Types, Rings, and Games

Wei Chen

Thesis submitted to The University of Nottingham
for the degree of Doctor of Philosophy

July 2012

To my parents.

Abstract

Algebraic equations on complex numbers and functional equations on generating functions are often used to solve combinatorial problems. But the introduction of common arithmetic operators such as subtraction and division always causes panic in the world of objects which are generated from constants by applying products and coproducts. Over the years, researchers have been endeavoring to interpretate some absurd calculations on objects which lead to meaningful combinatorial results.

This thesis investigates connections between *algebraic equations on complex numbers* and *isomorphisms of recursively defined objects*. We are attempting to work out conditions under which isomorphisms between recursively defined objects can be decided by equalities between polynomials on multi-variables with integers as coefficients.

As instances of recursively defined objects in computer science, especially in functional programming languages, *algebraic data types* are adopted as objectives of our research. By studying the algebraic structure of the quotient set of non-trivial[1] polynomial types under the least congruence relation that is generated from a given system of recursive type isomorphisms, we develop a sufficient and necessary condition under which this quotient set forms a *ring*. This is an extension of Fiore and Leinster's work that for a given single recursive type isomorphism, they gave a sufficient condition under which the set of non-trivial polynomial types forms a ring. Further, the *polynomial division*

[1] We consider all type expressions that are generated from 0 (empty type) and 1 (unit type) by applying products and coproducts as trivial polynomial types.

algorithm on multi-variables can be used to decide isomorphisms between non-trivial polynomial types.

On the other hand, combinatorial processes can be extracted from factorizations of polynomials. As an illustration, we invent and study an infinite class of one-person board games, so-called *replacement-set games*. There is a one-dimensional unbounded board which is divided into squares. The aim of these games is to move a checker from the initial square to the final square by using rules that are defined by a multiset of integers. It turns out that every solvable replacement-set game corresponds to a product of *cyclotomic polynomials* with at most one negative coefficient. An algorithm is derived to solve these games. That is, it restores combinatorial processes behind polynomial factorizations on one variable.

This research is interesting because it builds a bridge between applied mathematics and theoretical computer science. We believe that connections between algebraic equations on complex numbers and isomorphisms of recursively defined objects will introduce mature methods in applied mathematics, e.g. Gaussian elimination and Gröbner Basis, into theoretical computer science as bases of descriptions and analysis of recursively defined objects, e.g. data structures, languages, and algorithms. Specifically, some word problems can be decided by polynomial division algorithm on multi-variables. On the other hand, combinatorial explanations of algebraic equations on complex numbers can be extracted from proofs of isomorphisms between recursively defined objects.

However, when primitive recursions are introduced to produce isomorphisms between non-trivial polynomial inductive[2] types, the algebraic properties of the quotient set of non-trivial polynomial inductive types are still unclear. As for two-dimensional replacement-set games, whether there is an algorithm that can produce processes to solve these games is still unknown. Also, the connections between algebraic equations on complex numbers and functional equations on generating functions are obscure. All of these need more investigation in future.

[2] The type expressions are defined by least fixed points.

Acknowledgements

I would like to express great gratitude to my supervisor Prof. Roland C. Backhouse for accepting me as a PhD student and for his patient instruction over the last four years. His experience, knowledge, skills, and encouragement are beacons of my research adventure. His rigorous attitude to science will influence and inspire me for life.

I would also like to thank my internal examiner Dr. Natasha Alechina and external examiner Prof. Marcelo Fiore. I really appreciate Dr. Tom Leinster's comments which make me more productive.

Thanks to my parents for their patience and encouragement. I would also like to thank my friends in Nottingham: Thomas Anberree, Ruibin Bai, Florent Balestrieri, Lin Du, Feng Gu, Bartosz Neuman, Adam Rhodes, Neil Sculthorpe, Maddalena Strumia, Jerry Swan, John Woodward, Weiyan Xiang, and Song Zhang. I appreciate help from João F. Ferreira and Alexandra Ferreira when I arrived in Nottingham.

Contents

Chapter 1

Introduction

Recursively defined objects permeate all of computer science. Research into isomorphisms of recursively defined objects usually results in deeper understanding of their underlying computational and combinatorial models. For very different reasons, isomorphisms of recursively defined objects have been studied. Isomorphic objects are usually cardinality preserving. Hence, *generating functions* [Niv69, Hen74, SS78, GJ83, JBR88, GKP94, Sta97, Sta99] and *functional equations* [Fla85, BLL88, BLL98, FS08] are useful mathematical models to formalize recursively defined objects. The relevant literature is scattered in different branches of computer science. For instance, *Schützenberger's Methodology* [CS63, BLFR01] which determines generating functions of unambiguous context-free languages, *Combinatorial Species* [BLL88, BLL98] which is useful for formalizing data structures in functional equations, and *Flajolet's Symbolic Method* [Fla85, FS08] which translates specifications into functional equations for asymptotic analysis [VF90, Odl95]. These methods are reviewed in section 1.5 as related work.

Another interesting research line was developed by Blass [Bla95] and Fiore and Leinster [Fio04, FL05] from *Schanuel's Problem* [Sch91] and *Lawvere's Remark* [Law91] which is also known as *seven-trees-in-one* [Bla95]. The idea is to build connections between recursively defined objects and *algebraic equations* on *complex numbers*. Fiore and Leinster gave a condition on single recursive type isomorphisms under which the quotient set of non-trivial polynomial types forms a ring. It implies that the polynomial division algorithm on one variable can be used to decide isomorphisms between non-trivial polynomial types. More details are given in sections 1.1 and 1.2.

Following Fiore and Leinster's research, by taking *algebraic data types* [Mal90, Hoo96, BM96] as objectives of our investigation, we extend their results from a single recursive type isomorphism to a system of recursive type isomorphisms. This investigation builds connections between algebraic equation systems on complex numbers and recursive type isomorphism systems. It follows that the problem of deciding isomorphisms between non-trivial polynomial types given by recursive type isomorphism systems can be reduced to the problem of deciding equivalences between polynomials on multi-variables with integers as coefficients. The latter problem is effectively the *ideal membership problem* in computational algebra which can be solved by the polynomial division algorithm on multi-variables [BW98, CLS07]. This contribution is discussed in section 1.3.

Seven-trees-in-one has been turned into a one-person board game, namely *the nuclear pennies game* [Yor07, Pip07a, Pip07b]. As an illustration of the theory we have developed, we invent an infinite class of one-person board games which has the nuclear pennies game as an instance, so-called *replacement-set games* [BCF10, BCF11]. The aim of these games is to move a checker on a board n squares right using replacement rules given by some multiset of integers which represent relative positions. The interesting thing about these games is that they build connections between types and *cyclotomic polynomials* [Isa94, Lan02]. That is, every solvable non-trivial replacement-set corresponds to a product of cyclotomic polynomials with at most one negative coefficient. We also give several *ad-hoc* methods to construct subsets of all solvable non-trivial replacement-sets. As another contribution, a brief explanation about this is given in section 1.4.

1.1 Seven-Trees-In-One

The story started from a remark given by Lawvere [Law91]:

> *I was surprised to note that an isomorphism $x = 1 + x^2$ (leading to complex numbers as Euler characteristics if they don't collapse) always induces an isomorphism $x^7 = x$.*

An appropriate explanation of *Lawvere's Remark* is:

> A *binary tree* is an empty tree (1) or a pair of binary trees (x^2); there is an *isomorphism* between the set of seven-tuples of binary trees (x^7) and the set of binary trees (x).

This is also known as *seven-trees-in-one* as named by Blass. He gave an explicit coding between the set of seven-tuples of binary trees and the set of binary trees in [Bla95]. Notice that x in the above discussion is considered as an *object* not a number. However, it is interesting that if we take $x = 1 + x^2$ as an algebraic equation on *complex numbers*, by solving this equation, we have:

$$x = \cos\frac{\pi}{3} \pm \sin\frac{\pi}{3}i \ .$$

Further,

$$x^7 = x \times x^6 = x \times (\cos\frac{\pi}{3} + \sin\frac{\pi}{3}i)^6 = x \times (\cos 2\pi + \sin 2\pi i) = x \ .$$

It seems that there is a *short cut* to prove seven-trees-in-one by taking objects as complex numbers. But, anyone who wants to do this must *at least* explain the following *strange* phenomenon:

$$x^6 = (\cos\frac{\pi}{3} + \sin\frac{\pi}{3}i)^6 = \cos 2\pi + \sin 2\pi i = 1$$

is true in terms of complex numbers while the set of six-tuples of binary trees is *not* isomorphic to the set of the empty tree because cardinalities of both sides are different.

The cardinality of the set of six-tuples of binary trees is countable infinity while that of the set of the empty tree is one.

1.2 Rings from Quotient Semirings

For clarity, let us use capital letters, e.g. R, S and T, to denote objects and lower case letters, e.g. x, y and z, to denote complex numbers. We use the symbol $=$ for equalities between complex numbers and the symbol $\cong$ for isomorphisms between objects in a *distributive category* (see section 2.2) respectively. Operators $+$ and $\times$ are overloaded to denote respectively addition and multiplication of complex numbers or coproduct and product of objects. Their meanings will be clear from context.

In order to understand seven-trees-in-one and, more generally, connections between complex numbers and *recursively defined objects*, it is necessary to investigate the underlying algebraic structure of all objects which are generated from a finite set of recursively defined objects and the *terminal object* 1 by applying *products* and *coproducts*. This structure is effectively a *quotient semiring* with the *terminal object* 1 and the *initial object* 0 as *unit* and *zero* respectively.

As for seven-trees-in-one, for instance, the collection of all objects generated from T and 1 by applying products and coproducts is a quotient semiring under *the least congruence relation* generated from the isomorphism $T \cong 1 + T^2$. By the congruence relation, we mean an equivalence relation preserving products and coproducts.

In [Gat98], Gates showed that when polynomial $P(T)$ has at least one constant term and at least one nonconstant term, isomorphisms between objects in distributive category subject to $T \cong P(T)$ are decided by equalities in any semiring subject to $T = P(T)$. By using this result, in order to study seven-trees-in-one, we can focus on the semiring $\mathbb{N}[T]$ of all polynomials on T with natural numbers as coefficients with respect to the congruence relation $=_{1+T^2}$ which is generated from the identity $T = 1 + T^2$.

In [Bla95], Blass studied the semiring $\mathbb{N}[T]$ with respect to the identity $T = 1 + T^2$ and observed that $1 + T^3$ plays the role of *zero* for all polynomials in the quotient set $(\mathbb{N}[T] - \mathbb{N})/{=_{1+T^2}}$. Following Blass's research, Fiore and Leinster investigated the semiring $\mathbb{N}[T]$ with respect to the identity $T = 1 + T + T^2$ in [FL04]. They showed that $1 + T^2$ plays the role of zero in the quotient set $(\mathbb{N}[T] - \mathbb{N})/{=_{1+T+T^2}}$.

Fiore and Leinster generalised from these examples. Let α be a type which is generated from T and 1 by applying products and coproducts and suppose α has a term T^n for n at least 2. With respect to the isomorphism $T \cong 1+\alpha$, based on the theory of maximal subgroups within semigroups [Gre51], Fiore and Leinster showed that the quotient set of non-trivial polynomial types forms a *ring* [Fio04, FL05]. That is, *subtraction* is valid in the quotient semiring of non-trivial polynomial types.

Returning to the seven-trees-in-one. The above investigation shows a way to decide the isomorphism $T^7 \cong T$ without bothering to explicitly construct a coding between them. Recall that $1+T^3$ is a zero of polynomials in $(\mathbb{N}[T]-\mathbb{N})/{=_{1+T^2}}$. This leads to the result that T^3 is a *negative unit*. Further, we have that the quotient set $(\mathbb{N}[T]-\mathbb{N})/{=_{1+T^2}}$ has the same algebraic properties as the *quotient ring* $\mathbb{Z}[x]/(x = 1 + x^2)$. It is the quotient set of all polynomials with integers as coefficients under the equivalence relation which is given by the *principal ideal* generated from $x^2 - x + 1$. This quotient ring can be considered as the *ring extension* $\mathbb{Z}[\cos\frac{\pi}{3} \pm \sin\frac{\pi}{3}i]$ on the roots $\cos\frac{\pi}{3} \pm \sin\frac{\pi}{3}i$ of the equation $x = 1 + x^2$ as well. For instance, the equation:

$$x^6 - 1 = (\cos\frac{\pi}{3} \pm \sin\frac{\pi}{3}i)^6 - 1 = \cos 2\pi + \sin 2\pi i - 1 = 0$$

corresponds to the identity:

$$T^6 + T^3 \ =_{1+T^2}\ T^3 \times (1 + T^3) \ =_{1+T^2}\ 1 + T^3\ .$$

That is, 0, 1, and -1 correspond to $1 + T^3$, $2 + T^3$, and T^3 respectively. Then, seven-trees-in-one can be decided by the following factorization on $\mathbb{Z}[x]$:

$$x^7 - x = x \times (x^6 - 1) = x \times (x^3 - 1) \times (x^3 + 1) = x \times (x^3 - 1) \times (x + 1) \times (x^2 - x + 1)\ .$$

This factorization shows that x^6-1 and x^7-x are in the principal ideal generated from $x^2 - x + 1$. We have equations $x^6 = 1$ and $x^7 = x$. Accordingly, we get isomorphisms

$T^6 \cong 1+T^3$ and $T^7 \cong T$. Generally, Fiore and Leinster showed that the *polynomial division algorithm* on $\mathbb{Z}[x]$ can be used to decide isomorphisms between non-trivial polynomial types if subtraction is valid [Fio04, FL05].

From the categorical view, Fiore and Leinster's research gives an answer to the following problem posed by Schanuel [Sch91]:

> *Though ill-posed, the question is suggestive: a good answer should complete the diagram*
>
> $$\begin{tikzcd} S \arrow[r, hook] \arrow[d] & E \arrow[d] \\ \mathbb{N} \arrow[r, hook] & \mathbb{Z} \end{tikzcd}$$
>
> *where S is the category of finite sets; we seek an enlargement E, the isomorphism classes of which should give rise to all integers, rather than just natural numbers.*

That is, subtraction is valid on E, which coincides with Fiore and Leinster's result on types that constructs rings from quotient semirings.

1.3 Rings of Recursive Type Isomorphism Systems

Without loss of generality, as concrete representations of recursively defined objects, we choose *algebraic data types* [Mal90, Hoo96, BM96] as objectives of our investigation.

Fiore and Leinster's investigation is on the algebraic structure of the quotient set of non-trivial polynomial types which are generated from *one* recursively defined type T and the *unit type* 1 with respect to the least congruence relation generated from *one* type isomorphism $T \cong 1+\alpha$. Moreover, when α has a term T^n for n at least 2, there is a *ring isomorphism* between the above quotient set and the *polynomial quotient ring* which is based on the *principal ideal* generated from the polynomial $(T-(1+\alpha))$.

Inspired by Fiore and Leinster's research, it is natural to ask the following question:

> *Under what kind of condition does the quotient set of non-trivial polynomial types which are generated from a finite set $\mathfrak{T}$ of recursively defined types form a ring, with respect to the least congruence relation which is generated from a system $\mathfrak{S}$ of recursive type isomorphisms on $\mathfrak{T}$?*

Let $\cong_{\mathfrak{S}}$ be isomorphisms in the *free distributive category* (see section 2.2) on $\mathfrak{T}$ which is equipped with the system $\mathfrak{S}$. Let $\mathbb{N}[\mathfrak{T}]^{+}/{=_{\mathfrak{S}}}$ be the quotient set of the $\mathbb{N}[\mathfrak{T}] - \mathbb{N}$ under the least congruence relation $=_{\mathfrak{S}}$ which is generated from the system $\mathfrak{S}$. Since all semiring properties can be derived from this category, a straightforward consequence is that for all polynomials p and q in $\mathbb{N}[\mathfrak{T}]^{+}$, $p =_{\mathfrak{S}} q$ implies that $p \cong_{\mathfrak{S}} q$, written as $=_{\mathfrak{S}} \subseteq \cong_{\mathfrak{S}}$[1]. Thus, in order to answer the above question, we can focus on the algebraic structure of the quotient set $\mathbb{N}[\mathfrak{T}]^{+}/{=_{\mathfrak{S}}}$.

As an example, given the following system of recursively defined types:

$$\mathfrak{S} \quad \triangleq \quad \begin{cases} S \;\cong\; 1 + T^2 \; ; \\ T \;\cong\; 1 + S \times T \; , \end{cases}$$

we are interested in the algebraic structure of the quotient set $\mathbb{N}[S,T]^{+}/{=_{\mathfrak{S}}}$. *If the quotient set* $\mathbb{N}[S,T]^{+}/{=_{\mathfrak{S}}}$ *forms a ring*, then there is a *ring isomorphism*[2] between this quotient set and the polynomial quotient ring $\mathbb{Z}[x,y]/{=_{\mathcal{S}}}$ where $=_{\mathcal{S}}$ is the equivalence relation given by the *ideal* $I_{\mathcal{S}}$ which is generated from the algebraic equation system on complex numbers:

$$\mathcal{S} \quad \triangleq \quad \begin{cases} x = 1 + y^2 \; ; \\ y = 1 + x \times y \; . \end{cases}$$

That is, they have the same algebraic properties with regard to equalities and operators

[1] We still don't know whether $\cong_{\mathfrak{S}} \subseteq =_{\mathfrak{S}}$ is true or not. Its proof can be a generalization of Gates' result in [Gat98].

[2] By theorem 2.1.1, there is a ring epimorphism from $\mathbb{Z}[x,y]/{=_{\mathcal{S}}}$ to $\mathbb{N}[S,T]^{+}/{=_{\mathfrak{S}}}$. By the definition of $=_{\mathfrak{S}}$, we have $=_{\mathfrak{S}} \subseteq =_{\mathcal{S}}$. This ensures the ring epimorphism is injective as well.

defined on them, denoted as:

$$(\mathbb{Z}[x,y]/{=_{\mathcal{S}}},\ +,\ \times,\ I_{\mathcal{S}},\ 1{+}I_{\mathcal{S}},\ -1{+}I_{\mathcal{S}}) \longleftrightarrow (\mathbb{N}[S,T]^{+}/{=_{\mathfrak{G}}},\ +,\ \times,\ \overline{\Lambda_{\mathfrak{G}}},\ \overline{1+\Lambda_{\mathfrak{G}}},\ \overline{\eta_{\mathfrak{G}}})$$

where $\overline{\Lambda_{\mathfrak{G}}}$ and $\overline{\eta_{\mathfrak{G}}}$ are respectively equivalence classes of *zero* $\Lambda_{\mathfrak{G}}$ and *negative unit* $\eta_{\mathfrak{G}}$ with respect to the least congruence relation $=_{\mathfrak{G}}$. By this ring isomorphism, we have that for all polynomials p and q in $\mathbb{N}[S,T]^{+}$, $p-q \in I_{\mathcal{S}}$ implies that $p \cong_{\mathfrak{G}} q$.

To answer the above question is the *first* motivation of our research. The significance of this investigation is that it builds *connections* between systems of recursive type isomorphisms and systems of equations on complex numbers. That is, *with respect to defined equalities and operators, problems on types can be solved by taking them as complex numbers.*

Notice that the unit 1 and the zero 0 in $\mathbb{N}[\mathfrak{T}]$ are not in $\mathbb{N}[\mathfrak{T}]^{+}$. Hence, the quotient set $\mathbb{N}[\mathfrak{T}]^{+}/{=_{\mathfrak{G}}}$ inherits all properties of the semiring $\mathbb{N}[\mathfrak{T}]$ except for *zero* and *unit.* In order to answer the question we propose above, the *crucial step* is to construct a zero $\Lambda_{\mathfrak{G}}$ for $\mathbb{N}[\mathfrak{T}]^{+}$ with respect to the least congruence relation $=_{\mathfrak{G}}$. By doing this, we have that $\mathbb{N}[\mathfrak{T}]^{+}/{=_{\mathfrak{G}}}$ is a semiring with zero $\overline{\Lambda_{\mathfrak{G}}}$ and unit $\overline{1+\Lambda_{\mathfrak{G}}}$. Further, $\Lambda_{\mathfrak{G}}$ is so constructed that it is isomorphic to $1+\eta_{\mathfrak{G}}$ for some type $\eta_{\mathfrak{G}}$ in $\mathbb{N}[\mathfrak{T}]^{+}$. It follows that $\mathbb{N}[\mathfrak{T}]^{+}/{=_{\mathfrak{G}}}$ forms a quotient ring with $\overline{\eta_{\mathfrak{G}}}$ as negative unit.

We reproduce Fiore and Leinster's result in sections 3.1 and 3.2 to illustrate the above idea with the assumption on isomorphisms given by Fiore and Leinster in [Fio04, FL05]. That is, if the type α has a term T^n for n at least 2, then the quotient set $\mathbb{N}[T]^{+}/{=_{1+\alpha}}$ with $=_{1+\alpha}$ the least congruence relation generated from the isomorphism $T = 1+\alpha$ forms a ring. To prove that the quotient set $\mathbb{N}[T]^{+}/{=_{1+\alpha}}$ is a ring through constructing a zero $\Lambda_{1+\alpha}$ *simplifies* the proof given by Fiore in [Fio04] which is based on *Green's Relations* [Gre51] within semigroups.

As a *careful* extension, in section 3.3, we investigate mutually recursive type isomorphisms $\mathfrak{G}$ on two types S and T. We show that *if types S and T both generate 1, and they generate each other and one of their recursive definitions has a term with degree at* *least* 2*, then the quotient set* $\mathbb{N}[S,T]^{+}/{=_{\mathfrak{G}}}$ *forms a ring.* Here, for all types p and q, p generates q if and only if $p =_{\mathfrak{G}} q + r$ for some type r. For instance, returning to the previous example:

$$\mathfrak{G} \quad \triangleq \quad \begin{cases} S \cong 1 + T^2 \ ; \\ T \cong 1 + S \times T \ . \end{cases}$$

We have:

$$\begin{aligned} S &=_{\mathfrak{G}} 1 + T^2 =_{\mathfrak{G}} 1 + T \times (1 + S \times T) =_{\mathfrak{G}} 1 + T + T \times S \times T \ ; \\ T &=_{\mathfrak{G}} 1 + S \times T =_{\mathfrak{G}} 1 + S \times (1 + T^2) =_{\mathfrak{G}} 1 + S + S \times T^2 \ . \end{aligned}$$

It follows that $\mathfrak{G}$ satisfies the above condition. Further, the quotient set $\mathbb{N}[S,T]^{+}/{=_{\mathfrak{G}}}$ is a ring. The key step of the proof is the construction of the zero $\Lambda_{\mathfrak{G}}$ which is analogous with the construction of $\Lambda_{1+\alpha}$.

By generalising the above condition on mutually recursive type isomorphisms $\mathfrak{G}$ of two recursively defined types to a condition on systems of recursive type isomorphisms $\mathfrak{S}$ on a finite set $\mathfrak{T}$ of recursively defined types, we develop an algorithm to decide whether the quotient set $\mathbb{N}[\mathfrak{T}]^{+}/{=_{\mathfrak{S}}}$ forms a ring with respect to the least congruence relation generated from $\mathfrak{S}$ in section 3.4. The main results are given in theorem 3.4.1 which is an answer to the question we propose at the beginning of this section.

Notice that if $\mathbb{N}[\mathfrak{T}]^{+}/{=_{\mathfrak{S}}}$ forms a ring, then the *subtraction* is valid on this quotient set. As a reasonable extension, we may ask the following question:

> *How does one construct an extension of* $\mathbb{N}[\mathfrak{T}]^{+}/{=_{\mathfrak{S}}}$ *such that division is valid?*

In categorical view, we want to finish the following diagram:

$$\begin{tikzcd} S \arrow[r, hook] \arrow[d] & E \arrow[r, hook] \arrow[d] & F \arrow[d] \\ \mathbb{N} \arrow[r, hook] & \mathbb{Z} \arrow[r, hook] & \mathbb{Q} \end{tikzcd}$$

which is an extension of *Schanuel's Problem* [Sch91]. That is, we are looking for an enlargement F of category of finite sets which gives rise to all rational numbers.

The exploration starts from a concrete example. Return to the isomorphism $T \cong 1+T^2$. Fiore [private communication, 2010] identified an interesting isomorphism:

$$\mathit{List}\ T \quad \cong^{ind}_{1+T^2} \quad T\ .$$

In words, *the set of all finite lists of binary trees is isomorphic to the set of all binary trees.* Here, we use $\cong^{ind}_{1+T^2}$ to emphasize that the type T is the least fixed point $\mu X\,.\,(1+X\times X)$ and primitive recursions (on T and on $\mathit{List}\ T$) are allowed in the free distributive category on T. In our investigation into this isomorphism, we find that the following isomorphism:

$$\mathit{List}\ T \quad \cong^{ind}_{1+T^2} \quad 1+T^3$$

is true as well. Combining the above two isomorphisms, since $1+T^3$ plays the role of zero in $\mathbb{N}[T]^+/{=_{1+T^2}}$, the isomorphism:

$$T \quad \cong^{ind}_{1+T^2} \quad \mathit{List}\ T \quad \cong^{ind}_{1+T^2} \quad 1+T^3$$

we call *trees-in-zero*. By constructing explicitly functions behind this isomorphism, an appropriate proof of this isomorphism is given in section 3.5.

The interesting thing is that by introducing the *List* type, we can construct multiplicative inverses for non-trivial polynomial inductive types. For instance, considering the type isomorphism $T \cong 1+T^2$, we have that $1+T^3$, $3+T^3$, and T^3 play roles of zero, two, and negative one respectively. The productive inverse of $3+T^3$ is $\mathit{List}\ (1+T^3\times(3+T^3))$ which is verified by the following calculation:

$$\begin{aligned}
& (\mathit{List}\ (1+T^3\times(3+T^3)))\times(3+T^3)\\
\cong^{ind}_{1+T^2}\quad & (\mathit{List}\ (1+T^3+2T^3+T^6))\times(3+T^3)\\
\cong^{ind}_{1+T^2}\quad & (\mathit{List}\ (2T^3+T^6))\times(3+T^3)\\
\cong^{ind}_{1+T^2}\quad & (\mathit{List}\ (T^3+T^3\times(1+T^3)))\times(3+T^3)\\
\cong^{ind}_{1+T^2}\quad & (\mathit{List}\ (T^3))\times(3+T^3)
\end{aligned}$$

$$\begin{aligned}
&\cong^{ind}_{1+T^2} && (List\ (T^3)) + (List\ (T^3)) + (List\ (T^3)) + (List\ (T^3)) \times T^3 \\
&\cong^{ind}_{1+T^2} && 1 + (List\ (T^3)) \times T^3 + (List\ (T^3)) + (List\ (T^3)) \times (1 + T^3) \\
&\cong^{ind}_{1+T^2} && 1 + (List\ (T^3)) \times (1 + T^3) + (List\ (T^3)) \times (1 + T^3) \\
&\cong^{ind}_{1+T^2} && 1 + (1 + T^3)\ .
\end{aligned}$$

However, when the *List* type and primitive recursions are introduced into the free distributive category on the finite set $\mathfrak{T}$ of polynomial inductive types (which are defined as least fixed points) equipped with the system $\mathfrak{S}$ of recursive type isomorphisms, the algebraic structure of this category is still unclear. This needs more investigation.

1.4 Replacement-Set Games

The *second* motivation of our research is to generalize the isomorphism seven-trees-in-one. Notice that $T \cong 1 + T^2$ is not the only type isomorphism which generates the isomorphism $T^n \cong_{1+T^2} T$ for some positive natural number n. For instance, the type isomorphism $T \cong 1 + T + T^2$ generates the isomorphism $T^5 \cong_{1+T+T^2} T$, which was studied in [FL04, FL05, Fio04]. The question is:

> *How does one characterize the complete set of identities $T^k = \beta$ which generate $T^k = T^{n+k}$ for natural numbers k and n?*

In order to answer this question, we introduce an infinite class of one-person board games, so-called replacement-set games, in section 4.2. The aim of these games is to move a checker on a board n squares right according to replacement rules given by some multiset R of integers which corresponds to the identity $T^k = \beta$. For instance, the isomorphism seven-trees-in-one has been turned into *the nuclear pennies game* [Yor07, Pip07a, Pip07b] which is considered as a replacement-set game with *replacement-set* $R = \{\!|\ -1,\ 1\ |\!\}$ and *displacement* $n = 6$.

Notice that to construct the identity $T^k = \beta$ which generates $T^k =_\beta T^{n+k}$ is equivalent to constructing a solvable replacement-set game $(R,\ n)$ where R corresponds to the identity $T^k = \beta$. In section 4.3, we study trivial replacement-set games where $min.R \geqslant 0$ or $max.R \leqslant 0$. In section 4.4, an algorithm is constructed to solve non-trivial replacement-set games where $min.R < 0 < max.R$. Through these investigations, a necessary and sufficient condition on the identity $T^k = \beta$ is given.

It turns out that the solvable non-trivial replacement-sets correspond to products of *cyclotomic polynomials* [Isa94, Lan02] with at most one negative coefficient. This is an answer to the problem we propose. Further, by using properties of cyclotomic polynomials, several infinite classes of solvable non-trivial replacement-sets are explicitly constructed in section 4.5. As far as we are aware, it is still an open problem to explicitly construct the complete set of solvable non-trivial replacement-sets.

1.5 Related Work

Notice that our research is to formalise recursively defined objects as *algebraic equation systems* on *complex numbers*. In this section, let us review three relevant methods: *Schützenberger's Methodology, Combinatorial Species*, and *Flajolet's Symbolic Method*. They are based on *functional equation systems* over *generating functions*. And they translate recursively defined objects to generating functions for different purposes.

1.5.1 Schützenberger's Methodology for Formal Languages

The generating function for formal languages is the formal power series:

$$\langle \Sigma n \ :\ 0 \leqslant n \ :\ f_n x^n \rangle$$

with coefficients f_n as numbers of words of length n. Here, we use the notation

$$\langle \oplus i \ :\ R \ :\ P \rangle$$

for quantifiers (see section 2.5). For instance, the generating function of a^* is:

$$\frac{1}{1-x} \quad = \quad \langle \Sigma n \;:\; 0 \leqslant n \;:\; x^n \rangle \;.$$

That is, the number of words of length n is 1 for all natural numbers n.

Chomsky and Schützenberger discovered the method which translates unambiguous context-free languages into their generating functions [CS63]. It is well known that generating functions for unambiguous regular expressions are *rational generating functions* [Sta97, FS08]. However, not all rational generating functions are generating functions for unambiguous regular expressions. Given a rational generating function, whether it is a generating function for some unambiguous regular expression was investigated in [BLFR01].

Let us focus on *Schützenberger's Methodology.* The idea is: an unambiguous context-free grammar is translated into a system of functional equations over generating functions; by solving this algebraic system, one can get the corresponding generating function. And if the given grammar is regular, then its corresponding algebraic system degenerates into a *linear system.* For example, given the unambiguous regular expression $(aa+b)^*a$, its corresponding state transition system is as following:

$$\begin{cases} L_0 = bL_0 + aL_1 \;; \\ L_1 = aL_0 + 1 \;. \end{cases}$$

By replacing a and b by x, we get the following linear system:

$$\begin{cases} L_0 = xL_0 + xL_1 \;; \\ L_1 = xL_0 + 1 \;. \end{cases}$$

By solving this linear system, we get the generating function for $(aa+b)^*a$. That is,

$$F(x) = L_0 = \frac{x}{1-x-x^2} \;.$$

Notice that $F(x)$ defines a linear recurrence relation. Expanding $F(x)$ by using *power series* [Zor04], we have:

$$\begin{cases} f_0 = 0, & ; \\ f_1 = 1, & ; \\ f_n = f_{n-1} + f_{n-2}, & n \geqslant 2 \;. \end{cases}$$

That is, as for the language defined by the regular expression $(aa+b)^*a$, the *Fibonacci Numbers* are numbers of words of length n for all natural numbers n.

1.5.2 Combinatorial Species for Data Structures

A *species* of structures is a construction F, for each finite set U, to produce a finite set $F[U]$ which is independent of the nature of elements of U. In categorical terms, a species is a functor between categories of finite sets and bijections [BLL98]. Species with operators defined on them, e.g. *addition, multiplication, substitution*, and *differentiation*, construct *species algebra.* For every operator between species, there is a corresponding operator between their generating functions. Thus, specifications of combinatorial structures in species algebra can be translated into their generating functions directly.

