Università Degli Studi di Firenze

Dottorato di Ricerca in
Ingegneria Informatica e dell'Automazione

XV Ciclo

ECO method and Object Grammars: two methods for the enumeration of combinatorial objects.

Il Candidato

____

(Enrica Duchi)

Il Coordinatore

____

(Prof. Edoardo Mosca)

Acknowledgments

First, I want to thank my tutor Renzo Pinzani, that gave me the opportunity to work with him and the Group of Discrete Mathematics and Theoretical Computer Science in Florence. He made the department the interesting place it is for Combinatorics, attracting researchers from everywhere. Among other things, I remember with pleasure the nice dinners organized by him.

Thanks to Jean-Marc Fédou, that followed my work in Nice during these three years. I want to thank him not only for helping me with my research but also for his great hospitality. I spent many days with him and his family, they made me feel free to stay with them each time I needed. I thank very much his family too.

I met Simone Rinaldi four years ago, and since then I have been happy to work with him. I want to thank him for the helpful advices that he gave me during these years, and not only concerning my work.

I want to thank my research group in Florence and in Sienna: Elena Barcucci, Elisa Pergola, Elisabetta Grazzini, Luca Ferrari, Nicola Salvadori, Maddalena Poneti, Alberto Del Lungo, Sara Brunetti and Andrea Frosini. We had many interesting discussions during these years.

I want also thank the people that gave me the welcome at E.S.S.I. in Nice Sophia Antipolis.

Thanks to Srečko Brlek: I had the opportunity to work with him during the last year, in Florence. I remember the nice evenings spent together, talking about Mathematics and many other things.

I thank a lot Robert Sulanke, both for his suggestions on my thesis and for his great helpfulness to me. I really appreciated his will in listening my ideas, even if sometimes they were not clearly explained. I want also to thank him and his wife Joyce for their hospitality. I will never forget the beautiful days we spent together in San Francisco, up and down in the cable cars...

I thank very much Christine Garcia that, besides being a scientific collaborator, is also a friend to me. I really appreciate her patience in listening to me, even if sometimes my french was not so good. She and Jean-Marc tried to teach me how to be more ordinate and not to write in loose sheets. I'm still working on it... I take the opportunity to thank Christine's family too, with them I spent very nice days.

During this last year I had the opportunity to work with Gilles Schaeffer. I really appreciate his calm in dealing with a mathematical problem. I would like to be so confident that any problem has its solution. I want to make a special thank to Gilles, for having reread my thesis and for his useful suggestions.

My friend Emanuele. What can I say? During this three years we shared our good periods and our troubles. I thank him very much for his friendship, I think that this period would have been less nice without him.

Finally, I want to thank my parents, my brother Bruno, and my sister Bruna for the confidence they give me.

Sommario

La *combinatoria* è un'importante area della matematica, riguardante lo studio dell strutture discrete. Negli ultimi anni la combinatoria ha assunto un ruolo rilevante in molte discipline scientifiche, come l'informatica teorica, la fisica statistica e la biologia. Infatti, vari problemi che sorgono da queste discipline possono essere risolti utilizzando tecniche di combinatoria enumerativa. Questo è possibile quando tali problemi sono riconducibili allo studio di semplici oggetti, come grafi, alberi, cammini nel piano.

Questa tesi si colloca nell'ambito della combinatoria enumerativa e biettiva: facciamo uso di biiezioni e di un metodo per l'Enumerazione di Oggetti Combinatori (ECO) per risolvere dei problemi di combinatoria. In particolare, viene fatto un ampio studio sulle relazioni tra il metodo ECO ed un altro metodo ricorsivo per l'enumerazione di oggetti combinatori, quello delle grammatiche ad oggetti.

