ELSEVIER

Theoretical Computer Science 290 (2003) 1915–1929

Theoretical Computer Science

www.elsevier.com/locate/tcs

Object grammars and bijections

I. Dutour[a,*], J.M. Fédou[b]

[a] *LaBRI, Université Bordeaux I, Unité associée au CNRS UMR 5800, 351 cours de la Libération, 33405 Talence Cedex, France*

[b] *I3S-ESSI, Université de Nice, Unité associée au CNRS UPRES-A 6070, 650 Route des Colles, B.P. 145, 06903 Sophia-Antipolis Cedex, France*

Received 3 July 2000; received in revised form 26 September 2001; accepted 4 April 2002

Communicated by M. Nivat

Abstract

A new systematic approach for the specification of bijections between sets of combinatorial objects is presented. It is based on the notion of object grammars. Object grammars give recursive descriptions of objects and generalize context-free grammars. The study of a particular substitution in these object grammars confirms once more the key role of Dyck words in the domain of enumerative and bijective combinatorics.

Keywords: Bijective combinatorics; Objects grammars; Bijections; Substitution; Dyck words

1. Introduction

An *object grammar* defines classes of objects by means of terminal objects and certain types of operations applied to the objects. It is most often described with pictures. For instance, the standard decomposition of complete binary trees is an object grammar (Fig. 1).

The formalism given here for object grammars [10] generalizes the one for context-free grammars. It is akin to the work of Flajolet et al. [13] allowing for the specification of structures by grammars involving set, sequence and cycle constructions. One can also categorize object grammars as belonging to the domain of Universal Algebra and Magmas [12,16]. Finally, our approach is related to the Theory of Species [3,14]

* Corresponding author.

E-mail addresses: isabelle.dutour@labri.fr (I. Dutour), fedou@unice.fr (J.M. Fédou).

0304-3975/02/$ - see front matter

PII: S0304-3975(02)00368-7

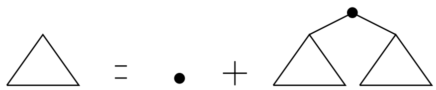

Fig. 1. Complete binary trees.

which gives a general approach to the description, construction, and enumeration of combinatorial structures on finite sets.

This paper outlines some methods using object grammars for *constructing bijections* between sets of objects. Another important application of object grammars is the *uniform random generation* of combinatorial objects. We described it precisely in [10,11]. Note that a different approach for recursively defining objects (based on "local" transformations) is also developed by Barcucci et al. [2]. The authors explore the same applications of their method named ECO: enumeration, bijection and random generation. The powers of object grammars and ECO method are quite different and complementary.

Building bijections between sets of objects is often useful in enumerative combinatorics. "Nice" bijections permit to translate a (difficult) problem into a standard one. A typical illustration of this principle is the DSV method [7,17], in which the enumeration of combinatorial objects is converted into the enumeration of words of a unambiguous algebraic language.

In this paper, we present methods for building bijections that are based on *unambiguous isomorphic object grammars*. In Section 2, we review the necessary definitions for object grammars. We then provide in Section 3 the precise notion of isomorphic object grammars. Two object grammars are isomorphic if they have the same "structure", i.e. the same number of sets of objects, terminal objects and object operations (with compatible definition domains). Isomorphic object grammars lead to "natural" bijections between the objects they generate.

In the sequel we study a transformation on object grammars called *S-iteration*. It consists in rewriting a grammar substituting one occurrence of a set of objects by its "equation". The S-iteration is similar to the *unfolding* transformation defined in the context of rewriting systems, partial evaluation technics and transformations of programs and regular systems (see for example [5,6]). Several aspects of our study of the S-iteration can be related to the work of Arnold and Dauchet concerning an equivalence relation on the class of recognizable sets of trees [1].

Section 4 is devoted to the definition of the S-iteration; Section 5 presents the main result of the paper and its consequences: from two one-dimensional unambiguous object grammars with at least one object operation of arity $\geqslant 2$, one can build two *isomorphic* object grammars by finite sequences of S-iterations. In particular, the set of Dyck words generated by the grammar $D=\varepsilon+xD\bar{x}D$ falls within this framework. Therefore, we are able to build bijections between Dyck words and all the objects that can be described by a one-dimensional unambiguous object grammar of degree $\geqslant 2$. This result confirms the crucial role of Dyck words in coding combinatorial objects (Viennot's observation in [17]). In Section 6, we prove this result. We finish by discussing some ideas and directions of research.

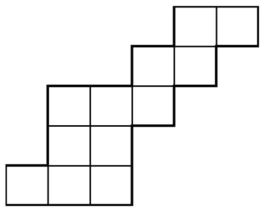

Fig. 2. A parallelogram polyomino.