For instance, considering the following definition of binary trees: a binary tree is an empty tree or an element followed by a pair of binary trees. The empty tree is interpreted as the *empty set species*, defined as:

$$1[U] \quad \triangleq \quad \begin{cases} \{U\}, & \text{if } U = \varnothing \ ; \\ \varnothing, & \text{if } U \neq \varnothing \ . \end{cases}$$

The element is characterized as the *singleton species*, defined as:

$$X[U] \quad \triangleq \quad \begin{cases} \{U\}, & \text{if } |U| = 1 \ ; \\ \varnothing, & \text{if } |U| \neq 1 \ . \end{cases}$$

Disjoint union and cartesian product are considered as species addition and multiplication respectively. Now, the binary tree can be represented by the functional equation:

$$B \quad = \quad 1 + X \cdot B^2 \ .$$

By solving this equation as a quadratic equation on complex numbers, we have:

$$B(x) = \frac{1 \pm \sqrt{1-4x}}{2} \ .$$

Since coefficients of the power series of $\frac{1+\sqrt{1-4x}}{2}$ contain negative natural numbers, $\frac{1+\sqrt{1-4x}}{2}$ is not the generating function of binary trees. Expanding $\frac{1-\sqrt{1-4x}}{2}$ by using power series, we have:

$$B(x) = \langle\, \Sigma n \ : \ 0 \leqslant n \ : \ \frac{1}{n+1}\binom{2n}{n} x^n \,\rangle \ .$$

The coefficients of the above series are numbers of binary trees with n nodes for all natural numbers n.

The significance of this generalisation is that it enables us to focus on algebraic operators and structures of species, without getting too involved in the details of operators between generating functions. Bergeron et al formalized tree-like data structures in species algebra, such as AVL trees and 2-3 Trees in [BLL98]. However, it is hard to solve the functional equations in a combinatorial sense.

1.5.3 Flajolet's Symbolic Method for Asymptotic Analysis

Analysis of data structures and algorithms involves specifications and asymptotic analysis of combinatorial structures. *The Symbolic Method* was developed by Flajolet to translate specifications into functional equations directly. These functional equations are over generating functions. Through analyzing generating functions, one can characterize statistical properties of data structures and algorithms.

Flajolet observed the relation between structural definitions of combinatorial structures and their functional equations [Fla85]. In his research, combinatorial constructors are *admissible* if they preserve cardinalities. And a *combinatorial class* is a closure set constructed from initial sets by admissible constructors which can be translated into functional equations explicitly. Further, complex analysis methods, for instance, *singularity analysis* and *saddle point analysis*, are applied to evaluate statistical properties of combinatorial structures [VF90, Odl95]. In [FS08], Flajolet and Sedgewick defined a set of elementary operators, e.g. *disjoint union, cartesian product, sequence, cycle, multiset*, and *powerset*, as constructors of admissible combinatorial classes. For every elementary operator, there is a corresponding operator on generating functions. This makes automated asymptotic analysis possible.

1.6 Applications

Data structures are in the center of computer science. Analysis and reasoning of data structures are vital in all aspects of computer science, e.g. program optimization, algorithm design and complexity analysis, and software testing and verification. In order to analyze and reason about data structures, we need to choose appropriate mathematical models to formalize them and to design suitable algebraic operators to manipulate them. Since we usually care about shapes of data structures rather than contents stored in them, functional equations on generating functions, as we have seen in section 1.5, are useful models to specify and manipulate data structures. Our research shows that many recursively defined data structures which are generated from constants by applying cartesian products and disjoint unions can be considered as algebraic equations on complex numbers and can be manipulated as polynomials on multi-variable with integers as coefficients. It follows that methods in applied mathematics can be used to analyze data structures.

To define and decide structural equalites between data structures is a fundamental problem in analysis and reasoning of data structures. Structural equalities are captured by isomorphisms in our research. Our setting for isomorphisms is free distributive category. This generalization ensures that deduced properties are true not only for data structures but also for functions and algorithms which are generated from constants by applying products and coproducts. More interestingly, isomorphisms between them can be automatically extracted from proofs of their corresponding algebraic equations. This is useful for program and data structures transformations that are usually required to preserve some structural properties. The first interesting application of the theory we have developed is that it predicts that there is an algorithm to decide whether a replacement-set game is solvable.

On the other hand, connections we have built between recursive polynomial type isomorphisms and algebraic equations show that it is possible to represent complex numbers and polynomials with integers as coefficients as recursive polynomial types. This gives a clue to prove properties in computable algebra by using automated theorem provers.

Chapter 2

Mathematical Preliminaries

In this chapter, we give a brief introduction to mathematics and notations used in this thesis. The concepts of algebraic structures, e.g. semirings, rings, and ideals, are needed to understand chapter 3. We give their definitions and some theorems without proofs in section 2.1. More information can be found in any textbook of algebra, for instance, [MB99] and [Lan02]. Very small part of knowledge on categories and initial algebra is used to characterize algebraic data types and define functions between types. We list relevant information in sections 2.2 and 2.3. More details can be found in [BM96, Hoo96]. Basic properties of cyclotomic polynomials [Isa94, Lan02] are given in section 2.4 which are used for the construction of solvable replacement-sets. Finally, in section 2.5, some examples are given to explain notations we use in this thesis. Similar notations are used in [Gri98, Kal90, GS94, Bac03].

2.1 Algebraic Structures

Let S be a set which is closed for the binary operator $\oplus$. The structure $(S, \oplus)$ is a *semigroup* if $\oplus$ is associative. That is, for all a, b, and c in S,

$$(a \oplus b) \oplus c \quad = \quad a \oplus (b \oplus c)\ .$$

If there is an element $1_\oplus$ in S satisfying that for all a in S,

$$a \oplus 1_\oplus \quad = \quad a \quad = \quad 1_\oplus \oplus a \ ,$$

then the structure $(S,\ \oplus,\ 1_\oplus)$ is a *monoid.* The element $1_\oplus$ is said to be the *unit* of this monoid. For instance, the set of all natural numbers equipped with the arithmetic addition forms a monoid with 0 as the unit. A monoid is *commutative* if $\oplus$ is commutative. That is, for all a and b in S,

$$a \oplus b \quad = \quad b \oplus a \ .$$

A *group* $(S,\ \oplus,\ 1_\oplus)$ is a monoid satisfying that for every element a in S, there is an element b in S such that

$$a \oplus b \quad = \quad 1_\oplus \quad = \quad b \oplus a \ .$$

We say that b is an *inverse* of a. A group is an *abelian* group if $\oplus$ is commutative. For example, the set of all integers equipped with arithmetic addition forms an abelian group.

A *semiring* $(S,\ \oplus,\ \otimes,\ 1_\oplus,\ 1_\otimes)$ is a set S equipped with two binary operators $\oplus$ and $\otimes$ which satisfies the following clauses:

a. $(S,\ \oplus,\ 1_\oplus)$ is a commutative monoid ;

b. $(S,\ \otimes,\ 1_\otimes)$ is a monoid ;

c. $\otimes$ distributes through $\oplus$, that is, for all a, b, and c in S,

$$\begin{aligned} a \otimes (b \oplus c) \quad &= \quad (a \otimes b) \oplus (a \otimes c) \ ; \\ (b \oplus c) \otimes a \quad &= \quad (b \otimes a) \oplus (c \otimes a) \ ; \end{aligned}$$

d. $1_\oplus$ is the *zero* of $\otimes$, that is, for all a in S,

$$a \otimes 1_\oplus \quad = \quad 1_\oplus \quad = \quad 1_\oplus \otimes a \ .$$

For instance, the set of all natural numbers equipped with arithmetic addition and multiplication forms a semiring, written as $(\mathbb{N},\ +,\ \times,\ 0,\ 1)$. The power set of the set Σ^* of all words generated from an alphabet Σ equipped with set union operator $\cup$ and concatenation operator $\cdot$ forms a semiring with the empty set $\varnothing$ and the set of the empty string ϵ as units of set union and concatenation respectively, denoted by $(\wp(\Sigma^*),\ \cup,\ \cdot,\ \varnothing,\ \{\epsilon\})$. As another example, the structure $(\mathbb{N}[x],\ +,\ \times,\ 0,\ 1)$ of all polynomials on the indeterminate x with natural numbers as coefficients which is equipped with polynomial addition and multiplication is a semiring as well. Generally, given a finite set $\mathfrak{I}$ of indeterminates, we use $\mathbb{N}[\mathfrak{I}]$ for the set of all polynomials which are generated from indeterminates in $\mathfrak{I}$ with natural numbers as coefficients. The structure $(\mathbb{N}[\mathfrak{I}],\ +,\ \times,\ 0,\ 1)$ forms a semiring. For instance, taking $\mathfrak{I}$ to be $\{x,\ y\}$, the polynomial $x^2 + 3xy$ is an element in $\mathbb{N}[\mathfrak{I}]$. A semiring is *commutative* if the binary operator $\otimes$ is commutative.

A *ring* is a semiring satisfying that the structure $(S,\ \oplus,\ 1_\oplus)$ forms an abelian group. The structure $(\mathbb{Z},\ +,\ \times,\ 0,\ 1)$ of all integers with arithmetic addition and multiplication is a ring which has 0, 1, and -1 as zero, unit, and negative unit respectively. Given a commutative ring K, the structure $(K[\mathfrak{I}],\ +,\ \times,\ 0,\ 1)$ of all polynomials on indeterminates in $\mathfrak{I}$ with coefficients in K is a *polynomial ring.*

Given a commutative ring K, an *ideal* I is a subgroup of K satisfying that

$$K \otimes I \subseteq I$$

where $\otimes$ is extended to sets. An ideal is said to be *principal* if it is generated from a fixed element. That is,

$$I = K \otimes \{a\}$$

for some element a in K. For instance, given a polynomial p in $\mathbb{Z}[x]$,

$$I_p \quad \triangleq \quad \mathbb{Z}[x] \otimes \{p\}$$

is a principal ideal of $\mathbb{Z}[x]$. Generally, given a finite set $\mathfrak{P}$ of polynomials in $K[\mathfrak{I}]$ for some commutative ring K and some finite set $\mathfrak{I}$ of indeterminates,

$$I_{\mathfrak{P}} \quad \triangleq \quad K[\mathfrak{I}] \otimes \mathfrak{P}$$

is an ideal of $K[\mathfrak{I}]$. Notice that an ideal I of a commutative ring K defines an equivalence relation $=_I$ on K. That is, for all a and b in K,

$$a \ =_I \ b \quad = \quad a - b \in I \ .$$

Further, the quotient set $K/{=_I}$ is a *quotient ring.* We use the notation $K[\mathfrak{I}]/{=_{\mathfrak{P}}}$ for the *polynomial quotient ring* under the equivalence relation defined by the ideal $I_{\mathfrak{P}}$. Notice that K itself is an ideal as well. The ideal $\{0\}$ and the ideal K are called *improper ideals.*

A *ring morphism* is a function between two rings which preserves $\oplus$ and $\otimes$ and unit. The *kernel* of a ring morphism is the set of elements which are mapped into zero.

Theorem 2.1.1 (Main Theorem on Quotient Ring)**.** For all ring morphisms $f : S \rightarrow R$ with an ideal I a subset of the kernel of f, there is a *unique* ring morphism $f' : S/{=_I} \rightarrow R$ satisfying that $f = f' \circ \rho$ with $\rho : S \rightarrow S/{=_I}$. In particular, if I is equal to the kernel of f, then f' is a *monomorphism.*

2.2 Categories

A *category* $\mathcal{C}$ is a collection of *objects* and *arrows* equipped with the *composite* operator $\circ$. Each arrow f connects two objects A and B which are domain and codomain of f respectively, written as $f : A \rightarrow B$. For all arrows $f : A \rightarrow B$ and $g : B \rightarrow C$, there is an arrow $g \circ f : A \rightarrow C$. For each object A, there is an identity arrow $id_A : A \rightarrow A$. The composite operator $\circ$ is associative and has identity arrows as units.

A *functor* F is a homomorphism between two categories. Given two categories $\mathcal{C}$ and $\mathcal{D}$, the functor $F : \mathcal{C} \rightarrow \mathcal{D}$ maps objects and arrows in $\mathcal{C}$ to objects and arrows in $\mathcal{D}$ respectively and preserves identities and the composite operator. Specifically, for all f :

$A \to B$, there is an arrow $Ff : FA \to FB$. And F satisfies that $F(f \circ g) = Ff \circ Fg$ and $F(id_A) = id_{FA}$. We write Id for the *identity functor*. We use the notation K_A for the *constant functor* whose codomain is a category consisted of only one object A and its identity arrow id_A.

A *terminal object*, written as 1, is an object satisfying that for each object A in $\mathcal{C}$, there is a unique arrow from A to 1. By duality, an *initial object*, denoted by 0, is an object satisfying that for each object A, there is a unique arrow from 0 to A.

A *product* of two objects A and B consists of an object $A \times B$ and two arrows $outl : A \times B \to A$ and $outr : A \times B \to B$ satisfying the universal property: given arrows $f : C \to A$ and $g : C \to B$, there is a unique arrow $f \vartriangle g : C \to A \times B$ such that

$$h = f \vartriangle g \quad \equiv \quad outl \circ h = f \ \wedge \ outr \circ h = g \ .$$

Given a category $\mathcal{C}$ having products for each pair of objects, the *product functor* $\mathcal{C} \times \mathcal{C} \to \mathcal{C}$ is defined by its mapping on arrows as: for all arrows $f : A \to C$ and $g : B \to D$,

$$f \times g \quad \triangleq \quad (f \circ outl) \vartriangle (g \circ outr) \ : \ A \times B \to C \times D \ .$$

A *coproduct* of two objects A and B consists of an object $A + B$ and two arrows $inl : A \to A + B$ and $inr : B \to A + B$ satisfying the universal property: given arrows $f : A \to C$ and $g : B \to C$, there is a unique arrow $f \triangledown g : A + B \to C$ such that

$$h = f \triangledown g \quad \equiv \quad h \circ inl = f \ \wedge \ h \circ inr = g \ .$$

Given a category $\mathcal{C}$ having coproducts for each pair of objects, the *coproduct functor* $\mathcal{C} + \mathcal{C} \to \mathcal{C}$ is defined as: for all arrows $f : A \to C$ and $g : B \to D$,

$$f + g \quad \triangleq \quad (inl \circ f) \triangledown (inr \circ g) \ : \ A + B \to C + D \ .$$

An arrow $f : A \to B$ is an *isomorphism* if there is an arrow $g : B \to A$ satisfying that

$$f \circ g = id_B \quad \wedge \quad g \circ f = id_A \ .$$

A *distributive category* is a category which has initial and terminal objects, binary products and coproducts, and satisfies that for all objects A, B, and C, arrows $0 \to A \times 0$ and $A \times B + A \times C \to A \times (B + C)$ are isomorphisms.

A *free distributive category* is a distributive category whose collection of objects is generated from a collection of objects, initial and terminal objects by binary products and coproducts. The collection of arrows in a free distributive category on a finite set of objects is generated from the following arrows by applying composite:

$$\begin{aligned}
&id_A : A \to A && (\mathit{identity})\\
&!_A : A \to 1 && (\mathit{terminal})\\
&\text{¡}_A : 0 \to A && (\mathit{initial})\\
&outl : A \times B \to A && (\mathit{projection})\\
&outr : A \times B \to B && (\mathit{projection})\\
&inl : A \to A + B && (\mathit{injection})\\
&inr : B \to A + B && (\mathit{injection})\\
&\delta : A \times (B + C) \to A \times B + A \times C && (\mathit{distribution})\\
&\zeta : A \times 0 \to 0 && (\mathit{zero})\\
&f \vartriangle g : C \to A \times B && (\mathit{product})\\
&h \triangledown i : A + B \to C && (\mathit{coproduct})
\end{aligned}$$

with

$$f : C \to A, \quad g : C \to B, \quad h : A \to C, \quad i : A \to B\ .$$

2.3 Initial Algebra

Let $F : \mathcal{C} \to \mathcal{C}$ be an endofunctor on the category $\mathcal{C}$. Given an object A in $\mathcal{C}$, an F-*algebra* on A is an arrow $f : FA \to A$. An F-*homomorphism* from the F-algebra

$f : FA \to A$ to the F-algebra $g : FB \to B$ is an arrow $h : A \to B$ satisfying that

$$h \circ f \quad = \quad g \circ Fh \ .$$

Let $\mathbf{Alg}(F)$ be the category with objects F-algebras and arrows F-homomorphisms. An initial F-algebra α is an initial object in $\mathbf{Alg}(F)$. For all F-algebras f, the arrow from α to f is called the *catamorphism*, denoted by $(\!|f|\!)$, which satisfies the universal property:

$$h \ = \ (\!|f|\!) \quad \equiv \quad h \circ \alpha \ = \ f \circ Fh \ .$$

Let F be a *polynomial endofunctor* which is constructed from identity and constant functors by finite products and coproducts. For instance, the type $\mathbb{B}$ of booleans is the initial $(K_1 + K_1)$-algebra and the type $\mathbb{N}$ of natural numbers is the initial $(K_1 + Id)$-algebra.

2.4 Cyclotomic Polynomials

The m-th *cyclotomic polynomial* is defined as:

$$\Phi.m \quad \triangleq \quad \langle \Pi k \ : \ 0 \leqslant k < m \ \wedge \ k \perp m \ : \ x - e^{\frac{2k\pi}{m} i} \rangle$$

where $k \perp m$ denotes that natural numbers k and m are coprime. The first several cyclotomic polynomials are as following:

$$\Phi.0 = 1 \ ;$$

$$\Phi.1 = x - 1 \ ;$$

$$\Phi.2 = x + 1 \ ;$$

$$\Phi.3 = x^2 + x + 1 \ ;$$

$$\Phi.4 = x^2 + 1 \ ;$$

$$\Phi.5 = x^4 + x^3 + x^2 + x + 1 \ ;$$

$$\Phi.6 = x^2 - x + 1 \ .$$

From this definition, we have that for all positive natural numbers a,

$$x^a - 1 \quad = \quad \langle \Pi k \;:\; 1 \leqslant k \leqslant a \;\wedge\; k \setminus a \;:\; \Phi.k \rangle$$

where $k \setminus a$ denotes that k divides a. With the aid of the *Möbius function* which is defined as:

$$\mu.n = \begin{cases} 0, & \text{if } p^2 \setminus n \text{ for some prime } p\,; \\ (-1)^r, & \text{if } n \text{ is a product of } r \text{ distinct primes ;} \\ 1, & \text{if } n = 1\ , \end{cases}$$

cyclotomic polynomials can be calculated by the following formula:

$$\Phi.m \quad = \quad \langle \Pi k \;:\; 1 \leqslant k \leqslant m \;\wedge\; k \setminus m \;:\; (x^k - 1)^{\mu.(\frac{m}{k})} \rangle\ .$$

For instance,

$$\Phi.6 = \frac{x^6 - 1}{x^3 - 1} \times \frac{x - 1}{x^2 - 1} = x^2 - x + 1\ .$$

For all prime numbers p,

$$\Phi.p \quad = \quad \langle \Sigma k \;:\; 0 \leqslant k < p \;:\; x^k \rangle\ .$$

And for natural numbers m and a,

$$\Phi.(m \times a^2) = [x^a/x]\, \Phi.(m \times a)\ \ .$$

2.5 Notations

We use the following proof notation:

$$\begin{array}{ll} & P \\ = & \{\ \ \text{Why } P \equiv Q\ ?\ \ \} \\ & Q \end{array}$$

where P and Q are predicates and the hint is given in the middle which is surrounded by curly braces.

We use the following notation:

$$\langle \oplus\, i \;:\; R \;:\; P \rangle$$

for the quantifier $\oplus$ where i is a dummy, R is the range of i, and P is the term which depends on the dummy.

The Dijkstra's *guarded command language* is used to formalize algorithms. As an example, in the following program:

```
{   $P$   }
**do**    $-x < y$   $\longrightarrow$   $y \;:=\; x + y$
[]        $y < -x$   $\longrightarrow$   $x \;:=\; x + y$
**od**
{   $Q$   }
```

the predicates P and Q are pre-condition and post-condition respectively, and $-x < y \;\to\; y \;:=\; x + y$ and $y < -x \;\to\; x \;:=\; x + y$ are non-deterministic guarded commands within a loop. Notice that non-deterministic guarded command is different from if-statement. Given a list of non-deterministic guarded commands, if more than one of them is true, then one of them is non-deterministically chosen to be executed. If none of them is true, the result is undefined.

We use the symbol $\triangleq$ to emphasize that its right-hand side is the definition of its left-hand side. The notation $f^{\cup}$ is for the inverse function of f. We use capital letters to denote objects, types, algebraic structures, and functors. Lower case letters are usually used for variables and functions. Greek letters are often used to denote polynomials, morphisms, and specific functions.

Chapter 3

Rings of Recursive Type Isomorphism Systems

Starting from the interesting isomorphism *seven-trees-in-one*, Fiore and Leinster gave a condition on single recursive types under which the set of non-trivial polynomial types forms a ring. In section 3.2, we reproduce Fiore and Leinster's result by constructing a zero for the quotient set of non-trivial polynomial types. The same idea is extended to recursive type isomorphism systems. In section 3.4, we give a sufficient and necessary condition on a given recursive type isomorphism system under which the set of non-trivial types forms a *ring*. The significance of this investigation is not only that its underlying algebraic structure is interesting, but also that it reveals connections between algebraic equation systems and recursive type isomorphism systems. This theory predicts that isomorphisms between types can be decided by the *polynomial division algorithm on multi-variables*. In section 3.5, we investigate another interesting isomorphism, so-called *trees-in-zero*.

3.1 Seven-Trees-In-One

Let us consider the type T of *binary trees*:

$$T \quad \triangleq \quad leaf \ | \ node\ (T,\ T)\ .$$

That is, a binary tree is a leaf or a pair of binary trees. Given constructors $leaf : 1 \to T$ and $node : T \times T \to T$, by using coproduct, this type definition declares the following function:

$$in \quad \triangleq \quad leaf \triangledown node \ : \ 1 + T \times T \to T\ .$$

This function is bijective. Its inverse function $in^{\cup} : T \to 1 + T \times T$ can be defined as:

$$in^{\cup} \circ (leaf \triangledown node) \quad = \quad inl \triangledown inr\ .$$

To get rid of unnecessary details, we write:

(3.1) $\quad T \ \cong \ 1 + T \times T\ .$

It is a surprise that there is an isomorphism between the type of binary trees and the type of seven-tuples of binary trees. That is,

$$T \ \cong \ T^7\ .$$

This is known as *Lawvere's Remark* [Law91] or *seven-trees-in-one* [Bla95].

In order to understand seven-trees-in-one better, let us look at its proof. An important and useful fact was given by Gates in [Gat98]. That is,

Theorem 3.1.1 ([Gat98])**.** Given a polynomial P having at least one constant term and at least one nonconstant term, then for two polynomials Q and R, the following are equivalent:

- $Q(T) = R(T)$ in any semiring such that $P(T) = T$;
- $Q(T) \cong R(T)$ in any distributive category such that $P(T) \cong T$.

Our setting for type isomorphisms is the free distributive category $\mathcal{C}[T]$ on T. Objects in $\mathcal{C}[T]$ are generated from T, 0 (initial object), and 1 (terminal object) by applying binary products and coproducts. Arrows in $\mathcal{C}[T]$ are generated from the following arrows by applying composite:

$$\begin{aligned}
&id_A : A \to A && (\textit{identity})\\
&!_A : A \to 1 && (\textit{terminal})\\
&¡_A : 0 \to A && (\textit{initial})\\
&outl : A \times B \to A && (\textit{projection})\\
&outr : A \times B \to B && (\textit{projection})\\
&inl : A \to A + B && (\textit{injection})\\
&inr : B \to A + B && (\textit{injection})\\
&\delta : A \times (B + C) \to A \times B + A \times C && (\textit{distribution})\\
&\zeta : A \times 0 \to 0 && (\textit{zero})\\
&f \vartriangle g : C \to A \times B && (\textit{product})\\
&h \triangledown i : A + B \to C && (\textit{coproduct})
\end{aligned}$$

with

$$f : C \to A, \quad g : C \to B, \quad h : A \to C, \quad i : A \to B \ .$$

Let P be an object in $\mathcal{C}[T]$. For all objects A and B in $\mathcal{C}[T]$, we say A is isomorphic to B subject to $T \cong P$, written as $A \cong_P B$, if there is an isomorphism between A and B in the category $\mathcal{C}[T]$ equipped with the axiom isomorphism $T \cong P$. That is, let in and $in^{\cup}$ be arrows between T and P, $A \cong_P B$ denotes that there is an isomorphism between A and B which is generated from the above arrows and in and $in^{\cup}$.

In order to use theorem 3.1.1, we introduce the following semiring. Let $\mathbb{N}[T]$ be the set of all polynomials in T with natural numbers as coefficients. The structure $(\mathbb{N}[T], +, \times, 0, 1)$ forms a semiring. Let $=_\beta$ be the least congruence relation on $\mathbb{N}[T]$ generated from the identity $T = \beta$ where β is a polynomial in $\mathbb{N}[T] - \mathbb{N}$ and satisfies that its constant term is not zero. That is, the relation $=_\beta$ is the least relation that includes the pair $(T,\ \beta)$ and is an equivalence relation which is preserved by polynomial products and additions.

With the above definitions, a straightforward consequence of theorem 3.1.1 is:

Corollary 3.1.2. Given β in $\mathbb{N}[T] - \mathbb{N}$ with constant term nonzero,

$$\cong_\beta \quad = \quad =_\beta \ .$$

Specifically, we have:

$$T \ \cong_{1+T^2} \ T^7 \quad \equiv \quad T \ =_{1+T^2} \ T^7 \ .$$

By using semiring properties and the identity $T = 1 + T^2$, we have:

$$(3.2) \qquad T + (1 + T^3) \ =_{1+T^2} \ T \quad \wedge \quad T \times (1 + T^3) \ =_{1+T^2} \ 1 + T^3$$

which follow respectively from

$$\begin{aligned} T + (1 + T^3) \ &=_{1+T^2} \ 1 + T \times (1 + T^2) \\ &=_{1+T^2} \ 1 + T^2 \\ &=_{1+T^2} \ T \end{aligned}$$

and

$$\begin{aligned} T \times (1 + T^3) \ &=_{1+T^2} \ T + T^4 \\ &=_{1+T^2} \ 1 + T^2 + T^4 \\ &=_{1+T^2} \ 1 + T^2 \times (1 + T^2) \\ &=_{1+T^2} \ 1 + T^3 \ . \end{aligned}$$

Let the notation $\mathbb{N}[T]^+$ denote the set $\mathbb{N}[T] - \mathbb{N}$. From the property (3.2), by induction on the structure of polynomials, we have that the polynomial $1 + T^3$ is a *zero* of the quotient set $\mathbb{N}[T]^+/{=_{1+T^2}}$ which is the set of equivalence classes of $\mathbb{N}[T]^+$ under the relation $=_{1+T^2}$. That is,

$$(3.3) \qquad \langle \forall p \ : \ p \in \mathbb{N}[T]^+ \ : \ p + (1 + T^3) \ =_{1+T^2} \ p \ \wedge \ p \times (1 + T^3) \ =_{1+T^2} \ 1 + T^3 \rangle \ .$$

As an example, the idempotence of $1 + T^3$ can be proven as follows:

$$\begin{array}{ll} & (1 + T^3) + (1 + T^3) \\ = & \{ \quad (3.2), \text{ particularly, } T \times (1 + T^3) \ =_{1+T^2} \ 1 + T^3, \text{ twice} \quad \} \\ & (1 + T^3) + T^2 \times (1 + T^3) \\ = & \{ \quad \text{semiring} \quad \} \\ & 1 + T^2 \times (T + (1 + T^3)) \\ = & \{ \quad (3.2), \text{ particularly, } T + (1 + T^3) \ =_{1+T^2} \ T \quad \} \\ & 1 + T^3 \ . \end{array}$$

By using the property (3.3), we prove seven-trees-in-one as follows:

$$\begin{array}{ll} & T \ \cong_{1+T^2} \ T^7 \\ = & \{ \quad \text{corollary 3.1.2} \quad \} \\ & T \ =_{1+T^2} \ T^7 \\ = & \{ \quad \text{Aiming to equalise both sides, we use (3.3) to add } T^4 \times (1 + T^3) \\ & \qquad \text{on the left side and } T \times (1 + T^3) \text{ on the right side.} \quad \} \\ & T + T^4 \times (1 + T^3) \ =_{1+T^2} \ T^7 + T \times (1 + T^3) \\ = & \{ \quad \text{semiring and reflexivity} \quad \} \\ & \mathit{true} \ . \end{array}$$

More interesting, from the property (3.3), we have:

$$\langle \forall p \ : \ p \in \mathbb{N}[T]^+ \ : \ p + T^3 \times p \ =_{1+T^2} \ 1 + T^3 \rangle \ .$$

That is, for all polynomials p in $\mathbb{N}[T]^+$, there is an *additive inverse* $T^3 \times p$. Thus, the quotient set $\mathbb{N}[T]^+/{=_{1+T^2}}$ forms a *ring*. The *zero* and *unit* of this ring are respectively equivalence classes of $1+T^3$ and $1+(1+T^3)$ under the least congruence relation $=_{1+T^2}$.

3.2 Recursive Type Isomorphisms

Generally, let us consider the following recursive type isomorphism:

$$(3.4) \qquad T \;\cong\; 1+\alpha$$

where α is a polynomial in $\mathbb{N}[T]$ with degree at least 2. Following Fiore and Leinster's lead [FL04, Fio04, FL05], in this section, let us show that the quotient set $\mathbb{N}[T]^{+}/{=_{1+\alpha}}$ forms a *ring*. The idea is to construct a zero $\Lambda_{1+\alpha}$ for $\mathbb{N}[T]^{+}/{=_{1+\alpha}}$. And $\Lambda_{1+\alpha}$ is so constructed that it is isomorphic to $1+\eta_{1+\alpha}$ for a polynomial $\eta_{1+\alpha}$ in $\mathbb{N}[T]^{+}$. Since $\eta_{1+\alpha}$ is effectively an *additive inverse* of the unit of $\mathbb{N}[T]^{+}/{=_{1+\alpha}}$, the quotient set $\mathbb{N}[T]^{+}/{=_{1+\alpha}}$ is a ring.

Lemma 3.2.1. There is a polynomial γ in $\mathbb{N}[T]$ such that

$$\alpha \;=_{1+\alpha}\; 1+2\alpha+\alpha^2+\gamma\ .$$

Proof. Notice that α has degree at least 2. Let us rewrite α as T^k+r for k at least 2 and r in $\mathbb{N}[T]$.

$$\begin{array}{ll} & T^k + r \\ =_{1+\alpha} & \{\ (3.4)\ \} \\ & (1+\alpha)^k + r \\ =_{1+\alpha} & \{\ 2 \leqslant k, \text{ by the Binomial Theorem,} \\ & \qquad (1+\alpha)^k \;=\; 1+2\alpha+\alpha^2+\delta, \text{ for some } \delta \text{ in } \mathbb{N}[T].\ \} \\ & 1+2\alpha+\alpha^2+\delta+r \\ =_{1+\alpha} & \{\ \text{renaming, } \gamma \;:=\; \delta+r\ \} \\ & 1+2\alpha+\alpha^2+\gamma\ . \end{array}$$

□

Define

$$(3.5) \qquad \Lambda_{1+\alpha} \;\triangleq\; 1+\alpha+\alpha^2+\gamma\ .$$

We get a crucial lemma.

Lemma 3.2.2. $\alpha + \Lambda_{1+\alpha} \; =_{1+\alpha} \; \alpha$.

Proof. It directly follows from the definition of $\Lambda_{1+\alpha}$ and lemma 3.2.1. □

By using this lemma, it is easy to prove the following properties:

(3.6) $T + \Lambda_{1+\alpha} \; =_{1+\alpha} \; T \quad \wedge \quad T \times \Lambda_{1+\alpha} \; =_{1+\alpha} \; \Lambda_{1+\alpha}$.