Il *metodo ECO*, introdotto da Pinzani et al., costruisce un oggetto mediante l'espansione locale di un altro più piccolo. Spesso accade che un operatore ECO, ϑ, faccia crescere gli oggetti di una classe $\mathcal{O}$ con una certa regolarità. In questo caso ϑ può essere facilmente descritto da una *regola di successione* Ω, che è un sistema costituito da un assioma e da un insieme di produzioni. L'assioma (a) rappresenta il numero di oggetti prodotti dall'oggetto più piccolo per mezzo dell'operatore ϑ. Una produzione di Ω è della forma $(k) \rightarrow (e_1(k))(e_2(k)) \ldots (e_k(k))$, dove (k) rappresenta il numero di oggetti $O_1, \ldots, O_k$ prodotti da un qualsiasi oggetto $O \in \mathcal{O}$ ed $e_i(k)$ rappresenta il numero di oggetti prodotti da O_i, per $i = 1 \ldots k$. Sia p il parametro secondo il quale ϑ fa crescere gli oggetti, allora $\Sigma = (\mathcal{O}, p, \vartheta, \Omega)$ viene detto un ECO-sistema.

Le *grammatiche ad oggetti*, introdotte da Fédou e Dutour, descrivono gli oggetti di $\mathcal{O}$ per mezzo di operazioni di composizione degli stessi, a partire da oggetti più piccoli. Una grammatica ad oggetti è rappresentata da $\langle \mathbb{O}, \mathbb{E}, \Phi, \mathcal{O} \rangle$, dove $\mathbb{O}$ è una famiglia finita di classi di oggetti, $\mathbb{E}$ è una famiglia finita di sottoclassi costituite dagli oggetti più piccoli di ogni classe appartenente ad $\mathbb{O}$, Φ è l'insieme delle operazioni sugli oggetti ed $\mathcal{O} \in \mathbb{O}$ è la classe generata dalla grammatica. Una grammatica ad oggetti si dice completa e non ambigua quando vengono generati tutti gli oggetti della classe $\mathcal{O}$ ma una volta sola.

La tesi inizia (**Capitolo 1**) con un'introduzione sulle strutture che vengono utilizzate, sul metodo ECO e sulle grammatiche ad oggetti. Poi si divide in due parti. La prima parte riguarda l'analisi di problemi riguardanti le regole di successione: il problema di individuare classi di regole di successione equivalenti ed il problema di trovare regole di successione che descrivano la sequenza definita da una ricorrenza lineare. La seconda parte contiene il risultato principale della tesi. Infatti si dimostra come una qualsiasi classe di oggetti generata da una grammatica ad oggetti completa e non ambigua, possa essere descritta da un ECO-sistema secondo un parametro lineare. Questo risultato è infine esteso al caso di parametri q-lineari naturali per le grammatiche unidimensionali.

Nei dettagli, la tesi è organizzata come segue. La prima parte è divisa in due capitoli. Nel paragrafo 2.1 del **Capitolo 2**, si introduce il problema dell'equivalenza fra regole di successione e si richiamano i risultati principali esistenti per le regole di successione finite e fattoriali. Nel paragrafo 2.2 si dimostra l'equivalenza di due insiemi infiniti di regole di successione, legate ai numeri di Catalan e di Schröder. La dimostrazione dell'equivalenza di queste regole è biettiva, nel senso che si determinano due diverse costruzioni ECO che descrivono la stessa classe di oggetti secondo lo stesso parametro. A tal fine, vengono presentate due

nuove costruzioni ECO per i cammini di Dyck e di Schröder (sottoparagrafi 2.2.1 e 2.2.2). Nel paragrafo 2.3 viene introdotto un insieme infinito di regole di successione che definiscono la sequenza dei numeri ballot. Anche in questo caso se ne dimostra l'equivalenza in maniera biettiva, a partire da un'interessante costruzione ECO sui cammini di Dyck (sottoparagrafo 2.3.2). Nel paragrafo 2.4 si introducono altri insiemi di regole di successione e se ne dimostra l'equivalenza calcolandone le funzioni generatrici. Nel paragrafo 3.3 del **Capitolo 3**, vengono introdotte le regole di successione negative al fine di definire l'operazione di sottrazione fra regole di successione e di determinare le regole di successione inverse, rispetto alle operazioni di prodotto e di semiprodotto introdotte nel paragrafo 3.2. Nel paragrafo 3.5 viene invece trattato il problema di determinare regole di successione associate a ricorrenze lineari. In particolare, nel sottoparagrafo 3.5.1 vengono introdotte delle regole di successione che descrivono un'ampia classe di ricorrenze lineari positive non decrescenti. Nel sottoparagrafo 3.5.2 mostriamo come sia possibile descrivere delle ricorrenze lineari per mezzo di regole di successione negative. In particolare, descriviamo il caso delle ricorrenze lineari a due termini.