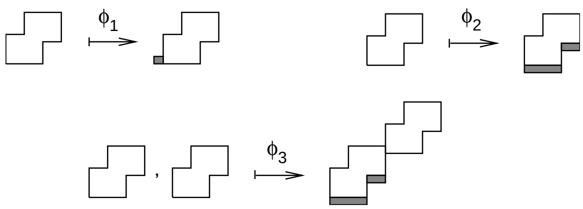

Fig. 3. Object operations on parallelogram polyominoes.

2. Object grammars

Let $\mathscr{E}$ be a family of sets of objects. An *object operation* (*in* $\mathscr{E}$) is a mapping $\phi : E_1 \times \cdots \times E_k \to E$, where $E \in \mathscr{E}$ and $E_i \in \mathscr{E}$ for i in $[1,k]$. It describes the way of building an object of E from k objects belonging to $E_1, \ldots, E_k$, respectively.

The *domain* of ϕ is $E_1 \times \cdots \times E_k$, denoted by $dom(\phi)$, the *codomain* is E, denoted by $codom(\phi)$ and the *image* is denoted by $Im(\phi)$. The ith projection E_i of $dom(\phi)$ is called a *component* of ϕ.

Example 2.1. A parallelogram polyomino can be defined as the surface lying between two North–East paths that are disjoint, except at their common ending points (see Fig. 2) [9]. Let E_{pp} be the set of parallelogram polynominoes.

The mappings ϕ_1, ϕ_2 and ϕ_3 illustrated in Fig. 3 are object operations in $\mathscr{E} = \{E_{pp}\}$. The operations ϕ_1 and ϕ_2 are operations of arity 1 ($E_{pp} \to E_{pp}$). The operation ϕ_1 glues a new cell at the left of the lowest cell of the first column of a polyomino. The operation ϕ_2 adds a new cell at the bottom of each column of a polyomino. The operation ϕ_3 is an operation of arity 2 ($E_{pp} \times E_{pp} \to E_{pp}$); it takes two polyominoes as argument, applies ϕ_2 to the first one and glues them by one cell: the top-cell of the last column of the first polyomino facing the bottom-cell of the first column of the second.

Definition 2.1. An object grammar is a 4-tuple $\langle \mathscr{E}, \mathscr{T}, \mathscr{P}, S \rangle$ where:

$\mathscr{E} = \{E_i\}_{i \in I}$ is a finite family of sets of objects. (I is a finite subset of $\mathbb{N}$),

$\mathscr{T} = \{T_{E_i}\}_{i \in I}$ is a finite family of finite subsets of sets of $\mathscr{E}$, $T_{E_i} \subset E_i$, whose elements are called *terminal objects*,

$\mathscr{P}$ is a set of object operations ϕ in $\mathscr{E}$,
S is a fixed set of $\mathscr{E}$ called the *axiom* of the grammar.

The *dimension* of an object grammar is the cardinality of $\mathscr{E}$.

Remark. Sometimes a 3-tuple $\langle \mathscr{E}, \mathscr{T}, \mathscr{P} \rangle$ is called also object grammar. The axiom is chosen later in $\mathscr{E}$.

In the following, the terms *grammar* and *operation* will often be used for *object grammar* and *object operation*, respectively.

The construction of an object can be described by its *derivation tree*: *internal nodes* are labelled with object operations and *leaves* with terminal objects. These derivation trees are comparable to the abstract trees within the theory of Compiling.

Let $G = \langle \mathscr{E}, \mathscr{T}, \mathscr{P} \rangle$ be an object grammar and $E \in \mathscr{E}$ a set of objects. An object o is said to be *generated* in G by E, if there is a derivation tree of G on E (i.e. the codomain of the label of the root is E) whose evaluation is o.

The set of objects generated by E in G is denoted by $\mathscr{O}_G(E)$. If S in $\mathscr{E}$ is chosen as the axiom of G, then $\mathscr{O}_G(S)$ is called the set of objects generated by G.

Example 2.2. Let us note $\square$ the one-cell polyomino. Here are two examples of object grammars:

$$G_1 = \langle \{E_{pp}\}, \{\{\square\}\}, \{\phi_1, \phi_2\}, E_{pp} \rangle$$

and

$$G_2 = \langle \{E_{pp}\}, \{\{\square\}\}, \{\phi_1, \phi_2, \phi_3\}, E_{pp} \rangle.$$

The parallelogram polyomino of Fig. 2 belongs to $\mathscr{O}_{G_2}(E_{pp})$, its derivation tree in G_2 is given in Fig. 4. The set $\mathscr{O}_{G_2}(E_{pp})$ is the set of parallelogram polyominoes.

The set $\mathscr{O}_{G_1}(E_{pp})$ is the set of *Ferrers diagrams*; it is a proper subset of parallelogram polyominoes.

By analogy to context-free grammars, an object grammar G is *unambiguous* if every object in $\mathscr{O}_G(S)$ has exactly one derivation tree. Unambiguity is an important property for building bijections.