That is,

$$
\begin{aligned}
& T + \Lambda_{1+\alpha} \\
=_{1+\alpha} \quad & \{ \quad (3.4) \quad \} \\
& (1+\alpha) + \Lambda_{1+\alpha} \\
=_{1+\alpha} \quad & \{ \quad \text{semiring and lemma 3.2.2} \quad \} \\
& 1+\alpha \\
=_{1+\alpha} \quad & \{ \quad (3.4) \quad \} \\
& T
\end{aligned}
$$

and

$$
\begin{aligned}
& T \times \Lambda_{1+\alpha} \\
=_{1+\alpha} \quad & \{ \quad (3.4) \text{ and semiring} \quad \} \\
& \Lambda_{1+\alpha} + \alpha \times \Lambda_{1+\alpha} \\
=_{1+\alpha} \quad & \{ \quad \text{definition (3.5) of } \Lambda_{1+\alpha} \quad \} \\
& 1 + \alpha + \alpha^2 + \gamma + \alpha \times \Lambda_{1+\alpha} \\
=_{1+\alpha} \quad & \{ \quad \text{semiring} \quad \} \\
& 1 + \alpha + \gamma + \alpha \times (\alpha + \Lambda_{1+\alpha}) \\
=_{1+\alpha} \quad & \{ \quad \text{lemma 3.2.2 and semiring} \quad \} \\
& 1 + \alpha + \alpha^2 + \gamma
\end{aligned}
$$

$$
\begin{array}{ll}
=_{1+\alpha} & \{\ \text{definition (3.5) of } \Lambda_{1+\alpha}\ \} \\
& \Lambda_{1+\alpha}\ .
\end{array}
$$

Notice that the quotient set $\mathbb{N}[T]^{+}/{=_{1+\alpha}}$ inherits all properties from the semiring $\mathbb{N}[T]$ except for the *unit* and *zero*. Using the property (3.6), by induction on the structure of polynomials, we have that $\Lambda_{1+\alpha}$ is a *zero* of $\mathbb{N}[T]^{+}/{=_{1+\alpha}}$. That is,

$$\langle \forall p\ :\ p \in \mathbb{N}[T]^{+}\ :\ p + \Lambda_{1+\alpha}\ =_{1+\alpha}\ p\ \wedge\ p \times \Lambda_{1+\alpha}\ =_{1+\alpha}\ \Lambda_{1+\alpha} \rangle\ .$$

And $1+\Lambda_{1+\alpha}$ is a *unit* of $\mathbb{N}[T]^{+}/{=_{1+\alpha}}$. A straightforward consequence is that $\mathbb{N}[T]^{+}/{=_{1+\alpha}}$ is a semiring. Recalling the definition (3.5) of $\Lambda_{1+\alpha}$, let us define

$$\eta_{1+\alpha} \quad \triangleq \quad \alpha + \alpha^2 + \gamma\ .$$

That is, $1 + \eta_{1+\alpha}\ =_{1+\alpha}\ \Lambda_{1+\alpha}$. Since $\Lambda_{1+\alpha}$ is a zero of $\mathbb{N}[T]^{+}/{=_{1+\alpha}}$, it is easy to see that

$$\langle \forall p\ :\ p \in \mathbb{N}[T]^{+}\ :\ p + \eta_{1+\alpha} \times p\ =_{1+\alpha}\ \Lambda_{1+\alpha} \rangle\ .$$

It follows that

Theorem 3.2.3. Given the identity $T = 1 + \alpha$ with α in $\mathbb{N}[T]$ having degree at least 2, the quotient semiring $\mathbb{N}[T]^{+}/{=_{1+\alpha}}$ forms a ring. The inverse of the unit is the equivalence class of $\eta_{1+\alpha}$ under the least congruence relation $=_{1+\alpha}$.

Return to the identity $T = 1 + T^2$. Recall that $1 + T^3$ and T^3 play the roles of zero and negative unit respectively in $\mathbb{N}[T]^{+}/{=_{1+T^2}}$. We can use polynomials $1 + T^3 + i$ ($i \geqslant 0$) and $T^3 \times (-i)$ ($i < 0$) in $\mathbb{N}[T]^{+}/{=_{1+T^2}}$ to represent integers i. Formally, let $\overline{1 + T^3}$ and $\overline{2 + T^3}$ be equivalence classes of $1 + T^3$ and $2 + T^3$ under the congruence relation $=_{1+T^2}$ respectively. We have the following *ring monomorphism* from the ring $\mathbb{Z}$ to the ring $\mathbb{N}[T]^{+}/{=_{1+T^2}}$:

$$\Theta\ :\ (\mathbb{Z},\ +,\ \times,\ 0,\ 1)\ \hookrightarrow\ (\mathbb{N}[T]^{+}/{=_{1+T^2}},\ +,\ \times,\ \overline{1+T^3},\ \overline{2+T^3})$$

which is defined as: for all integers i in $\mathbb{Z}$,

$$\Theta\,.\,i \quad \triangleq \quad \begin{cases} 1 + T^3 + i, & i \geqslant 0\ ; \\ T^3 \times (-i), & i < 0\ . \end{cases}$$

More interesting, we can specify a monomorphism from a *polynomial quotient ring* to the ring $\mathbb{N}[T]^+/{=_{1+T^2}}$. Specifically, let $=_{x-(1+x^2)}$ be the equivalence relation defined by the *principal ideal* $I_{x-(1+x^2)}$ which is generated from the polynomial $x-(1+x^2)$, that is,

$$I_{x-(1+x^2)} \quad \triangleq \quad \{\, p \times (x-(1+x^2)) \mid p \in \mathbb{Z}[x]\,\}\ .$$

We have that the quotient set $\mathbb{Z}[x]/{=_{x-(1+x^2)}}$ forms a *polynomial quotient ring.* The *kernel* of this quotient ring is the principal ideal $I_{x-(1+x^2)}$. By solving the equation:

$$x-(1+x^2)=0\ ,$$

we have that $x=\cos\frac{\pi}{3}\pm\sin\frac{\pi}{3}i$. Further, the above quotient ring can be considered as the *ring extension*:

$$\mathbb{Z}[\cos\frac{\pi}{3}\pm\sin\frac{\pi}{3}i]$$

of complex numbers $\cos\frac{\pi}{3}\pm\sin\frac{\pi}{3}i$.

Now, elements in the polynomial quotient ring $\mathbb{Z}[x]/{=_{x-(1+x^2)}}$ (or the ring extension $\mathbb{Z}[\cos\frac{\pi}{3}\pm\sin\frac{\pi}{3}i]$) can be represented by elements in the ring $\mathbb{N}[T]^+/{=_{1+T^2}}$ by the following *ring isomorphism*:

$$\begin{aligned}\Theta\ :\ &(\mathbb{Z}[x]/{=_{x-(1+x^2)}},\ +,\ \times,\ \overline{x-(1+x^2)},\ \overline{1+x-(1+x^2)})\\ &\qquad\leftrightarrow\ (\mathbb{N}[T]^+/{=_{1+T^2}},\ +,\ \times,\ \overline{1+T^3},\ \overline{2+T^3})\end{aligned}$$

which is defined as: for all polynomials p and q with natural numbers as coefficients,

$$\Theta\,.\,(\overline{p-q}) \quad \triangleq \quad \overline{[T/x]\,(1+x^3+p+x^3\times q)}\ .$$

From the definition of $=_\beta$, we have that for all polynomials p and q in $\mathbb{N}[T]^+$,

$$p\ =_\beta\ q \quad \Rightarrow \quad (\beta-T)\setminus(p-q)\ .$$

By theorem 2.1.1, Θ is an isomorphism. Hence, the range of Θ has the same algebraic properties as the polynomial quotient ring $\mathbb{Z}[x]/{=_{x-(1+x^2)}}$ or the ring extension $\mathbb{Z}[\cos\frac{\pi}{3}\pm\sin\frac{\pi}{3}i]$ with respect to defined operators and equalities.

From this property, for instance, we have:

$$
\begin{array}{cl}
 & (x-(1+x^2)) \setminus (x^6-1) \\
= & \{ \quad \text{definition of } =_{x-(1+x^2)} \quad \} \\
 & x^6 \;=_{x-(1+x^2)}\; 1 \\
= & \{ \quad \text{ring isomorphism} \quad \} \\
 & \Theta\,.\,x^6 \;=_{1+T^2}\; \Theta\,.\,1 \\
= & \{ \quad \text{definition of } \Theta \quad \} \\
 & 1+T^3+T^6 \;=_{1+T^2}\; 1+T^3+1 \\
= & \{ \quad 1+T^3 \text{ is a zero in } \mathbb{N}[T]^+/{=_{1+T^2}}. \quad \} \\
 & T^6 \;=_{1+T^2}\; 2+T^3 \\
= & \{ \quad \text{corollary 3.1.2} \quad \} \\
 & T^6 \;\cong_{1+T^2}\; 2+T^3\ ,
\end{array}
$$

and

$$
\begin{array}{cl}
 & (x-(1+x^2)) \setminus (x^7-x) \\
= & \{ \quad \text{definition of } =_{x-(1+x^2)} \quad \} \\
 & x^7 \;=_{x-(1+x^2)}\; x \\
= & \{ \quad \text{ring isomorphism} \quad \} \\
 & \Theta\,.\,x^7 \;=_{1+T^2}\; \Theta\,.\,x \\
= & \{ \quad \text{definition of } \Theta \quad \} \\
 & 1+T^3+T^7 \;=_{1+T^2}\; 1+T^3+T \\
= & \{ \quad 1+T^3 \text{ is a zero in } \mathbb{N}[T]^+/{=_{1+T^2}}. \quad \} \\
 & T^7 \;=_{1+T^2}\; T \\
= & \{ \quad \text{corollary 3.1.2} \quad \} \\
 & T^7 \;\cong_{1+T^2}\; T\ .
\end{array}
$$

Generally, given an isomorphism $T \cong 1+\alpha$ with the degree of α at least 2, by theorems 3.2.3 and 2.1.1 and the definition of $=_{1+\alpha}$, we can define the following ring isomorphism:

$$\begin{aligned}\Theta \ : \ & (\mathbb{Z}[T]/{=_{T-(1+\alpha)}},\ +,\ \times,\ \overline{T-(1+\alpha)},\ \overline{1+T-(1+\alpha)}) \\ & \quad \leftrightarrow \ (\mathbb{N}[T]^{+}/{=_{1+\alpha}},\ +,\ \times,\ \overline{\Lambda_{1+\alpha}},\ \overline{\eta_{1+\alpha}})\ .\end{aligned}$$

Combining with corollary 3.1.2, *isomorphisms between non-trivial objects in $\mathcal{C}[T]$ which is equipped with the axiom isomorphism $T \ \cong\ 1+\alpha$ can be decided by polynomial division algorithm on $\mathbb{Z}[x]$.* That is, for all non-trivial objects A and B in $\mathcal{C}[T]$,

$$A \ \cong_{1+\alpha}\ B \quad \equiv \quad (T-(1+\alpha)) \setminus (A-B)\ .$$

This coincides with Fiore and Leinster's result [Fio04, FL05].

3.3 Recursive Type Isomorphisms on Two Types

Notice that the construction of the zero $\Lambda_{1+\alpha}$ is a crucial step to the proof that $\mathbb{N}[T]^{+}/{=_{1+\alpha}}$ is a ring. Can we construct a zero for the quotient set of multi-variable polynomials under the least congruence relation generated from identities on two variables? Specifically, we use the notation $\mathbb{N}[S,T]$ for the set of all multi-variable polynomials in S and T with natural numbers as coefficients. Let α and β be polynomials in $\mathbb{N}[S,T]$ satisfying that α has a term T^m and β has a term S^n with m and n at least 2. Let the symbol $=_{\mathfrak{G}}$ denote the least congruence relation generated from the following identities:

$$\mathfrak{G} \quad \triangleq \quad \begin{cases} S \ = \ 1+\alpha\ ; \\ T \ = \ 1+\beta\ . \end{cases}$$

That is, the relation $=_{\mathfrak{G}}$ is the least congruence relation that includes pairs $(S,\ 1+\alpha)$ and $(T,\ 1+\beta)$, and is preserved by *polynomial products and additions.* Let the notation $\mathbb{N}[S,T]^{+}$ denote the set $\mathbb{N}[S,T]-\mathbb{N}$. We are interested in the structure of the quotient set $\mathbb{N}[S,T]^{+}/{=_{\mathfrak{G}}}$.

Analogous with the construction of $\Lambda_{1+\alpha}$, we have:

Lemma 3.3.1. There is a polynomial γ in $\mathbb{N}[S,T]$ such that

$$\alpha \;=_{\mathfrak{G}}\; 1 + 2\alpha + \alpha^2 + \beta + \beta^2 + \gamma \; .$$

Proof.

α

$=_{\mathfrak{G}}$ { α has a term T^m with m at least 2.
Rewrite α as $T^m + p$ with p in $\mathbb{N}[S,T]$. }

$T^m + p$

$=_{\mathfrak{G}}$ { $T \;=\; 1 + \beta$ }

$(1+\beta)^m + p$

$=_{\mathfrak{G}}$ { $2 \leqslant m$, by the Binomial Theorem,
$(1+\beta)^m \;=\; 1 + 2\beta + \beta^2 + q$ for some q in $\mathbb{N}[S,T]$. }

$1 + 2\beta + \beta^2 + p + q$

$=_{\mathfrak{G}}$ { β has a term S^n with n at least 2.
Rewrite β as $S^n + r$ with r in $\mathbb{N}[S,T]$. }

$1 + \beta + \beta^2 + S^n + p + q + r$

$=_{\mathfrak{G}}$ { $S \;=\; 1 + \alpha$ }

$1 + \beta + \beta^2 + (1+\alpha)^n + p + q + r$

$=_{\mathfrak{G}}$ { $2 \leqslant n$, by the Binomial Theorem,
$(1+\alpha)^n \;=\; 2\alpha + \alpha^2 + s$ for some s in $\mathbb{N}[S,T]$. }

$1 + 2\alpha + \alpha^2 + \beta + \beta^2 + p + q + r + s$

$=_{\mathfrak{G}}$ { renaming, $\gamma \;:=\; p + q + r + s$ }

$1 + 2\alpha + \alpha^2 + \beta + \beta^2 + \gamma \; .$

□

Define

$$(3.7) \qquad \Lambda_{\mathfrak{G}} \;\triangleq\; 1 + \alpha + \alpha^2 + \beta + \beta^2 + \gamma \; .$$

From lemma 3.3.1, we have that

(3.8) $\alpha + \Lambda_{\mathfrak{G}} \ =_{\mathfrak{G}} \ \alpha$.

Symmetrically, we can construct

$$\Lambda'_{\mathfrak{G}} \quad \triangleq \quad 1 + \alpha + \alpha^2 + \beta + \beta^2 + \gamma' \ .$$

with γ' in $\mathbb{N}[S, T]$ satisfying that

(3.9) $\beta + \Lambda'_{\mathfrak{G}} \ =_{\mathfrak{G}} \ \beta$.

Notice that $\Lambda_{\mathfrak{G}} \ =_{\mathfrak{G}} \ \Lambda'_{\mathfrak{G}}$. That is,

$\Lambda_{\mathfrak{G}}$

$=_{\mathfrak{G}}$ { $\Lambda_{\mathfrak{G}}$ has a term β and (3.9). }

$\Lambda_{\mathfrak{G}} + \Lambda'_{\mathfrak{G}}$

$=_{\mathfrak{G}}$ { $\Lambda'_{\mathfrak{G}}$ has a term α and (3.8). }

$\Lambda'_{\mathfrak{G}}$.

It follows that

Lemma 3.3.2. $\alpha + \Lambda_{\mathfrak{G}} \ =_{\mathfrak{G}} \ \alpha \quad \wedge \quad \beta + \Lambda_{\mathfrak{G}} \ =_{\mathfrak{G}} \ \beta$.

By using this lemma and the definition (3.7) of $\Lambda_{\mathfrak{G}}$, we have:

$$S + \Lambda_{\mathfrak{G}} \ =_{\mathfrak{G}} \ S \quad \wedge \quad T + \Lambda_{\mathfrak{G}} \ =_{\mathfrak{G}} \ T \quad \wedge \quad S \times \Lambda_{\mathfrak{G}} \ =_{\mathfrak{G}} \ \Lambda_{\mathfrak{G}} \quad \wedge \quad T \times \Lambda_{\mathfrak{G}} \ =_{\mathfrak{G}} \ \Lambda_{\mathfrak{G}} \ .$$

Let us prove $S + \Lambda_{\mathfrak{G}} \ =_{\mathfrak{G}} \ S$ and $S \times \Lambda_{\mathfrak{G}} \ =_{\mathfrak{G}} \ \Lambda_{\mathfrak{G}}$ as follows:

$S + \Lambda_{\mathfrak{G}}$

$=_{\mathfrak{G}}$ { $S \ = \ 1 + \alpha$ and semiring }

$1 + (\alpha + \Lambda_{\mathfrak{G}})$

$=_{\mathfrak{G}}$ { lemma 3.3.2 }

$1 + \alpha$

$=_{\mathfrak{G}}$ { $S \ = \ 1 + \alpha$ }

S

and

$$
\begin{array}{ll}
& S \times \Lambda_{\mathfrak{G}} \\
=_{\mathfrak{G}} & \{ \quad S \ = \ 1+\alpha \ \text{ and semiring } \quad \} \\
& \Lambda_{\mathfrak{G}} + \alpha \times \Lambda_{\mathfrak{G}} \\
=_{\mathfrak{G}} & \{ \quad \text{definition (3.7) of } \Lambda_{\mathfrak{G}} \quad \} \\
& 1+\alpha+\alpha^2+\beta+\beta^2+\gamma+\alpha\times\Lambda_{\mathfrak{G}} \\
=_{\mathfrak{G}} & \{ \quad \text{semiring} \quad \} \\
& 1+\alpha+\beta+\beta^2+\gamma+\alpha\times(\alpha+\Lambda_{\mathfrak{G}}) \\
=_{\mathfrak{G}} & \{ \quad \text{lemma 3.3.2 and semiring} \quad \} \\
& 1+\alpha+\alpha^2+\beta+\beta^2+\gamma \\
=_{\mathfrak{G}} & \{ \quad \text{definition (3.7) of } \Lambda_{\mathfrak{G}} \quad \} \\
& \Lambda_{\mathfrak{G}} \ .
\end{array}
$$

Similar arguments apply to the proofs of $T + \Lambda_{\mathfrak{G}} \ =_{\mathfrak{G}} \ T$ and $T \times \Lambda_{\mathfrak{G}} \ =_{\mathfrak{G}} \ \Lambda_{\mathfrak{G}}$.

Theorem 3.3.3. Let α and β be polynomials in $\mathbb{N}[S,T]$. Given the following identities:

$$
\mathfrak{G} \quad \triangleq \quad \begin{cases} S \ = \ 1+\alpha \ ; \\ T \ = \ 1+\beta \ . \end{cases}
$$

which satisfies that α has a term T^m and β has a term S^n with m and n at least 2, the quotient set $\mathbb{N}[S,T]^+/{=_{\mathfrak{G}}}$ forms a *ring*. The inverse of the unit is the equivalence class of $\eta_{\mathfrak{G}}$ under the congruence relation $=_{\mathfrak{G}}$ where $\eta_{\mathfrak{G}}$ is defined as:

$$
\eta_{\mathfrak{G}} \quad \triangleq \quad \alpha+\alpha^2+\beta+\beta^2+\gamma \ .
$$

That is, $\Lambda_{\mathfrak{G}} \ =_{\mathfrak{G}} \ 1+\eta_{\mathfrak{G}}$. Moreover,

$$
\langle \forall p \ : \ p \in \mathbb{N}[S,T]^+ \ : \ p+\Lambda_{\mathfrak{G}} \ =_{\mathfrak{G}} \ p \ \wedge \ p\times\Lambda_{\mathfrak{G}} \ =_{\mathfrak{G}} \ \Lambda_{\mathfrak{G}} \ \wedge \ p+\eta_{\mathfrak{G}}\times p \ =_{\mathfrak{G}} \ \Lambda_{\mathfrak{G}} \rangle \ .
$$

Generally, considering the following identities:

$$
\mathfrak{G} \quad \triangleq \quad \begin{cases} S \ = \ p_S \ ; \\ T \ = \ p_T \ , \end{cases}
$$

with p_S and p_T in $\mathbb{N}[S,T]^+$. (Note that if p_S or p_T has degree 0, then $\mathfrak{G}$ degenerates to an identity on one variable or constants.) We are interested in the problem of under what condition the quotient set $\mathbb{N}[S,T]^+/{=_\mathfrak{G}}$ forms a ring since the condition on $\mathfrak{G}$ given in theorem 3.3.3 seems too *strong*. For example, consider the following identities:

$$G \quad \triangleq \quad \begin{cases} S &= \; T^2 \; ; \\ T &= \; 1 + S \; . \end{cases}$$

Notice that

$$\begin{aligned} S \; &=_G \; T^2 \\ &=_G \; (1+S)^2 \\ &=_G \; 1 + 2S + S^2 \\ &=_G \; 1 + S + S^2 + T^2 \end{aligned}$$

and

$$\begin{aligned} T \; &=_G \; 1 + S \\ &=_G \; 1 + T^2 \\ &=_G \; 1 + (1+S)^2 \\ &=_G \; 1 + 1 + 2S + S^2 \; . \end{aligned}$$

Taking α and β to be $S + S^2 + T^2$ and $1 + 2S + S^2$ respectively, the polynomials α and β satisfy that α has a term T^m and β has a term S^n with m and n at least 2. From theorem 3.3.3, we have that the quotient set $\mathbb{N}[S,T]^+/{=_G}$ forms a ring under the least congruence relation $=_G$ generated from G.

We are going to relax the condition in theorem 3.3.3. For our purposes, let us define the binary relation $\rhd_\mathfrak{G}$ on $\mathbb{N}[S,T]$ as: for all p and q in $\mathbb{N}[S,T]$,

$$p \; \rhd_\mathfrak{G} \; q \quad \equiv \quad \langle \exists \; r \; : \; r \in \mathbb{N}[S,T] \; : \; p \; =_\mathfrak{G} \; q + r \rangle \; .$$

We say that p *generates* q with respect to $\mathfrak{G}$. From this definition and properties of the relation $=_\mathfrak{G}$, we have that the relation $\rhd_\mathfrak{G}$ is *reflexive, transitive, and compatible* *with products and additions.* Specifically, for all p, q, r, and s in $\mathbb{N}[S,T]$,

$$\begin{array}{lll} p \rhd_{\mathfrak{G}} p \; ; & & \\ p \rhd_{\mathfrak{G}} q \quad \wedge \quad q \rhd_{\mathfrak{G}} r & \Rightarrow & p \rhd_{\mathfrak{G}} r \; ; \\ p \rhd_{\mathfrak{G}} q \quad \wedge \quad r \rhd_{\mathfrak{G}} s & \Rightarrow & p+r \rhd_{\mathfrak{G}} q+s \quad \wedge \quad p \times r \rhd_{\mathfrak{G}} q \times s \; . \end{array}$$

Motivated by our investigation into the system G in the above example, the condition given in theorem 3.3.3 can be generalized to

$$(3.10) \qquad S \rhd_{\mathfrak{G}} 1 \quad \wedge \quad T \rhd_{\mathfrak{G}} 1 \quad \wedge \quad \langle \exists\, m,\, n \;:\; 2 \leqslant m,\, n \;:\; S \rhd_{\mathfrak{G}} T^m \,\wedge\, T \rhd_{\mathfrak{G}} S^n \rangle \; .$$

Further, we want to show that the condition (3.10) is equivalent to the condition that *S and T both generate the term 1, S and T generate each other, and at least one of p_S and p_T has degree at least* 2. Specifically, let us use deg for the degree of a given polynomial. For instance, $deg\,.\,(1+S^2T) \;=\; 3$. This condition is formalized as:

$$(3.11) \qquad (S \rhd_{\mathfrak{G}} 1 \wedge T \rhd_{\mathfrak{G}} 1) \;\wedge\; (S \rhd_{\mathfrak{G}} T \wedge T \rhd_{\mathfrak{G}} S) \;\wedge\; (2 \leqslant deg\,.\,p_S \vee 2 \leqslant deg\,.\,p_T) \; .$$

That is, our goal is to show that

$$(3.10) \quad \equiv \quad (3.11) \; .$$

Notice that

Lemma 3.3.4.

$$\begin{array}{ll} & (S \rhd_{\mathfrak{G}} T \quad \wedge \quad T \rhd_{\mathfrak{G}} S) \quad \wedge \quad (2 \leqslant deg\,.\,p_S \quad \vee \quad 2 \leqslant deg\,.\,p_T) \\ \Rightarrow & \langle \exists\, m,\, n \;:\; 2 \leqslant m,\, n \;:\; S \rhd_{\mathfrak{G}} T^m \;\wedge\; T \rhd_{\mathfrak{G}} S^n \rangle \; . \end{array}$$

Proof. Suppose that $2 \leqslant deg\,.\,p_S$. Since $S \;=\; p_S$, by the definition of $\rhd_{\mathfrak{G}}$, we have

$$\langle \exists\, a,\, b \;:\; 2 \leqslant a+b \;\wedge\; 0 \leqslant a,\, b \;:\; S \rhd_{\mathfrak{G}} T^a S^b \rangle \; .$$

Further,

$$
\begin{array}{ll}
& S \rhd_{\mathfrak{G}} T \quad \wedge \quad T \rhd_{\mathfrak{G}} S \quad \wedge \quad S \rhd_{\mathfrak{G}} T^a S^b \\
\Rightarrow & \{ \quad \rhd_{\mathfrak{G}} \text{ is transitive and compatible with products. Specifically,} \\
& \qquad T \rhd_{\mathfrak{G}} S \quad \Rightarrow \quad T^a \rhd_{\mathfrak{G}} S^a \quad \Rightarrow \quad T^a S^b \rhd_{\mathfrak{G}} S^{a+b} \, . \quad \} \\
& S \rhd_{\mathfrak{G}} T \quad \wedge \quad T \rhd_{\mathfrak{G}} S \quad \wedge \quad S \rhd_{\mathfrak{G}} S^{a+b} \\
\Rightarrow & \{ \quad \rhd_{\mathfrak{G}} \text{ is transitive and compatible with products. Specifically,} \\
& \qquad S \rhd_{\mathfrak{G}} T \quad \Rightarrow \quad S^{a+b} \rhd_{\mathfrak{G}} T^{a+b} \, . \quad \} \\
& T \rhd_{\mathfrak{G}} S \quad \wedge \quad S \rhd_{\mathfrak{G}} S^{a+b} \quad \wedge \quad S^{a+b} \rhd_{\mathfrak{G}} T^{a+b} \\
\Rightarrow & \{ \quad \text{transitivity and weakening} \quad \} \\
& S \rhd_{\mathfrak{G}} T^{a+b} \quad \wedge \quad T \rhd_{\mathfrak{G}} S^{a+b} \, .
\end{array}
$$

By symmetry, we have that

$$
S \rhd_{\mathfrak{G}} T \quad \wedge \quad T \rhd_{\mathfrak{G}} S \quad \wedge \quad T \rhd_{\mathfrak{G}} T^a S^b \qquad \Rightarrow \qquad S \rhd_{\mathfrak{G}} T^{a+b} \quad \wedge \quad T \rhd_{\mathfrak{G}} S^{a+b} \, .
$$

Recall that $2 \leqslant a+b$. We prove the lemma. □

From lemma 3.3.4,

$$
(3.10) \quad \Leftarrow \quad (3.11) \, .
$$

Notice that

$$
\begin{array}{ll}
& (S \rhd_{\mathfrak{G}} 1 \ \wedge \ T \rhd_{\mathfrak{G}} 1) \quad \wedge \quad \langle \exists\, m, n \ : \ 2 \leqslant m, n \ : \ S \rhd_{\mathfrak{G}} T^m \ \wedge \ T \rhd_{\mathfrak{G}} S^n \rangle \\
\Rightarrow & \{ \quad \rhd_{\mathfrak{G}} \text{ is transitive and compatible with products. Specifically,} \\
& \qquad T \rhd_{\mathfrak{G}} 1 \ \wedge \ 2 \leqslant m \quad \Rightarrow \quad T^{m-1} \rhd_{\mathfrak{G}} 1 \quad \Rightarrow \quad T^m \rhd_{\mathfrak{G}} T \, ; \\
& \qquad S \rhd_{\mathfrak{G}} 1 \ \wedge \ 2 \leqslant n \quad \Rightarrow \quad S^{n-1} \rhd_{\mathfrak{G}} 1 \quad \Rightarrow \quad S^n \rhd_{\mathfrak{G}} S \, . \quad \} \\
& (S \rhd_{\mathfrak{G}} 1 \quad \wedge \quad T \rhd_{\mathfrak{G}} 1) \quad \wedge \quad (S \rhd_{\mathfrak{G}} T \quad \wedge \quad T \rhd_{\mathfrak{G}} S) \, .
\end{array}
$$

Since degrees of p_S and p_T are at least 1,

$$
\neg\, (deg\,.\,p_S \geqslant 2 \vee deg\,.\,p_T \geqslant 2) \quad \Rightarrow \quad \neg\, \langle \exists\, m, n \ : \ 2 \leqslant m, n \ : \ S \rhd_{\mathfrak{G}} T^m \wedge T \rhd_{\mathfrak{G}} S^n \rangle \, .
$$

In words, if degrees of p_S and p_T are 1, then it is impossible to generate $S \;\vartriangleright_{\mathfrak{G}}\; T^m$ or $T \;\vartriangleright_{\mathfrak{G}}\; S^n$ for m and n at least 2. Combining the above discussions, we have that

$$(3.10) \quad \Rightarrow \quad (3.11) \; .$$

Therefore, (3.10) is equivalent to (3.11).

The advantage of (3.11) over (3.10) is that the function *deg* is easier to calculate than the relation $\vartriangleright_{\mathfrak{G}}$. How does one derive an algorithm to decide whether the conjunction $(S \;\vartriangleright_{\mathfrak{G}}\; 1 \;\wedge\; T \;\vartriangleright_{\mathfrak{G}}\; 1) \;\wedge\; (S \;\vartriangleright_{\mathfrak{G}}\; T \;\wedge\; T \;\vartriangleright_{\mathfrak{G}}\; S)$ is true or not? Before we study this problem, let us generalize the condition (3.11) on *identities on two variables* to the condition on *identity systems.*

3.4 Recursive Type Isomorphism Systems

Given a finite set $\mathfrak{T}$ of variables, let us consider the least congruence relation generated from the following system $\mathfrak{G}$ of identities:

$$\mathfrak{G} \quad \triangleq \quad \langle T \;:\; T \in \mathfrak{T} \;\wedge\; p_T \in \mathbb{N}[\mathfrak{T}]^+ \;:\; T \;=\; p_T \rangle$$

where we use the notation $\mathbb{N}[\mathfrak{T}]$ for the set of all multi-variable polynomials in variables from $\mathfrak{T}$ with natural numbers as coefficients. By generalizing the condition (3.11) in section 3.3, we have the condition that *for all types T in $\mathfrak{T}$, T generates the term* 1*, variables in $\mathfrak{T}$ generate each other, and at least one of p_T has degree at least* 2, written as:

$$(3.12) \; \langle \forall\, T \in \mathfrak{T} \;:\;: T \vartriangleright_{\mathfrak{G}} 1 \rangle \;\wedge\; \langle \forall\, T, R \in \mathfrak{T} \;:\;: T \vartriangleright_{\mathfrak{G}} R \rangle \;\wedge\; \langle \exists\, T \in \mathfrak{T} \;:\;: 2 \leqslant deg \,.\, p_T \rangle \; .$$

Generalizing the proof of that (3.10) is equivalent to (3.11), we have that the condition (3.12) is equivalent to

$$(3.13) \; \langle \forall\, T \in \mathfrak{T} \;:\;: T \;\vartriangleright_{\mathfrak{G}}\; 1 \rangle \quad \wedge \quad \langle \forall\, T, R \in \mathfrak{T} \;:\;: \langle \exists\, m \;: 2 \leqslant m :\; T \;\vartriangleright_{\mathfrak{G}}\; R^m \rangle \rangle \; .$$

It follows that for all variables T in $\mathfrak{T}$,

$$T \ \rhd_{\mathfrak{G}} \ 1 + \langle \Sigma \ R, \, m \ : R \in \mathfrak{T} \ \wedge \ 2 \leqslant m : \ R^m \rangle \ .$$

By generalizing the construction of $\Lambda_{\mathfrak{G}}$ in lemma 3.3.1 and the properties of $\Lambda_{\mathfrak{G}}$ in lemma 3.3.2, we can construct a zero $\Lambda_{\mathfrak{G}}$ for the quotient set $\mathbb{N}[\mathfrak{T}]^+/{=_{\mathfrak{G}}}$. The polynomial $\Lambda_{\mathfrak{G}}$ is so constructed that it is isomorphic to $1 + \eta_{\mathfrak{G}}$ for a polynomial $\eta_{\mathfrak{G}}$ in $\mathbb{N}[\mathfrak{T}]^+$. Further,

$$\langle \forall \, p \ : \ p \in \mathbb{N}[\mathfrak{T}]^+ \ : \ p + \Lambda_{\mathfrak{G}} \ =_{\mathfrak{G}} \ p \ \wedge \ p \times \Lambda_{\mathfrak{G}} \ =_{\mathfrak{G}} \ \Lambda_{\mathfrak{G}} \ \wedge \ p + \eta_{\mathfrak{G}} \times p \ =_{\mathfrak{G}} \ \Lambda_{\mathfrak{G}} \rangle \ .$$

Therefore, under the condition (3.12), the quotient set $\mathbb{N}[\mathfrak{T}]^+/{=_{\mathfrak{G}}}$ forms a ring.