Nella seconda parte, è presente il lavoro principale della tesi, cioè lo studio delle relazioni fra metodo ECO e grammatiche ad oggetti. Il **Capitolo 4** è dedicato alla dimostrazione che da una qualsiasi grammatica ad oggetti completa e non ambigua si può ottenere un ECO-sistema secondo un parametro lineare. In particolare, nel paragrafo 4.1 diamo le definizioni principali sulle grammmatiche ad oggetti e forniamo alcuni esempi. Nel paragrafo 4.2 introduciamo il concetto di parametri lineari e q-lineari su una grammatica ad oggetti. Nel paragrafo 4.3 dimostriamo che per una qualsiasi classe di oggetti $\mathcal{O}$, generata da una grammatica ad oggetti G, unidimensionale, completa e non ambigua, è sempre possibile determinare un ECO-sistema che la descriva secondo un parametro lineare. Per dimostrarlo, determiniamo una costruzione ECO per una particolare classe di alberi, in biiezione con gli alberi di derivazione di G, che sono a loro volta in biiezione con gli oggetti di $\mathcal{O}$. Più precisamente, nel sottoparagrafo 4.3.3 trattiamo il caso dei parametri lineari uniformi, nel sottoparagrafo 4.3.4 lo estendiamo al caso dei parameteri lineari. Nel paragrafo 4.4 trattiamo invece il caso più generale delle grammatiche multidimensionali, fornendo un'altra costruzione ECO per gli alberi associati ad una grammatica multidimensionale. Nel paragrafo 4.5, presentiamo una costruzione per la classe degli alberi associata alla grammatica dei poliomini convessi direzionati. Nel **Capitolo 5** viene affrontato il caso dei parametri q-lineari per grammatiche unidimensionali. In particolare, nel paragrafo 5.1 introduciamo il concetto di parametri q-naturali su una grammatica G e trasportiamo tali parametri sull'ECO-sistema associato a G. Nel paragrafo 5.2 diamo degli esempi e delle applicazioni. Nel paragrafo 5.3 mostriamo come sia utile avere una costruzione ECO associata ad una grammatica ad oggetti. In particolare, mostriamo che è possibile estendere una tale costruzione ECO per una classe di cammini, ad un' altra classe più difficile da trattare. Nel **Capitolo 6**, trattiamo il problema inverso, cioè quello di ottenere una grammatica a partire da un ECO-sistema. Tale problema viene risolto per una particolare classe di oggetti: i poliomini convessi. Nel paragrafo 6.1 si determina una costruzione ECO per questi oggetti, fino ad ora questa classe non era stata enumerata con il metodo ECO. Nei pararafi 6.3 e 6.4 , si determina una decomposizione algebrica nell'albero di generazione della regola di successione associata alla costruzione ECO. Questo permette di ottenere una grammatica per la classe dei poliomini convessi.

Résumé

La *combinatoire* est une branche importante des mathématiques, qui étudie les structures discrètes. Dans ces dernières années la combinatoire a eu un rôle utile dans de nombreuses disciplines scientifiques, comme l'informatique théorique, la physique statistique et la biologie. En effet, de nombreux problèmes issus de ces disciplines peuvent être traités en utilisant des techniques de combinatoire énumérative. Ceci est possible quand ces problèmes se ramènent à l'étude d'objets combinatoires simples, comme graphes, arbres ou chemins du plan.

