the sense that the only operations for manipulating recursive types are those of folding and unfolding. A motivation for restricting attention to this setting is that it amounts to considering isomorphisms that are realisable by finitistic programs (given by finite mappings between data-type patterns, see Figures 2 and 6 for examples). Thus, isomorphism in this setting implies isomorphism in the setting of inductive and/or coinductive types; which are, roughly, generic recursive types further admitting manipulations by iterators and/or coiterators, respectively (see, *e.g.*, [1]). The converse does not hold; for instance, there are recursive programs establishing an isomorphism between the types of binary trees and of lists of binary trees, but there is no finitistic one.

As we will see, the algebra of type isomorphism for generic recursive polynomial types is extremely rich; even when restricted to just one recursive type

$$\mathtt{T} \cong \mathrm{F}(\mathtt{T})$$

which is the case considered in the paper. We illustrate the situation with the type of (unlabelled) binary trees

$$\mathtt{B} \cong 1 + \mathtt{B}^2 \tag{2}$$

where 1 denotes the unit type and where the type $\mathtt{T}^n$ ($n \in \mathbb{N}$) denotes the n-fold product type $\mathtt{T} \times \ldots \times \mathtt{T}$ of the type $\mathtt{T}$.

*Research supported by an EPSRC Advanced Research Fellowship.

POPL'04 January 14–16, 2004, Venice, Italy.

The algebra of type isomorphism satisfies commutative and associative laws for sums and products, and the distributive law of products over sums establishing the isomorphism of the types in (1). Thus, for instance, we have that $\mathtt{B}^{n+1} \cong \mathtt{B}^n + \mathtt{B}^{n+2}$ for all $n \in \mathbb{N}$. Further, Lawvere and Schanuel noted in their investigations on Objective Number Theory that other unexpected algebraic manipulations, like the one to follow, are also valid: For the sake of argument, consider (2) in the form

$$-1 = \mathtt{B}^2 - \mathtt{B} \quad \text{and} \quad \mathtt{B} - 1 = \mathtt{B}^2$$

and by algebraic manipulation note that

$$-1 = \mathtt{B}^2 - \mathtt{B} = \mathtt{B}^3 \quad ,$$

and hence that

$$1 = (-1)^2 = \mathtt{B}^6 \tag{3}$$

and so also that

$$\mathtt{B} = \mathtt{B}^7 \quad .$$

There is in principle no reason for which these purely algebraic, abstract manipulations may make sense type theoretically. Indeed, the identity (3) does not; as there is not just one six tuple of binary trees. Astonishingly, the last conclusion is a valid isomorphism:

$$\mathtt{B} \cong \mathtt{B}^7 \quad . \tag{4}$$

This can be established in the algebra of type isomorphism following an heuristic unfolding/folding procedure which proceeds to cancel a $\mathtt{B}^n$ summand in a polynomial by unfolding a higher power of $\mathtt{B}$ in another summand until it produces $\mathtt{B}^{n+2}$ as a summand, that is then folded together with $\mathtt{B}^n$ into $\mathtt{B}^{n+1}$. An application of this procedure deducing (4) is given in Figure 1. In it we have underlined the summands being unfolded and folded in each step. Calculations in the algebra of type isomorphism yield isomorphisms in the type theory; one such, extracted from Figure 1, giving an explicit bijection between seven binary trees and one is presented in Figure 2; see also [3].

The deep reasons for which the calculation of Figure 1 is possible is that the types

$$[-\mathbf{1}] \stackrel{\text{def}}{=} \mathtt{B}^3 \ \text{ and } \ \mathbf{0} \stackrel{\text{def}}{=} 1 + [-\mathbf{1}]$$

satisfy the following isomorphisms

$$\mathbf{0} \times \mathtt{B} \cong \mathbf{0} \tag{5}$$

$$\mathtt{B} + \mathbf{0} \cong \mathtt{B} \tag{6}$$

$$\mathbf{0} + \mathbf{0} \cong \mathbf{0} \tag{7}$$

Indeed, using the isomorphism (5) we can establish the following one

$$\begin{aligned} & \mathtt{B} + \mathbf{0} \times (\mathtt{B} + \mathtt{B}^2 + \mathtt{B}^3) \\ \cong\ & \mathtt{B} + \mathbf{0} \times \mathtt{B} \times (\mathtt{B} + \mathtt{B}^2 + \mathtt{B}^3) \\ \cong\ & \mathtt{B} + \mathtt{B}^2 + \mathtt{B}^3 + \mathtt{B}^4 + \mathtt{B}^5 + \mathtt{B}^6 + \mathtt{B}^7 \\ \cong\ & \mathbf{0} \times (\mathtt{B} + \mathtt{B}^2 + \mathtt{B}^3) + \mathtt{B}^7 \end{aligned} \tag{8}$$

from which, further using the isomorphisms (6) and (7), we can cancel out $\mathbf{0} \times (\mathtt{B} + \mathtt{B}^2 + \mathtt{B}^3)$ to conclude (4). The main empirical contribution of this work is thus that:

> There are negative and even imaginary types.

The main technical problem addressed by this work is that of the decidability of type isomorphism in the above setting. We are not only interested in establishing whether or not two types are isomorphic but moreover in establishing their *explicit decidability*; *i.e.*, in providing non-exhaustive methods for constructing explicit isomorphisms (or proof derivations in the algebraic context) whenever they exist. The main technical contribution of the paper is that:

> In the type theory of sums and products with an inhabited generic recursive polynomial type
>
> $$\mathtt{T} \cong \mathsf{F}(\mathtt{T})$$
>
> the type isomorphism problem between polynomial types
>
> $$\mathsf{F}_1(\mathtt{T}) \stackrel{?}{\cong} \mathsf{F}_2(\mathtt{T})$$
>
> is explicitly decidable.

The above introductory remarks place this work in the context of programming language theory. However, it also has a place in the context of structural combinatorics, where one is interested in investigating bijections between combinatorial structures.

2 Type isomorphism

This section introduces the type theory of the paper.

2.1 The type theory

The type theory of sums, products, and generic recursive types has types given by the following grammar

$\mathsf{T} \in \mathcal{T}(\mathbb{V})$	$::=$	$\mathtt{X}$	(Type variables)
	$\mid$	$\mathtt{0}$	(Empty type)
	$\mid$	$\mathsf{T}_1 + \mathsf{T}_2$	(Binary sum types)
	$\mid$	$\mathtt{1}$	(Unit type)
	$\mid$	$\mathsf{T}_1 \times \mathsf{T}_2$	(Binary product types)
	$\mid$	$\rho \mathtt{X}.\mathsf{T}$	(Generic Recursive types)

where $\mathtt{X}$ ranges over (a countable set of) type variables $\mathbb{V}$. We use a non-standard operator for recursive types to emphasize that they are *generic*; rather than inductive or coinductive, for instance. The recursive type constructor $\rho\mathtt{X}.\mathsf{T}$ binds the free occurrence of $\mathtt{X}$ in T; accordingly types are identified up to alpha equivalence. The ordinary definitions of free variables, bound variables, and substitution apply.

The raw terms associated to the above types are given by the following grammar

t	$::=$	x	(Variables)
	$\mid$	$\bot_{\mathsf{T}}(t)$	(Absurd)
	$\mid$	$\iota_1^{\mathsf{T}}(t)$	(First injection)
	$\mid$	$\iota_2^{\mathsf{T}}(t)$	(Second injection)
	$\mid$	$\delta(t, x_1.t_1, x_2.t_2)$	(Discriminator)
	$\mid$	$\langle\rangle$	(Unit)
	$\mid$	$\langle t_1, t_2 \rangle$	(Pairing)
	$\mid$	$\pi_1(t)$	(First projection)
	$\mid$	$\pi_2(t)$	(Second projection)
	$\mid$	$\mathrm{fold}_{\rho\mathtt{X}.\mathsf{T}}(t)$	(Fold)
	$\mid$	$\mathrm{unfold}(t)$	(Unfold)

$$
\begin{array}{llllllllll}
\underline{B} & \cong 1+\underline{B^2} & \cong & 1+B+\underline{B^3} & \cong & \underline{1}+B+\underline{B^2}+B^4 & \cong & B+B+\underline{B^4} & \cong & B+\underline{B}+\underline{B^3}+B^5 \\
 & \cong B+B^2+\underline{B^5} & \cong & B+\underline{B^2}+\underline{B^4}+B^6 & \cong & \underline{B}+\underline{B^3}+B^6 & \cong & B^2+\underline{B^6} & \cong & B^2+\underline{B^5}+B^7 \\
 & \cong \underline{B^2}+\underline{B^4}+B^6+B^7 & \cong & B^3+\underline{B^6}+B^7 & \cong & \underline{B^3}+\underline{B^5}+B^7+B^7 & \cong & B^4+\underline{B^7}+B^7 & \cong & \underline{B^4}+\underline{B^6}+B^7+B^8 \\
 & \cong \underline{B^5}+\underline{B^7}+B^8 & \cong & \underline{B^6}+\underline{B^8} & \cong & B^7 & & & &
\end{array}
$$