Actually, the condition (3.12) is also a *necessary* condition as for that $\mathbb{N}[\mathfrak{T}]^+/{=_{\mathfrak{G}}}$ forms a ring with respect to polynomial addition and product. Suppose that $\mathbb{N}[\mathfrak{T}]^+/{=_{\mathfrak{G}}}$ is a ring. We can find polynomials $\Lambda_{\mathfrak{G}}$ and $\eta_{\mathfrak{G}}$ in $\mathbb{N}[\mathfrak{T}]^+$ which play the roles of zero and negative unit of $\mathbb{N}[\mathfrak{T}]^+/{=_{\mathfrak{G}}}$ respectively. Because for all T in $\mathfrak{T}$,

$$T \ =_{\mathfrak{G}} \ T + \Lambda_{\mathfrak{G}} \ =_{\mathfrak{G}} \ T + 1 + \eta_{\mathfrak{G}} \ ,$$

by the definition of $\rhd_{\mathfrak{G}}$, we have:

$$\langle \forall \, T \in \mathfrak{T} \ : : \ T \ \rhd_{\mathfrak{G}} \ 1 \rangle \ .$$

Notice that the structure $(\mathbb{N}[T]^+/{=_{\mathfrak{G}}}, \ +, \ \overline{\Lambda_{\mathfrak{G}}}, \ \overline{\eta_{\mathfrak{G}}})$ is an *additive group*. We have that for all T and R in $\mathfrak{T}$,

$$T \ =_{\mathfrak{G}} \ R + \eta_{\mathfrak{G}} \times R + T \quad \wedge \quad R \ =_{\mathfrak{G}} \ T + \eta_{\mathfrak{G}} \times T + R \ .$$

By the definition of $\rhd_{\mathfrak{G}}$, we have:

$$\langle \forall \, T, \, R \in \mathfrak{T} \ : : \ T \ \rhd_{\mathfrak{G}} \ R \rangle \ .$$

If the degree of p_T is 1 for all p_T in $\mathfrak{G}$, then for all p and q in $\mathbb{N}[\mathfrak{T}]^+$,

$$p \ =_{\mathfrak{G}} \ q \quad \Rightarrow \quad deg \, . \, p = deg \, . \, q \ .$$

However, since for all T in $\mathfrak{T}$,

$$\Lambda_{\mathfrak{S}} \times T \;=_{\mathfrak{S}}\; \Lambda_{\mathfrak{S}} \;,$$

we have:

$$deg\,.\,(\Lambda_{\mathfrak{S}} \times T) = deg\,.\,\Lambda_{\mathfrak{S}} + 1 \neq deg\,.\,\Lambda_{\mathfrak{S}} \;.$$

By contradiction,

$$\langle \exists\, T \in \mathfrak{T} \; : : \; 2 \leqslant deg\,.\,p_T \rangle \;.$$

Combining the above discussions,

Theorem 3.4.1. Given a system

$$\mathfrak{S} \quad \triangleq \quad \langle T \;:\; T \in \mathfrak{T} \;\wedge\; p_T \in \mathbb{N}[\mathfrak{T}]^+ \;:\; T \;=\; p_T \rangle$$

on the finite set $\mathfrak{T}$ of variables, the quotient set $\mathbb{N}[\mathfrak{T}]^+/{=_{\mathfrak{S}}}$ forms a ring if and only if

$$\langle \forall\, T \in \mathfrak{T} \; : : \; T \rhd_{\mathfrak{S}} 1 \rangle \;\wedge\; \langle \forall\, T, R \in \mathfrak{T} \; : : \; T \rhd_{\mathfrak{S}} R \rangle \;\wedge\; \langle \exists\, T \in \mathfrak{T} \; : : \; 2 \leqslant deg\,.\,p_T \rangle \;.$$

Now, we are going to derive an algorithm to decide the condition (3.12). In order to get better understanding of the condition (3.12), let us consider the following example:

$$H \quad \triangleq \quad \begin{cases} X \;=\; YZ\;; \\ Y \;=\; 1 + Z\;; \\ Z \;=\; X + Y\;. \end{cases}$$

We use $t \lessdot p$ to denote that the *monomial* t is a term of the polynomial p. For instance, $1 \lessdot 2 \times 1 + 2 \times YZ$ and $YZ \lessdot 2 \times 1 + 2 \times YZ$. Let $symb\,.\,t$ be the set of all constants and indeterminates appearing in the monomial t. For example, $symb\,.\,(1) \;=\; \{1\}$ and $symb\,.\,(YZ) \;=\; \{Y, Z\}$. We have the following property: for all variables T,

$$(3.14) \qquad T \rhd_{\mathfrak{S}} 1 \quad \equiv \quad \langle \exists\, t \;:\; t \lessdot p_T \;:\; \langle \forall\, s \;:\; s \in symb\,.\,t \;:\; s \rhd_{\mathfrak{S}} 1 \rangle\rangle \;.$$

That is, T *generates* 1 *if and only if there is a monomial* t *in* p_T *satisfying that all variables appearing in* t *generate* 1.

By using this property, we have that

$$
\begin{array}{ll}
 & Y \;\rhd_H\; 1 \\
= & \quad\{\quad (3.14) \text{ and } Y \;=\; 1 + Z \quad\} \\
 & 1 \;\rhd_H\; 1 \;\vee\; Z \;\rhd_H\; 1 \\
= & \quad\{\quad \rhd_H \text{ is reflexive} \quad\} \\
 & \mathit{true}\ .
\end{array}
$$

Further, let us prove $X \;\rhd_H\; 1$ and $Z \;\rhd_H\; 1$ as following:

$$
\begin{array}{ll}
 & X \;\rhd_H\; 1 \\
= & \quad\{\quad (3.14) \text{ and } X \;=\; YZ \quad\} \\
 & Y \;\rhd_H\; 1 \;\wedge\; Z \;\rhd_H\; 1 \\
= & \quad\{\quad Y \;\rhd_H\; 1 \text{ (from the above proof)} \quad\} \\
 & Z \;\rhd_H\; 1 \\
= & \quad\{\quad (3.14) \text{ and } Z \;=\; X + Y \quad\} \\
 & X \;\rhd_H\; 1 \;\vee\; Y \;\rhd_H\; 1 \\
= & \quad\{\quad Y \;\rhd_H\; 1 \text{ (from the above proof)} \quad\} \\
 & \mathit{true}\ .
\end{array}
$$

Motivated by the property (3.14), let us define the algorithm $\mathbf{E}\,.\,\mathfrak{S}$ as:

Algorithm 3.4.2.

```
$A := \varnothing$
do
    $B := A$
    for each type $T$ in $\mathfrak{T} - A$
        for each monomial $t \preccurlyeq p_T$
            if $\mathit{symb}\,.\,t \subseteq A \;\vee\; \mathit{symb}\,.\,t = \{1\}$
                $A := A \cup \{T\}$; break
while $A \; != \; B$ .
```

We have:

$$(3.15) \qquad \langle \forall\, T \in \mathfrak{T} \;::\; T \rhd_{\mathfrak{G}} 1 \rangle \qquad \equiv \qquad \mathbf{E} \,.\, \mathfrak{G} \;=\; \mathfrak{T} \,.$$

Further, let $\mathbf{G} \,.\, \mathfrak{G}$ be the graph (V, E) which is defined as:

$$V \;=\; \mathfrak{T} \qquad \wedge \qquad (T, R) \in E \quad \equiv \quad \langle \exists\, t \;:\; t \leqslant p_T \;:\; R \in symb \,.\, t \rangle \,.$$

We have:

Lemma 3.4.3. That all variables in $\mathfrak{T}$ generate 1 implies that

$$\langle \forall\, T, R \in \mathfrak{T} \;::\; T \rhd_{\mathfrak{G}} R \rangle \quad \equiv \quad \mathbf{G} \,.\, \mathfrak{G} \text{ is strongly connected} \,.$$

Proof. For all variables T and R in $\mathfrak{T}$, we have that

$$\begin{array}{ll}
 & T \rhd_{\mathfrak{G}} R \\
= & \{ \quad T \;=\; p_T \quad \} \\
 & \langle \exists\, t \;:\; t \leqslant p_T \;:\; t \rhd_{\mathfrak{G}} R \rangle \\
= & \{ \quad \text{the structure of monomials} \quad \} \\
 & \langle \exists\, t \;:\; t \leqslant p_T \;:\; \langle \exists\, S, p, q \;:\; S \in symb \,.\, t \;\wedge\; p, q \in \mathbb{N}[\mathfrak{T}] \;:\; pSq \rhd_{\mathfrak{G}} R \rangle\rangle \\
= & \{ \quad \langle \forall\, T \in \mathfrak{T} \;::\; T \rhd_{\mathfrak{G}} 1 \rangle \quad \Rightarrow \quad p \rhd_{\mathfrak{G}} 1 \quad \wedge \quad q \rhd_{\mathfrak{G}} 1 \quad \} \\
 & \langle \exists\, t \;:\; t \leqslant p_T \;:\; \langle \exists\; S \;:\; S \in symb \,.\, t \;:\; S \rhd_{\mathfrak{G}} R \rangle\rangle \\
= & \{ \quad \text{the definition of } \mathbf{G} \,.\, \mathfrak{G} \quad \} \\
 & (T, S) \in E \quad \wedge \quad S \rhd_{\mathfrak{G}} R \,.
\end{array}$$

That is, $T \rhd_{\mathfrak{G}} R$ is equivalent to that there is a path from T to R in the graph $\mathbf{G} \,.\, \mathfrak{G}$. By the definition of *strongly connected directed graphs*, we prove the lemma. □

Combining (3.15) and lemma 3.4.3, the condition (3.12) is equivalent to

$$(3.16) \qquad \mathbf{E} \,.\, \mathfrak{G} \;=\; \mathfrak{T} \quad \wedge \quad \mathbf{G} \,.\, \mathfrak{G} \text{ is strongly connected} \quad \wedge \quad \langle \exists\, T \in \mathfrak{T} \;::\; 2 \leqslant deg \,.\, p_T \rangle \,.$$

Returning to our example system H. It is easy to show that the graph

$$\mathbf{G}\,.\,H \quad \triangleq \quad (\{X,\,Y,\,Z\},\,\{(X,\,Y),\,(X,\,Z),\,(Y,\,Z),\,(Z,\,X),\,(Z,\,Y)\})$$

is strongly connected. Notice that $\mathbf{E}\,.\,\mathfrak{S} = \{X,\,Y,\,Z\}$ and $deg\,.\,(YZ) = 2$. That is, H satisfies the condition (3.16). Let us show that the quotient set $\mathbb{N}[X,\,Y,\,Z]^{+}/{=_H}$ is a ring through constructing the zero Λ_H. We have that

$$\begin{aligned} Z &=_H X+Y \\ &=_H YZ+Y \\ &=_H (1+Z)Z+Y \\ &=_H Z+Z^2+Y \\ &=_H Z+(X+Y)^2+Y \\ &=_H Z+(X+Y)^2+1+Z \\ &=_H 1+2Z+2XY+X^2+Y^2\ . \end{aligned}$$

By using this derived identity, we get

$$\begin{aligned} X &=_H YZ \\ &=_H (1+Z)Z \\ &=_H Z+Z^2 \\ &=_H 1+2Z+2XY+X^2+Y^2+Z^2 \end{aligned}$$

and

$$\begin{aligned} Y &=_H 1+Z \\ &=_H 1+1+2Z+2XY+X^2+Y^2 \\ &=_H 1+1+2Z+2XY+X^2+(1+Z)^2 \\ &=_H 1+2+4Z+2XY+X^2+Z^2\ . \end{aligned}$$

Taking α, β, and γ to be polynomials $2Z{+}2XY{+}X^2{+}Y^2{+}Z^2$, $2{+}4Z{+}2XY{+}X^2{+}Z^2$, and $2Z+2XY+X^2+Y^2$ respectively. That is, the derived identities

$$\begin{cases} X =_H 1+\alpha\ ; \\ Y =_H 1+\beta\ ; \\ Z =_H 1+\gamma\ . \end{cases}$$

satisfy that α, β, and γ have terms Y^2 and Z^2, X^2 and Z^2, and X^2 and Y^2 respectively. By generalizing lemmas 3.3.1 and 3.3.2, we have that there is a polynomial p in $\mathbb{N}[X,Y,Z]$ satisfying that

$$\Lambda_H \quad \triangleq \quad 1+\alpha+\beta+\gamma+\alpha^2+\beta^2+\gamma^2+p$$

is a zero of the quotient set $\mathbb{N}[X,\ Y,\ Z]^+/{=_H}$. Further, the quotient set $\mathbb{N}[X,\ Y,\ Z]^+/{=_H}$ forms a ring. The inverse of the unit is the equivalence class of

$$\eta_H \quad \triangleq \quad \alpha+\beta+\gamma+\alpha^2+\beta^2+\gamma^2+p$$

under the least congruence relation $=_H$.

In summary, we derive an algorithm to decide the condition (3.12). That is,

Corollary 3.4.4. Given a system $\mathfrak{S}$ of identities on the finite set $\mathfrak{T}$ of variables, the quotient set $\mathbb{N}[\mathfrak{T}]^+/{=_{\mathfrak{S}}}$ forms a ring if and only if

$$\mathbf{E}\,.\,\mathfrak{S} \ = \ \mathfrak{T} \quad \wedge \quad \mathbf{G}\,.\,\mathfrak{S} \text{ is strongly connected} \quad \wedge \quad \langle \exists\, T \in \mathfrak{T} \ :: \ 2 \leqslant deg\,.\,p_T \rangle\ .$$

We now suppose that $\mathfrak{S}$ satisfies the above condition in corollary 3.4.4. Let $I_{\mathfrak{S}}$ and $=_{I_{\mathfrak{S}}}$ be respectively an ideal in $\mathbb{Z}[\mathfrak{T}]$ generated from $\mathfrak{S}$ and the equivalence relation defined by $I_{\mathfrak{S}}$. Let $=_{\mathfrak{S}}$ be the least congruence relation generated from $\mathfrak{S}$. Notice that for all p and q in $\mathbb{N}[\mathfrak{T}]^+$:

$$p \ =_{\mathfrak{S}} \ q \quad \Rightarrow \quad p-q \ \in \ I_{\mathfrak{S}}\ .$$

By theorem 2.1.1, we can define the following ring isomorphism:

$$\Theta \ : \ (\mathbb{Z}[\mathfrak{T}]/{=_{I_{\mathfrak{S}}}},\ +,\ \times,\ I_{\mathfrak{S}},\ 1+I_{\mathfrak{S}}) \ \leftrightarrow \ (\mathbb{N}[\mathfrak{T}]^+/{=_{\mathfrak{S}}},\ +,\ \times,\ \overline{\Lambda_{\mathfrak{S}}},\ \overline{\Lambda_{\mathfrak{S}}+1})\ .$$

That is, for all polynomials p and q in $\mathbb{N}[\mathfrak{T}]$,

$$\Theta\,.\,(p-q) \quad \triangleq \quad \overline{\Lambda_{\mathfrak{S}}+p+\eta_{\mathfrak{S}}\times q}\ .$$

Further, *polynomial division algorithm on* $\mathbb{Z}[\mathfrak{T}]$ *can be used to decide identities in* $\mathbb{N}[\mathfrak{T}]^+/{=_{\mathfrak{S}}}$.

We would like to use polynomial division algorithm to decide isomorphisms $\cong_{\mathfrak{G}}$ in the free distributive category $\mathcal{C}[\mathfrak{T}]$ on $\mathfrak{T}$ which is equipped with axiom isomorphisms:

$$\mathfrak{G} \quad \triangleq \quad \langle T \ : \ T \in \mathfrak{T} \ \wedge \ p_T \in \mathcal{C}[\mathfrak{T}] \ : \ T \ \cong \ p_T \rangle \ .$$

Notice that for all objects A, B, and C in $\mathcal{C}[\mathfrak{T}]$, the following semiring properties are derivable from arrows in $\mathcal{C}[\mathfrak{T}]$.

$$A + (B + C) \ \cong \ (A + B) + C \qquad \text{(associativity of coproduct)}$$

$$A + B \ \cong \ B + A \qquad \text{(symmetry of coproduct)}$$

$$A + 0 \ \cong \ A \ \cong \ 0 + A \qquad \text{(unit of coproduct)}$$

$$A \times (B \times C) \ \cong \ (A \times B) \times C \qquad \text{(associativity of product)}$$

$$A \times B \ \cong \ B \times A \qquad \text{(symmetry of product)}$$

$$A \times 1 \ \cong \ A \ \cong \ 1 \times A \qquad \text{(unit of product)}$$

$$A \times 0 \ \cong \ 0 \ \cong \ 0 \times A \qquad \text{(zero of product)}$$

$$A \times (B + C) \ \cong \ A \times B + A \times C \qquad \text{(distribution)}$$

A straightforward consequence of these properties is:

Corollary 3.4.5 (Soundness). $\quad =_{\mathfrak{G}} \ \subseteq \ \cong_{\mathfrak{G}}$.

Combining with the above ring isomorphism Θ, we have that for all polynomials p and q in $\mathbb{N}[\mathfrak{T}]^+$,

$$p - q \ \in \ I_{\mathfrak{G}} \quad \Rightarrow \quad p \ \cong_{\mathfrak{G}} \ q$$

when $\mathfrak{G}$ satisfies the condition in corollary 3.4.4. That is, polynomial division algorithm on $\mathbb{Z}[\mathfrak{T}]$ can be used to decide isomorphisms $\cong_{\mathfrak{G}}$.

However, we don't know whether the completeness:

$$\cong_{\mathfrak{G}} \quad \subseteq \quad =_{\mathfrak{G}}$$

is true or not when $\mathfrak{G}$ is not a single isomorphism $T \cong P(T)$ such that P satisfies the condition in theorem 3.1.1. Its proof can be a generalization of Gates' result in [Gat98]. Until now, we don't know how to prove it.

On the other hand, when $\mathfrak{G}$ doesn't satisfy the condition in corollary 3.4.4, whether $\cong_{\mathfrak{G}}$ can be decided is not clear. In [Fio04], Fiore discussed linear cases with respect to single recursive types. As for recursive type systems, more investigations are needed.

3.5 Trees-In-Zero

Return to the type T of binary trees, i.e., the least fixed point $\mu X\,.\,(1 + X \times X)$. Let $\mathit{List}\ T$ be the type of all finite lists of T, i.e., the least fixed point $\mu X\,.\,(1 + T \times X)$. In this section, we are going to show that T is isomorphic to $\mathit{List}\ T$ and $\mathit{List}\ T$ is isomorphic to $1 + T^3$. Here, isomorphisms denote that there are inverse functions which are generated from arrows in the free distributive category $\mathcal{C}[T]$ on T equipped with the axiom isomorphism $T \cong 1 + T^2$ by applying functional compositions and primitive recursions (on T or on $\mathit{List}\ T$). We write:

$$T \quad \cong^{\mathit{ind}}_{1+T^2} \quad \mathit{List}\ T \quad \cong^{\mathit{ind}}_{1+T^2} \quad 1 + T^3\ .$$

Since $1 + T^3$ is a zero of $\mathbb{N}[T]^+/{=_{1+T^2}}$, we refer to this isomorphism as *trees-in-zero*.

A crucial property used in our proof of trees-in-zero is:

$$(3.17) \qquad (\mathit{List}\ (T^6)) \times (1 + T^3) \quad \cong^{\mathit{ind}}_{1+T^2} \quad 1 + T^3\ .$$

For clarity, we give its proof in section 3.5.1. Define Σ_n as:

$$\Sigma_n \quad \triangleq \quad \langle\, \Sigma\, i\ :\ 0 \leqslant i < n\ :\ T^i\, \rangle\ .$$

Another useful property is:

$$\mathit{List}\ T \quad \cong^{\mathit{ind}}_{1+T^2} \quad \Sigma_n \times (\mathit{List}\ (T^n))$$

which follows from:

$$\begin{array}{ll} & \mathit{List}\ T \\ \cong^{\mathit{ind}}_{1+T^2} & \{\ \ \text{definition of } \mathit{List}\ T\ \ \} \\ & \mu X\,.\,(1 + T \times X) \\ \cong^{\mathit{ind}}_{1+T^2} & \{\ \ \text{generalization of square rule (specifically, } \mu f \cong \mu(f^n))\ \ \} \end{array}$$

$$
\begin{array}{ll}
& \mu X \,.\, (\Sigma_n + T^n \times X) \\
\cong^{ind}_{1+T^2} & \{ \quad f := (\Sigma_n \times),\ g := (1+) \circ (T^n \times),\ h := (\Sigma_n +) \circ (T^n \times), \\
& \quad \text{Since } f \text{ is a lower adjoint and } f \circ g = h \circ f, \\
& \quad \text{we have that } f \circ \mu g = \mu h. \quad \} \\
& \Sigma_n \times \mu X \,.\, (1 + T^n \times X) \\
\cong^{ind}_{1+T^2} & \{ \quad \text{definition of } \mathit{List}\ (T^n) \quad \} \\
& \Sigma_n \times (\mathit{List}\ (T^n)) \ .
\end{array}
$$

Further, since

$$
\begin{array}{ll}
& \Sigma_n \\
\cong^{ind}_{1+T^2} & \{ \quad \text{definition of } \Sigma_n \quad \} \\
& 1 + T + T^2 + T^3 + T^4 + T^5 \\
\cong^{ind}_{1+T^2} & \{ \quad 1 + T^3 \text{ is a zero of } \mathbb{N}[T]^+/{=_{1+T^2}}. \quad \} \\
& T + T^2 + T^4 + T^5 \\
\cong^{ind}_{1+T^2} & \{ \quad \text{semiring} \quad \} \\
& T + T^4 + (1 + T^3) \times T^2 \\
\cong^{ind}_{1+T^2} & \{ \quad 1 + T^3 \text{ is a zero of } \mathbb{N}[T]^+/{=_{1+T^2}}. \quad \} \\
& T + T^4 \ ,
\end{array}
$$

we get

$$
\mathit{List}\ T \quad \cong^{ind}_{1+T^2} \quad (\mathit{List}\ (T^6)) \times (T + T^4) \ .
$$

Then, by using the property (3.17), we have:

$$
\begin{array}{ll}
& (\mathit{List}\ (T^6)) \times (T + T^4) \\
\cong^{ind}_{1+T^2} & \{ \quad \text{semiring} \quad \} \\
& (\mathit{List}\ (T^6)) \times (1 + T^3) \times T \\
\cong^{ind}_{1+T^2} & \{ \quad 1 + T^3 \text{ is a zero of } \mathbb{N}[T]^+/{=_{1+T^2}}. \quad \} \\
& (\mathit{List}\ (T^6)) \times (1 + T^3)
\end{array}
$$

$\cong^{ind}_{1+T^2} \quad \{ \quad (3.17) \quad \}$

$\quad 1 + T^3$

and

$\quad (List\ (T^6)) \times (T + T^4)$

$\cong^{ind}_{1+T^2} \quad \{ \quad \text{semiring and } List\ T \quad \cong^{ind}_{1+T^2} \quad 1 + T \times List\ T \quad \}$

$\quad T + (List\ (T^6)) \times T^7 + (List\ (T^6)) \times T^4$

$\cong^{ind}_{1+T^2} \quad \{ \quad \text{semiring} \quad \}$

$\quad T + (List\ (T^6)) \times (1 + T^3) \times T^4$

$\cong^{ind}_{1+T^2} \quad \{ \quad 1 + T^3 \text{ is a zero of } \mathbb{N}[T]^+/{=_{1+T^2}}. \quad \}$

$\quad T + (List\ (T^6)) \times (1 + T^3)$

$\cong^{ind}_{1+T^2} \quad \{ \quad (3.17) \quad \}$

$\quad T + (1 + T^3)$

$\cong^{ind}_{1+T^2} \quad \{ \quad 1 + T^3 \text{ is a zero of } \mathbb{N}[T]^+/{=_{1+T^2}}. \quad \}$

$\quad T\ .$

Combining the above results, we get a proof of trees-in-zero.

3.5.1 Catamorphisms

Now, let us prove the property (3.17) by explicitly constructing mutually inverse functions $\llparenthesis f_{T^6} \rrparenthesis$ and $\llparenthesis f_{T^6} \rrparenthesis^\cup$ which are shown in the following diagram:

$$\begin{tikzcd}
(List\ (T^6)) \times (1 + T^3) \arrow[r, shift left, "in^\cup"] \arrow[d, shift left, "\llparenthesis f_{T^6} \rrparenthesis"] & (1 + T^3) + T^6 \times ((List\ (T^6)) \times (1 + T^3)) \arrow[l, shift left, "in"] \arrow[d, shift left, "id_{1+T^3} + id_{T^6} \times \llparenthesis f_{T^6} \rrparenthesis"] \\
1 + T^3 \arrow[u, shift left, "\llparenthesis f_{T^6} \rrparenthesis^\cup"] \arrow[r, shift left, "f^\cup_{T^6}"] & (1 + T^3) + T^6 \times (1 + T^3) \arrow[l, shift left, "f_{T^6}"] \arrow[u, shift left, "id_{1+T^3} + id_{T^6} \times \llparenthesis f_{T^6} \rrparenthesis^\cup"]
\end{tikzcd}$$

Here, functions in, $in^{\cup}$, f_{T^6}, and $f^{\cup}_{T^6}$ are so constructed that they satisfy:

$$\begin{aligned} in \circ in^{\cup} \quad &= \quad id \quad = \quad in^{\cup} \circ in \ ; \\ f_{T^6} \circ f^{\cup}_{T^6} \quad &= \quad id \quad = \quad f^{\cup}_{T^6} \circ f_{T^6} \ . \end{aligned}$$

By using these functions, we can define $(\!| f_{T^6} |\!)$ and $(\!| f_{T^6} |\!)^{\cup}$ respectively as the following *catamorphism* and its inverse:

$$\begin{aligned} (\!| f_{T^6} |\!) \quad &\triangleq \quad f_{T^6} \circ (id_{1+T^3} + id_{T^6} \times (\!| f_{T^6} |\!)) \circ in^{\cup} \ ; \\ (\!| f_{T^6} |\!)^{\cup} \quad &\triangleq \quad in \circ (id_{1+T^3} + id_{T^6} \times (\!| f_{T^6} |\!)^{\cup}) \circ f^{\cup}_{T^6} \ . \end{aligned}$$

Since $(\!| f_{T^6} |\!)$ and $(\!| f_{T^6} |\!)^{\cup}$ are recursively defined, we need to show that they all *terminate*. The terminations of $(\!| f_{T^6} |\!)$ and $(\!| f_{T^6} |\!)^{\cup}$ are decided by the definitions of $in^{\cup}$ and $f^{\cup}_{T^6}$ respectively.

Specifically, from the definition of the star operator, by *explicitly* defining constructors

$$one \ : \ 1 + T^3 \rightarrow (List\ (T^6)) \times (1 + T^3)$$

and

$$pcons \ : \ T^6 \times ((List\ (T^6)) \times (1 + T^3)) \rightarrow (List\ (T^6)) \times (1 + T^3) \ ,$$

functions in and $in^{\cup}$ can be defined as:

$$\begin{aligned} in \quad &\triangleq \quad one \triangledown pcons \ ; \\ in^{\cup} \circ (one \triangledown pcons) \quad &\triangleq \quad inl \triangledown inr \ . \end{aligned}$$

They satisfy:

$$in \circ in^{\cup} = id = in^{\cup} \circ in \ .$$

Define the bound function

$$length \ : \ (List\ (T^6)) \times (1 + T^3) \rightarrow \mathbb{N}$$

as:

$$\begin{aligned} length \quad (one\ a) \quad &= \quad zero \\ length \quad (pcons\ (p,\ ps)) \quad &= \quad succ\ (length \ \ ps) \ . \end{aligned}$$

Because the function $in^{\cup}$ decreases the length of its input, we have that *the length of the input of* $(\!| f_{T^6} |\!)$ *is decreasing after each unfolding.* Further, $(\!| f_{T^6} |\!)$ terminates when the length of its input is zero.

Since $1 + T^3$ is a zero of $\mathbb{N}[T]^+/{\cong_{1+T^2}}$, we have that

$$1 + T^3 \ \cong_{1+T^2}\ (1 + T^3) + T^6 \times (1 + T^3) \ .$$

Functions f_{T^6} and $f^{\cup}_{T^6}$ can be constructed from one of proofs of this isomorphism. For instance, suppose that we have the following functions:

$$\begin{aligned}
&tn\_prod \ : \ (\,n : \mathbb{N}^+) \to T^n \times (1 + T^3) \to 1 + T^3 \ ;\\
&tn\_prod^{\cup} \ : \ (\,n : \mathbb{N}^+) \to 1 + T^3 \to T^n \times (1 + T^3) \ ;\\
&idem\_n \ : \ (\,n : \mathbb{N}^+) \to n \times (1 + T^3) \to 1 + T^3 \ ;\\
&idem\_n^{\cup} \ : \ (\,n : \mathbb{N}^+) \to 1 + T^3 \to n \times (1 + T^3) \ .
\end{aligned}$$

And they satisfy that for all n in $\mathbb{N}^+$,

$$\begin{aligned}
&(tn\_prod\ n) \ \circ \ (tn\_prod^{\cup}\ n) \quad = \quad id \quad = \quad (tn\_prod^{\cup}\ n) \ \circ \ (tn\_prod\ n) \ ;\\
&(idem\_n\ n) \ \circ \ (idem\_n^{\cup}\ n) \quad = \quad id \quad = \quad (idem\_n^{\cup}\ n) \ \circ \ (idem\_n\ n) \ .
\end{aligned}$$

From the following proof:

$$\begin{aligned}
&\qquad (1 + T^3) + T^6 \times (1 + T^3)\\
\cong_{1+T^2}&\qquad \{ \quad (id_{1+T^3} + (tn\_prod \ \ 6)) \ \text{ and } \ (id_{1+T^3} + (tn\_prod^{\cup} \ \ 6)) \quad \}\\
&\qquad (1 + T^3) + (1 + T^3)\\
\cong_{1+T^2}&\qquad \{ \quad (idem\_n \ \ 2) \ \text{ and } \ (idem\_n^{\cup} \ \ 2) \quad \}\\
&\qquad 1 + T^3 \ ,
\end{aligned}$$

functions f_{T^6} and $f^{\cup}_{T^6}$ can be constructed as:

$$\begin{aligned}
f_{T^6} \quad &\triangleq \quad (idem\_n \ \ 2) \ \circ \ (id_{1+T^3} + (tn\_prod \ \ 6)) \ ;\\
f^{\cup}_{T^6} \quad &\triangleq \quad (id_{1+T^3} + (tn\_prod^{\cup} \ \ 6)) \ \circ \ (idem\_n^{\cup} \ \ 2) \ .
\end{aligned}$$

Functions appearing in hints of the above proof work as witnesses. Details about them are given in the following sections.

3.5.2 Product-Zero Functions

Firstly, from the following proof:

$$\begin{array}{ll}
 & T \times (1 + T^3) \\
\cong_{1+T^2} & \{ \quad \text{semiring} \quad \} \\
 & T + T^4 \\
\cong_{1+T^2} & \{ \quad T \;\cong\; 1 + T^2 \quad \} \\
 & (1 + T^2) + T^4 \\
\cong_{1+T^2} & \{ \quad \text{semiring} \quad \} \\
 & 1 + T^2 \times (1 + T^2) \\
\cong_{1+T^2} & \{ \quad T \;\cong\; 1 + T^2 \quad \} \\
 & 1 + T^3 \; ,
\end{array}$$

we can construct the function

$$t\_prod \;:\; T \times (1 + T^3) \rightarrow 1 + T^3$$

and its inverse

$$t\_prod^{\cup} \;:\; 1 + T^3 \rightarrow T \times (1 + T^3)$$

as

$$\begin{array}{lll}
t\_prod \quad (leaf,\ inl\ \bullet) & = & inl\ \bullet \\
t\_prod \quad (node\ (a,\ b),\ inl\ \bullet) & = & inr\ (a,\ b,\ leaf) \\
t\_prod \quad (a,\ inr\ (b,\ c,\ d)) & = & inr\ (a,\ b,\ node\ (c,\ d))
\end{array}$$

and

$$\begin{array}{lll}
t\_prod^{\cup} \quad (inl\ \bullet) & = & (leaf,\ inl\ \bullet) \\
t\_prod^{\cup} \quad (inr\ (a,\ b,\ leaf)) & = & (node\ (a,\ b),\ inl\ \bullet) \\
t\_prod^{\cup} \quad (inr\ (a,\ b,\ node\ (c,\ d))) & = & (a,\ inr\ (b,\ c,\ d))
\end{array}$$

respectively. Notice that these functions are effectively *composite functions* which are generated from *the identity function, functions corresponding to semiring properties, and given functions behind the isomorphism $T \cong 1 + T^2$ by applying finite function compositions, products, and coproducts.* For example, let

$$\begin{aligned} s1 &: T \times (1 + T^3) \to T + T^4 \\ s1^{\cup} &: T + T^4 \to T \times (1 + T^3) \end{aligned}$$

and

$$\begin{aligned} s2 &: (1 + T^2) + T^4 \to 1 + T^2 \times (1 + T^2) \\ s2^{\cup} &: 1 + T^2 \times (1 + T^2) \to (1 + T^2) + T^4 \end{aligned}$$

be respectively functions corresponding to steps in the above proof which have "semiring" as hints. Let

$$f \; : \; 1 + T^2 \to T$$

and

$$f^{\cup} \; : \; T \to 1 + T^2$$

be given functions behind the isomorphism $T \cong 1 + T^2$. Functions t_prod and $t\_prod^{\cup}$ can be defined as the following function compositions:

$$\begin{aligned} t\_prod &\triangleq (id_1 + id_{T^2} \times f) \circ s2 \circ (f^{\cup} + id_{T^4}) \circ s1 \; ; \\ t\_prod^{\cup} &\triangleq s1^{\cup} \circ (f + id_{T^4}) \circ s2^{\cup} \circ (id_1 + id_{T^2} \times f^{\cup}) \; . \end{aligned}$$

Name the constructors of T after

$$leaf \triangleq f \circ inl$$

and

$$node \triangleq f \circ inr$$

respectively. Then, simplify the above definitions. The resulting functions are as same as we have shown before. For the convenience of the termination proof of $(\!| f_{T^6} |\!)^{\cup}$, we prefer to use their explicit definitions.