One can also define several *normal forms* for object grammars: *reduced*, 1-2 or *complete*. The *reduced* and 1-2 forms extend usual normal forms of context-free grammars: the reduced and Chomsky normal form. A grammar is said to be reduced if every set of objects E in $\mathscr{E}$ is accessible from the axiom and $\mathscr{O}_G(E) \neq \emptyset$; it is said to be in 1-2 form if all its operations are of arity 1 or 2. The *complete* form is specific for object grammars. A grammar is said to be complete if $\mathscr{O}_G(E) = E$ for every set of objects E in $\mathscr{E}$ (generally $\mathscr{O}_G(E) \subseteq E$). For example, the grammar G_2 previously defined is complete while G_1 is not. The details on transformations of object grammars into normal forms are given in [10].

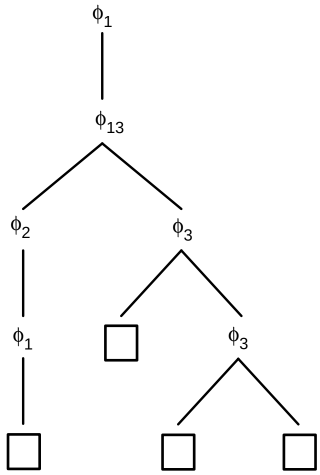

Fig. 4. A derivation tree in G_2.

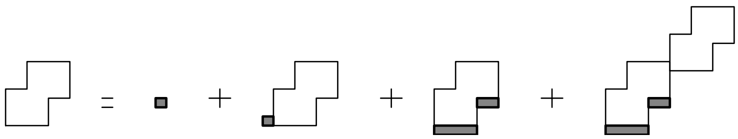

Fig. 5. Schematic object grammar for parallelogram polyominoes.

Another definition

A complete, unambiguous object grammar $G = \langle \mathscr{E}, \mathscr{T}, \mathscr{P}, S \rangle$ can be described as a *system of equations* Σ involving set of objects, terminal objects and object operations, or as a *system of graphic equations*. The equations describe the decomposition of a set of objects into a disjoint union of terminal objects and images of operations

$$\Sigma = \left\{ E_i = \sum_{e_i \in T_{E_i}} e_i + \sum_{codom(\phi)=E_i} \phi(E_{i_{1,\phi}}, \ldots, E_{i_{k,\phi}}) \right\}_{i=1,\ldots,n}.$$

For example, the equation for the grammar G_2 generating parallelogram polynominoes previously defined is

$$E_{pp} = \square + \phi_1(E_{pp}) + \phi_2(E_{pp}) + \phi_3(E_{pp}, E_{pp}).$$

A schematic representation of this grammar is given in Fig. 5.

3. Isomorphic object grammars and bijections

3.1. Characteristic system of polynomials

The *characteristic system of polynomials* associated with an object grammar characterizes its structure. For example, the characteristic polynomial of the grammar for parallelogram polynominoes schematized in Fig. 5 is $1+2x+x^2$: the coefficient of x^i is the number of object operations of arity i and the constant term is the number of terminal objects.

Let X_n be the set of commutative variables $\{x_1,\dots,x_n\}$ and $\mathbb{N}[X_n]$ the semi-ring of polynomials in the variables $x_1,\dots,x_n$ having coefficients in $\mathbb{N}$. An element of this semi-ring is written

$$P(x_1,\dots,x_n)=\sum_{i_1,\dots,i_n} c_{i_1,\dots,i_n} x_1^{i_1}\dots x_n^{i_n}=\sum_{\boldsymbol{i}} c_{\boldsymbol{i}} x^{\boldsymbol{i}},$$

where only a finite number of coefficients $c_{i_1,\dots,i_n}$ are non-zero.

The n-tuple $(i_1,\dots,i_n)$ is denoted by $\boldsymbol{i}$, the notation $x^{\boldsymbol{i}}$ stands for $x_1^{i_1}\dots x_n^{i_n}$ and $c_{\boldsymbol{i}}=\langle x^{\boldsymbol{i}},P\rangle$ is the coefficient of $x^{\boldsymbol{i}}$ in P.

Definition 3.1. Let $G=\langle\mathscr{E},\mathscr{T},\mathscr{P}\rangle$ be an n-dimensional object grammar. Let $\sigma:\mathscr{E}\to X_n$ be a bijection associating with each set of objects a variable of X_n. Let us simplify the notations assuming that $\sigma(E_i)=x_i$ for all i in $[1,\dots,n]$.

The characteristic system of polynomials of G, denoted by $\mathscr{S}^\sigma_{cp}(G)$, is a set of pairs (x_i,P_i) where P_i is the polynomial of $\mathbb{N}[X_n]$ built as follows: the right-hand side of the equation defining E_i is transformed linearly by translating
- every terminal object e_i into the value 1,
- every expression $\phi(E_{i_{1,\phi}},\dots,E_{i_{k,\phi}})$ into the monomial $x_{i_{1,\phi}}\dots x_{i_{k,\phi}}$.