Figure 1. $B \cong 1+B^2 \Rightarrow B \cong B^7$

```
datatype B = e | m of B * B  ;

val fold: B * B * B * B * B * B * B -> B = fn t => case t of

    ( e, e, e, e, e, e, e ) => e

  | ( b1, m(e,e), e, e, e, e, e ) => m(b1, e)

  | ( m(b1,b2) , e , e , e , e , e , e ) => m(b1,m(b2,e))

  | ( b1, m(m(b2,b3),e), e, e, e, e, e ) => m(b1,m(b2,m(b3,e)))

  | ( b1, m(b2,m(b3,b4)), e , e , e , e , e ) => m(b1,m(b2,m(b3,m(b4,e))))

  | ( b1, b2, m(b3,b4), e, e, e, e ) => m(b1,m(b2,m(b3,m(b4,m(e,e)))))

  | ( b1, b2, b3, m(b4,b5), e, e, e ) => m(b1,m(b2,m(b3,m(b4,m(m(b5,e),e)))))

  | ( b1, b2, b3, b4, m(b5,b6), e, e ) => m(b1,m(b2,m(b3,m(b4,m(m(b5,m(b6,e)),e)))))

  | ( b1, b2, b3, b4, b5, b6, m(b7,b8) ) => m(b1,m(b2,m(b3,m(b4,m(m(b5,m(b6,m(b7,b8))),e)))))

  | ( b1, b2, b3, b4, b5, m(b6,b7), e ) => m(b1,m(b2,m(b3,m(b4,m(b5,m(b6,b7))))))  ;
```

Figure 2. Seven binary trees in one

where x ranges over (a countable set of) variables. As usual, terms are identified up to alpha equivalence; the discriminator δ stands for the usual case analysis operation on sum types and binds the free occurrences of x_i in t_i ($i = 1,2$). Again, the ordinary definitions of free variables, bound variables, and substitution apply.

A context is a finite set of variables together with associated types. Well-typed terms in context are given by derivable judgements $\Gamma \vdash t : T$, where Γ is a context, t is a term, and T is a type; they are given according to the typing rules of Figure 3.

The equational theory that provides the notion of term equality in the type theory is given by the rules in Figure 4. This equational theory splits naturally in congruence rules, η and β rules for sums and products, and the mutual inverse of folding and unfolding.

The notion of type isomorphism is intrinsic to the type theory and given below.

DEFINITION 1. *Let $\mathbb{T} \subseteq \mathcal{T}(\mathbb{V})$ be a set of types. Two types $T_1, T_2 \in \mathbb{T}$ are said to be isomorphic in the restriction of the type theory to types in $\mathbb{T}$, if there exist terms $x_1 : T_1 \vdash t_2 : T_2$ and $x_2 : T_2 \vdash t_1 : T_1$ of this type theory such that in it the judgements $x_1 : T_1 \vdash t_1[{}^{t_2}/x_2] = x_1 : T_1$ and $x_2 : T_2 \vdash t_2[{}^{t_1}/x_1] = x_2 : T_2$ are derivable.*

EXAMPLE 2.

1. *Every type T is isomorphic to the type $T+0$ via the terms $x : T+0 \vdash \delta(\ x\ ,\ x_1.x_1\ ,\ x_2.\bot_T(x_2)\ ) : T$ and $x : T \vdash \iota_1^0(x) : T+0$.*

2. *For every type T, the types $T \times 0$ and 0 are isomorphic via the terms $x : T \times 0 \vdash \pi_2(x) : 0$ and $x : 0 \vdash \langle \bot_T(x), x\rangle : T \times 0$.*

3. *For all types T, T_1, T_2, the types $(T_1+T_2) \times T$ and $(T_1 \times T)+(T_2 \times T)$ are isomorphic via the terms*

$$
\begin{array}{l}
x : (T_1+T_2) \times T \\
\quad \vdash \delta(\ \ \pi_1(x)\ \ , \\
\qquad\qquad x_1.\iota_1^{T_2 \times T}(\langle x_1, \pi_2(x)\rangle)\ \ , \\
\qquad\qquad x_2.\iota_2^{T_1 \times T}(\langle x_2, \pi_2(x)\rangle)\ \ ) : (T_1 \times T)+(T_2 \times T)
\end{array}
$$

and

$$
\begin{array}{l}
x : (T_1 \times T)+(T_2 \times T) \\
\quad \vdash \delta(\ \ x\ \ , \\
\qquad\qquad x_1.\langle \iota_1^{T_2}(\pi_1(x_1)), \pi_2(x_1)\rangle\ \ , \\
\qquad\qquad x_2.\langle \iota_2^{T_1}(\pi_1(x_2)), \pi_2(x_2)\rangle\ \ ) : (T_1+T_2) \times T \quad .
\end{array}
$$

2.2 The algebra of type isomorphism

To understand type isomorphism, and in particular study its algebra, it is convenient to consider the type theory in categorical form through its associated classifying syntactic category.

$$\frac{}{\Gamma, x : T, \Gamma' \vdash x : T} \qquad \frac{\Gamma \vdash t : 0}{\Gamma \vdash \bot_T(t) : T}$$

$$\frac{\Gamma \vdash t : T_i}{\Gamma \vdash \iota_i^{T_j}(t) : T_1 + T_2} \ (i,j = 1,2; i \neq j) \qquad \frac{\Gamma \vdash t : T_1 + T_2 \qquad \Gamma, x_i : T_i \vdash t_i : T \quad (i = 1,2)}{\Gamma \vdash \delta(t, x_1.t_1, x_2.t_2) : T}$$

$$\frac{}{\Gamma \vdash \langle\rangle : 1} \qquad \frac{\Gamma \vdash t_i : T_i \quad (i = 1,2)}{\Gamma \vdash \langle t_1, t_2 \rangle : T_1 \times T_2} \qquad \frac{\Gamma \vdash t : T_1 \times T_2}{\Gamma \vdash \pi_i(t) : T_i} \ (i = 1,2)$$

$$\frac{\Gamma \vdash t : T[^{\rho X.T}/_X]}{\Gamma \vdash \mathrm{fold}_{\rho X.T}(t) : \rho X.T} \qquad \frac{\Gamma \vdash t : \rho X.T}{\Gamma \vdash \mathrm{unfold}(t) : T[^{\rho X.T}/_X]}$$

Figure 3. Typing rules

DEFINITION 3. *For a set of types $\mathbb{T} \subseteq \mathcal{T}(\mathbb{V})$, we let $\mathcal{C}(\mathbb{T})$ be the category with objects given by types in $\mathbb{T}$ and morphisms $T_1 \to T_2$ given by equivalence classes $[x : T_1 \vdash t : T_2]$ of derivable judgements in the type theory restricted to types in $\mathbb{T}$ under the equivalence identifying $(x : T_1 \vdash t : T_2)$ and $(x' : T_1 \vdash t' : T_2)$ iff the judgement $x : T_1 \vdash t = t'[^x/_{x'}] : T_2$ is derivable in the type theory restricted to types in $\mathbb{T}$. Composition is by substitution*

$$[x' : T_2 \vdash t' : T_3] \circ [x : T_1 \vdash t : T_2] = [x : T_1 \vdash t'[^t/_{x'}] : T_3]$$

with identities given by $[x : T \vdash x : T]$.

Thus, two types are isomorphic in the restriction of the type theory to types in the set $\mathbb{T}$ (in the sense of Definition 3) iff they are isomorphic in $\mathcal{C}(\mathbb{T})$ (in the categorical sense).

Below we will be interested in studying type isomorphism in the type theory restricted to a set of types closed under sums and products. It will be useful thus to introduce the following definition and notation: The closure of a set of types $\mathbb{T} \subseteq \mathcal{T}(\mathbb{V})$ under sums and products, is denoted $[\mathbb{T}]$ and is defined inductively by the following rules

$$\frac{T \in \mathbb{T}}{T \in [\mathbb{T}]} \qquad 0 \in [\mathbb{T}] \qquad \frac{T_1, T_2 \in [\mathbb{T}]}{T_1 + T_2 \in [\mathbb{T}]}$$

$$1 \in [\mathbb{T}] \qquad \frac{T_1, T_2 \in [\mathbb{T}]}{T_1 \times T_2 \in [\mathbb{T}]}$$

2.2.1 Sums and products

Type isomorphism without generic recursive types is well-understood.

PROPOSITION 4. *For a set of type variables $\mathbb{V}$, the category $\mathcal{C}[\mathbb{V}]$ is the free distributive category on $\mathbb{V}$.*

Type isomorphism for sum and product types is completely axiomatised by the equational rules in Figure 5.

The complete equational axiomatisation for sum and product types of Figure 5 splits into congruence rules and axioms stating that the structures $([\mathbb{V}], 0, +)$ and $([\mathbb{V}], 1, \times)$ are commutative monoids with products distributing over sums; that is, that $([\mathbb{V}], 0, +, 1, \times)$ is a so-called commutative rig (= ri*n*g without *n*egatives).

It follows that every type in $[\mathbb{V}]$ is isomorphic to a polynomial type, and that type isomorphism is decidable. Indeed, two types are isomorphic iff they have the same polynomial form. We have the following corollary.

COROLLARY 5 (DECIDABILITY). *For a set of type variables $\mathbb{V}$, type isomorphism in $\mathcal{C}[\mathbb{V}]$ is explicitly decidable.*

2.2.2 Sums, products, and a generic recursive polynomial type

We move now to consider the algebra of type isomorphism for the type theory of sums and products over an inhabited generic recursive polynomial type.