Further, by using t_prod and $t\_prod^{\cup}$ as bases, we can recursively define functions

$$tn\_prod \ : \ (n : \mathbb{N}^{+}) \rightarrow T^{n} \times (1 + T^{3}) \rightarrow 1 + T^{3}$$

and

$$tn\_prod^{\cup} \ : \ (n : \mathbb{N}^{+}) \rightarrow 1 + T^{3} \rightarrow T^{n} \times (1 + T^{3})$$

as

$$\begin{array}{lcl} tn\_prod \ \ (succ\ zero) & = & t\_prod \\ tn\_prod \ \ (succ\ n) & = & t\_prod \ \circ \ (id_{T} \times (tn\_prod\ n)) \end{array}$$

and

$$\begin{array}{lcl} tn\_prod^{\cup} \ \ (succ\ zero) & = & t\_prod^{\cup} \\ tn\_prod^{\cup} \ \ (succ\ n) & = & (id_{T^{n}} \times t\_prod^{\cup}) \ \circ \ (tn\_prod^{\cup}\ n) \end{array}$$

respectively. By mathematical induction on n in the above definitions, we have that for all n in $\mathbb{N}^{+}$,

$$(tn\_prod \ \ n) \ \circ \ (tn\_prod^{\cup} \ \ n) \quad = \quad id \quad = \quad (tn\_prod^{\cup} \ \ n) \ \circ \ (tn\_prod \ \ n) \ .$$

3.5.3 Idempotence Functions

Similarly, from the following proof:

$$\begin{array}{ll} & (1 + T^{3}) + (1 + T^{3}) \\ \cong_{1+T^{2}} & \{ \ \text{semiring and } T \ \cong \ 1 + T^{2} \ \} \\ & (1 + T^{3}) + (1 + T^{2} + T^{4}) \\ \cong_{1+T^{2}} & \{ \ \text{semiring and } T \ \cong \ 1 + T^{2} \ \} \\ & 1 + T^{3} + T + T^{4} \\ \cong_{1+T^{2}} & \{ \ \text{semiring and } T \ \cong \ 1 + T^{2} \ \} \\ & 1 + T^{2} + T^{4} \\ \cong_{1+T^{2}} & \{ \ \text{semiring and } T \ \cong \ 1 + T^{2} \ \} \\ & 1 + T^{3} \ , \end{array}$$

we can construct the function

$$\mathit{idem} \;:\; (1 + T^3) + (1 + T^3) \to 1 + T^3$$

and its inverse

$$\mathit{idem}^{\cup} \;:\; 1 + T^3 \to (1 + T^3) + (1 + T^3)$$

as

$$\begin{aligned}
&\mathit{idem} \quad (\mathit{inl}\ (\mathit{inl}\ \bullet)) \quad = \quad \mathit{inl}\ \bullet \\
&\mathit{idem} \quad (\mathit{inl}\ (\mathit{inr}\ (a,\ b,\ c))) \quad = \quad \mathit{inr}\ (a,\ \mathit{node}(b,\ c),\ \mathit{leaf}) \\
&\mathit{idem} \quad (\mathit{inr}\ (\mathit{inl}\ \bullet)) \quad = \quad \mathit{inr}\ (\mathit{leaf},\ \mathit{leaf},\ \mathit{leaf}) \\
&\mathit{idem} \quad (\mathit{inr}\ (\mathit{inr}\ (a,\ b,\ \mathit{leaf}))) \quad = \quad \mathit{inr}\ (\mathit{node}\ (a,\ b),\ \mathit{leaf},\ \mathit{leaf}) \\
&\mathit{idem} \quad (\mathit{inr}\ (\mathit{inr}\ (a,\ b,\ \mathit{node}\ (c,\ d)))) \quad = \quad \mathit{inr}\ (a,\ b,\ \mathit{node}\ (c,\ d))
\end{aligned}$$

and

$$\begin{aligned}
&\mathit{idem}^{\cup} \quad (\mathit{inl}\ \bullet) \quad = \quad \mathit{inl}\ (\mathit{inl}\ \bullet) \\
&\mathit{idem}^{\cup} \quad (\mathit{inr}\ (a,\ \mathit{node}(b,\ c),\ \mathit{leaf})) \quad = \quad \mathit{inl}\ (\mathit{inr}\ (a,\ b,\ c)) \\
&\mathit{idem}^{\cup} \quad (\mathit{inr}\ (\mathit{leaf},\ \mathit{leaf},\ \mathit{leaf})) \quad = \quad \mathit{inr}\ (\mathit{inl}\ \bullet) \\
&\mathit{idem}^{\cup} \quad (\mathit{inr}\ (\mathit{node}\ (a,\ b),\ \mathit{leaf},\ \mathit{leaf})) \quad = \quad \mathit{inr}\ (\mathit{inr}\ (a,\ b,\ \mathit{leaf})) \\
&\mathit{idem}^{\cup} \quad (\mathit{inr}\ (a,\ b,\ \mathit{node}\ (c,\ d))) \quad = \quad \mathit{inr}\ (\mathit{inr}\ (a,\ b,\ \mathit{node}\ (c,\ d)))
\end{aligned}$$

respectively. Using the identity function, idem, and $\mathit{idem}^{\cup}$ as bases, we have functions

$$\mathit{idem\_n} \;:\; (\,n : \mathbb{N}^{+}) \to n \times (1 + T^3) \to 1 + T^3$$

and

$$\mathit{idem\_n}^{\cup} \;:\; (\,n : \mathbb{N}^{+}) \to 1 + T^3 \to n \times (1 + T^3)\ ,$$

defined recursively as:

$$\begin{aligned}
&\mathit{idem\_n} \quad (\mathit{succ}\ \mathit{zero}) \quad = \quad \mathit{id}_{1+T^3} \\
&\mathit{idem\_n} \quad (\mathit{succ}\ n) \quad = \quad \mathit{idem}\ \circ\ (\mathit{id}_{1+T^3} + (\mathit{idem\_n}\ \ n))
\end{aligned}$$

and

$$\begin{aligned} \mathit{idem\_n}^{\cup} \quad (\mathit{succ\ zero}) &= \mathit{id}_{1+T^3} \\ \mathit{idem\_n}^{\cup} \quad (\mathit{succ\ n}) &= (\mathit{id}_{1+T^3} + (\mathit{idem\_n}^{\cup} \quad n)) \circ \mathit{idem}^{\cup} . \end{aligned}$$

By mathematical induction on n, we have that for all n in $\mathbb{N}^+$,

$$(\mathit{idem\_n} \quad n) \circ (\mathit{idem\_n}^{\cup} \quad n) = \mathit{id} = (\mathit{idem\_n}^{\cup} \quad n) \circ (\mathit{idem\_n} \quad n) .$$

3.5.4 Termination

In order to *finish* our proof of trees-in-zero, we still need to show that $(\!| f_{T^6} |\!)^{\cup}$ *terminates*. For reader's convenience, we repeat the definitions of $(\!| f_{T^6} |\!)^{\cup}$ and $f_{T^6}^{\cup}$ as following:

$$\begin{aligned} (\!| f_{T^6} |\!)^{\cup} &\triangleq \mathit{in} \circ (\mathit{id}_{1+T^3} + \mathit{id}_{T^6} \times (\!| f_{T^6} |\!)^{\cup}) \circ f_{T^6}^{\cup} ; \\ f_{T^6}^{\cup} &\triangleq (\mathit{id}_{1+T^3} + (\mathit{tn\_prod}^{\cup} \quad 6)) \circ (\mathit{idem\_n}^{\cup} \quad 2) . \end{aligned}$$

The termination of $(\!| f_{T^6} |\!)^{\cup}$ is decided by $f_{T^6}^{\cup}$. Specifically, if the output of $f_{T^6}^{\cup}$ matches with the pattern "inl _", then $(\!| f_{T^6} |\!)^{\cup}$ always terminates. This is the *base* case. Suppose that the output of $f_{T^6}^{\cup}$ matches with the pattern "inr (_, a)". If there is a bound function

$$\mathit{size} \ : \ 1 + T^3 \to \mathbb{N}$$

satisfying that the size of a is less than the size of the input to $f_{T^6}^{\cup}$, and for the base case, the size of the input to $f_{T^6}^{\cup}$ is zero, then the termination of $(\!| f_{T^6} |\!)^{\cup}$ is established.

Before defining the bound function *size*, in order to get better understanding of the problem, let us do case analysis on inputs to $f_{T^6}^{\cup}$. From the definition of $\mathit{idem\_n}^{\cup}$, we have that $(\mathit{idem\_n}^{\cup} \quad 2)$ is effectively same as $\mathit{idem}^{\cup}$. Thus, by using patterns appearing in the definition of $\mathit{idem}^{\cup}$, we can write $f_{T^6}^{\cup}$ explicitly as:

$$\begin{aligned} f_{T^6}^{\cup} \quad (\mathit{inl} \bullet) &= \mathit{inl}\ (\mathit{inl} \bullet) \\ f_{T^6}^{\cup} \quad (\mathit{inr}\ (a,\ \mathit{node}\ (b,\ c),\ \mathit{leaf})) &= \mathit{inl}\ (\mathit{inr}\ (a,\ b,\ c)) \\ f_{T^6}^{\cup} \quad (\mathit{inr}\ (\mathit{leaf},\ \mathit{leaf},\ \mathit{leaf})) &= \mathit{inr}\ (\mathit{leaf}^6,\ \mathit{inl} \bullet) \end{aligned}$$

$$f^{\cup}_{T^6} \quad (inr\ (a,\ leaf,\ leaf)) \quad = \quad inr\ ((a,\ leaf^5),\ inl\ \bullet)$$

$$f^{\cup}_{T^6} \quad (inr\ (a,\ b,\ node\ (c,\ d))) \ = \ inr\ ((tn\_prod^{\cup}\ \ 6)\ (inr\ (a,\ b,\ node\ (c,\ d))))\ .$$

For the first and the second patterns in the above definition, we have:

$$(\!|\, f_{T^6} \,|\!)^{\cup} \quad (inl\ \bullet) \quad = \quad one\ (inl\ \bullet)$$

$$(\!|\, f_{T^6} \,|\!)^{\cup} \quad (inr\ (a,\ node\ (b,\ c),\ leaf)) \quad = \quad one\ (inr\ (a,\ b,\ c))\ .$$

The third and fourth patterns will be reduced to the first pattern after unfolding. That is,

$$(\!|\, f_{T^6} \,|\!)^{\cup} \quad (inr\ (leaf,\ leaf,\ leaf)) \quad = \quad pcons\ (leaf^6,\ ((\!|\, f_{T^6} \,|\!)^{\cup}\ \ (inl\ \bullet)))$$

$$(\!|\, f_{T^6} \,|\!)^{\cup} \quad (inr\ (a,\ leaf,\ leaf)) \quad = \quad pcons\ ((a,\ leaf^5),\ ((\!|\, f_{T^6} \,|\!)^{\cup}\ \ (inl\ \bullet)))\ .$$

Let us consider the last pattern. Recall the definition of $t\_prod^{\cup}$, repeated as following:

$$t\_prod^{\cup} \quad (inl\ \bullet) \quad = \quad (leaf,\ inl\ \bullet)$$

$$t\_prod^{\cup} \quad (inr\ (a,\ b,\ leaf)) \quad = \quad (node\ (a,\ b),\ inl\ \bullet)$$

$$t\_prod^{\cup} \quad (inr\ (a,\ b,\ node\ (c,\ d))) \quad = \quad (a,\ inr\ (b,\ c,\ d))\ .$$

Notice that when the input of $t\_prod^{\cup}$ is not "$inl\ \bullet$", *the depth of the third tree decreases after each unfolding of* $t\_prod^{\cup}$. Since $(tn\_prod^{\cup}\ \ 6)$ is recursively defined on $t\_prod^{\cup}$, as for the last pattern, $f^{\cup}_{T^6}$ *decreases the depth of the third tree as well.*

Based on above discussions, with the aid of the function

$$depth\ :\ T \rightarrow \mathbb{N}^{+}$$

which is defined as:

$$depth \quad leaf \quad = \quad succ\ zero$$

$$depth \quad (node\ (a,\ b)) \quad = \quad succ\ (max \quad (depth \quad a)\ (depth \quad b))\ ,$$

we can define the bound function $size$ as:

$$size \quad (inl\ \bullet) \quad = \quad zero$$

$$size \quad (inr\ (\_,\ node\ (\_,\ \_),\ leaf)) \quad = \quad zero$$

$$size \quad (inr\ (a,\ b,\ c)) \quad = \quad depth \quad c\ .$$

Here, the first and second patterns in the above definition correspond to the first and second patterns in the explicit definition of $f^{\cup}_{T^6}$ respectively. The third pattern is used to capture the property that $f^{\cup}_{T^6}$ decreases the depth of the third tree. Notice that when the third tree matches the pattern "$leaf$", the size of the input is one. This corresponds to the third and fourth patterns in the explicit definition of $f^{\cup}_{T^6}$ which terminate after unfolding once.

The $size$ function meets our requirement: when the output of $f^{\cup}_{T^6}$ matches with the pattern "$inl\ \_$", its value on the input to $f^{\cup}_{T^6}$ is zero; and if the output of $f^{\cup}_{T^6}$ matches with the pattern "$inr\ (\_,\ a)$", then the size of a is less than the size of the input to $f^{\cup}_{T^6}$. Hence, $(\!| f_{T^6} |\!)^{\cup}$ terminates.

In summary, we prove the isomorphism trees-in-zero by using $List\ T$ as a bridge. The crucial step is the explicit construction of the isomorphism between $(List\ (T^6)) \times (1+T^3)$ and $1 + T^3$. Technically, we construct them as a catamorphism and its inverse followed by their termination proofs.

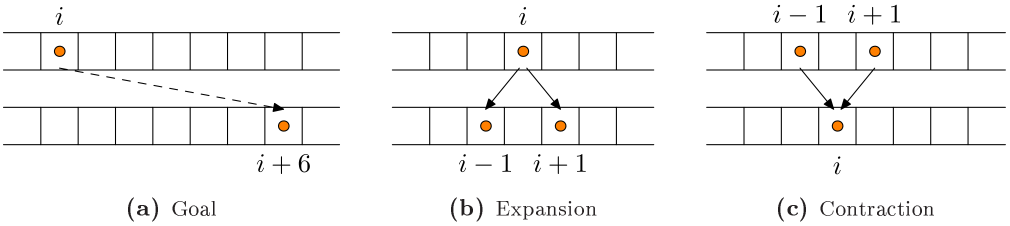
Chapter 4

Replacement-Set Games

It is interesting to notice that the isomorphism seven-trees-in-one can be illustrated by a one-person board game, so-called “the nuclear pennies game”. In this chapter, we introduce an infinite class of one-person board games which has the nuclear pennies game as an instance. This class of games we call *replacement-set games*. Through developing an algorithm to solve these games, we construct a necessary and sufficient condition on the polynomial β under which identities $T^k = \beta$ generate $T^k \ =_\beta \ T^{k+n}$ for natural numbers k and n. It is a surprise that this condition builds connections between type isomorphisms and *products of cyclotomic polynomials*. Further, by using properties of cyclotomic polynomials, we construct several infinite classes of solvable replacement-set games. However, it is still an open problem to construct the complete set of solvable replacement-sets.

4.1 The Nuclear Pennies Game

The seven-trees-in-one isomorphism has been turned into a game called the "nuclear pennies game" [Pip07a, Pip07b]. There is an unbounded one-dimensional board which is divided into squares. Initially there is only one checker on one of squares. The goal is to move this checker six squares to the right leaving all other squares empty. Index these squares by integers. There are two types of atomic moves: *expansions* and *contractions*. An expansion on square i is to replace a checker on square i by adding one checker on each of the two squares $i-1$ and $i+1$. A contraction to square i is that two checkers, one on square $i-1$ and one on square $i+1$, are replaced by adding one checker on square i. This game can be illustrated by the following figures:



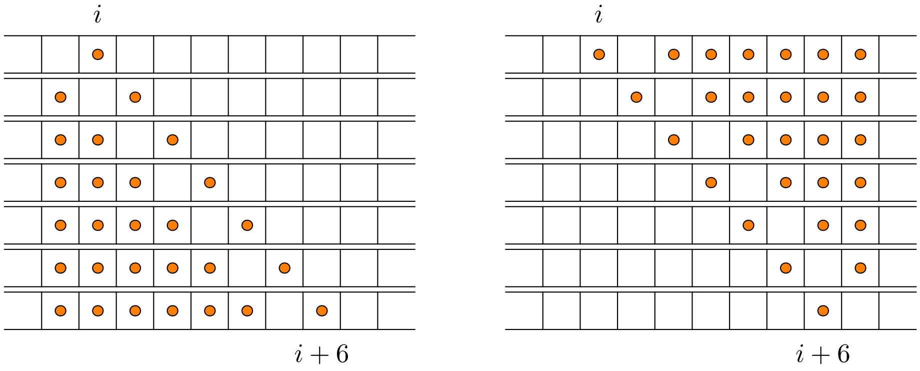
(a) Goal **(b)** Expansion **(c)** Contraction

Figure 4.1: The Nuclear Pennies Game

The connection between seven-trees-in-one and the nuclear pennies game is easy to see if one considers an atomic move as replacing $T^{i-1} \times T$ by $T^{i-1} \times (1 + T^2)$ or vice-versa.

Notice that expansions are reversed procedures of contractions. There is a symmetric solution to the nuclear pennies game. The solution can be decomposed into two stages: the first stage is to ensure that there is a checker on the square six squares right to the starting square and, symmetrically, there is a checker on the square six squares left to the finishing square; and the second stage is to connect the above two intermediate states.

Achieving the first stage is easy. It is shown in the following figure. In fig 4.2a, six expansions are used to ensure that a checker is added on the square six squares to the right of the starting square. Symmetrically, in fig 4.2b, working from bottom to top, six expansions ensure that a checker is added on the square six squares to the left of the finishing square.



(a) Initial Phase

(b) Final Phase

Figure 4.2: The Nuclear Pennies Game — The First Stage

The second stage is to connect two intermediate states: the bottom state in fig 4.2a and the top state in fig 4.2b. A possible solution is shown in the following figure:

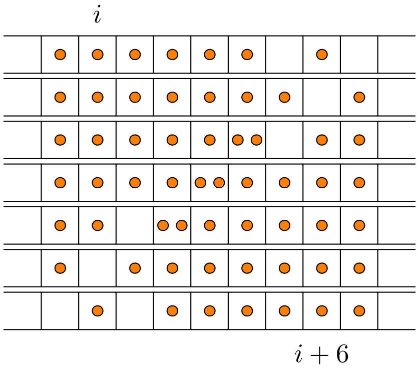


Figure 4.3: The Nuclear Pennies Game — The Second Stage

In fig 4.3, two intermediate states are repeated. The first and last moves (expansion on square $i+6$ and, symmetrically, expansion on square i) are used to ensure that there is a checker on square $i+5$ and a checker on square $i+1$ respectively . Then, expansions on squares $i+5$ and $i+4$ are used to produce the middle state and, symmetrically, expansions on squares $i+1$ and $i+2$ are used to produce the middle state as well. These expansions are from powers appearing in the following polynomial factorization:

$$T^6 - 1 \quad = \quad (T^4 + T^3 - T - 1) \times (T^2 - T + 1) \ .$$

More explanations about the connection between the above expansions and this factorization will be given in later sections.

4.2 Replacement-Set Games

The seven-trees-in-one isomorphism is not an isolated example. For instance, the isomorphism $T \cong_{1+T+T^2} T^5$ appearing in [Fio04] and several classes of isomorphisms which are similar with the seven-trees-in-one isomorphism given in [FL05]. We propose to construct identities $T^k = \beta$ which generate $T^k =_\beta T^{n+k}$ for polynomials β in $\mathbb{N}[T]$ for natural numbers k and n. This is equivalent to solving the following games: there is an unbounded one-dimensional board divided into squares with only a checker on one of squares; the goal is to move this checker to the square which is n squares to the right of the starting square leaving all other square empty; and atomic moves of this game are identified by a *multiset* R. Specifically, an expansion on square i is to replace a checker on square i by adding one checker into each square in the multiset $\{\!| \, i + a \mid a \in R \, |\!\}$ and a contraction is the reversed procedure.

The connection between $T^k \cong \beta$ and the multiset R is straightforward. Let A be the multiset of powers appearing in the polynomial β. We have:

$$R \quad \triangleq \quad \{\!| \, a - k \mid a \in A \, |\!\} \ .$$

That is, the multiset R captures the relative replacement squares to the square k. As an example, given an identity $T^5 \cong T^2 + T^5 + T^8$, the corresponding multiset R is $\{\!| -3,\ 0,\ 3 \, |\!\}$.

We call this class of games the *replacement-set games*. And the multiset R is called the *replacement-set* and n is called the *displacement*. The nuclear pennies game, for instance, is corresponding to the replacement-set game with replacement-set $\{\!| -1,\ 1 \, |\!\}$ and displacement 6.

In this chapter, we focus on the following problems: (a) given a replacement-set game, to decide whether there is a valid sequence of expansions and contractions which solves this game; (b) to construct such a sequence if the game is solvable; (c) given a displacement n, to construct all solvable replacement-sets.

For questions (a) and (b), we will give complete answers in sections 4.3 and 4.4. In section 4.5, we will give partial answers to question (c). By partial answers, we mean that some interesting subsets of the set of all solvable replacement-sets are constructed by using properties of *products of cyclotomic polynomials*.

4.3 Trivial Replacement-Set Games

Let $min.R$ and $max.R$ be the least and the greatest element of R respectively. A replacement-set game $(R,\ n)$ is *trivial* if $min.R \geqslant 0$ or $max.R \leqslant 0$. Because either there is no solution or there are trivial solutions to these games. More details are given as follows.

Suppose that $min.R$ is 0. There is a solution to the game $(R,\ n)$ if and only if n is 0. When n is 0, the solution to the game is trivial — doing nothing or a *valid* sequence of expansions and contractions satisfying that the number of expansions is the same as the number of contractions, and for all prefixes of this sequence, the number of contractions is at most the number of expansions. For instance, write Ei and Ci for expansion on square i and contraction to square i respectively, the sequence

$$[\,E0,\ E1,\ C0,\ E2,\ E0,\ C2,\ C1,\ C0\,]$$

is a solution to the replacement-set game $(\{\!|\, 0,\ 1\,|\!\},\ 0)$ with the initial checker on square 0. If n is not zero, since the initial checker is always on the board, there is no solution to the game. The similar argument applies to the case: $max.R$ is 0.

Suppose that $min.R$ is greater than 0. Without loss of generality, assume that the initial checker is placed on square 0. We want to show that there is a solution to the replacement-set game $(R,\ n)$ if and only if $R = \{\!\!\{\, a \,\}\!\!\}$ and a divides n. The if-part is easy to prove. When n is zero, the only solution is an empty sequence. If n is not zero, an appropriate solution is the following sequence of expansions:

$$[\, E0,\ Ea,\ E2a,\ \ldots,\ E(\frac{n}{a} - 1)a\,]\ .$$

In order to prove the only-if-part, we need some formalization. Checkers on the board at some stage of a game, called a *state*, can be modelled by a polynomial with natural numbers as coefficients. For instance, the state with two checkers on square 2 and one checker on square 3 can be characterized by the polynomial $2T^2 + T^3$. For all states p, an expansion on square i is *valid* if the coefficient of T^i in p is not zero and $i \geqslant 0$. A valid contraction is a reversed procedure of a valid expansion. Define the polynomial β_R as:

$$\beta_R \quad \triangleq \quad \beta \times T^{-k} \quad = \quad \langle \Sigma\, i \ :\ i \in R \ :\ T^i \rangle$$

where $T^k \ \cong\ \beta$. A valid expansion on square i can be formalized as the following state transition:

$$p \quad \rightarrow \quad p - T^i + T^i \times \beta_R\ .$$

Symmetrically, a valid contraction to square i is formalized as the state transition:

$$p \quad \rightarrow \quad p + T^i - T^i \times \beta_R\ .$$

A solution to the game is a finite sequence of valid expansions and contractions from the starting state 1 to the finishing state T^n. Given a state p, after a valid expansion on square i, a polynomial $T^i \times (\beta_R - 1)$ is added to p and, symmetrically, after a valid contraction, a polynomial $-T^i \times (\beta_R - 1)$ is added to p. Thus, that a game is solvable implies that there is a polynomial z in $\mathbb{Z}[T]$ satisfying that $1 + z \times (\beta_R - 1) \ =\ T^n$. That is, $\beta_R - 1$ divides $T^n - 1$ on $\mathbb{Z}[T]$, written as:

$$(\beta_R - 1) \setminus (T^n - 1)\ .$$

From the polynomial long division algorithm, if $\beta_R - 1$ divides $T^n - 1$, then $\beta_R - 1$ can be rewritten as $T^a + r - 1$ for a positive natural number a and a polynomial r in $\mathbb{N}[T]$. And the polynomial r satisfies that r is 0 or its degree is less than a and its constant term is zero. We have:

Lemma 4.3.1. $(T^a + r - 1) \setminus (T^n - 1) \quad \equiv \quad r = 0 \ \wedge \ a \setminus n \ .$

Proof. The if-part is established by the following fact:

$$T^n - 1 \quad = \quad (T^a - 1) \times \langle \Sigma\, i \ : \ 0 \leqslant i < \frac{n}{a} \ : \ T^{ia} \rangle \ .$$

Let us show the only-if-part. If $T^a + r - 1$ divides $T^n - 1$, then $T^a + r - 1$ is a product of cyclotomic polynomials. Since the only cyclotomic polynomial having negative constant term is $T - 1$ and the constant term of $T^a + r - 1$ is negative, $T - 1$ is a factor of $T^a + r - 1$. Recall that a is a positive natural number. If a is 1 or 2, it is straightforward that $r = 0$ and $a \setminus n$. When a is greater than 2, we have that there is a polynomial p in $\mathbb{Z}[T]$ with degree less than $a - 1$ and constant term zero such that

$$(T - 1) \times (T^{a-1} + p + 1) \quad = \quad T^a + r - 1 \ .$$

By simplifying the above equation, we have:

$$\begin{array}{ll}
 & T \times p + T - T^{a-1} - p \ = \ r \\
= & \{ \quad r \text{ is in } \mathbb{N}[T] \text{ with degree less than } a \text{ and constant term zero.} \quad \} \\
 & \langle \forall\, i \ : \ 1 \leqslant i \leqslant a - 1 \ : \ [T^i](T \times p + T - T^{a-1} - p) \geqslant 0 \rangle \\
= & \{ \quad \text{The degree of } p \text{ is less than } a - 1 \text{ and its constant term is zero.} \quad \} \\
 & 1 \geqslant [T]p \ \wedge \ \langle \forall\, i \ : \ 2 \leqslant i \leqslant a - 2 \ : \ [T^{i-1}]p \geqslant [T^i]p \rangle \ \wedge \ [T^{a-2}]p \geqslant 1 \\
= & \{ \quad \text{transitivity and } [x \leqslant y \leqslant x \ \equiv \ x = y] \quad \} \\
 & \langle \forall\, i \ : \ 1 \leqslant i \leqslant a - 2 \ : \ [T^i]p = 1 \rangle \ .
\end{array}$$

Further, $T \times p + T - T^{a-1} - p \ = \ 0 \ = \ r$. Also, from the polynomial long division algorithm,

$$(T^a - 1) \setminus (T^n - 1) \quad \equiv \quad a \setminus n \ .$$

We prove the lemma. □

By using lemma 4.3.1, we have that β_R must be T^a with a divides n if the game is solvable. This completes the proof of: when $min.R$ is greater than 0, there is a solution to the replacement-set game $(R,\ n)$ if and only if $R = \{\!|\, a \,|\!\}$ and a divides n. Symmetrically, a similar argument applies to the case: $max.R$ is less than 0.

In summary, for trivial replacement-set games $(R,\ n)$: *(a) if $min.R$ or $max.R$ is 0, then the game is solvable if and only if n is* 0*; (b) if $min.R$ is greater than* 0 *or $max.R$ is less than* 0*, then the game is solvable if and only if $R = \{\!|\, i \,|\!\}$ and i divides n with i an integer.*

4.4 Non-trivial Replacement-Set Games

Now, let us consider non-trivial replacement-set games. That is, the replacement set R satisfies that $min.R < 0 < max.R$.

4.4.1 Polynomials

Without loss of generality, we assume that only squares indexed by natural numbers are used and the initial checker is placed on the square m where $-m$ is the smallest element in R. With this assumption, checkers on the board can be modelled by a polynomial in $\mathbb{N}[T]$. For all states p, we say that there is a *valid expansion* on square i if the coefficient of T^i in p is not zero and $i \geqslant m$. Notice that a valid expansion on square i is to replace a checker on square i and to add one checker to each square in $\{\!|\, i+a \mid a \in R \,|\!\}$. Define the polynomial β_R as:

$$(4.1) \qquad \beta_R \quad \triangleq \quad \langle \Sigma i \ :\ i \in R \ :\ T^{i+m} \rangle \ .$$

By using this polynomial, a valid expansion on square i can be characterized by the following state transition:

$$p \quad \rightarrow \quad p - T^i + T^{i-m} \times \beta_R \ .$$

Symmetrically, a valid contraction to square i is corresponding to the state transition:

$$p \quad \rightarrow \quad p + T^i - T^{i-m} \times \beta_R \ .$$

Notice that $\beta = \beta_R \times T^{k-m}$. For instance, given an isomorphism $T^5 \cong T^2 + T^5 + T^8$, the corresponding multiset is $\{\!| -3,\ 0,\ 3 |\!\}$ and β_R is $1 + T^3 + T^6$. We have that $T^2 + T^5 + T^8 = (1 + T^3 + T^6) \times T^{5-3}$. The constant term of β_R is not zero. This property is useful later. Thus, in this subsection, we prefer β_R to β.

Recall that we assume the initial checker is placed on square m with $-m$ the smallest number in R. A solution to the game is a finite sequence of valid expansions and contractions from the starting state T^m to the finishing state T^{m+n}. Given a state p, after a valid expansion on square i, a polynomial $T^{i-m} \times (\beta_R - T^m)$ is added to p and, symmetrically, after a valid contraction, a polynomial $-T^{i-m} \times (\beta_R - T^m)$ is added to p. Thus, that a game is solvable implies that there is a polynomial z in $\mathbb{Z}[T]$ satisfying that $T^m + z \times (\beta_R - T^m) = T^{n+m}$. That is, $\beta_R - T^m$ divides $(T^n - 1) \times T^m$. Let us do case analysis on m. If m is zero, we have that $\beta_R - T^m$ divides $T^n - 1$. When m is not zero, since the constant term of β_R is not zero, we have that the common factor of $\beta_R - T^m$ and T^m is 1. Further, $\beta_R - T^m$ divides $T^n - 1$. Therefore, we establish a necessary condition for the problem that a given non-trivial replacement-set game $(R,\ n)$ has a solution. That is,

$$(\beta_R - T^m) \setminus (T^n - 1)$$

where the backslash symbol denotes divisibility on $\mathbb{Z}[T]$.