We obtain the characteristic system of polynomials

$$\mathscr{S}^\sigma_{cp}(G)=\left\{(x_i,P_i)/P_i=|T_{E_i}|+\sum_{codom(\phi)=E_i} x_{i_{1,\phi}}\dots x_{i_{k,\phi}}\right\}_{i=1,\dots,n}.$$

Remark. The number of terminal objects is $|T_{E_i}|=\langle x^{\mathbf{0}},P_i\rangle$. For $\boldsymbol{j}=(j_1,\dots,j_n)\neq\mathbf{0}$ the coefficient $\langle x^{\boldsymbol{j}},P_i\rangle$ is the number of operations ϕ with $codom(\phi)=E_i$ and such that the components of $dom(\phi)$ are $E_{j_1},\dots,E_{j_n}$ (without order).

Example 3.1. The first grammar of Fig. 8, denoted here by $G_{E_{cdp}}$, generates the so-called *convex directed polyominoes* (see [4]). Its characteristic system of polynomials is

$$\begin{aligned}\mathscr{S}^\sigma_{cp}(G_{E_{cdp}})=\{&(x_1,1+x_1+x_1x_3+x_2),\\ &(x_2,x_1+x_2+x_2x_3),\\ &(x_3,1+2x_3+{x_3}^2)\}.\end{aligned}$$

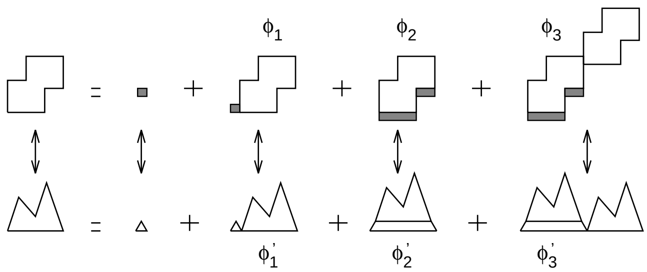

Fig. 6. Parallelogram polyominoes and Dyck paths.

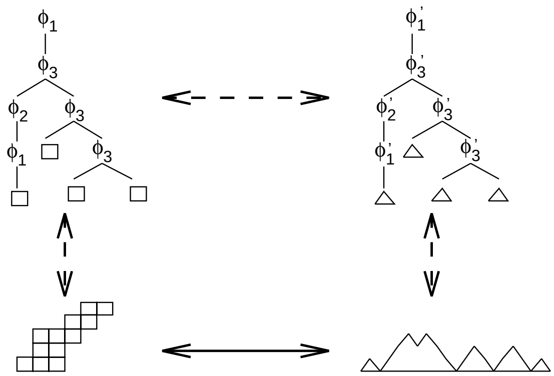

Fig. 7. Bijection between a parallelogram polyomino and a Dyck path.

3.2. Isomorphic object grammars

The notion of *isomorphic* object grammars permits to describe *natural* bijections between sets of objects.

Definition 3.2. Let $G=\langle \mathscr{E},\mathscr{T},\mathscr{P}\rangle$ and $G'=\langle \mathscr{E}',\mathscr{T}',\mathscr{P}'\rangle$ be two n-dimensional object grammars and $X_n=\{x_1,\ldots,x_n\}$ a set of commutative variables. The grammars G and G' are said to be *isomorphic* if there exist two bijections $\sigma:\mathscr{E}\to X_n$ and $\sigma':\mathscr{E}'\to X_n$ such that

$$\mathscr{S}^{\sigma}_{cp}(G)=\mathscr{S}^{\sigma'}_{cp}(G').$$

If two grammars are isomorphic, we can establish a one-to-one correspondence between their terminal objects and between their operations. We can therefore bijectively transform their derivation trees (see Fig. 7). We then obtain the following result:

Theorem 3.3. *Let G and G′ be two isomorphic object grammars. If G and G′ are unambiguous, then one can build a "natural" bijection between objects generated by G and objects generated by G′.*