More precisely, we consider the categories $\mathcal{C}[R]$ where $R = \rho X.F$ is a generic recursive type in which F is a type in $[X]$. A universal characterisation of these categories follows.

PROPOSITION 6. *For $X \in \mathbb{V}$ and $F \in [X]$, the category $\mathcal{C}[\rho X.F]$ is the free distributive category on a generator $R = \rho X.F$ equipped with a generic isomorphism $R \cong F[^R/_X]$.*

For the purpose of using a result of Gates [8] characterising type isomorphism in the setting we are considering we need further restrict to inhabited types.

DEFINITION 7. *A type T is said to be* inhabited *iff there exists a term t such that the judgement $\vdash t : T$ is derivable.*

PROPOSITION 8. *For $X \in \mathbb{V}$ and $F \in [X]$, the type $\rho X.F$ is inhabited iff the polynomial form of the type $F[^0/_X]$ is not the empty type 0.*

In the setting of this section, which is indeed the context for the rest of the paper, we have the following characterisation of type isomorphism.

THEOREM 9 (GATES [8]). *Let $R = \rho X.F$ be an inhabited type where $F \in [X]$. In $\mathcal{C}[R]$, two types are isomorphic iff they are provably equal by the equational rules in Figure 5 from the axiom $R \approx F[^R/_X]$.*

Note the important fact that since by the equational rules of Figure 5 every type in $[R]$ has a computable isomorphic polynomial form on R, the study of the explicit decidability of type isomorphisms $T_1 \cong T_2$ in $\mathcal{C}[R]$ for inhabited $R = \rho X.F$ with $F \in [X]$ reduces, by invoking the above theorem, to establishing the explicit decidability of $P_1 \approx P_2$ between polynomial types P_1, P_2 on $T = \rho X.P$, where

$$\Gamma \vdash t = t : T \qquad \frac{\Gamma \vdash t = t' : T}{\Gamma \vdash t' = t : T} \qquad \frac{\Gamma \vdash t_1 = t_2 : T \qquad \Gamma \vdash t_2 = t_3 : T}{\Gamma \vdash t_1 = t_3 : T}$$

$$\frac{\Gamma \vdash t = t' : \mathtt{0}}{\Gamma \vdash \bot_T(t) = \bot_T(t') : T}$$

$$\frac{\Gamma \vdash t = t' : T_i}{\Gamma \vdash \iota_i^{T_j}(t) = \iota_i^{T_j}(t') : T_1 + T_2} \quad (i,j = 1,2;\, i \neq j) \qquad \frac{\Gamma \vdash t = t' : T_1 + T_2 \qquad \Gamma, x_i : T_i \vdash t_i = t_i' : T \quad (i = 1,2)}{\Gamma \vdash \delta(t, x_1.t_1, x_2.t_2) = \delta(t', x_1.t_1', x_2.t_2') : T}$$

$$\frac{\Gamma \vdash t_i = t_i' : T_i \quad (i = 1,2)}{\Gamma \vdash \langle t_1, t_2 \rangle = \langle t_1', t_2' \rangle : T_1 \times T_2} \qquad \frac{\Gamma \vdash t = t' : T_1 \times T_2}{\Gamma \vdash \pi_i(t) = \pi_i(t') : T_i} \quad (i = 1,2)$$

$$\frac{\Gamma \vdash t = t' : T[{}^{\rho X.T}/X]}{\Gamma \vdash \mathrm{fold}_{\rho X.T}(t) = \mathrm{fold}_{\rho X.T}(t') : \rho X.T} \qquad \frac{\Gamma \vdash t = t' : \rho X.T}{\Gamma \vdash \mathrm{unfold}(t) = \mathrm{unfold}(t') : T[{}^{\rho X.T}/X]}$$

$$\Gamma \vdash \bot_T(t) = t' : T \qquad \Gamma \vdash \delta(\iota_i^{T_j}(t), x_1.t_1, x_2.t_2) = t_i[{}^t/x_i] : T \quad (i,j = 1,2;\, i \neq j)$$

$$\Gamma \vdash \delta(t, x_1.t'[{}^{\iota_1^{T_2}(x_1)}/x], x_2.t'[{}^{\iota_2^{T_1}(x_2)}/x]) = t'[{}^t/x] : T$$

$$\Gamma \vdash t = \langle\rangle : \mathtt{1} \qquad \Gamma \vdash \pi_i(\langle t_1, t_2 \rangle) = t_i : T_i \quad (i = 1,2) \qquad \Gamma \vdash t = \langle \pi_1(t), \pi_2(t) \rangle : T_1 \times T_2$$

$$\Gamma \vdash \mathrm{unfold}(\mathrm{fold}_{\rho X.T}(t)) = t : T[{}^{\rho X.T}/X] \qquad \Gamma \vdash \mathrm{fold}_{\rho X.T}(\mathrm{unfold}(t)) = t : \rho X.T$$

Figure 4. Equational rules

$P \in [X]$ is the polynomial form of F, under the equational theory of commutative rigs (Figure 5) from the axiom $\mathtt{T} \approx P[{}^{\mathtt{T}}/X]$. This is the situation that we will restrict to in the sequel.

3 Algebra

We have just seen that the key to understanding type isomorphism for sums and products over an inhabited generic recursive polynomial type is the algebra of commutative rigs. In fact, other related algebraic structures also play a salient role here. This section gives the basic background and the necessary results on the various commutative algebras that arise.

Semigroups, monoids, and groups. A *semigroup* consists of a set together with an associative binary operation. A *monoid* is a semigroup with a neutral element. A *group* is a monoid in which every element has an inverse. These algebraic structures are said to be commutative whenever the binary operation is.

Subsets of semigroups closed under the binary operation are referred to as *subsemigroups*. A subsemigroup that happens to be a monoid/group is referred to as a *submonoid*/*subgroup*.

Rigs and rings. A *semirig* $(R, +, 1, \cdot)$ is given by an additive commutative semigroup $(R, +)$ and a multiplicative monoid $(R, 1, \cdot)$ satisfying the distributive laws

$$(x + y) \cdot z = x \cdot z + y \cdot z$$
$$z \cdot (x + y) = z \cdot x + z \cdot y$$

for all $x, y, z \in R$. A *rig* is a semirig for which its additive semigroup is a monoid, say with neutral element 0, satisfying the laws

$$0 \cdot x = 0 = x \cdot 0$$

for all elements x. A *ring* is a rig in which the additive monoid is a group. These algebraic structures are said to be commutative whenever the multiplication is.

The set of natural numbers $\mathbb{N}$ with the usual operations of addition and multiplication is a rig, but not a ring. The set of integers $\mathbb{Z}$ is a ring.

For a rig/ring $(R, 0, +, 1, \cdot)$, the set of polynomials $R[x]$ on a generator x with the usual operations of addition and multiplication is also a rig/ring.

Further rigs/rings can be constructed by quotients: for a relation $\sim$ on a rig/ring R, we let $R/{\sim}$ be the quotient rig/ring of R by the least congruence containing $\sim$. As it is customary, for a subset S of R we write R/S for $R/{\sim_S}$ where $x \sim_S 0$ for all $x \in S$. Examples of such rings are the rings of modular integers $\mathbb{Z}_c \cong \mathbb{Z}/(c)$ for $1 \leq c \in \mathbb{N}$, and the ring of Gaussian integers $\mathbb{Z}[i] \cong \mathbb{Z}[x]/(x^2 + 1)$.

Other examples of quotient rigs that we will encounter below are the rig of degrees

$$\mathrm{Deg} = \{ -\infty, 0, 1, \ldots, n, \ldots \} \cong \mathbb{N}[x]/{\sim_{\mathrm{deg}}}$$

where $p \sim_{\mathrm{deg}} q$ iff $\deg(p) = \deg(q)$, and the rig of three dimensions

$$D_3 = \{ 0, 1, \infty \} \cong \mathbb{N}[x]/{\sim_3}$$

where $p \sim_3 q$ iff either $\deg(p) = \deg(q) = 0$ or both $\deg(p)$ and $\deg(q)$ are positive.

$$\mathsf{T} \approx \mathsf{T} \qquad \frac{\mathsf{T} \approx \mathsf{T}'}{\mathsf{T}' \approx \mathsf{T}} \qquad \frac{\mathsf{T}_1 \approx \mathsf{T}_2 \quad \mathsf{T}_2 \approx \mathsf{T}_3}{\mathsf{T}_1 \approx \mathsf{T}_3}$$

$$\frac{\mathsf{T}_1 \approx \mathsf{T}_1' \quad \mathsf{T}_2 \approx \mathsf{T}_2'}{\mathsf{T}_1 + \mathsf{T}_2 \approx \mathsf{T}_1' + \mathsf{T}_2'} \qquad \frac{\mathsf{T}_1 \approx \mathsf{T}_1' \quad \mathsf{T}_2 \approx \mathsf{T}_2'}{\mathsf{T}_1 \times \mathsf{T}_2 \approx \mathsf{T}_1' \times \mathsf{T}_2'}$$

$$\mathsf{T} + 0 \approx \mathsf{T} \qquad (\mathsf{T}_1 + \mathsf{T}_2) + \mathsf{T}_3 \approx \mathsf{T}_1 + (\mathsf{T}_2 + \mathsf{T}_3) \qquad \mathsf{T}_1 + \mathsf{T}_2 \approx \mathsf{T}_2 + \mathsf{T}_1$$

$$\mathsf{T} \times 1 \approx \mathsf{T} \qquad (\mathsf{T}_1 \times \mathsf{T}_2) \times \mathsf{T}_3 \approx \mathsf{T}_1 \times (\mathsf{T}_2 \times \mathsf{T}_3) \qquad \mathsf{T}_1 \times \mathsf{T}_2 \approx \mathsf{T}_2 \times \mathsf{T}_1$$

$$0 \times \mathsf{T} \approx 0 \qquad (\mathsf{T}_1 + \mathsf{T}_2) \times \mathsf{T} \approx (\mathsf{T}_1 \times \mathsf{T}) + (\mathsf{T}_2 \times \mathsf{T})$$

Figure 5. Complete axiomatisation of isomorphism for sum and product types

A non-standard example is the rig/ring of valued monomials $R\lceil x \rceil$ in a a rig/ring R, with underlying set given by $\{c\,x^n \mid c \in R, n \in \mathbb{N}\} \subseteq R[x]$, with addition given by

$$c\,x^m + d\,x^n = \begin{cases} c\,x^m & \text{, if } m > n \\ d\,x^n & \text{, if } m < n \\ (c+d)\,x^m & \text{, if } m = n \end{cases}$$

and multiplication given as for polynomials.