4.4.2 An Algorithmic Solution

We want to show that the above condition is a sufficient condition as well, by constructing an algorithm to produce a sequence of valid expansions and contractions from the starting state T^m to the finishing state T^{n+m} provided that $\beta_R - T^m$ is a factor of $T^n - 1$.

Notice that *the game* $(R,\ n)$ *with the initial checker on square* m *is solvable if and only* *if the game* $(\{\!| \frac{i}{gcd.R} \mid i \in R |\!\}, \frac{n}{gcd.R})$ *with the initial checker on square* $\frac{m}{gcd.R}$ *is solvable.* Here, $gcd.R$ is the greatest common divisor of elements in R.

The *if-part* is established directly through replacing all expansions on squares i by expansions on squares $i \times gcd.R$ and all contractions to squares i by contractions to squares $i \times gcd.R$. Recall that a valid expansion on square i is to replace a checker on square i by adding one checker to each square in $\{\!| i + a \mid a \in R |\!\}$ and a valid contraction is a reversed procedure of a valid expansion. Also, $-m$ is in R since $-m$ is the least element in R. We have that the set of all squares i on which a checker can be placed during a game (R, n) with the initial checker on square m is the set of *linear combinations* of elements in R. More precisely, they are *multiples of the greatest common divisor of* R. Thus, given a solution to game (R, n) with the initial checker on square m, it is valid to replace all expansions on squares i by expansions on squares $\frac{i}{gcd.R}$ and all contractions to squares i by contractions to squares $\frac{i}{gcd.R}$. This completes the proof of the *only-if-part*.

It follows that a solution to a game (R, n) with $gcd.R \neq 1$ can be constructed from a solution to the game $(\{\!| \frac{i}{gcd.R} \mid i \in R |\!\}, \frac{n}{gcd.R})$ through replacing square indexes i by $i \times gcd.R$. Without loss of generality, let us consider the problem of constructing a sequence of valid expansions and contractions from the starting state T^m to the finishing state T^{n+m} provided that $\beta_R - T^m$ is a factor of $T^n - 1$ and $gcd.R = 1$.

Reviewing The Nuclear Pennies Game

The idea is embodied in the solution to the nuclear pennies game. The corresponding replacement-set game is identified by the pair $(\{\!| -1,\ 1 |\!\},\ 6)$. Since the least element in the replacement-set $\{\!| -1,\ 1 |\!\}$ is -1, the starting square m is 1 and $\beta_{\{\!| -1,\ 1 |\!\}}$ is $1+T^2$. By using polynomials to characterize states of a game, the solution to the nuclear pennies game is formalized as follows:

$\{ \quad p = T \quad \}$

expansions on squares from 1 to 7 ;

$\{ \quad p = T + \langle \Sigma j \ : \ 1 \leqslant j \leqslant 7 \ : \ T^{j-1} \times (1 + T^2 - T) \rangle \quad \}$

expansions on squares 6 and 5 ;

$$\{ \quad p = T + \langle \Sigma j \ : \ 1 \leqslant j \leqslant 7 \ : \ T^{j-1} \times (1 + T^2 - T) \rangle \\ + (T^{6-1} + T^{5-1}) \times (1 + T^2 - T) \quad \}$$

$$\{ \quad p = T^7 + \langle \Sigma j \ : \ 1 \leqslant j \leqslant 7 \ : \ T^{j-1} \times (1 + T^2 - T) \rangle \\ + (T^{3-1} + T^{2-1}) \times (1 + T^2 - T) \quad \}$$

contractions to squares 3 and 2 ;

$\{ \quad p = T^7 + \langle \Sigma j \ : \ 1 \leqslant j \leqslant 7 \ : \ T^{j-1} \times (1 + T^2 - T) \rangle \quad \}$

contractions to squares from 1 to 7 .

$\{ \quad p = T^7 \quad \}$

Because $1 + T^2 - T$ is a factor of $T^6 - 1$, from the following factorization:

$$T^6 - 1 \quad = \quad (T^4 + T^3 - T - 1) \times (T^2 - T + 1) \ ,$$

we have that for all polynomial γ in $\mathbb{N}[T]$,

$$T + (\gamma + T^2 + T) \times (1 + T^2 - T) \quad = \quad T^7 + (\gamma + T^5 + T^4) \times (1 + T^2 - T) \ .$$

It follows that the two middle states are equal. Here, we take γ to be

$$\langle \Sigma j \ : \ 1 \leqslant j \leqslant 7 \ : \ T^{j-1} \rangle \ .$$

Further, the solution to the nuclear pennies game can be considered as the construction of two valid expansion sequences starting from states T and T^7 respectively satisfying:

(a) their corresponding polynomial characterizations are equal;

(b) the resulting states ensure respectively that expansions on squares T^6 and T^5, and expansions on squares T^3 and T^2 are valid.

Specifically, the two valid expansion sequences are

$$[\, E1,\ E2,\ E3,\ E4,\ E5,\ E6,\ E7\,]$$

and

$$[\, E7,\ E6,\ E5,\ E4,\ E3,\ E2,\ E1\,]\ .$$

Notice that an expansion on square i adds one checker on each of squares $i-1$ and $i+1$. Hence, expansions on $i-1$ and $i+1$ following an expansion on i are valid. This property ensures the validity of the above expansion sequences. Further, they have the same polynomial characterization:

$$\langle \Sigma j \ :\ 1 \leqslant j \leqslant 7 \ :\ T^{j-1} \times (1 + T^2 - T) \rangle\ .$$

And the resulting states after the above expansions starting respectively from states T and T^7 are as follows:

$$\begin{aligned}
S1 &= T + \langle \Sigma j \ :\ 1 \leqslant j \leqslant 7 \ :\ T^{j-1} \times (1 + T^2 - T) \rangle \\
&= \langle \Sigma j \ :\ 0 \leqslant j \leqslant 6 \ :\ T^j \rangle + T^8\ ; \\
S2 &= T^7 + \langle \Sigma j \ :\ 1 \leqslant j \leqslant 7 \ :\ T^{j-1} \times (1 + T^2 - T) \rangle \\
&= 1 + \langle \Sigma j \ :\ 2 \leqslant j \leqslant 8 \ :\ T^j \rangle\ .
\end{aligned}$$

Let the notation $[T^k]p$ denote the coefficient of T^k in state p. We have:

$$[T^6]S1 > 0 \quad \wedge \quad [T^5]S1 > 0$$

and

$$[T^3]S2 > 0 \quad \wedge \quad [T^2]S2 > 0\ .$$

That is, the above condition (b) is satisfied.

Then, we can do expansions on squares 6 and 5, and expansions on squares 3 and 2 respectively to get the middle state. Recall that contraction is the reversed procedure of expansion. The solution to the nuclear pennies game can be captured by an expansion sequence followed by a contraction sequence, shown as:

$$[\,E1,\ E2,\ E3,\ E4,\ E5,\ E6,\ E7,\ E6,\ E5,\ C3,\ C2,\ C1,\ C2,\ C3,\ C4,\ C5,\ C6,\ C7\,]\ .$$

The Algorithm Outline

Generally, suppose that the given non-trivial replacement-set game $(R,\ n)$ satisfies that $\beta_R - T^m$ divides $T^n - 1$ and the initial checker is on square m with $-m$ the least element in R. Define the partial order $\sqsubseteq$ on $\mathbb{N}[T]$ as: for all polynomials p and q,

$$p \ \sqsubseteq\ q \quad \triangleq \quad \langle \forall\, i \in \mathbb{N} \ :\ 1 \leqslant [T^i]p \ :\ [T^i]p \leqslant [T^i]q \rangle \ .$$

One can construct *least* polynomials δ and ρ in the poset $(\mathbb{N}[T],\ \sqsubseteq)$ satisfying that for all polynomials γ in $\mathbb{N}[T]$,

$$(4.2) \qquad T^m + (\gamma + \delta) \times (\beta_R - T^m) \quad = \quad T^{m+n} + (\gamma + \rho) \times (\beta_R - T^m)$$

by using the polynomial long division algorithm on $\mathbb{Z}[T]$.

Inspired by the symmetric solution to the nuclear pennies game, if we have two valid expansion sequences γ_m and γ_{m+n} starting from T^m and T^{m+n} respectively which satisfy:

(a) they have the same polynomial characterization $\gamma \times (\beta_R - T^m)$;

(b) $\delta \times T^m \ \sqsubseteq\ T^m + \gamma \times (\beta_R - T^m) \quad \wedge \quad \rho \times T^m \ \sqsubseteq\ T^{m+n} + \gamma \times (\beta_R - T^m)$,

the algorithm to solve the non-trivial replacement-set game $(R,\ n)$ with the initial checker on square m can be constructed as:

Algorithm 4.4.1.

```
$\{ \quad p = T^m \quad \}$
expansions on squares in $\gamma_m$ ;
$\{ \quad p = T^m + \gamma \times (\beta_R - T^m) \quad \}$
expansions on squares in the set of powers appearing in $\delta \times T^m$ ;
$\{ \quad p = T^m + (\gamma + \delta) \times (\beta_R - T^m) \quad \}$
$\{ \quad p = T^{m+n} + (\gamma + \rho) \times (\beta_R - T^m) \quad \}$
contractions to squares in the set of powers appearing in $\rho \times T^m$ ;
$\{ \quad p = T^{m+n} + \gamma \times (\beta_R - T^m) \quad \}$
contractions to squares in $\gamma_{m+n}$ .
$\{ \quad p = T^{m+n} \quad \}$
```

The equation (4.2) and the above condition (a) ensure that two intermediate states are equal. The above condition (b) ensures that expansions according to δ and contractions according to ρ are valid.

Constructing Valid Expansion Sequences

Assuming that $gcd.R = 1$, we now consider the problem of constructing valid expansion sequences γ_m and γ_{m+n} satisfying the above conditions. Notice that the equation (4.2) can be rewritten as:

$$T^{m+n} - T^m \quad = \quad (\delta - \rho) \times (\beta_R - T^m) \ .$$

And degrees and codegrees (the least powers) of polynomials on both sides of this equation are respectively same. Recall that $min.R < 0 < max.R$. According to the definition (4.1) of β_R, the degree of $\beta_R - T^m$ is greater than m and the codegree is 0. Thus, the degree of $\delta - \rho$ is less than n and its codegree is m. Further, we have:

$$\delta \times T^m \sqsubseteq K \times \langle \Sigma i \ : \ 2m \leqslant i < m+n \ : \ T^i \rangle$$
$$\rho \times T^m \sqsubseteq K \times \langle \Sigma i \ : \ 2m \leqslant i < m+n \ : \ T^i \rangle$$

where K *is the greatest coefficient of* δ *and* ρ. By using these properties, the above conditions (a) and (b) which the valid expansion sequences γ_m and γ_{m+n} should satisfy can be refined to:

(a) they have the same polynomial characterization $\gamma \times (\beta_R - T^m)$;

(b) $K \times \langle \Sigma i \ : \ 2m \leqslant i < m+n \ : \ T^i \rangle \sqsubseteq T^m + \gamma \times (\beta_R - T^m) \quad \wedge$
$K \times \langle \Sigma i \ : \ 2m \leqslant i < m+n \ : \ T^i \rangle \sqsubseteq T^{m+n} + \gamma \times (\beta_R - T^m)$.

Recall that the set of all squares on which one checker can be placed during a game $(R,\ n)$ with the initial checker on square m is the set of all multiples of the greatest common divisor of R. With assumptions $gcd.R = 1$ and $min.R < 0 < max.R$, by extending *the Euclidean Algorithm*, it is possible to produce a *compound expansion* L_i which satisfies the following property: given a state p with $[T^i]p > 0$, one can get a state p' by the following transition:

$$(4.3) \qquad p \quad \xrightarrow{L_i} \quad p'$$

satisfying that

$$(4.4) \qquad [T^{i-1}]p' > 0 \quad \wedge \quad [T^{i+1}]p' > 0 \ .$$

For clarity, we will give the algorithm which produces L_i later. Now, by using L_i, we construct the following compound expansion sequences:

$$(4.5) \qquad [\, L_m,\ L_{m+1},\ \cdots,\ L_{m+n} \,]$$

and

$$(4.6) \qquad [\, L_{m+n},\ L_{m+n-1},\ \cdots,\ L_m \,] \ .$$

The property (4.4) ensures that the above sequences are *valid* expansion sequences from the starting state T^m and the finishing state T^{m+n} respectively. Let $l_i \times (\beta_R - T^m)$ be the polynomial characterization of the transition (4.3). The corresponding polynomial characterizations of sequences (4.5) and (4.6) are equal to:

$$\gamma' \times (\beta_R - T^m) \quad \triangleq \quad \langle \Sigma i \ : \ m \leqslant i \leqslant m+n \ : \ l_i \rangle \times (\beta_R - T^m) \ .$$

Further, from the property (4.4), we have:

$$\langle \Sigma i \ : \ m \leqslant i < m+n \ : \ T^i \rangle \ \sqsubseteq \ T^m + \gamma' \times (\beta_R - T^m) \ ;$$
$$\langle \Sigma i \ : \ m < i \leqslant m+n \ : \ T^i \rangle \ \sqsubseteq \ T^{m+n} + \gamma' \times (\beta_R - T^m) \ .$$

Because

$$[T^m](T^m + \gamma' \times (\beta_R - T^m)) > 0 \quad \wedge \quad [T^{m+n}](T^{m+n} + \gamma' \times (\beta_R - T^m)) > 0 \ ,$$

sequences (4.5) and (4.6) can be repeated K times respectively. We now take γ_m and γ_{m+n} to be:

(4.7) $\quad [\, L_m,\ L_{m+1},\ \cdots\ ,\ L_{m+n}\,]^K$

and

(4.8) $\quad [\, L_{m+n},\ L_{m+n-1},\ \cdots\ ,\ L_m\,]^K$

respectively. Let γ be $K \times \gamma'$. Based on the above discussion, $\gamma \times (\beta_R - T^m)$ is the polynomial characterization of γ_m and γ_{m+n}, and

$$K \times \langle \Sigma i \ : \ m \leqslant i < m+n \ : \ T^i \rangle \ \sqsubseteq \ T^m + \gamma \times (\beta_R - T^m) \ ;$$
$$K \times \langle \Sigma i \ : \ m < i \leqslant m+n \ : \ T^i \rangle \ \sqsubseteq \ T^{m+n} + \gamma \times (\beta_R - T^m) \ .$$

Since m is greater than 0 ($-m$ is the least element of R and $min.R < 0 < max.R$), we have:

$$\langle \Sigma i \ : \ 2m \leqslant i < m+n \ : \ T^i \rangle \ \sqsubseteq \ \langle \Sigma i \ : \ m \leqslant i < m+n \ : \ T^i \rangle \ ;$$
$$\langle \Sigma i \ : \ 2m \leqslant i < m+n \ : \ T^i \rangle \ \sqsubseteq \ \langle \Sigma i \ : \ m < i \leqslant m+n \ : \ T^i \rangle \ .$$

Combining the above results, the refined conditions (a) and (b) are satisfied. This completes the construction of γ_m and γ_{m+n}.

Constructing Compound Expansions

We now focus on the construction of the compound expansion L_i which is a valid expansion sequence satisfying that given a state with at least one checker on square i, the resulting state after the sequence of expansions has at least one checker on each of squares $i-1$ and $i+1$.

Recall that all squares on which a checker can be placed during a game $(R,\ n)$ with the initial checker on square m are *linear combinations of elements in* R as well as *multiples of the greatest common divisor of* R. In particular, since $min.R < 0 < max.R$ and $gcd.R = 1$, by extending the *Euclidean Algorithm*, we can construct multisets A and B whose elements are from R satisfying:

$$i + \langle \Sigma a \ : \ a \in A \ : \ a \rangle \ = \ i - gcd.R \ = \ i - 1 \ ;$$

$$i + \langle \Sigma b \ : \ b \in B \ : \ b \rangle \ = \ i + gcd.R \ = \ i + 1 \ .$$

Further, the compound expansion L_i is constructed by *serializing* A and B. Let us illustrate the above idea by a simple example. For instance, taking R to be the multiset $\Lbag -3,\ 5 \Rbag$. Following the procedure of the Euclidean Algorithm, multisets A and B can be constructed as follows:

x	y	A	B
-3	5	$\Lbag -3 \Rbag$	$\Lbag 5 \Rbag$
-3	2	$\Lbag -3 \Rbag$	$\Lbag -3,\ 5 \Rbag$
-1	2	$\Lbag -3,\ -3,\ 5 \Rbag$	$\Lbag -3,\ 5 \Rbag$
-1	1	$\Lbag -3,\ -3,\ 5 \Rbag$	$\Lbag -3,\ -3,\ -3,\ 5,\ 5 \Rbag$

By serializing multisets A and B, we get the following valid expansion sequence:

$$[\, i,\ i-3,\ i-3-3,\ i+5,\ i+5+5,\ i+5+5-3,\ i+5+5-3-3\,] \ .$$

It is composed of a single expansion on square i followed by two subsequences:

$$[\, i-3,\ i-3-3\,]$$

and

$$[\, i+5,\ i+5+5,\ i+5+5-3,\ i+5+5-3-3\,] \ .$$

These two subsequences are constructed from multisets A and B respectively. Note that after the expansion on square i, there is at least one checker on each of squares $i-3$ and $i-5$. And after expansions on squares $i-3-3$ and $i+5+5-3-3$, there is at least one checker on each of squares $i-1 = i-3-3+5$ and $i+1 = i+5+5-3-3-3$. Hence, the above sequence satisfies the requirement on the compound expansion L_i with regard to $R = \{\!|\ -3,\ 5\ |\!\}$.

Generally, we give the algorithm to construct multisets A and B as follows:

Algorithm 4.4.2.

```
{   $min.R < 0 < max.R$   }
$x,\ y\ :=\ min.R,\ max.R$ ;
$Q,\ A,\ B\ :=\ set.R - \{x,\ 0,\ y\},\ \{\!|\ x\ |\!\},\ \{\!|\ y\ |\!\}$ ;
{   Invariant: $\langle \Sigma a\ :\ a \in A\ :\ a \rangle\ =\ x\ \ \wedge\ \ \langle \Sigma b\ :\ b \in B\ :\ b \rangle\ =\ y$
          $\wedge\ \ gcd.(Q\ \cup\ \{x,\ y\}) = gcd.R$   }
do    $x \neq -gcd.R\ \ \vee\ \ y \neq gcd.R\ \ \longrightarrow$
    do    $-x < y\ \ \longrightarrow\ \ y,\ B\ :=\ x+y,\ A\ \uplus\ B$
    []    $y < -x\ \ \longrightarrow\ \ x,\ A\ :=\ x+y,\ A\ \uplus\ B$
    od ;
    {   $-x = y\ \ \wedge\ \ \langle \Sigma a\ :\ a \in A\ :\ a \rangle\ =\ x\ \ \wedge\ \ \langle \Sigma b\ :\ b \in B\ :\ b \rangle\ =\ y$   }
    if    $Q \neq \varnothing\ \ \wedge\ \ min.Q < 0\ \ \longrightarrow\ \ x\ :=\ min.Q$ ;
                                              $Q,\ A\ :=\ Q - \{\!|\ x\ |\!\},\ \{\!|\ x\ |\!\}$
    []    $Q \neq \varnothing\ \ \wedge\ \ max.Q > 0\ \ \longrightarrow\ \ y\ :=\ max.Q$ ;
                                              $Q,\ B\ :=\ Q - \{\!|\ y\ |\!\},\ \{\!|\ y\ |\!\}$
    []    $Q = \varnothing\ \ \longrightarrow$  skip
    fi
od
{   $\langle \Sigma a\ :\ a \in A\ :\ a \rangle\ =\ -gcd.R\ \ \wedge\ \ \langle \Sigma b\ :\ b \in B\ :\ b \rangle\ =\ gcd.R$   }
```

This algorithm effectively calculates $gcd.R$ and $-gcd.R$ by extending the Euclidean Algorithm. We use the corresponding set $set.R$ of the multiset R to avoid unnecessary computation. It is worth to mentioning that 0 is removed from R to get rid of *possible meaningless computation.*

The *serializations* of A and B are respectively done by ordering elements in $i + A$ and $i + B$ (addition is extended to sets) then forming sequences of their partial sums. Further, by arbitrarily interleaving these two sequences with an expansion on square i headed, we get the needed compound expansion. In particular, by using the property:

$$\langle \Sigma a \ : \ a \in A \ : \ a \rangle \ < \ 0 \ < \ \langle \Sigma b \ : \ b \in B \ : \ b \rangle$$

which is maintained through the above algorithm, we give a specific algorithm to construct the compound expansion L_i from multisets A and B as follows:

Algorithm 4.4.3.

```
{ $\langle \Sigma a : a \in A : a \rangle = -gcd.R \quad \wedge \quad \langle \Sigma b : b \in B : b \rangle = gcd.R$
  $\wedge \quad [T^i]p > 0 \quad \wedge \quad i \geqslant m$ }
$j,\ k \ := \ min.A,\ max.B$ ;
$A,\ B \ := \ A - \{\!| min.A |\!\},\ B - \{\!| max.B |\!\}$ ;
$L_i,\ p \ := \ [i],\ p + T^{i-m} \times (\beta_R - T^m)$ ;
{ Invariant: $[T^{j+i}]p > 0 \quad \wedge \quad [T^{k+i}]p > 0 \quad \wedge \quad j \leqslant -gcd.R < 0 < gcd.R \leqslant k$
$\wedge \quad j + \langle \Sigma a : a \in A : a \rangle = -gcd.R \quad \wedge \quad k + \langle \Sigma b : b \in B : b \rangle = gcd.R$ }
do    $A \neq \varnothing$   ⟶   $L_i \ := \ L_i \ +\!\!+ \ [j+i]$ ;
                         $p \ := \ p + T^{j+i-m} \times (\beta_R - T^m)$ ;
                         $j \ := \ j + min.A$ ;
                         $A \ := \ A - \{\!| min.A |\!\}$
[]    $B \neq \varnothing$   ⟶   $L_i \ := \ L_i \ +\!\!+ \ [k+i]$ ;
                         $p \ := \ p + T^{k+i-m} \times (\beta_R - T^m)$ ;
                         $k \ := \ k + max.B$ ;
                         $B \ := \ B - \{\!| max.B |\!\}$
```

```
**od**
{ $[T^{i-gcd.R}]p > 0 \quad \wedge \quad [T^{i+gcd.R}]p > 0$ }
```

Here, the initial checker is supposed to be on square m. We assume that i is at least m. This ensures that the running state p is a polynomial with natural numbers as coefficients. Variables j and k are used to record partial sums. We always choose the minimum of A and the maximum of B for increases on j and k respectively. This maintains the property:

$$j \leqslant -gcd.R < 0 < gcd.R \leqslant k$$

which avoids arguments on the case $j = k$. At the end of the algorithm, the property:

$$[T^{i-gcd.R}]p > 0 \quad \wedge \quad [T^{i+gcd.R}]p > 0$$

ensures that the resulting sequence L_i satisfies our requirement (4.4) on the compound expansion.

Until now, we have finished the construction of the algorithm to solve non-trivial replacement-set games. Combining our arguments in section 4.4.1, we have:

Theorem 4.4.4. A given non-trivial replacement-set game $(R,\ n)$ with the initial checker on square m is solvable if and only if

$$(\beta_R - T^m) \setminus (T^n - 1)$$

where β_R is defined by equation (4.1). And, when the game is solvable, an appropriate solution is given by algorithm 4.4.1 where γ_m and γ_{m+n} are equations (4.7) and (4.8) respectively with the compound expansion L_i produced by algorithms 4.4.2 and 4.4.3.

Combining results we get for trivial replacement-set games, we give complete answers to questions (a) and (b) proposed in section 4.2.

4.4.3 The Normalization

We call a sequence of expansions and contractions which solves a non-trivial replacement-set game a *valid* sequence. And a valid sequence consisting of a sequence of expansions followed by a sequence of contractions we call a *normal* sequence. The careful reader may notice that we always construct normal sequences to solve non-trivial replacement-set games. It is natural to ask whether for all valid sequences L, there is a normal sequence which solves the same game as L does. The answer to this question is yes. In this subsection, let us show that there is a valid sequence if and only if there is a normal sequence.

For our purposes, we define the binary relation $\mathcal{E}$ on $\mathbb{N}[T]$ as: for all states p and q,

$$p\ \mathcal{E}\ q \quad \equiv \quad \langle \exists\, i\ :\ i \geqslant m\ \wedge\ [T^i]p > 0\ :\ p + T^{i-m} \times (\beta_R - T^m) = q \rangle\ .$$

That is, there is a valid expansion from states p to q. Let $\mathcal{F}$ be the converse relation of $\mathcal{E}$ and $\mathcal{I}$ be the identity relation. We have:

Lemma 4.4.5. $\mathcal{F}\mathcal{E} \quad \subseteq \quad \mathcal{E}\mathcal{F}\ \cup\ \mathcal{I}\ .$

Proof. For all states p and q,

$$\begin{array}{ll} & p\ \mathcal{F}\mathcal{E}\ q \\ = & \{\ \text{definitions of } \mathcal{E} \text{ and } \mathcal{F} \text{ and relation composition}\ \} \\ & \langle \exists\, r,\, i,\, j\ :\ r \in \mathbb{N}[T]\ \wedge\ i \geqslant m\ \wedge\ j \geqslant m\ \wedge\ [T^i]r > 0\ \wedge\ [T^j]r > 0\ : \\ & \qquad p = r + T^{i-m} \times (\beta_R - T^m)\ \wedge\ r + T^{j-m} \times (\beta_R - T^m) = q \rangle\ . \end{array}$$

If $i = j$, we have that $p = q$. That is, $\mathcal{F}\mathcal{E} \subseteq \mathcal{I}$. Suppose that $i \neq j$. Notice that

$$\begin{array}{ll} & i \neq j \quad \wedge \quad [T^i]r > 0 \quad \wedge \quad [T^j]r > 0 \\ \Rightarrow & \{\ \text{An expansion on square } i\ (j) \\ & \qquad \text{does not remove any checker on square } j\ (i).\ \} \\ & [T^j](r + T^{i-m} \times (\beta_R - T^m)) > 0 \quad \wedge \quad [T^i](r + T^{j-m} \times (\beta_R - T^m)) > 0 \end{array}$$

$$= \quad \{ \quad p = r + T^{i-m} \times (\beta_R - T^m) \ \wedge \ r + T^{j-m} \times (\beta_R - T^m) = q \quad \}$$

$$[T^j]p > 0 \quad \wedge \quad [T^i]q > 0$$

$$= \quad \{ \quad \text{expansions on squares } j \text{ and } i \text{ in } p \text{ and } q \text{ respectively}$$

$$\text{and } p = r + T^{i-m} \times (\beta_R - T^m) \ \wedge \ r + T^{j-m} \times (\beta_R - T^m) = q \quad \}$$

$$[T^j]p > 0 \quad \wedge \quad [T^i]q > 0$$

$$\wedge \quad p + T^{j-m} \times (\beta_R - T^m) = r + (T^{i-m} + T^{j-m}) \times (\beta_R - T^m)$$

$$\wedge \quad r + (T^{i-m} + T^{j-m}) \times (\beta_R - T^m) = q + T^{i-m} \times (\beta_R - T^m)$$

$$= \quad \{ \quad \text{wittness} : \ r + (T^{i-m} + T^{j-m}) \times (\beta_R - T^m) \quad \}$$

$$\langle \exists\, r,\, i,\, j \ : \ r \in \mathbb{N}[T] \ \wedge \ i \geqslant m \ \wedge \ j \geqslant m \ \wedge \ [T^j]p > 0 \ \wedge \ [T^i]q > 0 \ :$$

$$p + T^{j-m} \times (\beta_R - T^m) = r = q + T^{i-m} \times (\beta_R - T^m) \rangle$$

$$= \quad \{ \quad \text{definitions of } \mathcal{E} \text{ and } \mathcal{F} \text{ and relation composition} \quad \}$$

$$p \ \mathcal{E}\mathcal{F} \ q \ .$$

This completes the proof. □

Recall that contraction is the reverse procedure of expansion. The set of all sequences of expansions and contractions can be identified by the regular expression $(\mathcal{E} \ \cup \ \mathcal{F})^*$ where concatenation is replaced by relation composition. Likely, the set of all sequences consisted of a sequence of expansions followed by a sequence of contractions is specified by $\mathcal{E}^*\mathcal{F}^*$. Then, the statement that there is a valid sequence if and only if a normal sequence can be generalized to:

Theorem 4.4.6. For all states p and q,

$$\langle \exists\, \mathcal{L} \ : \ \mathcal{L} \in (\mathcal{E} \ \cup \ \mathcal{F})^* \ : \ p \ \mathcal{L} \ q \rangle \quad \equiv \quad \langle \exists\, \mathcal{L} \ : \ \mathcal{L} \in \mathcal{E}^*\mathcal{F}^* \ : \ p \ \mathcal{L} \ q \rangle \ .$$

Proof. Let us show $(\mathcal{E} \ \cup \ \mathcal{F})^* = \mathcal{E}^*\mathcal{F}^*$ as follows:

$$(\mathcal{E} \ \cup \ \mathcal{F})^* = \mathcal{E}^*\mathcal{F}^*$$

$$= \quad \{ \quad \mathcal{E}^*\mathcal{F}^* \ \subseteq \ (\mathcal{E} \ \cup \ \mathcal{F})^* \text{ and anti-symmetry} \quad \}$$

$$(\mathcal{E} \ \cup \ \mathcal{F})^* \ \subseteq \ \mathcal{E}^*\mathcal{F}^*$$

$$\Leftarrow \qquad \{ \quad (\mathcal{E} \cup \mathcal{F})^* = \langle \mu X :: \mathcal{I} \cup X(\mathcal{E} \cup \mathcal{F}) \rangle \quad \}$$

$$\mathcal{I} \cup \mathcal{E}^*\mathcal{F}^*(\mathcal{E} \cup \mathcal{F}) \subseteq \mathcal{E}^*\mathcal{F}^*$$

$$= \qquad \{ \quad \mathcal{I} \subseteq \mathcal{E}^*\mathcal{F}^* \text{ and } \mathcal{E}^*\mathcal{F}^*\mathcal{F} \subseteq \mathcal{E}^*\mathcal{F}^* \quad \}$$

$$\mathcal{E}^*\mathcal{F}^*\mathcal{E} \subseteq \mathcal{E}^*\mathcal{F}^*$$

$$\Leftarrow \qquad \{ \quad \text{monotonicity and } \mathcal{E}^*\mathcal{E}^* = \mathcal{E}^* \quad \}$$

$$\mathcal{F}^*\mathcal{E} \subseteq \mathcal{E}^*\mathcal{F}^*$$

$$\Leftarrow \qquad \{ \quad \mathcal{E}\mathcal{F}^* \cup \mathcal{F}^* \subseteq \mathcal{E}^*\mathcal{F}^* \text{ and transitivity} \quad \}$$

$$\mathcal{F}^*\mathcal{E} \subseteq \mathcal{E}\mathcal{F}^* \cup \mathcal{F}^*$$

$$\Leftarrow \qquad \{ \quad \mathcal{F}^*\mathcal{E} = \langle \mu X :: \mathcal{E} \cup \mathcal{F}X \rangle \quad \}$$

$$\mathcal{E} \cup \mathcal{F}(\mathcal{E}\mathcal{F}^* \cup \mathcal{F}^*) \subseteq \mathcal{E}\mathcal{F}^* \cup \mathcal{F}^*$$

$$= \qquad \{ \quad \mathcal{E} \subseteq \mathcal{E}\mathcal{F}^* \text{ and } \mathcal{F}\mathcal{F}^* \subseteq \mathcal{F}^* \quad \}$$

$$\mathcal{F}\mathcal{E}\mathcal{F}^* \subseteq \mathcal{E}\mathcal{F}^* \cup \mathcal{F}^*$$

$$\Leftarrow \qquad \{ \quad \text{lemma 4.4.5, monotonicity, and transitivity} \quad \}$$

$$(\mathcal{E}\mathcal{F} \cup \mathcal{I})\mathcal{F}^* \subseteq \mathcal{E}\mathcal{F}^* \cup \mathcal{F}^*$$

$$= \qquad \{ \quad \mathcal{E}\mathcal{F}\mathcal{F}^* \subseteq \mathcal{E}\mathcal{F}^* \text{ and } \mathcal{I}\mathcal{F}^* = \mathcal{F}^* \quad \}$$

$$\mathit{true}\ .$$

□

4.5 Constructing Solvable Replacement-Sets

In this section, we focus on the question (c) proposed in section 4.2: given a displacement n, to construct all solvable replacement-sets R. By *solvable replacement-set*, we mean that the replacement-set game $(R,\ n)$ is solvable.