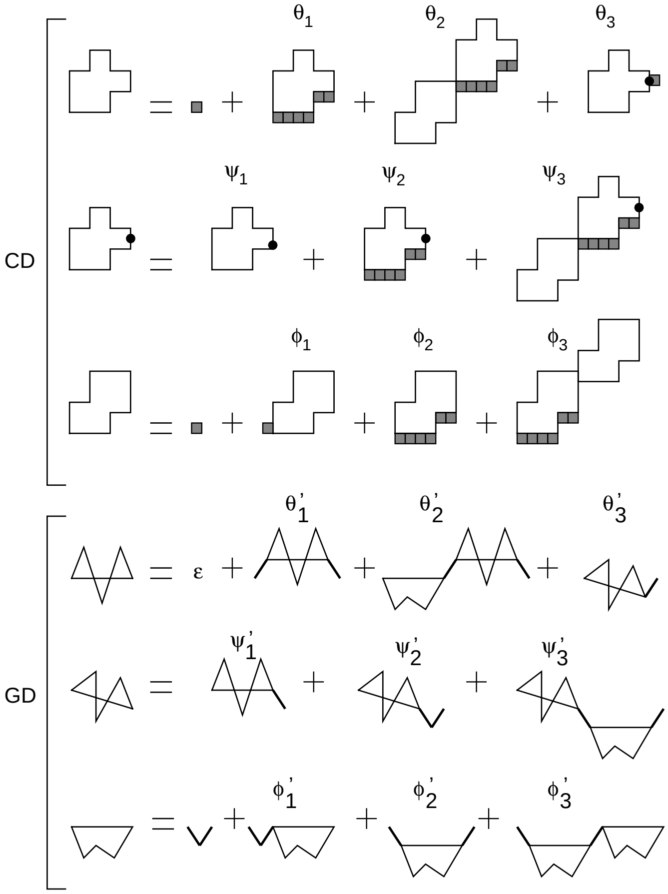

Fig. 8. Convex directed polyominoes and bilateral Dyck paths.

Example 3.2. Isomorphic object grammars for families of polyominoes and families of paths are given in Figs. 6–8. Fig. 6 describes a well-known bijection between parallelogram polyominoes and Dyck paths (Delest and Fédou [8] or Delest and Viennot [9]): parallelogram polynominoes of perimeter $2n+2$ correspond to Dyck paths of length $2n$. A new bijection between convex directed polyominoes and bilateral Dyck paths is presented in Fig. 8. It allows to enumerate them according to their perimeter (see also Bousquet-Mélou [4] and Lin and Chang [15]): the number of convex directed polyominoes of perimeter $2n+4$ is $\binom{2n}{n}$, the number of bilateral Dyck paths of length $2n$.

4. *S*-iteration

Example 4.1. The grammars (a) and (b) of Fig. 9 are not isomorphic (their characteristic polynomials are $1+x+x^3$ and $1+x^2$, respectively). If we substitute the set

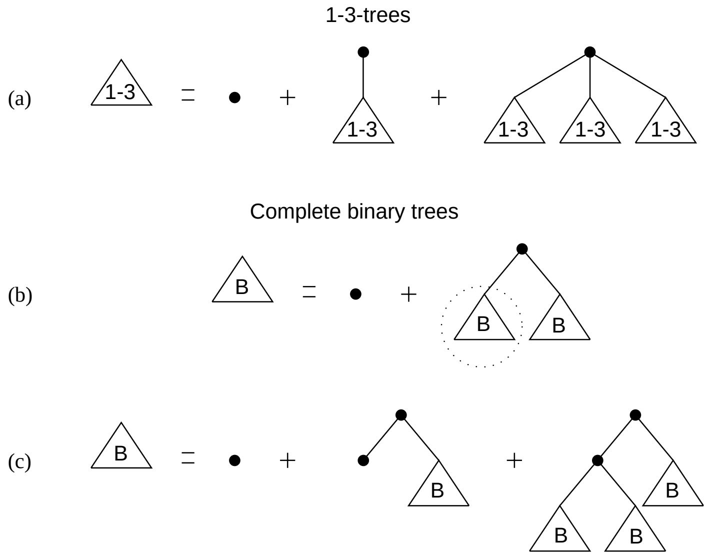

Fig. 9. Complete binary trees and 1-3-trees.

of objects surrounded by a dashed line in (b) by the whole equation, and expand the result, we obtain the grammar (c) which is now isomorphic to the grammar (a).

This transformation can also be described on the characteristic polynomials. Its corresponds to substituting an occurrence of x by $1+x^2$.

We give a formalization of this transformation. We call it *S-iteration* (*S* for *S*ubstitution).

Definition 4.1. Let G be an n-dimensional object grammar

$$G=\left\{E_i=\sum_{e_i\in T_{E_i}} e_i+\sum_{codom(\phi)=E_i}\phi(E_{i_{1,\phi}},\ldots,E_{i_{k,\phi}})\right\}_{i=1,\ldots,n}.$$

Let ψ be an object operation with $codom(\psi)=E_m$ and such that one of the components of $dom(\psi)$ is the set of objects E_j. An *S-iteration* of G consists in substituting, in the equation defining E_m, the expression

$$\psi(E_{m_{1,\psi}},\ldots,E_j,\ldots,E_{m_{k,\psi}})$$

by

$$\sum_{e_j\in T_{E_j}}\psi(E_{m_{1,\psi}},\ldots,e_j,\ldots,E_{m_{k,\psi}})$$

$$+\sum_{codom(\phi)=E_j}\psi(E_{m_{1,\psi}},\ldots,\phi(E_{j_{1,\phi}},\ldots,E_{j_{k,\phi}}),\ldots,E_{m_{k,\psi}}).$$

The new grammar G' generates the same set of objects as G, for every $E_i : \mathcal{O}_{G'}(E_i) = \mathcal{O}_G(E_i)$ for all $i \in [1, n]$.