More rigs/rings can be constructed by the product of such, with operations given pointwise. In particular, the binary product of R and R' is denoted $R \times R'$.

A rig homomorphism is a function between rigs that commutes with the operations. Examples are: the head homomorphism $h : R[x] \to R\lceil x \rceil$ mapping a polynomial to its head (*i.e.*, its valued monomial of highest degree); the degree homomorphism $\deg : R\lceil x \rceil \to \mathrm{Deg}$ mapping a valued monomial to its degree; and the dimension homomorphism $\dim : \mathrm{Deg} \to D_3$ identifying all the positive degrees.

For polynomials $p_1, p_2 \in \mathbb{N}[x]$, there is a bijective correspondence between elements r in a rig R such that $p_1(r) = p_2(r)$ and rig homomorphisms $\mathbb{N}[x]/(p_1(x) = p_2(x)) \to R$. Indeed, the universal rig homomorphism induced by such an element r is given by the mapping $p \mapsto p(r)$; whilst the universal element associated to a rig homomorphism is obtained by evaluating at x.

3.1 Cancellation and inverses in commutative semigroups

Arguments like the passage from (8) to (4) in the introduction are intimately related to cancellation properties. This section studies cancellation, and the more general notion of invertibility, in the context of commutative semigroups.

3.1.1 Cancellation in commutative semigroups

We give a cancellation lemma for submonoids of commutative semigroups. The central concept needed for this purpose is the analogue of the natural precongruence on a monoid induced by its binary operation. For semigroups, however, this relation is typically just a transitive congruence.

DEFINITION 10. *For elements a, x, y of a semigroup $(S, +)$, we let*

$$a : x \lhd y \quad \Longleftrightarrow \quad x + a = y$$

and define $\lhd$ as the transitive relation on S given by

$$x \lhd y \quad \Longleftrightarrow \quad a : x \lhd y \textit{ for some } a \quad .$$

Further, we let $<>$ be the equivalence relation on S given by

$$x <> y \quad \Longleftrightarrow \quad x = y \textit{ or both } x \lhd y \textit{ and } y \lhd x$$

and let $\langle x \rangle \subseteq S$ be the equivalence class of x under $<>$.

(The equivalence relation $<>$ corresponds to Green's relations $\mathcal{H}$, $\mathcal{L}$, $\mathcal{R}$, $\mathcal{J}$, $\mathcal{D}$ all of which coincide in the case of commutative semigroups; see, *e.g.*, [11].)

PROPOSITION 11. 1. *In a semigroup $(S, +)$ with $a : x \lhd y$ and $b : y \lhd z$ we have that*

$$a + b : x \lhd z \quad .$$

2. *In a commutative semigroup $(S, +)$ with $a : x \lhd y$ and $b : u \lhd v$ we have that*

$$a + b : x + u \lhd y + v \quad .$$

Further, for a semigroup $(S, \cdot)$ with $\cdot$ distributing over $+$, we also have that

$$x\,b + a\,u + a\,b : x\,u \lhd y\,v \quad .$$

A simple cancellation lemma follows.

LEMMA 12 (CANCELLATION). *Let $(M, +, e)$ be a submonoid of a commutative semigroup $(S, +)$. For $x, y \in M$ and $a \in S$ such that $a \lhd e$ the following cancellation law holds:*

$$x + a = y + a \quad \Longrightarrow \quad x = y \quad .$$

PROOF. For $a' : a \lhd e$ we have that $x = x + e = x + a + a' = y + a + a' = y + e = y$. ☐

3.1.2 Groups within commutative semigroups

In many situations, like the ones we will encounter in Section 4, cancellation in semigroups is a manifestation of subgroup structure. A fundamental general result in semigroup theory due to Green [10], the commutative version of which we give below, gives an analysis of this phenomenon.

THEOREM 13. *Let* $(S,+)$ *be a commutative semigroup, and let* $H = \langle h \rangle \subseteq S$ *for* $h \in S$. *The following statements are equivalent.*

1. *The relationship* $h+h \lhd h$ *holds.*
2. *There are* $k,\ell \in H$ *such that* $k+\ell \in S$ *is in* H.
3. H *is a subsemigroup of* S.
4. H *is a maximal subgroup of* S.
5. H *contains an idempotent.*

PROOF. $(1 \Rightarrow 2)$ Because h and $h+h$ are in H.

$(2 \Rightarrow 3)$ Let $k,\ell,k+\ell \in H$. For $x,y \in H$, since $x+y <> k+\ell$, we have $x+y \in H$.

$(3 \Rightarrow 4)$ For $k,\ell \in H$, let $x : k+\ell \lhd k$ and $y : k \lhd \ell$, and define $e = \ell + x$. We have that $e \in H$ because $\ell \lhd e \lhd k$. Further, e is idempotent because

$$e+e = \ell+x+k+y+x = k+y+x = e\ .$$

Thus, for $d \in H$ with $z : e \lhd d$, we have that

$$d+e = e+z+e = e+z = d$$

and hence that $(H,e,+)$ is a submonoid of S. It is also a subgroup because $d \lhd e$ for every $d \in H$.

Let H be a subgroup of a group K which is a subgroup of S. As $e \in H$ is the neutral element of K, we have that $e \lhd k \lhd e$ for all $k \in K$, from which it follows that $K = H$.

$(4 \Rightarrow 5)$ As groups have exactly one idempotent: the neutral element.

$(5 \Rightarrow 1)$ Because for e an idempotent in H, we have that $h+h <> e+e = e$. □

We extend the above result to the case of semirigs.

PROPOSITION 14. *Let* e *be an idempotent in a commutative semigroup* $(S,+)$ *and let* $(S,\cdot,1)$ *be a monoid with* $\cdot$ *distributing over* $+$. *The structure* $(\langle e \rangle, e, +, u, \cdot)$ *with* $1 \lhd u$ *is a ring iff* $1+e^2 \lhd e$ *and* $u = 1+e$.

PROOF. $(\Rightarrow)$ Since $1 \lhd u$, we have that $1+e \lhd u+e = u$ and hence that $1+e^2 = 1+e \in \langle e \rangle$. Thus, $1+e = (1+e)u = u+eu = u+e = u$.

$(\Leftarrow)$ By Theorem 13, the structure $(\langle e \rangle, e, +)$ is a commutative group.

Since by hypothesis $1 \lhd e$, it follows that $e \lhd 1+e \lhd e+e = e$ and hence that $1+e \in \langle e \rangle$. Further, we have that $e^2 \in \langle e \rangle$ because by hypothesis $e^2 \lhd e$ and also $1 \lhd e$, which yields $e \lhd e^2$. Thus, for any $x,y \in \langle e \rangle$, since $xy <> e^2$, we have that $xy \in \langle e \rangle$.

For $x \in \langle e \rangle$, we have that

$$\begin{aligned} xe &= x(e+e) = xe+xe \\ ex &= (e+e)x = ex+ex \end{aligned}$$

from which we conclude that $xe = ex = e$. Consequently, the following calculations, with $x \in \langle e \rangle$,

$$\begin{aligned} x(1+e) &= x+xe = x+e = x \\ (1+e)x &= x+ex = x+e = x \end{aligned}$$

show that $1+e$ is a neutral element for multiplication.

The distributivity of $\cdot$ over $+$ in $\langle e \rangle$ is inherited from that of S. □

4 Isomorphisms of generic recursive polynomial types

As observed at the end of Section 2, the study of type isomorphism in $\mathcal{C}[\rho X.F]$ with $F \in [X]$ inhabited amounts to the study of equality of polynomials in the quotient polynomial rig $\mathbb{N}[x]/(x = p(x))$ with $p \in \mathbb{N}[x]$ such that $p(0) \neq 0$. This section studies these rigs.

The theory is developed in two parts. First, we deal with the case of polynomials representing generic trees. The more sophisticated case of linear polynomials respectively representing generic lists, generic naturals, and generic constants is considered next.