As for trivial replacement-sets $(\mathit{min}.R \geqslant 0$ or $\max.R \leqslant 0)$, answers to the above question are trivial: (a) if $\mathit{min}.R$ or $\mathit{max}.R$ is 0, the replacement-set R is solvable if and only if n is 0; (b) if $\mathit{min}.R > 0$ or $\mathit{max}.R < 0$, then the replacement-set R is solvable if and only if $R = \{\!| \, i \, |\!\}$ satisfying that i divides n with i an integer. This has been shown in section 4.3.

Considering non-trivial replacement-sets ($min.R < 0 < max.R$). Theorem 4.4.4 tells us that the replacement-set R is solvable if and only if

$$(\beta_R - T^m) \setminus (T^n - 1)$$

where $-m$ is the least element of R and

$$\beta_R \quad \triangleq \quad \langle \Sigma\, i \;:\; i \in R \;:\; T^{i+m} \rangle \;.$$

That is, there is a corresponding between factors $\beta_R - T^m$ of $T^n - 1$ and non-trivial solvable replacement-sets R. Recall that β_R is a polynomial with natural numbers as coefficients. The polynomial $\beta_R - T^m$ is a polynomial with natural numbers as coefficients ($T^m \sqsubseteq \beta_R$) or a polynomial with only one negative coefficient ($T^m \not\sqsubseteq \beta_R$). The first class of replacement-sets we call *monotonic solvable replacement-sets.* Because in an expansion on square i, the number of checkers on square i will not change and the number of checkers on board increases strictly. We call the second class of replacement-sets *true solvable replacement-sets.* Because in an expansion on square i, a checker on square i is always truly replaced (the number of checkers on square i decreases by one) and the number of checkers on board does not change or increases strictly.

It is well-known that factors of $T^n - 1$ are products of *cyclotomic polynomials* (irreducible factors of $T^n - 1$ on $\mathbb{Z}[T]$). By using properties of cyclotomic polynomials, we construct two infinite classes of monotonic solvable replacement-sets in subsection 4.5.1 and one infinite class of true solvable replacment-sets in subsection 4.5.2. Unfortunately, as far as we know, it is still an *open problem* to completely characterize the set of all non-trivial solvable replacement-sets.

For our purposes, we introduce a specific class of products of cyclotomic polynomials. Given positive natural numbers a and b, we define $\Gamma.(a,\ b)$ as:

$$\Gamma.(a,\ b) \quad \triangleq \quad \langle\, \Sigma\, k \;:\; 0 \leqslant k < a \;:\; T^{k \times b} \,\rangle \;.$$

From the geometric series and the following property of cyclotomic polynomials:

$$(4.9) \qquad T^n - 1 \quad = \quad \langle \Pi k \ : \ 1 \leqslant k \leqslant n \ \wedge \ k \setminus n \ : \ \Phi.k \rangle \ ,$$

we have:

$$(4.10) \qquad \Gamma.(a,\ b) = \frac{T^{a\times b} - 1}{T^b - 1} = \langle \Pi k \ : \ 1 \leqslant k \leqslant a \times b \ \wedge \ k \setminus (a \times b) \ \wedge \ \neg(k \setminus b) \ : \ \Phi.k \rangle \ .$$

4.5.1 Monotonic Solvable Replacement-Sets

Notice that coefficients of $\Gamma.(a,\ b)$ are natural numbers. Given a finite set $\mathcal{A}$ of pairs of positive natural numbers, from the property (4.10), we have that the following product:

$$(4.11) \qquad \langle \Pi (a,\ b) \ : \ (a,\ b) \in \mathcal{A} \ : \ \Gamma.(a,\ b) \rangle$$

is a product of cyclotomic polynomials with natural numbers as coefficients.

However, not all such products are factors of $T^n - 1$ for some positive natural number n. From the property (4.9), we have that a factor of $T^n - 1$ is a product of *non-repeated cyclotomic polynomials.* For instance,

$$\Phi_2 \times \Phi_3 \quad = \quad (T + 1) \times (T^2 + T + 1)$$

is a factor of $T^6 - 1$, while

$$\Phi_2 \times \Phi_2 \quad = \quad (T + 1) \times (T + 1)$$

is not a factor of any $T^n - 1$. Also, $\Gamma.(a,\ b)$ is a product of cyclotomic polynomials indexed by elements in the set $\overline{a \times b} - \overline{b}$ which follows from the property (4.10). Here, we use $\overline{a \times b}$ and $\overline{b}$ for sets of positive divisors of $a \times b$ and b respectively and symbol $-$ for set difference. We have that if the set $\mathcal{A}$ satisfies:

$$(4.12) \qquad \langle \forall (a,\ b), (c,\ d) \in \mathcal{A} \ :: \ (\overline{a \times b} - \overline{b}) \ \cap \ (\overline{c \times d} - \overline{d}) = \varnothing \rangle \ ,$$

then the product (4.11) is a factor of $T^n - 1$ for some positive natural number n with natural numbers as coefficients.

Further, let p be a product in form (4.11) with the set $\mathcal{A}$ satisfying the condition (4.12). Let m be a positive natural number less than the degree of p. (Choosing m in this way ensures that the resulting replacement-set R is non-trivial. That is, $min.R < 0 < max.R$.) We define β_R as:

$$\beta_R \quad \triangleq \quad p + T^m \ .$$

The corresponding multiset R of β_R is a monotonic solvable replacement-set with the displacement n positive multiples of the least common multiple of all elements in

$$\langle \cup\,(a,\ b) \in \mathcal{A}\ ::\ \overline{a \times b} - \overline{b} \rangle \ .$$

Let us look at an infinite class of examples. Take n to be a positive square-free natural number. We define $\mathcal{A}$ as:

$$\mathcal{A} \quad \triangleq \quad \{\,(p,\ 1) \mid p \text{ is a prime factor of } n.\,\} \ .$$

Since for all primes p and q with $p \neq q$,

$$(\overline{p \times 1} - \overline{1}) \ \cap \ (\overline{q \times 1} - \overline{1}) \quad = \quad \{p\} \ \cap \ \{q\} = \varnothing \ ,$$

we have that $\mathcal{A}$ satisfies the condition (4.12). Further, β_R is constructed from $\mathcal{A}$ as:

$$\beta_R \quad \triangleq \quad \langle \Pi\,(p,\ 1) \in \mathcal{A}\ ::\ \Gamma.(p,\ 1) \rangle + T^m$$

where m satisfies that

$$0 < m < \langle \Sigma\,(p,\ 1) \in \mathcal{A}\ ::\ (p-1) \rangle \ .$$

The corresponding multiset R of β_R is a monotonic solvable replacement-set. As an instance of this class, by taking n to be 6 and m to be 2, we have:

$$\begin{aligned} \beta_R &= \Gamma.(3,\ 1) \times \Gamma.(2,\ 1) + T^2 \\ &= (T^2 + T + 1) \times (T + 1) + T^2 \\ &= T^3 + 3T^2 + 2T + 1 \ . \end{aligned}$$

The corresponding multiset R is:

$$\{\!|\ -2,\ 2 * (-1),\ 3 * 0,\ 1\ |\!\}$$

where $-2,\ -1,\ 0$, and 1 are relative positions to the position $m = 2$. Further, the game $(R,\ n)$ is solvable which follows from:

$$\beta_R - T^m \quad = \quad \Gamma.(3,\ 1) \times \Gamma.(2,\ 1) \quad = \quad \Phi_3 \times \Phi_2 \quad = \quad \frac{T^6 - 1}{\Phi_1 \times \Phi_6}$$

and theorem 4.4.4.

Lots of other examples can be constructed as above by using the product (4.11) with $\mathcal{A}$ satisfying the condition (4.12). We list some of them in the following table:

$\mathcal{A}$	$\beta_R - T^m$	n
$\{(k,\ 1)\}$	$\frac{T^k-1}{T-1} = \langle \Sigma i\ :\ 0 \leqslant i < k\ :\ T^i \rangle$	k
$\{(2,\ 2k)\}$	$\frac{T^{4k}-1}{T^{2k}-1} = T^{2k} + 1$	$4k$
$\{(a,\ a^{k-1})\}$	$\frac{T^{a^k}-1}{T^{a^{k-1}}-1} = \langle \Sigma i\ :\ 0 \leqslant i < a\ :\ T^{i \times a^{k-1}} \rangle$	a^k
$\{(3,\ 1),\ (2,\ 8)\}$	$\frac{T^3-1}{T-1} \times \frac{T^{16}-1}{T^8-1} = (T^2 + T + 1) \times (T^8 + 1)$	48

The second class in the above table was previously identified by Marcelo Fiore [private communication, 2010]. The third class was studied in [BCF10].

Notice that the condition (4.12) on the set $\mathcal{A}$ is equivalent to:

$$(4.13) \qquad \langle \forall\, (a,\ b),\ (c,\ d) \in \mathcal{A}\ ::\ ((a \times b)\ \nabla\ (c \times d)) \setminus b \quad \vee \quad ((a \times b)\ \nabla\ (c \times d)) \setminus d \rangle$$

where the symbol ∇ denotes the greatest common divisor. The advantage of (4.13) over (4.12) is that the computation of the greatest common divisor is cheaper than the construction of the set of positive divisors. Now, let us show they are equivalent as follows. By duality between propositional operators and set operators, we have that for all sets $A,\ B,\ C$, and D:

$$(A - B)\ \cap\ (C - D) = \varnothing \quad \equiv \quad (A\ \cap\ C)\ \subseteq\ B \quad \vee \quad (A\ \cap\ C)\ \subseteq\ D$$

which follows from that for all propositions $p,\ q,\ r$, and s:

$$\begin{aligned} & \neg((p\ \wedge\ \neg q)\ \wedge\ (r\ \wedge\ \neg s)) \\ = \quad & \{\ \text{De Morgan rule}\ \} \\ & \neg p\ \vee\ q\ \vee\ \neg r\ \vee\ s \end{aligned}$$

$$= \quad \{ \text{ idempotency and commutativity } \}$$

$$\neg p \vee \neg r \vee q \vee \neg p \vee \neg r \vee s$$

$$= \quad \{ \text{ De Morgan rule } \}$$

$$\neg (p \wedge r) \vee q \vee \neg (p \wedge r) \vee s$$

$$= \quad \{ \text{ defintion of } \Rightarrow \ \}$$

$$((p \wedge r) \Rightarrow q) \vee ((p \wedge r) \Rightarrow s) \ .$$

Instantiating A, B, C, and D by $\overline{a \times b}$, $\overline{b}$, $\overline{c \times d}$, and $\overline{d}$, we have that

$$(\overline{a \times b} - \overline{b}) \cap (\overline{c \times d} - \overline{d}) = \varnothing \quad \equiv \quad (\overline{a \times b} \cap \overline{c \times d}) \subseteq \overline{b} \quad \vee \quad (\overline{a \times b} \cap \overline{c \times d}) \subseteq \overline{d} \ .$$

Further, since for all positive natural numbers a and b, $\overline{a} \cap \overline{b} = \overline{a \nabla b}$ and $a \setminus b \equiv \overline{a} \subseteq \overline{b}$. We get:

$$(\overline{a \times b} - \overline{b}) \cap (\overline{c \times d} - \overline{d}) = \varnothing \quad \equiv \quad ((a \times b)\ \nabla\ (c \times d)) \setminus b \quad \vee \quad ((a \times b)\ \nabla\ (c \times d)) \setminus d \ .$$

This completes the proof.

In summary, we have:

Theorem 4.5.1. Given a finite set $\mathcal{A}$ of pairs of positive natural numbers which satisfies:

$$\langle \forall (a,\ b), (c,\ d) \in \mathcal{A} \ :: \ ((a \times b)\ \nabla\ (c \times d)) \setminus b \quad \vee \quad ((a \times b)\ \nabla\ (c \times d)) \setminus d \rangle \ ,$$

the corresponding multiset R of

$$\beta_R \quad \triangleq \quad \langle \Pi (a,\ b) \in \mathcal{A} \ :: \ \Gamma.(a,\ b) \rangle + T^m$$

with $0 < m < deg.\beta_R$ is a monotonic solvable replacement-set with displacement n positive multiples of the least common multiple of elements in

$$\langle \cup (a,\ b) \in \mathcal{A} \ :: \ \overline{a \times b} - \overline{b} \rangle \ .$$

Is theorem 4.5.1 the only way to define monotonic solvable replacement-sets? The answer is *no*. In our investigation, we notice an interesting example:

$$\frac{T^{15} - 1}{T^5 - 1} \times \frac{T^8 - 1}{T^3 - 1} = \Gamma.(3,\ 5) \times \frac{\Gamma.(8,\ 1)}{\Gamma.(3,\ 1)} = \Phi_{15} \times \Phi_8 \times \Phi_4 \times \Phi_2 \ .$$

Although

$$\Phi_{15} \quad = \quad T^8 - T^7 + T^5 - T^4 + T^3 - T + 1$$

is *not* a cyclotomic polynomial with natural numbers as coefficients,

$$\Phi_{15} \times \Phi_8 \times \Phi_4 \times \Phi_2 \quad = \quad T^{15} + T^{12} + T^{10} + T^9 + T^6 + T^5 + T^3 + 1$$

is a product of cyclotomic polynomials with natural numbers as coefficients.

Generally, we are interested in the problem of finding a natural number c which satisfies that

$$\frac{T^{a \times b} - 1}{T^b - 1} \times \frac{T^c - 1}{T^a - 1} \quad = \quad \Gamma.(a,\ b) \times \frac{\Gamma.(c,\ 1)}{\Gamma.(a,\ 1)} \tag{4.14}$$

is a product of cyclotomic polynomials with natural numbers as coefficients provided that positive natural numbers a and b are *coprime*, written as $a \perp b$.

Notice that by using the geometric series, we have:

$$\frac{1}{1 - T^a} \quad = \quad \langle \Sigma i \ : \ 0 \leqslant i \ : \ T^{a \times i} \rangle \ .$$

Further, the formula (4.14) can be rewritten as:

$$\begin{array}{ll}
& \langle \Sigma i \ : \ 0 \leqslant i < a \ : \ T^{b \times i} \rangle \times \langle \Sigma i \ : \ 0 \leqslant i \ : \ T^{a \times i} \rangle \times (1 - T^c) \\
= & \{ \ \text{polynomial arithmetic} \ \} \\
& \langle \Sigma k,\ j \ : \ 0 \leqslant k \ \wedge \ 0 \leqslant j < a \ : \ T^{a \times k + b \times j} \rangle \ - \\
& \langle \Sigma k,\ i \ : \ 0 \leqslant k \ \wedge \ 0 \leqslant i < a \ : \ T^{a \times k + b \times i + c} \rangle \\
= & \{ \ \text{factorization on powers} \ \} \\
& \langle \Sigma k,\ j \ : \ 0 \leqslant k \ \wedge \ 0 \leqslant j < a \ : \ T^{a \times (k + \lfloor \frac{b \times j}{a} \rfloor) + (b \times j) \ mod \ a} \rangle \ - \\
& \langle \Sigma k,\ i \ : \ 0 \leqslant k \ \wedge \ 0 \leqslant i < a \ : \ T^{a \times (k + \lfloor \frac{b \times i + c}{a} \rfloor) + (b \times i + c) \ mod \ a} \rangle \\
= & \{ \ \text{renaming: } k := k - \lfloor \frac{b \times j}{a} \rfloor \text{ and } k := k - \lfloor \frac{b \times i + c}{a} \rfloor \text{ respectively} \ \} \\
& \langle \Sigma k,\ j \ : \ \lfloor \dfrac{b \times j}{a} \rfloor \leqslant k \ \wedge \ 0 \leqslant j < a \ : \ T^{a \times k + (b \times j) \ mod \ a} \rangle \ - \\
& \langle \Sigma k,\ i \ : \ \lfloor \dfrac{b \times i + c}{a} \rfloor \leqslant k \ \wedge \ 0 \leqslant i < a \ : \ T^{a \times k + (b \times i + c) \ mod \ a} \rangle \ .
\end{array}$$

Considering the last line in the above calculation. By comparing terms on both sides of the minus operator, we have that the formula (4.14) is a product of cyclotomic polynomials with natural numbers as coefficients if and only if there is an *injective* function

$$f \;:\; [0,\ a) \to [0,\ a)$$

satisfying that

$$(4.15) \qquad \langle \forall i \;:\; 0 \leqslant i < a \;:\; (b \times i + c)\ \mathit{mod}\ a = (b \times f.i)\ \mathit{mod}\ a \ \wedge\ b \times f.i \leqslant b \times i + c \rangle \ .$$

Our goal is to construct a function f satisfying the above property.

A useful property is:

Lemma 4.5.2. For all positive natural numbers a, b, i, and j,

$$a \perp b \quad \Rightarrow \quad (((b \times i)\ \mathit{mod}\ a = (b \times j)\ \mathit{mod}\ a) \quad \equiv \quad (i\ \mathit{mod}\ a = j\ \mathit{mod}\ a)) \ .$$

Proof.

$$\begin{array}{ll}
 & (b \times i)\ \mathit{mod}\ a = (b \times j)\ \mathit{mod}\ a \\
= & \{ \quad \text{definition of modulo} \quad \} \\
 & a \setminus ((i - j) \times b) \\
= & \{ \quad a \perp b \quad \} \\
 & a \setminus (i - j) \\
= & \{ \quad \text{definition of modulo} \quad \} \\
 & i\ \mathit{mod}\ a = j\ \mathit{mod}\ a \ .
\end{array}$$

□

This property implies that numbers from 0 to $a - 1$ appear one and only time in remainders $(b \times i)\ \mathit{mod}\ a$ for $0 \leqslant i < a$ provided $a \perp b$.

Further, let r be a number satisfying that $(b \times r)\ mod\ a = 1$. We have:

$$
\begin{array}{ll}
 & (b \times i + c)\ mod\ a = (b \times f.i)\ mod\ a \\
= & \quad \{ \quad (b \times r)\ mod\ a = 1 \quad \} \\
 & (b \times i + c \times ((b \times r)\ mod\ a))\ mod\ a = (b \times f.i)\ mod\ a \\
= & \quad \{ \quad [\,(i + j)\ mod\ a = (i\ mod\ a + j\ mod\ a)\ mod\ a\,] \text{ and} \\
 & \qquad\quad [\,(i \times j)\ mod\ a = ((i\ mod\ a) \times j)\ mod\ a\,] \quad \} \\
 & (b \times (i + c \times r))\ mod\ a = (b \times f.i)\ mod\ a \\
= & \quad \{ \quad \text{lemma 4.5.2 and } a \perp b \quad \} \\
 & (i + c \times r)\ mod\ a = f.i\ mod\ a \\
= & \quad \{ \quad f.i \in [0,\ a) \quad \} \\
 & (i + c \times r)\ mod\ a = f.i\ .
\end{array}
$$

Let us define f as: for all i in $[0,\ a)$,

$$f.i \quad \triangleq \quad (i + c \times r)\ mod\ a\ .$$

Notice that for all i_0 and i_1 in $[0,\ a)$,

$$
\begin{array}{ll}
 & f.i_0 = f.i_1 \\
= & \quad \{ \quad \text{definition of } f \quad \} \\
 & (i_0 + c \times r)\ mod\ a = (i_1 + c \times r)\ mod\ a \\
= & \quad \{ \quad \text{definition of modulo} \quad \} \\
 & a \backslash (i_0 + c \times r - i_1 - c \times r) \\
= & \quad \{ \quad \text{arithmetic and definition of modulo} \quad \} \\
 & i_0\ mod\ a = i_1\ mod\ a \\
= & \quad \{ \quad 0 \leqslant i_0,\ i_1 < a \quad \} \\
 & i_0 = i_1\ .
\end{array}
$$

That is, f is *bijective*.

It follows that the condition (4.15) is equivalent to:

$$\langle \forall i \ : \ 0 \leqslant i < a \ : \ b \times ((i + c \times r) \ mod \ a) \leqslant b \times i + c \rangle \ .$$

Also,

$$\begin{array}{ll}
 & \langle \forall i \ : \ 0 \leqslant i < a \ : \ b \times ((i + c \times r) \ mod \ a) \leqslant b \times i + c \rangle \\
= & \{ \quad \text{arithmetic} \quad \} \\
 & \langle \forall i \ : \ 0 \leqslant i < a \ : \ b \times ((i + c \times r) \ mod \ a - i) \leqslant c \rangle \\
= & \{ \quad -a < (i + c \times r) \ mod \ a - i < a \quad \} \\
 & \langle \forall i \ : \ 0 \leqslant i < a \ : \ b \times (((i + c \times r) \ mod \ a - i) \ mod \ a) \leqslant c \rangle \\
= & \{ \quad 0 \leqslant i < a \quad \} \\
 & \langle \forall i \ : \ 0 \leqslant i < a \ : \ b \times (((i + c \times r) \ mod \ a - i \ mod \ a) \ mod \ a) \leqslant c \rangle \\
= & \{ \quad [\,(i - j) \ mod \ a = (i \ mod \ a - j \ mod \ a) \ mod \ a\,] \quad \} \\
 & \langle \forall i \ : \ 0 \leqslant i < a \ : \ b \times ((c \times r) \ mod \ a) \leqslant c \rangle \\
= & \{ \quad \text{distribution and unit of } \forall \quad \} \\
 & b \times ((c \times r) \ mod \ a) \leqslant c \ .
\end{array}$$

We have that the formula (4.14) is a product of cyclotomic polynomials with natural numbers as coefficients if and only if

$$(4.16) \qquad b \times ((c \times r) \ mod \ a) \leqslant c$$

where r satisfies that $(b \times r) \ mod \ a = 1$.

Of course, we need to make sure that the formula (4.14) is a factor of $T^n - 1$ for some positive natural number n. By applying similar arguments which are used in the proof of theorem 4.5.1, we get the following condition:

$$(\overline{a \times b} - \overline{b}) \ \cap \ (\overline{c} - \overline{a}) \quad = \quad \varnothing \ ,$$

that is,

$$((a \times b) \ \nabla \ c) \setminus a \quad \vee \quad ((a \times b) \ \nabla \ c) \setminus b \ ,$$

under which the formula (4.14) is a factor of $T^n - 1$. Notice that

$$\begin{array}{ll}
 & ((a \times b) \mathbin{\nabla} c) \backslash a \\
= & \{ \quad \text{definition of the greatest common divisor} \quad \} \\
 & (a \times b) \mathbin{\nabla} c \quad = \quad (a \times b) \mathbin{\nabla} c \mathbin{\nabla} a \\
= & \{ \quad (a \times b) \mathbin{\nabla} a = a \quad \} \\
 & (a \times b) \mathbin{\nabla} c \quad = \quad a \mathbin{\nabla} c \\
= & \{ \quad \text{definition of the greatest common divisor} \quad \} \\
 & \langle \forall k \;:\; 1 < k \;:\; k \backslash (a \times b) \;\wedge k \backslash c \quad \equiv \quad k \backslash a \;\wedge\; k \backslash c \rangle \\
= & \{ \quad [p \;\wedge\; q \quad \equiv \quad p \;\wedge\; r \quad \equiv \quad (p \quad \Rightarrow \quad (q \quad \equiv \quad r))] \quad \} \\
 & \langle \forall k \;:\; 1 < k \;:\; k \backslash c \quad \Rightarrow \quad (k \backslash (a \times b) \quad \equiv \quad k \backslash a) \rangle \\
= & \{ \quad a \perp b \;\wedge\; (k \backslash (a \times b) \;\equiv\; k \backslash a) \text{ implies } \neg(k \backslash b) \text{ and transitivity} \quad \} \\
 & \langle \forall k \;:\; 1 < k \;:\; k \backslash c \quad \Rightarrow \quad \neg(k \backslash b) \rangle \\
= & \{ \quad \text{definition of the coprime} \quad \} \\
 & c \perp b \;.
\end{array}$$

Similarly,

$$((a \times b) \mathbin{\nabla} c) \backslash b \quad \equiv \quad c \perp a \;.$$

Combining with the condition (4.16), we get:

Theorem 4.5.3. For all positive natural numbers a, b, and c with $a \perp b$, the product

$$\frac{T^{a \times b} - 1}{T^b - 1} \times \frac{T^c - 1}{T^a - 1} \quad = \quad \Gamma.(a,\ b) \times \frac{\Gamma.(c,\ 1)}{\Gamma.(a,\ 1)}$$

is a factor of $T^n - 1$ with natural numbers as coefficients for some positive natural number n if and only if

$$b \times ((c \times r) \mathit{\ mod\ } a) \leqslant c \;\wedge\; ((c \perp a) \;\vee\; (c \perp b))$$

with r satisfying that $(b \times r) \mathit{\ mod\ } a = 1$.

Let us give an example to finish this subsection. Taking a and b to be 5 and 6 respectively. We have $(6 \times 1)\ mod\ 5 = 1$. Thus, we can choose r to be 1. Notice that $c = 11$ satisfies that

$$6 \times ((11 \times 1)\ mod\ 5) \leqslant 11 \ \wedge\ 11 \perp 6\ .$$

By theorem 4.5.3, the following product

$$\begin{aligned} & \frac{T^{5\times 6}-1}{T^6-1} \times \frac{T^{11}-1}{T^5-1} \quad = \quad \Gamma.(a,\ b) \times \frac{\Gamma.(c,\ 1)}{\Gamma.(a,\ 1)} \\ = \quad & \Phi_{30} \times \Phi_{15} \times \Phi_{10} \times \Phi_{11} \\ = \quad & T^{30} + T^{25} + T^{24} + T^{20} + T^{18} + T^{15} + T^{12} + T^{10} + T^6 + T^5 + 1 \end{aligned}$$

is a factor of $T^{330} - 1$ with natural numbers as coefficients.

4.5.2 True Solvable Replacement-Sets

Recall that a true solvable replacement-set is based on a polynomial $\beta_R - T^m$ with only one negative coefficient which divides $T^n - 1$ for some positive natural number n. For instance, the nuclear pennies game $(\{\!|\ -1,\ 1\ |\!\},\ 6)$ is a true solvable replacement-set game based on the polynomial $T^2 + 1 - T$. We are wondering whether this is the only instance of true solvable replacement-set games. The answer to this question is *no.* In this subsection, we give an infinite class of true solvable replacement-sets, although its construction is in an ad-hoc way.

Lemma 4.5.4. For all distinct primes p and q, the product

$$\Phi_{p\times q} \times \Gamma.(p \times q - p - q,\ 1)$$

is a factor of $T^n - 1$ with only one negative coefficient. Moreover, n is a positive multiple of the greatest common multiple of $p \times q$ and $p \times q - p - q$.

Proof. By definitions of Φ and Γ, the above product can be rewritten as:

$$\langle \Pi k \ :\ k = p \times q \ \vee\ (k \setminus (p \times q - p - q) \ \wedge\ k \neq 1) \ :\ \Phi_k \rangle\ .$$

Since $p\times q$ is greater than $p\times q-p-q$, we have that $p\times q$ is not a divisor of $p\times q-p-q$. Further, there is no repeated cyclotomic polynomials in the above product. Thus, it is a factor of T^n-1 with n positive multiples of the least common multiple of $p\times q$ and $p\times q-p-q$.

We now show that the above product has only one negative coefficient. Notice that

$$\Phi_{p\times q}\times\Gamma.(p\times q-p-q,\ 1)+T^{p\times q-p-q}$$

$$=\qquad\{\quad \text{definitions of } \Phi \text{ and } \Gamma \quad\}$$

$$\frac{T^{p\times q}-1}{T^p-1}\times\frac{T-1}{T^q-1}\times\frac{T^{p\times q-p-q}-1}{T-1}+T^{p\times q-p-q}$$

$$=\qquad\{\quad \text{polynomial arithmetic} \quad\}$$

$$\frac{T^{2\times p\times q-p-q}-T^{p\times q}-T^{p\times q-p-q}+1}{(T^p-1)\times(T^q-1)}+\frac{T^{p\times q}-T^{p\times q-q}-T^{p\times q-p}+T^{p\times q-p-q}}{(T^p-1)\times(T^q-1)}$$

$$=\qquad\{\quad \text{polynomial arithmetic} \quad\}$$

$$\frac{T^{(p-1)\times q+(q-1)\times p}-T^{(p-1)\times q}-T^{(q-1)\times p}+1}{(T^p-1)\times(T^q-1)}$$

$$=\qquad\{\quad \text{factorization} \quad\}$$

$$\frac{T^{(q-1)\times p}-1}{T^p-1}\times\frac{T^{(p-1)\times q}-1}{T^q-1}$$

$$=\qquad\{\quad \text{definition of } \Gamma \quad\}$$

$$\Gamma.(q-1,\ p)\times\Gamma.(p-1,\ q)\ .$$

That is,

$$\Phi_{p\times q}\times\Gamma.(p\times q-p-q,\ 1)+T^{p\times q-p-q}$$

has no negative coefficients. Thus, if the $(\,p\times q-p-q\,)$-th coefficient of

$$\Gamma.(q-1,\ p)\times\Gamma.(p-1,\ q)$$

is 0, then $\Phi_{p\times q}\times\Gamma.(p\times q-p-q,\ 1)$ has only one negative coefficient. By the definition of Γ and polynomial multiplication, powers of monomials in $\Gamma.(q-1,\ p)\times\Gamma.(p-1,\ q)$ are $i\times p+j\times q$ for $0\leqslant i<q-1$ and $0\leqslant j<p-1$. But,

$$i\times p+j\times q=p\times q-p-q$$

$$=\qquad\{\quad \text{arithmetic} \quad\}$$

$$
\begin{array}{ll}
 & (i+1)\times p + (j+1)\times q = p\times q \\
= & \{\quad p \text{ and } q \text{ are primes} \quad\} \\
 & (i+1=q \ \wedge\ j+1=0) \ \vee\ (i+1=0 \ \wedge\ j+1=p) \\
= & \{\quad 0\leqslant i<q-1 \text{ and } 0\leqslant j<p-1 \quad\} \\
 & \mathit{false}\ .
\end{array}
$$

That is, the $(\,p\times q-p-q\,)$-th coefficient of $\Gamma.(q-1,\ p)\times\Gamma.(p-1,\ q)$ is indeed 0. This completes the proof. □

As an example, by taking p and q to be 2 and 5 respectively, we get:

$$
\begin{array}{ll}
 & \Phi_{2\times 5}\times\Gamma.(2\times 5-2-5,\ 1) \\
= & \Phi_{10}\times\Gamma.(3,\ 1) \\
= & \dfrac{T^{10}-1}{T^{5}-1}\times\dfrac{T-1}{T^{2}-1}\times\dfrac{T^{3}-1}{T-1} \\
= & \dfrac{(T^{5}+1)\times(T^{2}+T+1)}{T+1} \\
= & T^{6}+T^{4}-T^{3}+T^{2}+1
\end{array}
$$

is a factor of $T^{30}-1$. Further, $(\{\!|\,-3,\ -1,\ 1,\ 3\,|\!\},\ 30)$ is a true solvable replacement-set game.

Chapter 5

Conclusion

Lawvere's Remark is the origin of this thesis. So far as we are aware, Blass gave the first explanation of Lawvere's Remark. As a milestone, Fiore and Leinster's result generalizes Lawvere's Remark with respect to *single* recursive type isomorphims.

Inspired by Fiore and Leinster's research, we extend their results to recursive type isomorphism systems on a finite set of types. We give a sufficient and necessary condition under which a given recursive type isomorphism system forms a *ring*. This theory shows that some isomorphisms between objects can be decided by using *polynomial division algorithm on multi-variables*.

Another interesting aspect of Lawvere's Remark is that it can be illustrated by a one-person board game — the nuclear pennies game. Fiore and Leinster's results predict that there is a solution to the nuclear pennies game. However, how one derives an algorithm to produce such a solution is not clear. We introduce an infinite class of one-person board games, so-called *replacement-set games*, which has the nuclear pennies game as an instance. An algorithm is constructed to give solutions to these games when they are solvable.

Until now, our theory has built a clear connection between algebraic equations on complex numbers and recursive type isomorphism systems. The significance of this connection is that methods in computational algebra can be introduced as *short cuts* to solve some problems on recursively defined objects which are in every corner of computer science. Conversely, we can give algorithmic explanations to some calculations in computational algebra.

5.1 Further Work

5.1.1 Primitive Recursions on Inductive Types

As we have shown in section 3.5, some interesting isomorphisms can be constructed when the *List* type and primitive recursions are introduced to the free distributive category $\mathcal{C}[T]$ on the inductive type T of binary trees. Generally, we are interested in the algebraic structure of the free distributive category $\mathcal{C}[\mathfrak{T}]$ on the finite set $\mathfrak{T}$ of inductive types equipped with the system $\mathfrak{S}$ of inductive type isomorphisms and primitive recursions. It seems that multiplicative inverses of non-trivial polynomial inductive types can be constructed. This needs more investigation.