Definition 4.2. An object grammar built from the object grammar G by a finite sequence of S-iterations is called an *iterate* of G. By *sequence of S-iterations*, we mean that every substitution consists in replacing a set of objects of the current grammar by its equation in the *initial* grammar G.

The S-iteration can be also defined on the characteristic system of polynomials. Let $\mathcal{S}^{\sigma}_{cp}(G)$ be the characteristic system of polynomials of G (with $\sigma(E_i) = x_i$)

$$\mathcal{S}^{\sigma}_{cp}(G) = \left\{ (x_i, P_i)/P_i = |T_{E_i}| + \sum_{codom(\phi)=E_i} x_{i_{1,\phi}} \dots x_{i_{k,\phi}} \right\}_{i=1,\dots,n}.$$

One obtains the characteristic system of polynomials for G', denoted $\mathcal{S}^{\sigma}_{cp}(G')$, as follows: in the polynomial P_m of $\mathcal{S}^{\sigma}_{cp}(G)$, the variable x_j occurring in the monomial $x_{m_{1,\psi}} \dots x_j \dots x_{m_{k,\psi}}$ is replaced by the polynomial P_j, and then we expand the result to obtain again a sum of monomials.

Definition 4.3. Two object grammars G_1 and G_2 are said to be *S-isomorphic* if one can build two isomorphic object grammars generating the same sets of objects as initial grammars by finite sequences of S-iterations from G_1 and G_2.

Characteristic systems of polynomials are said to be S-isomorphic if finite sequences of S-iterations lead to the same system of polynomials.

5. *S*-isomorphic one-dimensional grammars

We come now to the main result of the paper (Theorem 5.1) and its consequences.

We consider one-dimensional grammars, i.e. grammars defined by means of only one set of objects. The *degree* of such a grammar is the maximum arity of its operations.

5.1. Object grammars of degree $\geqslant 2$

Theorem 5.1. *Let G_1 and G_2 be two one-dimensional object grammars of degree $\geqslant 2$. Then G_1 and G_2 are S-isomorphic.*

Proof. The proof is constructive. See Section 6. □

Severals examples of S-isomorphic grammars are given in the present paper. The grammars in Fig. 6 are (S-)isomorphic and we do not need any S-iteration to describe a bijection between parallelogram polynomials and Dyck paths. The grammars (a) and

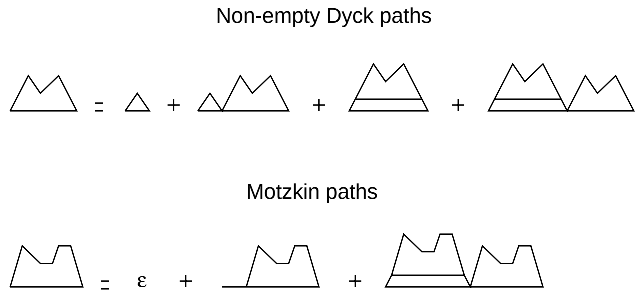

Fig. 10. Standard grammars for non-empty Dyck paths and Motzkin paths.

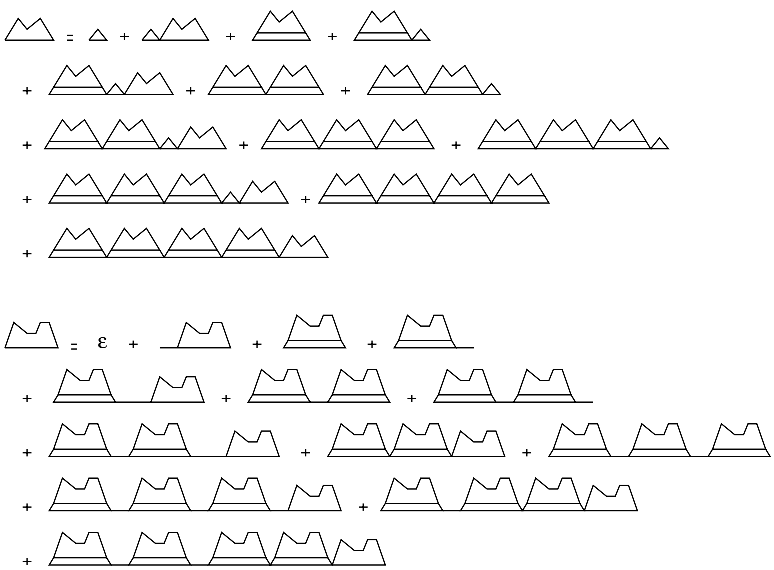

Fig. 11. Isomorphic grammars for non-empty Dyck paths and Motzkin paths.