The development in each case follows the same outline. First, the internal structure of the rigs is investigated in detail; subsequently cancellation and representation results are obtained. The representations are rig embeddings (*i.e.*, injective rig homomorphisms) of the form

$$\mathbb{N}[x]/(x = p(x)) \hookrightarrow R \times \mathbb{Z}[x]/(x - p(x))$$

where R is an appropriate parameter rig that stratifies the quotient polynomial ring $\mathbb{Z}[x]/(x-p(x))$. From these representations the decidability of the word problem easily follows. Indeed, the parameter rigs R have a trivially decidable equality, whilst decision procedures for the word problem in the quotient polynomial ring $\mathbb{Z}[x]/(x-p(x))$ are well-known from the theory of Gröbner bases. Finally, the constructive nature of the methods, which provide explicit constructions for the provable equations in the rigs and hence type isomorphisms between the associated types, is illustrated with example applications.

4.1 Generic trees

This section is concerned with type isomorphism for the type theory of sums and products over a type

$$\boxed{T \cong P(T) \qquad (P \in \mathbb{N}[T], \deg(P) \geq 2, P(0) \neq 0)}$$

of *generic trees*.

We start with a careful analysis of the relation $\lhd$ in the associated rigs. In particular, we first establish that all monomials of positive degree are equivalent under the associated equivalence $<>$.

PROPOSITION 15. *Let* $p \in \mathbb{N}[x]$ *be such that* $\deg(p) \geq 2$ *and* $p(0) \neq 0$.

1. *For all* $n,m \in \mathbb{N}$,

$$\left(\sum_{i=0}^{m-1} x^{n+i}\right) \cdot q(x) : x^n \lhd x^{n+m}$$

where $q(x) = p(x) - 1 \in \mathbb{N}[x]$.

2. *For* $n > d = \deg(p)$,
$$x^{n-d}\, q(x) : x^n \lhd x^{n-d+1}$$
where $q(x) = p(x) - x^d \in \mathbb{N}[x]$.

3. *For a monomial* x^n *in* p *and* $0 \le \ell \le n$,
$$1 + q(x) + \left(\sum_{i=n-\ell}^{n-1} x^i \right) \cdot (x^n + q(x)) : x^{n-\ell} \lhd x$$
where $q(x) = p(x) - (1 + x^n) \in \mathbb{N}[x]$.

The above proposition is extended from monomials to polynomials.

LEMMA 16. *Let* $p \in \mathbb{N}[x]$ *be such that* $\deg(p) \ge 2$ *and* $p(0) \ne 0$. *For all* $q, q' \in \mathbb{N}[x]$ *with* $\deg(q') \ge 1$, *we have that* $q \lhd q'$ *in* $\mathbb{N}[x]/(x = p(x))$. *Further, a witness for this relationship can be computed.*

PROOF. We first observe that $2x \lhd x$. Indeed, if $p(0) \ge 2$, we have that
$$x\, q(x) : 2x \lhd x^2$$
where $q(x) = p(x) - 2 \in \mathbb{N}[x]$, and hence we can compute a witness for
$$2x \lhd x^2 \lhd x$$
using Propositions 11 and 15(3). Otherwise $p(0) = 1$, and with respect to a monomial x^n with $n \ge 1$ in p, we can compute a witnesses for
$$2x \lhd x + x^{n+1} = x(1 + x^n) \lhd x^2 \lhd x$$
again using Propositions 11 and 15(3). Hence, using also Propositions 15(1) and 15(2), we can compute a witness for
$$\sum_{i \in I} x^{n_i} \lhd \sum_{i \in I} x \lhd x \quad .$$

Further, since for any $q' \in \mathbb{N}[x]$ with $\deg(q') \ge 1$, we can easily compute a witness for
$$x \lhd q'(x)$$
using Propositions 11 and 15(1), we are done. □

We obtain the striking result, noted by Steve Schanuel [12], that the polynomials of positive degree form a subring of the ambient semirig. This gives precise mathematical meaning to additive inverses of, or negative, types—recall the introduction.

COROLLARY 17. *For* $p \in \mathbb{N}[x]$ *with* $\deg(p) \ge 2$ *and* $p(0) \ne 0$, *the equivalence class* $\langle x \rangle$ *in the additive semigroup* $\mathbb{N}[x]/(x = p(x))$ *is given by the subset*
$$\{ q \in \mathbb{N}[x]/(x = p(x)) \mid \deg(q) \ge 1 \} \quad .$$

Moreover, for $w(x) : 2x \lhd x$ *in* $\mathbb{N}[x]/(x = p(x))$, *the structure*
$$(\langle x \rangle \,,\, \mathbf{0} \,,\, + \,,\, \mathbf{1} \,,\, \cdot\,)$$
where
$$\mathbf{0} = x + w(x) \quad \textit{and} \quad \mathbf{1} = 1 + \mathbf{0}$$
is a ring.

PROOF. The first part follows from Lemma 16; the second part from Theorem 13 and Proposition 14. □

LEMMA 18 (CANCELLATION). *Let* $p \in \mathbb{N}[x]$ *with* $\deg(p) \ge 2$ *and* $p(0) \ne 0$. *For* $t_1, t_2, q \in \mathbb{N}[x]$ *with* $\deg(t_1), \deg(t_2) \ge 1$, *the following cancellation law holds in* $\mathbb{N}[x]/(x = p(x))$*:*
$$t_1 + q = t_2 + q \quad \Longrightarrow \quad t_1 = t_2 \quad .$$

PROOF. Use Lemma 12 and Corollary 17. □

We now present the first representation result. The parameter rig in this case is the rig of three dimensions: one for the null polynomial, one for polynomials of degree zero, and the other one for polynomials of degree greater than or equal one.

THEOREM 19 (REPRESENTATION). *For* $p \in \mathbb{N}[x]$ *with* $\deg(p) \ge 2$ *and* $p(0) \ne 0$, *the universal rig homomorphism*
$$\mathbb{N}[x]/(x = p(x)) \longrightarrow D_3 \times \mathbb{Z}[x]/(x - p(x))$$
induced by $(\infty, x) \in D_3 \times \mathbb{Z}[x]/(x - p(x))$ *is injective.*

That is, $t_1 = t_2$ *in* $\mathbb{N}[x]/(x = p(x))$ *iff either (1)* $\deg(t_1) = \deg(t_2) \le 0$ *and* $t_1 = t_2$ *in* $\mathbb{N}$, *or (2)* $\deg(t_1), \deg(t_2) \ge 1$ *and* $t_1 = t_2$ *in* $\mathbb{Z}[x]/(x - p(x))$.

PROOF. $(\Rightarrow)$ Note that for $t_1, t_2 \in \mathbb{N}$, the equality $t_1 = t_2$ holds in $\mathbb{Z}[x](x - p(x))$ iff it holds in $\mathbb{N}$.

$(\Leftarrow)$ Let $t_1, t_2 \in \mathbb{N}[x]$.

If $\dim(t_1) = \dim(t_2) = 0$ then $t_1 = t_2 = 0$.

If $\dim(t_1) = \dim(t_2) = 1$ and $t_1 = t_2$ in $\mathbb{Z}[x]/(x = p(x))$ then $t_1 = t_2$ in $\mathbb{N}$.

If $\dim(t_1) = \dim(t_2) = \infty$ then $\deg(t_1), \deg(t_2) \ge 1$. Further, if $t_1 = t_2$ in $\mathbb{Z}[x]/(x - p(x))$ then there exist $q_1, q_2 \in \mathbb{N}[x]$ such that
$$t_1 - t_2 = (q_1 - q_2) \cdot (x - p(x)) \tag{9}$$
in $\mathbb{Z}[x]$ and hence such that
$$t_1 + q_1 \cdot p(x) + q_2 \cdot x = t_2 + q_1 \cdot x + q_2 \cdot p(x)$$
in $\mathbb{N}[x]$. Thus,
$$t_1 + (q_1 + q_2) \cdot x = t_2 + (q_1 + q_2) \cdot x$$
in $\mathbb{N}[x]/(x = p(x))$, and by the cancellation Lemma 18, we have
$$t_1 = t_2$$
in $\mathbb{N}[x]/(x = p(x))$. □

PROPOSITION 20. *For* $p \in \mathbb{N}[x]$ *with* $\deg(p) \ge 2$ *and* $p(0) \ne 0$, *we have the following description*
$$\mathbb{N}[x]/(x = p(x)) \cong \mathbb{N} \uplus \langle x \rangle$$
where, further,
$$\langle x \rangle \cong \mathbb{Z}[x]/(x - p(x)) \quad .$$

The representation gives a simple decidability test for the word problem and the method further allows the explicit construction of proof derivations in the algebra, and hence of isomorphisms in the type theory.

COROLLARY 21 (DECIDABILITY). *For $p \in \mathbb{N}[x]$ with $\deg(p) \geq 2$ and $p(0) \neq 0$, the word problem in $\mathbb{N}[x]/(x = p(x))$, and hence also type isomorphism in $\mathcal{C}[\rho X.\mathsf{F}]$ where $\mathsf{F} \in [X]$ has polynomial form p, are explicitly decidable.*

PROOF. Theorem 19 provides the following decidability test: $t_1 = t_2$ in $\mathbb{N}[x]/(x = p(x))$ iff either $t_1 = t_2$ in $\mathbb{N}$, or $\deg(t_1), \deg(t_2) \geq 1$ and t_1 and t_2 have the same normal form in $\mathbb{Z}[x]$ under the reduction rule

$$h(p) \rightsquigarrow x - (p(x) - h(p)) \quad .$$

Further, for $t_1 = t_2$ in $\mathbb{N}[x]/(x = p(x))$ we may divide $t_1 - t_2$ by $x - p(x)$ in $\mathbb{Z}[x]$ to obtain the situation (9) and proceed as in there to produce a derivation of the equality. $\square$

We conclude the section with two example applications: one revisiting isomorphism between seven binary trees and one [3], and the other one analysing an imaginary-unit type [6].