5.1.2 Construction of Solvable Replacement-Sets

The set of all solvable replacement-set games can be characterized by the set of all products of cyclotomic polynomials with at most one negative coefficient. By using properties of cyclotomic polynomials, several ad-hoc methods are developed to construct some subsets of the set of all solvable replacement-sets. However, the problem of constructing the complete set of all solvable replacement-set games is still open.

5.1.3 Two-Dimensional Replacement-Set Games

Let us consider two-dimensional replacement-set games. For instance, the following system of recursive type isomorphisms:

$$\begin{cases} S \;\cong\; 1 + S^2 \times T \; ; \\ T \;\cong\; 1 + S \times T^2 \; . \end{cases}$$

can be considered as the two-dimensional replacement-set game:

$$\{\,(1,\ 0)\,\} \;\leftrightarrow\; \{\,(0,\ 0),\ (2,\ 1)\}$$

$$\{\,(0,\ 1)\,\} \;\leftrightarrow\; \{\,(0,\ 0),\ (1,\ 2)\}$$

where terms $S^m \times T^n$ are characterized by vectors $(m,\ n)$ for all natural numbers m and n.

Our theory predicts that to move an initial checker from square $(1,\ 0)$ to square $(0,\ 1)$ by using the above replacement rules is possible. This can be proved by the factorization:

$$x - y = x \times (x \times y^2 - y + 1) - y \times (x^2 \times y - x + 1) \; . \tag{5.1}$$

Also, we can verify it using the following calculation on types:

$$\begin{aligned} S \;&\cong\; 1 + S^2 \times T \\ &\cong\; 1 + S \times T + S^3 \times T^2 \\ &\cong\; 1 + T + S^2 \times T^2 + S^3 \times T^2 \\ &\cong\; 2 + S \times T^2 + S^2 \times T^2 + S^3 \times T^2 \\ &\cong\; 2 + S \times T^2 + S^2 \times T^2 + S^2 \times T^3 + S^3 \times T^2 \\ &\cong\; 2 + S^2 \times T + S^2 \times T^2 + S^2 \times T^3 \\ &\cong\; 1 + S + S^2 \times T^2 + S^2 \times T^3 \\ &\cong\; 1 + S \times T + S^2 \times T^3 \\ &\cong\; 1 + S \times T^2 \\ &\cong\; T \; . \end{aligned}$$

How to derive an algorithm to give a solution to a solvable two-dimensional replacement-set game could be a further research topic. On the other hand, in order to get factorizations such as (5.1), we need a polynomial division algorithm on $\mathbb{Z}[x, y]$. Whether this idea works for general case needs more investigation.

Appendix A

One-Dimensional Replacement-Set Games

The algorithm to solve one-dimensional replacement-set games is implemented in Haskell as following:

```
-- Create : 06/02/2010
-- Last Modification : 03/09/2011
-- Author : Wei Chen
-- Facility : University of Nottingham
-- Description: One-Dimensional Replacement-Set Games

import Prelude hiding (min, max, drop, repeat, pred, seq)

-- A. Polynomial Arithmetic with Integer as Coefficients

-- A polynomial is represented by a list of integer pairs.
-- For every pair, the first coordinate is the coefficient
-- and the second coordinate is the power.

type Poly = [(Int, Int)]
```

```
-- polynomial addition

plus :: Poly → Poly → Poly
plus p [] = p
plus [] q = q
plus (x:p) (y:q)
  | (snd x) < (snd y) = x : plus p (y:q)
  | (snd x) > (snd y) = y : plus (x:p) q
  | (snd x) == (snd y)=
  if n == 0 then plus p q
  else (n, snd x) : plus p q
    where n = fst x + fst y

-- polynomial subtraction

minus :: Poly → Poly → Poly
minus p q = plus p (zip (map (((-1)*) ∘ fst) q) (map snd q))

-- polynomial multiplication

mult :: Poly → Poly → Poly
mult p [] = []
mult [] q = []
mult (x:p) q =
 plus (time x q) (mult p q)
  where
   time x [] = []
   time x (y:q) =
            ((fst x) * (fst y), (snd x) + (snd y)) : time x q

-- the degree of a polynomial

deg :: Poly → Int
deg = snd ∘ last
```

```
-- the codegree of a polynomial

cod :: Poly → Int
cod = snd ∘ head

-- the coefficient of the highest term of a polynomial

hcof :: Poly → Int
hcof = fst ∘ last

-- polynomial division

quotient :: Poly → Poly → Poly
quotient [] q = []
quotient p [] = []
quotient p q =
  if (deg p) < (deg q)
                || (hcof p ‘mod‘ hcof q ≠ 0) then []
  else plus r (quotient (minus p (mult r q)) q)
    where r = [(hcof p ‘div‘ hcof q, deg p - deg q)]

remainder :: Poly → Poly → Poly
remainder p q = minus p (mult (quotient p q) q)

-- B. Cyclotomic Polynomials

-- prime number test

prime :: Int → Bool
prime 1 = False
prime n = least_divisor_from 2 == n
  where { least_divisor_from d =
    if d == n || n ‘mod‘ d == 0 then d
    else least_divisor_from (d + 1) }
```

```
-- the number of prime divisors of a given natural number

prime_divisor :: Int → Int
prime_divisor 0 = 0
prime_divisor 1 = 0
prime_divisor n =
  if prime n then 1
  else iter 2 0
  where { iter d c
    | d == n = c
    | n ‘mod‘ d == 0 && prime d = iter (d + 1) (c + 1)
    | otherwise = iter (d + 1) c }

-- square-free test
-- e.g. 12 is not square-free, since 2^2 is a factor of 12.

square_free :: Int → Bool
square_free 0 = True
square_free 1 = True
square_free n =
  if prime n then True
  else iter 0 1 2 n
  where { iter pre cur d m
    | pre == cur = False
    | m == 1 = True
    | m ‘mod‘ d == 0 && prime d = iter cur d d (m ‘div‘ d)
    | otherwise = iter pre cur (d + 1) m }

-- the Möbius Function

mu :: Int → Int
mu n
  | n == 1 = 1
  | square_free n = (-1) ^ (prime_divisor n)
  | otherwise = 0
```

```
-- cyclotomic polynomials by using the Möbius Function

phi :: Int → Poly
phi n = iter 1 [(1,0)] [(1,0)]
  where { iter d p q
    | d > n = quotient p q
    | n ‘mod‘ d == 0
      && mu (n ‘div‘ d) == 1 = iter (d + 1) (mult p r) q
    | n ‘mod‘ d == 0
      && mu (n ‘div‘ d) == -1 = iter (d + 1) p (mult q r)
    | otherwise = iter (d + 1) p q
    where r = [(-1,0), (1,d)] }

-- C. Constructing Compound Expansions

-- the least element of a list

min :: [Int] → Int
min [x] = x
min (x:y:p) = if x < y then min (x:p)
              else min (y:p)

-- the greatest element of a list

max :: [Int] → Int
max [x] = x
max (x:y:p) = if x > y then max (x:p)
              else max (y:p)

-- membership test

member :: Int → [Int] → Bool
member x [] = False
member x (y:p) = if x == y then True
```

```
                 else member x p

-- set difference

diff :: [Int] → [Int] → [Int]
diff [] b = []
diff (x:p) b = if member x b then diff p b
               else x:(diff p b)

-- the base set of a multiset

set :: [Int] → [Int]
set [] = []
set (x:p) = if member x p then set p
            else x:(set p)

-- Euclidean Algorithm for lists of integers

ggcd :: [Int] → Int
ggcd [x] = x
ggcd (x:p) = gcd x (ggcd p)
  where { gcd m n
    | m == n = m
    | m == 0 = n
    | n == 0 = m
    | m < n = gcd m (n - m)
    | m > n = gcd (m - n) n }

-- multisets for compound expansions

-- The input is the replacement-set R
-- satisfying that min.R < 0 < max.R.
-- The outputs are two multisets A and B.
-- satisfying that the sum of A is -1 and the sum of B is 1.
```

```
eggcd :: [Int] → ([Int], [Int])
eggcd r = iter (diff (set r) [min r, max r])
               [min r] [max r] (min r) (max r)
  where { iter q a b x y
    | x == -g && y == g = (a,b)
    | -x < y = iter q a (a++b) x (x+y)
    | -x > y = iter q (a++b) b (x+y) y
    | q ≠ []
      && (min q) < 0 = iter (diff q [min q])
                            [min q] b (min q) y
    | q ≠ []
      && (max q) > 0 = iter (diff q [max q])
                            a [max q] x (max q)
    | q == []  = iter q a b x y
    where g = ggcd (map abs (set r)) }

-- remove an element from a list

drop :: Int → [Int] → [Int]
drop x (y:p) = if x == y then p
               else y:(drop x p)

-- serialization

serial :: ([Int], [Int]) → [Int]
serial (a,b) = iter [0] (drop (min a) a) (drop (max b) b)
                        (min a) (max b)
  where { iter l a b i j
    | a == [] && b == [] = l
    | a ≠ [] = iter (l++[i]) (drop (min a) a) b
                     (i+(min a)) j
    | b ≠ [] = iter (l++[j]) a (drop (max b) b)
                     i (j+(max b)) }

-- the greatest coefficient of a polynomial
```

```
max_coef :: Poly → Int
max_coef [x] = fst x
max_coef (x:y:p) = if fst x < fst y then max_coef (y:p)
                   else max_coef (x:p)

-- repeat a list for n times

repeat :: [Int] → Int → [Int]
repeat l 0 = []
repeat l 1 = l
repeat l n = l ++ (repeat l (n-1))

-- constructing a polynomial from a multiset of powers

pol :: [Int] → Poly
pol [] = []
pol (x:p) = plus [(1,x)] (pol p)

-- the multiset of all powers of a list

pow :: Poly → [Int]
pow [] = []
pow (x:p) = (iter (abs (fst x)) [snd x]) ++ (pow p)
  where
    iter 1 p = p
    iter n (x:p) = iter (n-1) (x:(x:p))

-- D. Expansion and Contraction Sequences

-- The inputs are a replacement-set and a displacement
-- which is supposed to be greater than 0.
-- The output is a solution sequence when the game is solvable.

seq :: [Int] → Int → ([Int], [Int])
seq [] m = error "There is no valid seq!\n"
seq [x] m = if x > 0 && m ‘mod‘ x == 0
              then ([ i | i ← [0..m-1], i ‘mod‘ x == 0], [])
            else if x < 0 && m ‘mod‘ x == 0
              then ([], [ i | i ← [1..m], i ‘mod‘ x == 0])
            else error "There is no valid seq!\n"
seq r m =
  if min r < 0 && max r > 0
     && remainder [(-1,0),(1,m)] (minus (pol r) [(1,0)]) == []
  then iter 0 (serial (eggcd r)) (serial (eggcd r))
  else error "There is no valid seq!\n"
    where {
      iter k a b
        | k < m = iter (k+g) (a++h) (h++b)
        | otherwise =
            ((repeat a c) ++ la, reverse ((repeat b c) ++ lb))
      where
        l = (serial (eggcd r))
        g = ggcd (map abs (set r))
        h = (map (+(k+g)) l)
        u = (quotient [(-1,0),(1,m)] (minus (pol r) [(1,0)]))
        la = pow (filter ((>0).fst) u)
        lb = pow (filter ((<0).fst) u)
        c = max_coef u }


-- E. Interface


show_poly [] = "0"
show_poly (x:p)
  | fst x == 1
    && snd x ≠ 0 = "T^" ++ show (snd x) ++ iter p
  | fst x == 1
    && snd x == 0 = "1" ++ iter p
  | fst x == -1
    && snd x ≠ 0 = " - "
```

```
          ++ "T^" ++ show (snd x) ++ iter p
 | fst x == -1
  && snd x == 0 = " - 1" ++ iter p
 | snd x ≠ 0 =
   show (fst x) ++ "T^" ++ show (snd x) ++ iter p
 | snd x == 0 = show (fst x) ++ iter p
 where
  iter [] = ""
  iter (x:p)
   | fst x == 1
    && snd x ≠ 0 = " + "
            ++ "T^" ++ show (snd x) ++ iter p
   | fst x == 1
    && snd x == 0 = " + 1" ++ iter p
   | fst x == -1
    && snd x ≠ 0 = " - "
            ++ "T^" ++ show (snd x) ++ iter p
   | fst x == -1
    && snd x == 0 = " - 1" ++ iter p
   | fst x > 1
    && snd x ≠ 0 = " + " ++ show (fst x)
            ++ "T^" ++ show (snd x) ++ iter p
   | fst x > 1
    && snd x == 0 = " + " ++ show (fst x) ++ iter p
   | fst x < -1
    && snd x ≠ 0 = " - " ++ show (abs (fst x))
            ++ "T^" ++ show (snd x) ++ iter p
   | fst x < -1
    && snd x == 0 = " - "
            ++ show (abs (fst x)) ++ iter p

out :: [([Int],Poly)] → IO()
out [] = putStr ""
out [x] = putStr (show (fst x))
     ≫ putStr "\t" ≫ putStrLn (show_poly (snd x))
```

```
out (x:p) = putStr (show (fst x))
          ≫ putStr "\t" ≫ putStrLn (show_poly (snd x))
          ≫ (out p)


-- F. Main Function


move :: [Int] → Int → IO()
move ms n = out (iter (seq ms n) [([], pol [0])])
  where
    iter ([], []) ps =  reverse ps
    iter ([], (x:cs)) (p:ps) =
      iter ([], cs)
           (([x],
             (plus (minus (snd p)
                          (pol (map (+x) ms)))
                   (pol [x]))) : (p : ps))
    iter ((x:es), cs) (p:ps) =
      iter (es, cs)
           (([x],
             (minus (plus (snd p)
                          (pol (map (+x) ms)))
                    (pol [x]))) : (p : ps))
```

Appendix B

Trees-In-Zero

Following the idea in section 3.5, a proof of trees-in-zero is given by the following programs implemented in Haskell.

```
-- Create : 21/04/2011
-- Last Modification : 05/09/2011
-- Author : Wei Chen
-- Facility : University of Nottingham
-- Description: Trees-In-Zero

import Prelude hiding (id)

-- A. Components

-- unit type

data I = Unit deriving (Show, Eq, Ord)

-- binary tree

data T = Leaf | Node T T deriving (Show, Eq, Ord)
```

Appendix B: Trees-In-Zero

```
-- coproduct

data Sum a b = Inl a | Inr b deriving (Show, Eq, Ord)

type  1 + T^3 = Sum I (T, (T, T))

type  1 + T + T^2 + T^3 + T^4 + T^5 = Sum I
                (Sum T
                   (Sum (T,T)
                      (Sum (T,(T,T))
                         (Sum (T,(T,(T,T)))
                            (T,(T,(T,(T,T))))))))

type  T + T^4  = Sum T (T,(T,(T,T)))

-- B. Semiring Functions

s5to1_4 :: 1 + T + T^2 + T^3 + T^4 + T^5 → T + T^4
s5to1_4 (Inl Unit) = Inl Leaf
s5to1_4 (Inr (Inl a)) = Inr (a,(Leaf, (Leaf, Leaf)))
s5to1_4 (Inr (Inr (Inl (a,b)))) = Inl (Node a b)
s5to1_4 (Inr (Inr (Inr (Inl (a,(b,c))))))
        = Inr (a,(Node b c, (Leaf, Leaf)))
s5to1_4 (Inr (Inr (Inr (Inr (Inl (a,(b,(c,d))))))))
        = Inr (a,(b,(Node c d, Leaf)))
s5to1_4 (Inr (Inr (Inr (Inr (Inr (a,(b,(c,(d,e)))))))))
        = Inr (a,(b,(c, Node d e)))

s5to1_4_i :: 1 + T + T^2 + T^3 + T^4 + T^5 → T + T^4
s5to1_4_i (Inl Leaf) = Inl Unit
s5to1_4_i (Inr (a,(Leaf, (Leaf, Leaf)))) = Inr (Inl a)
s5to1_4_i (Inl (Node a b)) = Inr (Inr (Inl (a,b)))
s5to1_4_i (Inr (a,(Node b c, (Leaf, Leaf))))
          = (Inr (Inr (Inr (Inl (a, (b, c))))))
s5to1_4_i (Inr (a,(b,(Node c d, Leaf))))
```

```
          = (Inr(Inr(Inr(Inr(Inl(a,(b,(c,d))))))))
s5to1_4_i (Inr (a,(b,(c, Node d e))))
          = (Inr(Inr(Inr(Inr(Inr(a,(b,(c,(d,e)))))))))

s1_4tol :: $T + T^4 \to 1 + T^3$
s1_4tol (Inl Leaf) = Inl Unit
s1_4tol (Inl (Node a b)) = Inr (a, (b, Leaf))
s1_4tol (Inr (a, (b, (c, d)))) = Inr (a, (b, Node c d))

s1_4tol_i :: $1 + T^3 \to T + T^4$
s1_4tol_i (Inl Unit) = Inl Leaf
s1_4tol_i (Inr (a, (b, Leaf))) = Inl (Node a b)
s1_4tol_i (Inr (a, (b, Node c d))) = Inr (a, (b, (c, d)))

st6l_1_4to1_t6l :: $(T^6)^* \times (T + T^4) \to T + (T^6)^* \times (1+T^3)$
st6l_1_4to1_t6l ([], Inl a) = Inl a
st6l_1_4to1_t6l (x, Inr (Leaf, (Leaf, (Leaf, Leaf))))
                = Inr (x, Inl Unit)
st6l_1_4to1_t6l ((a,(b,(c,(d,(e,f))))):x, Inl g)
                = Inr (x, Inr (a, (b,
                                    (Node c
                                      (Node d
                                        (Node e
                                          (Node f g)))))))
st6l_1_4to1_t6l (x, Inr (a, (b, (c, Node d e))))
                = Inr (x, Inr (a, (b,
                                    (Node c
                                      (Node d
                                        (Node e Leaf))))))
st6l_1_4to1_t6l (x, Inr (a, (b, (Node c d, Leaf))))
                = (Inr (x, Inr (a, (b, (Node c
                                         (Node d Leaf))))))
st6l_1_4to1_t6l (x, Inr (a, (Node b c, (Leaf, Leaf))))
                = Inr (x, Inr (a, (b, (Node c Leaf))))
st6l_1_4to1_t6l (x, Inr (Node a b, (Leaf, (Leaf, Leaf))))
```

```
                = Inr (x, Inr (a, (b, Leaf)))
```

st6l_1_4to1_t6l_i :: $T + (T^6)^* \times (1 + T^3) \to (T^6)^* \times (T + T^4)$

```
st6l_1_4to1_t6l_i (Inl a) =  ([], Inl a)
st6l_1_4to1_t6l_i (Inr (x, Inl Unit))
                 = (x, Inr (Leaf, (Leaf, (Leaf, Leaf))))
st6l_1_4to1_t6l_i (Inr (x, Inr (a, (b,
                                     (Node c
                                       (Node d
                                         (Node e
                                           (Node f g))))))))
                 = ((a,(b,(c,(d,(e,f))))) : x, Inl g)
st6l_1_4to1_t6l_i (Inr (x, Inr (a, (b,
                                     (Node c
                                       (Node d
                                         (Node e Leaf)))))))
                 = (x, Inr (a, (b, (c, Node d e))))
st6l_1_4to1_t6l_i (Inr (x, Inr (a, (b,
                                     (Node c
                                       (Node d Leaf))))))
                 = (x, Inr (a, (b, (Node c d, Leaf))))
st6l_1_4to1_t6l_i (Inr (x, Inr (a, (b, (Node c Leaf)))))
                 = (x, Inr (a, (Node b c, (Leaf, Leaf))))
st6l_1_4to1_t6l_i (Inr (x, Inr (a, (b, Leaf))))
                 = (x, Inr (Node a b, (Leaf, (Leaf, Leaf))))
```

s1lto1 :: $T + (1 + T^3) \to T$

```
s1lto1 (Inl a) = Node a Leaf
s1lto1 (Inr (Inl Unit)) = Leaf
s1lto1 (Inr (Inr (a, (b, c)))) = Node a (Node b c)
```

s1lto1_i :: $T \to T + (1 + T^3)$

```
s1lto1_i (Node a Leaf) = Inl a
s1lto1_i Leaf = Inr (Inl Unit)
s1lto1_i (Node a (Node b c)) = Inr (Inr (a, (b, c)))
```

```
-- C. Catamorphisms

ltot6l_i :: $1+T^3 \rightarrow 1+T^3 + T^6 \times (1+T^3)$
ltot6l_i (Inl Unit) = Inl (Inl Unit)
ltot6l_i (Inr (a,(b, Leaf))) = Inl (Inr(a,(b, Leaf)))
ltot6l_i (Inr (a,(b, Node c d))) = Inl (Inr(a,(b,Node c d)))
ltot6l_i (Inr (a,(b, (Node c Leaf))))
  = Inr ((a,(b,(c,(Leaf, (Leaf, Leaf))))), Inl Unit)
ltot6l_i (Inr (a,(b,(Node c
                      (Node d Leaf)))))
  = Inr ((a,(b,(c,(d,(Leaf, Leaf))))), Inl Unit)
ltot6l_i (Inr (a,(b,
                   (Node c
                     (Node d
                       (Node e Leaf))))))
  = Inr ((a,(b,(c,((Node d e), (Leaf, Leaf))))), Inl Unit)
ltot6l_i (Inr (a, (b,
                    (Node c
                      (Node d
                        (Node e
                          (Node f Leaf)))))))
  = Inr ((a,(b,(c,(d,((Node e f), Leaf))))), Inl Unit)
ltot6l_i (Inr (a, (b,
                    (Node c
                      (Node d
                        (Node e
                          (Node f
                            (Node g Leaf))))))))
  = Inr ((a,(b,(c,(d,(e, Node f g))))), Inl Unit)
ltot6l_i (Inr (a, (b,
                    (Node c
                      (Node d
                        (Node e
                          (Node f
                            (Node g
                              (Node h i)))))))))
  = Inr ((a,(b,(c,(d,(e,f))))), Inr (g,(h,i)))

ltot6l :: (1+T^3) +  T^6  ×  (1+T^3) → 1+T^3
ltot6l (Inl (Inl Unit)) = Inl Unit
ltot6l (Inl (Inr (a,(b, Leaf)))) = Inr (a,(b, Leaf))
ltot6l (Inl (Inr (a,(b, Node c d)))) = Inr (a,(b, Node c d))
ltot6l (Inr ((a,(b,(c,(Leaf, (Leaf, Leaf))))), Inl Unit))
       = Inr (a,(b,(Node c Leaf)))
ltot6l (Inr ((a,(b,(c,(d,(Leaf, Leaf))))), Inl Unit))
       = Inr (a,(b,(Node c
                 (Node d Leaf))))
ltot6l (Inr ((a,(b,(c,((Node d e), (Leaf, Leaf))))), Inl Unit))
       = Inr (a,(b,(Node c
                     (Node d
                       (Node e Leaf)))))
ltot6l (Inr ((a,(b,(c,(d,((Node e f), Leaf))))), Inl Unit))
       = Inr (a,(b,(Node c
                     (Node d
                       (Node e
                         (Node f Leaf))))))
ltot6l (Inr ((a,(b,(c,(d,(e,Node f g))))), Inl Unit))
       = Inr (a,(b,(Node c
                     (Node d
                       (Node e
                         (Node f
                           (Node g Leaf)))))))
ltot6l (Inr ((a,(b,(c,(d,(e,f))))), Inr (g,(h,i))))
       = Inr (a,(b,(Node c
                     (Node d
                       (Node e
                         (Node f
                           (Node g
                                (Node h i))))))))

-- identity function

id :: a → a
id x = x

-- function coproduct

add :: (a → b) → (c → d) → (Sum a c → Sum b d)
add f g (Inl x) = Inl (f x)
add f g (Inr x) = Inr (g x)

-- function product

time :: (a → b) → (c → d) → ((a, c) → (b, d))
time f g (x, y) = (f x, g y)

inT6List :: (1+T^3) +  T^6  × ((T^6)* × (1+T^3)) → (T^6)* × (1+T^3)
inT6List (Inl a) = ([], a)
inT6List (Inr (a, (x, b))) = (a:x, b)

inT6List_i :: (T^6)* × (1+T^3) → (1+T^3) +  T^6  × ((T^6)* × (1+T^3))
inT6List_i ([], a) = Inl a
inT6List_i (a:x, b) = Inr (a, (x, b))

cat_t6ltol :: (T^6)* × (1+T^3) → 1+T^3
cat_t6ltol = ltot6l ∘ (add id (time id cat_t6ltol)) ∘ inT6List_i

cat_t6ltol_i :: 1+T^3 → (T^6)* × (1+T^3)
cat_t6ltol_i = inT6List ∘ (add id (time id cat_t6ltol_i)) ∘ ltot6l_i

-- D. Between Middle and T

t6ls1_4tot :: (T^6)* × ( T + T^4) → T
```

```
t6ls1_4tot = s1lto1 $\circ$ (add id cat_t6ltol) $\circ$ st6l_1_4to1_t6l



t6ls1_4tot_i :: $T \rightarrow (T^6)^* \times (T + T^4)$
t6ls1_4tot_i = st6l_1_4to1_t6l_i $\circ$ (add id cat_t6ltol_i) $\circ$ s1lto1_i

-- E. Between Middle and $1 + T^3$

t6ls1_4tol :: $(T^6)^* \times (T + T^4) \rightarrow 1 + T^3$
t6ls1_4tol = cat_t6ltol $\circ$ (time id s1_4tol)

t6ls1_4tol_i :: $1 + T^3 \rightarrow (T^6)^* \times (T + T^4)$
t6ls1_4tol_i = (time id s1_4tol_i) $\circ$ cat_t6ltol_i

-- F. Main Functions

main :: $T \rightarrow 1 + T^3$
main = t6ls1_4tol $\circ$ t6ls1_4tot_i

main_i :: $1 + T^3 \rightarrow T$
main_i = t6ls1_4tot $\circ$ t6ls1_4tol_i
```

References

[Bac03] Roland C. Backhouse. *Program Construction: Calculating Implementations from Specifications.* Wiley, 2003.

[BCF10] Roland Backhouse, Wei Chen, and João F. Ferreira. The algorithmics of solitaire-like games. In *Proceedings of the 10th international conference on Mathematics of Program Construction*, MPC'10, pages 1–18, Berlin, Heidelberg, 2010. Springer-Verlag.

[BCF11] Roland Backhouse, Wei Chen, and João F. Ferreira. The algorithmics of solitaire-like games. *Science of Computer Programming*, Special Issue, 2011.

[Bla95] Andreas Blass. Seven trees in one. *Journal of Pure and Applied Algebra*, 103(1):1–21, 1995.

[BLFR01] E. Barcucci, A. Del Lungo, A. Frosini, and S. Rinaldi. A technology for reverse-engineering a combinatorial problem from a rational generating function. *Adv. Appl. Math.*, 26(2):129–153, 2001.

[BLL88] François Bergeron, Gilbert Labelle, and Pierre Leroux. Functional equations for data structures. In *STACS '88: Proceedings of the 5th Annual Symposium on Theoretical Aspects of Computer Science*, pages 73–80, London, UK, 1988. Springer-Verlag.

[BLL98] F. Bergeron, G. Labelle, and P. Leroux. *Combinatorial Species and Tree-like Structures.* Encyclopedia of Mathematics and Its Applications 67. Cambridge University Press, Cambridge, UK, 1998.

[BM96] Richard Bird and Oege De Moor. *Algebra of Programming.* Prentice Hall, London, 1996.

[BW98] Bruno Buchberger and Franz Winkler, editors. *Gröbner Bases and Applications.* Cambridge University Press, 1998.

[CLS07] David Cox, John Little, and Donal O' Shea. *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra.* Springer-Verlag, Berlin, third edition, 2007.

[CS63] N. Chomsky and M. P. Schützenberger. The algebraic theory of context-free languages. In P. Braffort and D. Hirschberg, editors, *Computer Programming and Formal Languages*, pages 118–161. North Holland, 1963.

[Fio04] Marcelo Fiore. Isomorphisms of generic recursive polynomial types. *SIGPLAN Not.*, 39(1):77–88, 2004.

[FL04] Marcelo Fiore and Tom Leinster. An objective representation of the gaussian integers. *Journal of Symbolic Computation*, 37:707–716, 2004.

[FL05] Marcelo Fiore and Tom Leinster. Objects of categories as complex numbers. *Advances in Mathematics*, 190(2):264–277, January 2005.

[Fla85] Philippe Flajolet. Elements of a general theory of combinatorial structures. In *FCT '85: Fundamentals of Computation Theory*, pages 112–127, London, UK, 1985. Springer-Verlag.

[FS08] Philippe Flajolet and Robert Sedgewick. *Analytic Combinatorics.* To be published by Cambridge University Press, web edition, 2008.

[Gat98] R. Gates. On the generic solution to $p(x) \cong x$ in distributive categories. *Journal of Pure and Applied Algebra*, 125:191–212, 1998.

[GJ83] I. P. Goulden and D. M. Jackson. *Combinatorial Enumeration.* John Wiley & Sons, Inc., New York, 1983.

[GKP94] Ronald L. Graham, Donald E. Knuth, and Oren Patashnik. *Concrete Mathematics: A Foundation for Computer Science.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1994.

[Gre51] J. A. Green. On the structure of semigroups. *The Annals of Mathematics, Second Series*, 54(1):163–172, 1951.

[Gri98] David Gries, editor. *The Science of Programming.* Springer-Verlag, 1998.

[GS94] David Gries and Fred B. Schneider. *A Logical Approach to Discrete Math.* Springer-Verlag, 1994.

[Hen74] Peter Henrici. *Applied and Computational Complex Analysis, Volume 1.* Pure and applied mathematics. John Wiley & Sons, Inc., New York, 1974.

[Hoo96] Paul Hoogendijk. *A Generic Theory of Data Types.* PhD thesis, Department of Computer Science, Eindhoven University of Technology, Netherlands, 1996.

[Isa94] I. Martin Isaacs. *Algebra: A Graduate Course.* Wadsworth Inc., 1994.

[JBR88] Jr. Jean Berstel and Christophe Reutenauer. *Rational Series and Their Languages.* Springer-Verlag New York, Inc., New York, NY, USA, 1988.

[Kal90] Anne Kaldewaij. *Programming: The Derivation of Algorithms.* Prentice Hall, 1990.

[Lan02] Serge Lang. *Algebra.* Springer Science+Business Media, New York, revised third edition, 2002.

[Law91] F. W. Lawvere. Some thoughts on the future of category theory. *Lecture Notes in Mathematics*, 1488:1–13, 1991.

[Mal90] G. R. Malcolm. *Algebraic Data Types and Program Transformation.* PhD thesis, Department of Computer Science, Groningen University, Netherlands, 1990.

[MB99] Saunders MacLane and Garrett Birkhoff. *Algebra.* AMS Chelsea Publishing, Providence, Rhode Island, third edition, 1999.

[Niv69] Ivan Niven. Formal power series. *The American Mathematical Monthly*, 76(8):871–889, 1969.

[Odl95] A. M. Odlyzko. Asymptotic enumeration methods. *Handbook of Combinatorics (Vol. 2)*, pages 1063–1229, 1995.

[Pip07a] Dan Piponi. Arboreal isomorphisms from nuclear pennies, September 2007. Blog post available at http://blog.sigfpe.com/2007/09/arboreal-isomorphisms-from-nuclear.html.

[Pip07b] Dan Piponi. Using thermonuclear pennies to embed complex numbers as types, October 2007. Blog post available at http://blog.sigfpe.com/2007/10/using-thermonuclear-pennies-to-embed.html.

[Sch91] Stephen. H. Schanuel. Negative sets have euler characteristic and dimension. *Lecture Notes in Mathematics*, 1488:379–385, 1991.

[SS78] Arto Salomaa and Matti Soittola. *Automata-Theoretical Aspects of Formal Power Series*. Springer-Verlag New York, Inc., New York, NY, USA, 1978.

[Sta97] Richard P. Stanley. *Enumerative Combinatorics, Volume 1*. Cambridge Studies in Advanced Mathematics 49. Cambridge University Press, Cambridge, England, 1997.

[Sta99] Richard P. Stanley. *Enumerative Combinatorics, Volume 2*. Cambridge Studies in Advanced Mathematics 62. Cambridge University Press, Cambridge, England, 1999.

[VF90] Jeffrey Scott Vitter and Philippe Flajolet. Average-case analysis of algorithms and data structures. *Handbook of Theoretical Computer Science (Vol. A): Algorithms and Complexity*, pages 431–524, 1990.

[Yor07] Brent Yorgey. The nuclear pennies game. http://www.mathelesstraveled.com /?p=80, October 2007.

[Zor04] Vladimir A. Zorich. *Mathematical Analysis (Vol. I)*. Springer-Verlag, Berlin, 2004.

Index