(b) in Fig. 9 require only one S-iteration to become isomorphic and to describe a bijection between complete binary trees and 1-3-trees. Finally, the standard grammars for non-empty Dyck paths and Motzkin paths (see Fig. 10) are S-isomorphic by virtue of Theorem 5.1. Indeed, Fig. 11 shows isomorphic grammars built via a sequence of S-iterations constructed in the proof of Theorem 5.1 (see Section 6).

Theorem 5.1 implies that we can build bijections between all the sets of objects that can be generated by a one-dimensional unambiguous object grammars of degree $\geqslant 2$. A set of objects which is well known in the field of combinatorics falls in this framework: the set of Dyck words, generated by the standard grammar $D = \varepsilon + xD\bar{x}D$.

Corollary 5.2. *One can build "natural" bijections between the set of Dyck words and all the classes of objects that can be described by a one-dimensional unambiguous object grammar of degree* $\geqslant 2$.

Hence, severals objects are "naturally" close to Dyck words and this confirms the crucial role of these words in coding combinatorial objects.

5.2. Object grammars of degree 1

Two object grammars of degree 1 having characteristic polynomials $K = k_0 + k_1 x$ and $Q = q_0 + q_1 x$, respectively, $(k_0, k_1, q_0, q_1 \geqslant 1)$ are S-isomorphic if and only if

$$\frac{k_1 - 1}{k_0} = \frac{q_1 - 1}{q_0}.$$

Thus, every positive rational number corresponds to a class of object grammars of degree 1 (see [10]).

6. Proof of the main theorem

We will prove Theorem 5.1 in terms of characteristic polynomials (the full details of the proof are given in [10]). Polynomials associated with one-dimensional object grammars are polynomials of degree $\geqslant 1$ in $\mathbb{N}[X]$ with a non-zero constant term. We denote this set of polynomials by $\mathbb{N}[X]_{OG}$.

In order to prove Theorem 5.1, we have to prove that all the polynomials of degree $\geqslant 2$ in $\mathbb{N}[X]_{OG}$ are pairwise S-isomorphic. The proof is constructive. The formula we shall establish in the following describes the finite sequences of S-iterations that lead to the same polynomial.

First we state a preliminary lemma which permits to consider only *uniformly non-zero polynomials*, i.e. polynomials P in $\mathbb{N}[X]_{OG}$ such that $\langle x^i, P\rangle \neq 0$ for $0 \leqslant i \leqslant deg(P)$.

Lemma 6.1. *Every polynomial of degree* $\geqslant 2$ *in* $\mathbb{N}[X]_{OG}$ *admits a uniformly non-zero iterate.*

Notation. Let K and P be polynomials in $\mathbb{N}[X]_{OG}$. Then, $S_i^P(K)$ consists in replacing an occurrence of x in a monomial x^i of K by the polynomial P.

Let K and Q be two uniformly non-zero polynomials of degree $\geqslant 2$ in $\mathbb{N}[X]_{OG}$. To check that they are S-isomorphic, we prove that there are two integers r et s, and integers $N_1, \ldots, N_r$ and $M_1, \ldots, M_s$ such that

$$(S_r^K)^{N_r} \ldots (S_1^K)^{N_1}(K) = (S_s^Q)^{M_s} \ldots (S_1^Q)^{M_1}(Q).$$

For r and s fixed, this leads to a linear system in the N_i and M_j. One can prove that, if K has degree n and Q has degree m with

$$K = \sum_{i=0}^{n} k_i x^i \quad \text{and} \quad Q = \sum_{i=0}^{m} q_i x^i$$

then we can take: $r = m+1$, $s = n+1$,

$$\begin{pmatrix} N_1 \\ N_2 \\ N_3 \\ \vdots \\ N_{m+1} \end{pmatrix} = \begin{pmatrix} -1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} + i \begin{pmatrix} q_0 \\ q_1 - 1 \\ q_2 \\ \vdots \\ q_m \end{pmatrix}$$

$$\text{and} \quad \begin{pmatrix} M_1 \\ M_2 \\ M_3 \\ \vdots \\ M_{n+1} \end{pmatrix} = \begin{pmatrix} -1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} + i \begin{pmatrix} k_0 \\ k_1 - 1 \\ k_2 \\ \vdots \\ k_n \end{pmatrix}$$

with $i \in \mathbb{N} \backslash \{0\}$.

Remark. There are technical details if the degree of K (or Q) equals 2. In this case, we consider $S_2^Q(Q)$ instead of Q (or $S_2^K(K)$ instead of K) so that $\langle x, Q\rangle \geqslant 2$ (or $\langle x, K\rangle \geqslant 2$).