EXAMPLE 22 (SEVEN BINARY TREES IN ONE). *We consider x^7 in $\mathbb{N}[x]/(x = x^2 + 1)$. Under the reduction rule $x^2 \rightsquigarrow x - 1$ we have that*

$$\begin{array}{rcl} x^7 & \rightsquigarrow & x(x-1)^3 \\ & = & x^4 - 3x^3 + 3x^2 - x \\ & \rightsquigarrow & (x-1)^2 - 3x(x-1) + 3(x-1) - x \\ & = & -2x^2 + 3x - 2 \\ & \rightsquigarrow & -2(x-1) + 3x - 2 \\ & = & x \end{array}$$

and hence that $x^7 = x$ in $\mathbb{N}[x]/(x = x^2 + 1)$. We will now follow the methods of the section to give a derivation of this identity.

First observe that $x^4 : 2x \lhd x$, as the following calculation shows

$$x^4 + 2x = x^4 + x^2 + x + 1 = x^3 + x + 1 = x^2 + 1 = x$$

so that

$$x^4 + x = x^4 + x^2 + 1 = x^3 + 1 \stackrel{\text{def}}{=} \mathbf{0} \quad . \qquad (10)$$

Second, the identities

$$x^n \cdot \mathbf{0} = \mathbf{0} \ (n \in \mathbb{N}) \quad \text{and} \quad x^n + \mathbf{0} = x^n \ (1 \leq n \in \mathbb{N})$$

hold, and one can easily give derivations as follows:

1. *$x \cdot \mathbf{0} = \mathbf{0}$ as shown in (10) and, by induction, $x^{n+1} \cdot \mathbf{0} = x^n \cdot (x \cdot \mathbf{0}) = x^n \cdot \mathbf{0} = \mathbf{0}$ for all $n \in \mathbb{N}$.*

2. *$x + \mathbf{0} = x^3 + x + 1 = x^2 + 1 = x$ and $x^{n+1} + \mathbf{0} = x^n \cdot (x + \mathbf{0}) = x^{n+1}$ for all $n \in \mathbb{N}$.*

Finally, using the division algorithm

$$x^7 - x = \Big((x^2 + x) - (x^5 + x^4)\Big) \cdot \Big(x - (x^2 + 1)\Big)$$

in $\mathbb{Z}[x]$, and hence

$$\begin{array}{l} x^7 + (x^4 + x^3 + x + 1) \cdot x^2 \\ \quad = x^7 + (x^5 + x^4) \cdot x + (x^2 + x) \cdot (x^2 + 1) \\ \quad = x + (x^5 + x^4) \cdot (x^2 + 1) + (x^2 + x) \cdot x \\ \quad = x + (x^4 + x^3 + x + 1) \cdot x^2 \end{array} \qquad (11)$$

in $\mathbb{N}[x]/(x = 1 + x^2)$. Further, since

$$x^4 + x^3 + x + 1 = \mathbf{0} \quad ,$$

the derivation (11) can be expanded to a derivation of $x^7 = x$ by precomposing it with

$$x^7 = x^7 + \mathbf{0} = x^7 + (x^4 + x^3 + x + 1) \cdot x^2$$

and postcomposing it with

$$x + (x^4 + x^3 + x + 1) \cdot x^2 = x + \mathbf{0} = x \quad .$$

An isomorphism obtained from this derivation is given in Figure 6.

EXAMPLE 23 (THE IMAGINARY-UNIT TYPE). *Let $\mathbb{Z}[i]$ be the ring of* Gaussian integers*, with underlying set*

$$\{ m + n\,i \mid m, n \in \mathbb{Z} \}$$

and operations as for the complex numbers.

The results of this section imply that the mapping associating $p \in \mathbb{N}[x]/(x = 1 + x + x^2)$ to $p \in \mathbb{N}$ if $\deg(p) \leq 0$, and to $p(i) \in \mathbb{Z}[i]$ otherwise, yields a bijection

$$\mathbb{N}[x]/(x = 1 + x + x^2) \quad \cong \quad \mathbb{N} \uplus \mathbb{Z}[i] \quad .$$

It follows that the isomorphisms satisfied by the imaginary-unit type

$$\mathtt{I} = \rho X. 1 + X + X^2$$

are characterised by the identities satisfied by the imaginary unit i. Indeed, for $X \in \mathbb{V}$, if $\mathsf{T}_1, \mathsf{T}_2 \in [X]$ are types with polynomial form $p_1, p_2 \in \mathbb{N}[X]$ then

$\mathsf{T}_1(\mathtt{I}) \cong \mathsf{T}_2(\mathtt{I})$ *in* $\mathcal{C}[\mathtt{I}]$

iff

either $\deg(p_1), \deg(p_2) \leq 0$ *or* $\deg(p_1), \deg(p_2) \geq 1$, *and* $p_1(i) = p_2(i)$ *in* $\mathbb{Z}[i]$.

Thus, for instance, we have the following isomorphisms

$$1 + \mathtt{I}^2 \cong \mathtt{I} + \mathtt{I}^3 \,, \quad 2 + \mathtt{I}^2 \cong \mathtt{I}^4 \,, \quad \mathtt{I} \cong \mathtt{I}^5$$

in $\mathcal{C}[\mathtt{I}]$.

4.2 Linear generic recursive types

We turn attention to type isomorphism for the type theory of sums and products over an inhabited linear generic recursive type:

$$\boxed{\mathtt{L} \cong c + d \times \mathtt{L} \qquad (1 \leq c \in \mathbb{N}, d \in \mathbb{N})}$$

A key difference with the previous situation of generic trees is that there are invariants (like having the same head or degree) that hold for the generic recursive type and its unfolding, and hence also for all isomorphic types. Indeed, we have the following simple property.

PROPOSITION 24. *For $p, q \in \mathbb{N}[x]$ and for $t_1 = t_2$ in $\mathbb{N}[x]/(p(x) = q(x))$, if $h(p) = h(q)$ then $h(t_1) = h(t_2)$, and if $\deg(p) = \deg(q)$ then $\deg(t_1) = \deg(t_2)$.*

It is thus natural to consider three different subcases: when $1 \leq c \in \mathbb{N}$ and $2 \leq d \in \mathbb{N}$ (Section 4.2.1) in which case the degree is an invariant; when $1 \leq c \in \mathbb{N}$ and $d = 1$ (Section 4.2.2) in which

```
datatype B = e | m of B * B  ;

val encode: B * B * B * B * B * B * B -> B = fn t => case t of

    ( e, e, e, e, e, e, e ) => m(e,e)

  | ( e, e, e, e, e, e, m(b1,b2) ) => m(m(m(m(m(e,e),e),b1),e),b2)

  | ( e, e, e, e, e, m(b1,b2), e ) => m(e,m(b1,b2))

  | ( e, e, e, e, e, m(b1,b2), m(b3,b4) ) => m(m(m(m(m(e,e),m(b1,b2)),b3),e),b4)

  | ( e, e, e, e, m(e,e), e, e ) => e

  | ( e, e, e, e, m(e,e), m(b1,b2), e ) => m(m(e,b1),b2)

  | ( e, e, e, e, m(e,m(b1,b2)), b3, e ) => m(m(m(m(e,b1),b2),e),b3)

  | ( e, e, e, e, m(e,e), b1, m(b2,b3) ) => m(m(m(e,b1),b2),b3)

  | ( e, e, e, e, m(e,m(b1,b2)), b3, m(b4,b5) ) => m(m(m(m(e,b1),b2),m(b3,b4)),b5)

  | ( e, e, e, e, m(m(b1,b2),b3), b4, b5 ) =>  m(m(m(m(m(e,e),b1),b2),m(b3,b4)),b5)

  | ( e, e, e, m(b1,b2), b3, b4, b5 ) => m(m(m(m(m(e,m(e,b1)),b2),b3),b4),b5)

  | ( e, e, m(b1,b2), b3, b4, b5, b6 ) => m(m(m(m(m(e,m(m(e,b1),b2)),b3),b4),b5),b6)

  | ( e, m(b1,b2), b3, b4, b5, b6, b7 ) => m(m(m(m(m(e,m(m(m(e,b1),b2),b3)),b4),b5),b6),b7)

  | ( m(e,b1), b2, b3, b4, b5, b6, b7 ) => m(m(m(m(m(m(b1,b2),b3),b4),b5),b6),b7)

  | ( m(m(b1,b2),b3), b4, b5, b6, b7, b8, b9 ) => m(m(m(m(m(e,m(m(m(m(b1,b2),b3),b4),b5)),b6),b7),b8),b9)   ;
```

Figure 6. Seven binary trees in one

case the head is an invariant; and when $1 \leq c \in \mathbb{N}$ and $d = 0$ (Section 4.2.3) in which case the situation is trivial.

4.2.1 Generic lists

A linear generic recursive type of the form:

$$\boxed{\mathtt{L} \cong c + d \times \mathtt{L} \qquad (1 \leq c \in \mathbb{N}, 2 \leq d \in \mathbb{N})}$$

is referred to as a *generic list*.

The structure of the rig $\mathbb{N}[x]/(x = c + d\,x)$ is intricate. In particular, there are different additive idempotents of each degree. These are defined below, and their properties established afterwards.

DEFINITION 25. *Let* $1 \leq c \in \mathbb{N}$ *and* $2 \leq d \in \mathbb{N}$. *For* $1 \leq n \in \mathbb{N}$, *we set*

$$\mathbf{0}_n \;=\; x^{n-1} \cdot \big(c + (d-1)\,x\big) \quad .$$

Note that $\mathbf{0}_{n+k} = x^k\,\mathbf{0}_n$ for all $1 \leq n \in \mathbb{N}$ and $k \in \mathbb{N}$.

PROPOSITION 26. *Let* $1 \leq c \in \mathbb{N}$ *and* $2 \leq d \in \mathbb{N}$.

1. *For every* $1 \leq m \leq n \in \mathbb{N}$,

$$x^n + \mathbf{0}_m \;=\; x^n \quad \text{in } \mathbb{N}[x]/(x = c + d\,x)$$

and hence for all $p \in \mathbb{N}[x]$ *with* $\deg(p) \geq m$,

$$p + \mathbf{0}_m \;=\; p \quad \text{in } \mathbb{N}[x]/(x = c + d\,x) \quad .$$

2. *For every* $1 \leq n \in \mathbb{N}$,

$$\langle x^n \rangle = \{\, q \in \mathbb{N}[x]/(x = c + d\,x) \mid \deg(q) = n \,\} \tag{12}$$

is a maximal subgroup of the additive semigroup $\mathbb{N}[x]/(x = c + d\,x)$ *with neutral element* $\mathbf{0}_n$.

PROOF. (1) We first show by induction that $x^n + \mathbf{0}_1 = x^n$ for all $1 \leq n \in \mathbb{N}$. Indeed, $x + \mathbf{0}_1 = x$ by definition of $\mathbf{0}_1$ and

$$\begin{aligned} x^{n+1} + \mathbf{0}_1 &= c\,x^n + d\,x^{n+1} + \mathbf{0}_1 \\ &= c\,x^n + d\,x^{n+1} && \text{, by induction} \\ &= x^{n+1} \end{aligned}$$

for all $1 \leq n \in \mathbb{N}$. Thus, for $1 \leq m \leq n \in \mathbb{N}$, we have that $x^n + \mathbf{0}_m = x^{m-1} \cdot (x^{n-m+1} + \mathbf{0}_1) = x^{m-1} \cdot x^{n-m+1} = x^n$.

(2) By Proposition 24, the inclusion $(\subseteq)$ in (12) holds. As for the reverse inclusion, we already know from the previous part that $\mathbf{0}_m \lhd p$ for all $p \in \mathbb{N}[x]$ with $1 \leq m \leq \deg(p)$; further since

$$x^n \lhd \mathbf{0}_n \quad \text{, for all } 1 \leq n \in \mathbb{N}$$

and since by the previous part

$$\mathbf{0}_n : \mathbf{0}_m \lhd \mathbf{0}_n \quad \text{, for all } 1 \le m \le n \in \mathbb{N}$$

we also have that, for $p = \sum_i c_i x^i$,

$$p \lhd \sum_i c_i \mathbf{0}_i = \mathbf{0}_{\deg(p)} \quad .$$

The rest of the statement follows by Theorem 13, because for all $1 \le n \in \mathbb{N}$ each $\mathbf{0}_n \in \langle x^n \rangle$ is an idempotent. □

Building on the above, we deduce cancellation, representation, and decidability results.

LEMMA 27 (CANCELLATION). *Let $1 \le c \in \mathbb{N}$ and $2 \le d \in \mathbb{N}$. For $t_1, t_2, q \in \mathbb{N}[x]$ with $0 \le \deg(q) \le \deg(t_1) = \deg(t_2)$, the following cancellation law holds in $\mathbb{N}[x]/(x = c + d\,x)$*

$$t_1 + q = t_2 + q \quad \Longrightarrow \quad t_1 = t_2 \quad .$$

PROOF. The case in which $\deg(q) = \deg(t_1) = \deg(t_2) = 0$ holds because, for all $m, n \in \mathbb{N}$, we have $m = n$ in $\mathbb{N}[x]/(x = c + d\,x)$ iff $m = n$ in $\mathbb{N}$; for all other cases, use Lemma 12 and Proposition 26, noticing that $q \lhd \mathbf{0}_{\deg(q)} \lhd \mathbf{0}_{\deg(t_1)}$. □

The parameter rig of the representation is the rig of degrees.

THEOREM 28 (REPRESENTATION). *For $1 \le c \in \mathbb{N}$ and $2 \le d \in \mathbb{N}$, the universal rig homomorphism*

$$\mathbb{N}[x]/(x = c + d\,x) \longrightarrow \mathrm{Deg} \times \mathbb{Z}[x]/(c + (d-1)\,x)$$

induced by $(1, x) \in \mathrm{Deg} \times \mathbb{Z}[x]/(c + (d-1)\,x)$ is injective.

That is, $t_1 = t_2$ in $\mathbb{N}[x]/(x = c + d\,x)$ iff $\deg(t_1) = \deg(t_2)$ and $t_1 = t_2$ in $\mathbb{Z}[x]/(c + (d-1)\,x)$.

PROOF. ($\Rightarrow$) Use Proposition 24.

($\Leftarrow$) Let $t_1, t_2 \in \mathbb{N}[x]$.

If $\deg(t_1) = \deg(t_2) = -\infty$ then $t_1 = t_2 = 0$.

If $\deg(t_1) = \deg(t_2) = 0$ and $t_1 = t_2$ in $\mathbb{Z}[x]/(c + (d-1)\,x)$ then $t_1 = t_2$ in $\mathbb{N}$.

If $\deg(t_1) = \deg(t_2) \ge 1$ and $t_1 = t_2$ in $\mathbb{Z}[x]/(c + (d-1)\,x)$ then there exist $q_1, q_2 \in \mathbb{N}[x]$ such that

$$t_1 - t_2 = (q_1 - q_2) \cdot (x - (c + d\,x)) \tag{13}$$

in $\mathbb{Z}[x]$ and hence such that

$$t_1 + q_1 \cdot (c + d\,x) + q_2 \cdot x = t_2 + q_1 \cdot x + q_2 \cdot (c + d\,x)$$

in $\mathbb{N}[x]$. Thus,

$$t_1 + (q_1 + q_2) \cdot x = t_2 + (q_1 + q_2) \cdot x$$

in $\mathbb{N}[x]/(x = c + d\,x)$, and by the cancellation Lemma 27, we have

$$t_1 = t_2$$

in $\mathbb{N}[x]/(x = c + d\,x)$. □

PROPOSITION 29. *For $1 \le c \in \mathbb{N}$ and $2 \le d \in \mathbb{N}$, we have the following description*

$$\mathbb{N}[x]/(x = c + d\,x) \quad \cong \quad \mathbb{N} \uplus \left(\biguplus_{1 \le n \in \mathbb{N}} \langle x^n \rangle \right) \quad .$$

As in the previous case of generic trees, we also obtain an explicit decidability result for generic lists.

COROLLARY 30 (DECIDABILITY). *For $1 \le c \in \mathbb{N}$ and $2 \le d \in \mathbb{N}$, the word problem in $\mathbb{N}[x]/(x = c + d\,x)$, and hence also type isomorphism in $\mathcal{C}[\rho X.\, F]$ where $F \in [X]$ has polynomial form $c + d\,X$, are explicitly decidable.*

PROOF. Theorem 28 provides the following decidability test: $t_1 = t_2$ in $\mathbb{N}[x]/(x = c + d\,x)$ iff $\deg(t_1) = \deg(t_2)$, and t_1 and t_2 have the same normal form in $\mathbb{Z}[x]$ under the reduction rule $(d-1)\,x \rightsquigarrow -c$.

Further, for $t_1 = t_2$ in $\mathbb{N}[x]/(x = c + d\,x)$ we may divide $t_1 - t_2$ by $c + (d-1)\,x$ in $\mathbb{Z}[x]$ to obtain the situation (13) and proceed as in there to produce a derivation of the equality. □

To conclude the section, we exemplify the theory for a negative-unit type [15].

EXAMPLE 31 (THE NEGATIVE-UNIT TYPE). *The results of this section imply that the mapping associating $p \in \mathbb{N}[x]/(x = 1 + 2\,x)$ to $p \in \mathbb{N}$, if $\deg(p) \le 0$, and to $(\deg(p) - 1, p(-1)) \in \mathbb{N} \times \mathbb{Z}$ otherwise, yields a bijection*

$$\mathbb{N}[x]/(x = 1 + 2\,x) \quad \cong \quad \mathbb{N} \uplus (\mathbb{N} \times \mathbb{Z}) \quad .$$

It follows that the isomorphisms satisfied by the negative-unit type

$$\mathtt{M} = \rho X.\, 1 + 2 \times X$$

are characterised by the identities satisfied by the negative unit -1. Indeed, for $X \in \mathbb{V}$, if $T_1, T_2 \in [X]$ are types with polynomial form $p_1, p_2 \in \mathbb{N}[x]$ then

$$\begin{array}{ll} & T_1(\mathtt{M}) \cong T_2(\mathtt{M}) \textit{ in } \mathcal{C}[\mathtt{M}] \\ \textit{iff} & \\ & \deg(p_1) = \deg(p_2) \textit{ and } p_1(-1) = p_2(-1) \textit{ in } \mathbb{Z}\,. \end{array}$$

Thus, for instance, we have the following isomorphisms

$$\begin{array}{lll} \sum_{i=0}^{n} \mathtt{M}^i & \cong \mathtt{M}^n & (n \in \mathbb{N} \textit{ even}) \\ \sum_{i=0}^{n} \mathtt{M}^i & \cong 1 + \mathtt{M}^n & (n \in \mathbb{N} \textit{ odd}) \end{array}$$

in $\mathcal{C}[\mathtt{M}]$.