Example 6.1. Let us consider a grammar for non-empty Dyck paths with characteristic polynomial $1+2x+x^2$ and a grammar for Motzkin paths with characteristic polynomial $1+x+x^2$ (Fig. 10). Applying the previous formula, we obtain the following common iterate:

$$1 + 3x + 3x^2 + 3x^3 + 2x^4 + x^5.$$

The corresponding isomorphic grammars are given in Fig. 11.

7. Conclusions and perspectives

Many questions are still unanswered. Our main theorem deals with one-dimensional object grammars. What happens for grammars of higher dimension?

One can also apply other transformations to object grammars. It would be especially interesting to study the operation consisting in deleting a terminal object. Thus, we would obtain correspondences between sets of objects that would be one to one “except for a finite number of elements”.

Take, as an example, the standard grammar for Dyck words $D=\varepsilon+xD\bar{x}D$. Plugging in $D=\varepsilon+D_+$, one obtains

$$D_+ = x\bar{x} + x\bar{x}D_+ + xD_+\bar{x} + xD_+\bar{x}D_+,$$

which is a grammar for non-empty Dyck words.

The characteristic polynomial $1+x^2$ is transformed into

$$(1+(1+y)^2)-1 = 1+2y+y^2,$$

which is the characteristic polynomial of the grammar for parallelogram polynominoes. Accordingly, we find again the standard bijection between parallelogram polynominoes and non-empty Dyck words.

Combining this new iteration and the S-iteration may, perhaps, allow to build simpler isomorphic grammars, which could be used to study parameters preserved by bijections.

Acknowledgements

We would like to thank Mireille Bousquet-Mélou, Srećko Brlek, Jacques Labelle and Christophe Reutenauer for their useful comments and corrections.

References

[1] A. Arnold, M. Dauchet, Une relation d'équivalence décidable sur la classe des forêts reconnaissables, Math. Systems Theory 12 (1978) 103–128.

[2] E. Barcucci, A. Del Lungo, E. Pergola, R. Pinzani, ECO: a methodology for the enumeration of combinatorial objects, J. Differential Equations Appl. 5 (4–5) (1999) 435–490.

[3] F. Bergeron, G. Labelle, P. Leroux, Combinatorial Species and Tree-Like Structures, Cambridge University Press, Cambridge, 1997.

[4] M. Bousquet-Mélou, Une bijection entre les polyominos convexes dirigés et les mots de Dyck bilatères, Inform. Théor. Appl. 26 (3) (1992) 205–219.

[5] R.M. Burstall, J. Darlington, A transformation system for developing recursive programs, J. ACM 24 (1) (1977) 44–67.

[6] B. Courcelle, Equivalences and transformations of regular systems—applications to recursive program schemes and grammars, Theoret. Comput. Sci. 42 (1986) 1–122.

[7] M. Delest, Langages algébriques: à la frontière entre la Combinatoire et l'Informatique, in: Actes du 6ème Colloque Séries Formelles et Combinatoire Algébrique, DIMACS, Rutgers University, USA, 1994, pp. 69–78.

[8] M. Delest, J.M. Fédou, Attribute grammars are useful for combinatorics, Theoret. Comput. Sci. 98 (1992) 65–76.

[9] M. Delest, X.G. Viennot, Algebraic languages and polyominoes enumeration, Theoret. Comput. Sci. 34 (1984) 169–206.

[10] I. Dutour, Grammaires d'objets: énumération, bijections et génération aléatoire, Ph.D. Thesis, Université Bordeaux I, 1996.

[11] I. Dutour, J.-M. Fédou, Object grammars and random generation, Discrete Math. Theoret. Comput. Sci. 2 (1998) 47–61 (http://dmtcs.loria.fr).

[12] S. Eilenberg, J.B. Wright, Automata in general algebra, Inform. and Control 11 (1967) 452–470.

[13] P. Flajolet, P. Zimmermann, B. Van Cutsem, A calculus for the random generation of combinatorial structures, Theoret. Comput. Sci. 132 (1994) 1–35.
[14] A. Joyal, Une théorie combinatoire des séries formelles, Adv. Math. 42 (1) (1981) 1–82.
[15] K.Y. Lin, S.J. Chang, Rigourous results for the number of convex polygons on the square and honeycomb lattices, J. Phys. A: Math. Gen. 21 (1988) 2635–2642.
[16] J. Mezei, J.B. Wright, Algebraic automata and context-free sets, Inform. and Control 11 (1967) 3–29.
[17] X.G. Viennot, Enumerative combinatorics and algebraic languages, in: L. Budach (Ed.), Proc. FCT'85, Lecture Notes in Computer Science, Vol. 199, Springer, Berlin, 1985, pp. 450–464.