4.2.2 Generic naturals

A linear generic recursive type of the form:

$$\boxed{\mathtt{N} \cong c + \mathtt{N} \qquad (1 \le c \in \mathbb{N})}$$

is referred to as a *generic natural*. For these types, cancellation and representation results can be given directly.

LEMMA 32 (CANCELLATION). *For $p, q \in \mathbb{N}[x]$, if $\deg(p) > \deg(q)$ then $p + c\,q = p$ in $\mathbb{N}[x]/(x = c + x)$ for all $1 \le c \in \mathbb{N}$.*

PROOF. Observe first that for $1 \le n \in \mathbb{N}$, we have that $x^n + c = x^n$ in $\mathbb{N}[x]/(x = c + x)$. Indeed, the assertion is trivially true for $n = 1$, and for $n > 1$, we have that

$$\begin{array}{rcll} x^n + c & = & x^n + c\,x^{n-1} + c & \\ & = & x^n + c\,x^{n-1} & \text{, by induction} \\ & = & x^n & \end{array}$$

and hence that

$$x^n + c\,x^m \quad = \quad x^m(x^{n-m}+c) \quad = \quad x^n$$

for all $0 \leq m < n \in \mathbb{N}$. Thus, for $0 \leq n_i < n \in \mathbb{N}$ with i ranging over a finite set I,

$$x^n + c\sum_{i \in I} x^{n_i} \quad = \quad x^n$$

and the result follows. □

THEOREM 33 (REPRESENTATION). *For $1 \leq c \in \mathbb{N}$, the universal rig homomorphism*

$$\mathbb{N}[x]/(x = c + x) \longrightarrow \mathbb{N}\lceil x\rceil \times \mathbb{Z}_c[x]$$

induced by $(x, x) \in \mathbb{N}\lceil x\rceil \times \mathbb{Z}_c[x]$ is injective.

That is, $t_1 = t_2$ in $\mathbb{N}[x]/(x = c + x)$ iff $h(t_1) = h(t_2)$ and $t_1 = t_2$ in $\mathbb{Z}_c[x]$.

PROOF. ($\Rightarrow$) Use Proposition 24.

($\Leftarrow$) Let $t_1, t_2 \in \mathbb{N}[x]$ with $h(t_1) = h(t_2)$ be such that $t_1 = t_2$ in $\mathbb{Z}_c[x] \cong \mathbb{Z}[x]/(c)$. There exist $q_1, q_2 \in \mathbb{N}[x]$ with $\deg(q_i) \leq \deg(q_1 - q_2) = \deg(t_1 - t_2) < \deg(t_j)$ for $i, j \in \{1, 2\}$ such that $t_1 - t_2 = c\,(q_1 - q_2)$ in $\mathbb{Z}[x]$ and hence such that $t_1 + c\,q_2 = t_2 + c\,q_1$ in $\mathbb{N}[x]$. Thus, since $\deg(t_1) > \deg(q_2)$ and $\deg(t_2) > \deg(q_1)$, it follows by Lemma 32 that $t_1 = t_2$ in $\mathbb{N}[x]/(x = c + x)$. □

PROPOSITION 34. *For $1 \leq c \in \mathbb{N}$, we have the following description: $\mathbb{N}[x]/(x = c + x)$ is isomorphic to*

$$\left\{ \sum_{i=0}^{n} c_i x^i \in \mathbb{N}[x] \mid c_n \neq 0 \text{ and } \forall 1 \leq i < n.\, c_i \in \mathbb{Z}_c \right\}.$$

The representation theorem provides the following decidability test: $t_1 = t_2$ in $\mathbb{N}[x]/(x = c + x)$ iff $h(t_1) = h(t_2)$, and t_1 and t_2 have the same normal form in $\mathbb{N}[x]$ under the reduction rule $c \rightsquigarrow 0$.

COROLLARY 35 (DECIDABILITY). *For $1 \leq c \in \mathbb{N}$, the word problem in $\mathbb{N}[x]/(x = c + x)$, and hence also type isomorphism in $\mathcal{C}[\rho X.\, F]$ where $F \in [x]$ has polynomial form $c + X$, are explicitly decidable.*

PROOF. Given $t_1, t_2 \in \mathbb{N}[x]$, if $h(t_1) \neq h(t_2)$ or c does not divide every coefficient of $t_1 - t_2 \in \mathbb{Z}[x]$ then $t_1 \neq t_2$ in $\mathbb{N}[x]/(x = c + x)$. Otherwise, $t_1 = t_2$ in $\mathbb{N}[x]/(x = c + x)$ and a derivation of this identity can be obtained by expressing $t_1 - t_2$ as $c\,(q_1 - q_2)$ for $q_i \in \mathbb{N}[x]$ $(i = 1, 2)$ as in the proof of Theorem 33 and using Lemma 32 to give a derivation of the form

$$t_1 \;=\; t_1 + c\,q_2 \;=\; t_2 + c\,q_1 \;=\; t_2$$

in $\mathbb{N}[x]/(x = c + x)$. □

4.2.3 Generic constants

For completeness we include the trivial case of *generic constants*:

$$\boxed{\mathtt{C} \cong c \qquad (1 \leq c \in \mathbb{N})}$$

THEOREM 36 (REPRESENTATION). *For $1 \leq c \in \mathbb{N}$, the universal rig homomorphism*

$$\mathbb{N}[x]/(x = c) \longrightarrow \mathbb{N}$$

induced by $c \in \mathbb{N}$ is bijective.

That is, $t_1 = t_2$ in $\mathbb{N}[x]/(x = c)$ iff $t_1(c) = t_2(c)$ in $\mathbb{N}$.

COROLLARY 37 (DECIDABILITY). *For $1 \leq c \in \mathbb{N}$, the word problem in $\mathbb{N}[x]/(x = c)$, and hence also type isomorphism in $\mathcal{C}[\rho X.\, F]$ where $F \in [x]$ has polynomial form c, are explicitly decidable.*

Acknowledgements. The results of Section 4.1, though not the proofs, are from joint work with Tom Leinster [7], to whom I am grateful for our collaboration. I am also grateful to Bill Lawvere for inspiring conversations on Objective Number Theory, and to Jimmie Lawson and Mike Mislove for bringing up the literature on semigroups to my attention.

5 References

[1] M. Abadi and M. P. Fiore. Syntactic considerations on recursive types. In *11th Annual Symposium on Logic in Computer Science*, pages 242–252. IEEE, Computer Society Press, 1996.

[2] R. Amadio and L. Cardelli. Subtyping recursive types. *ACM Transactions on Programming Languages and Systems*, 15(4):575–631, 1993.

[3] A. Blass. Seven trees in one. *Journal of Pure and Applied Algebra*, 103:1–21, 1995.

[4] R. Di Cosmo. *Isomorphisms of types: from λ-calculus to information retrieval and language design*. Birkhauser, 1995.

[5] R. Di Cosmo, F. Pottier, and D. Rémy. Subtyping recursive types modulo associative commutative products. Preprint available on-line, 2003.

[6] M. P. Fiore and T. Leinster. An objective representation of the Gaussian integers. To appear in the *Journal of Symbolic Computation*. (A preliminary preprint appeared in `arXiv:math.RA/0211454`, 2002).

[7] M. P. Fiore and T. Leinster. Objects of categories as complex numbers. To appear in *Advances in Mathematics*. (A preliminary preprint appeared in `arXiv:math.CT/0212377`, 2002).

[8] R. Gates. On the generic solution to $P(X) \cong X$ in distributive categories. *Journal of Pure and Applied Algebra*, 125:191–212, 1998.

[9] J. Gil. Subtyping arithmetical types. In *28th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pages 276–289. ACM, 2001.

[10] J. A. Green. On the structure of semigroups. *Annals of Mathematics*, 54:163–172, 1951.

[11] G. Lallement. *Semigroups and combinatorial applications*. Wiley, 1979.

[12] F. W. Lawvere. Private communication. Workshop on Domain Theory (DOMAIN 2002), Copenhagen, Denmark, July 2002.

[13] J. Palsberg and T. Zhao. Efficient and flexible matching of recursive types. In *15th Annual Symposium on Logic in Computer Science*, pages 388–398. IEEE, Computer Society Press, 2000.

[14] B. Pierce. *Types and Programming Languages*. The MIT Press, 2002.

[15] S. H. Schanuel. Negative sets have Euler characteristic and dimension. In *Proc. Como 1990*, volume 1488 of *Lecture Notes in Mathematics*, pages 379–385. Springer-Verlag, 1991.