Cette thèse se situe dans la domaine de la combinatoire énumérative et bijective: nous utilisons des bijections et une méthode pour l'Énumération d'Objets Combinatoires (ECO) pour résoudre des problèmes de combinatoire. En particulier nous étudions en profondeur les relations entre la méthode ECO et une autre méthode récursive pour l'énumération d'objets combinatoires, celle des grammaires objets.

La *méthode ECO*, introduite par Pinzani et al., construit chaque objet en faisant localement grandir un autre objet plus petit. Souvent l'opérateur ECO, ϑ, fait grandir les objets d'une classe $\mathcal{O}$ avec une certaine régularité. Dans ce cas ϑ peut être facilement décrit à l'aide d'une *règle de succession* Ω, qui est un système formé d'un axiome et d'un ensemble de productions. L'axiome (a) représente le nombre d'objets produits par l'objet le plus petit au moyen de l'opérateur ϑ. Une production de Ω est de la forme $(k) \rightarrow (e_1(k))(e_2(k))\dots(e_k(k))$, où (k) représente le nombre d'objets $O_1, \dots, O_k$ produit à partir de $O \in \mathcal{O}$ et $e_i(k)$ représente le nombre d'objets qui seront produits à leur tour à partir de O_i, pour $i = 1 \dots k$. Soit p le paramètre selon lequel ϑ fait grandir les objets, alors $\Sigma = (\mathcal{O}, p, \vartheta, \Omega)$ est appelé un ECO-système.

Les *grammaires d'objets*, introduites par Fédou et Dutour, décrivent les objets de $\mathcal{O}$ au moyen d'opérations de compositions de ceux-ci, à partir des objets plus petits. Une grammaire d'objets est représenté par $\langle \mathbb{O}, \mathbb{E}, \Phi, \mathcal{O} \rangle$, où $\mathbb{O}$ est une famille finie de classe d'objets, $\mathbb{E}$ est une famille finie de sous-classe construite à partir des objets plus petit de chaque class appartenant à $\mathbb{O}$, Φ est l'ensemble des opérations sur les objets et $\mathcal{O} \in \mathbb{O}$ est la classe engendrée par la grammaire. Une grammaire d'objets est dite complète et non ambigue quand tous les objets $\mathcal{O}$ sont engendrés une et une seule fois.

La thèse commence (Chapitre 1) par une introduction sur les structures qui sont ensuite utilisées, sur la méthode ECO et sur les grammaires d'objets. Puis elle se divise en deux parties. La première (Chapitre 2-3) analyse des problèmes sur les règles de succession: le problème de montrer des règles de succession équivalentes et celui de trouver des règles de succession qui décrivent une suite définie par une récurrence linéaire donnée. La seconde partie (Chapitre 4-6) convient le résultat principal de la thèse. En effet on y démontre comment une classe quelconque d'objets engendrée par une grammaire d'objet complète et non ambigue peut être décrite par un ECO-système suivant un paramètre linéaire. Ce résultat est enfin étendu au cas des paramètres q-linéaires naturels pour les grammaires unidimensionelles.

Dans le détail la thèse est organisée comme suit. La première partie est divisée en deux chapitre. Dans la section 2.1 du **chapitre 2**, on introduit le problème de l'équivalence entre règle de succession et on rapelle les principaux résultats existants pour les règles de succession finies et factorielles. Dans la section 2.2 on démontre l'équivalence de deux ensembles infinis de règles de succession, liées aux nombres de Catalan et de Schröder. La méthode utilisée est bijective, c'est-à-dire que l'équivalence de deux règles est montrée en donnant deux constructions ECO (une associée à chaque règle) qui décrivent la même classe d'objets selon le

même paramètres. Dans ce but, on présente deux nouvelles constructions ECO des chemins de Dyck et de Schröder (sections 2.2.1 et 2.2.2). Dans la section 2.3 est introduit un ensemble infini de règles de succession qui définissent la suite des nombres de ballot. Dans ce cas aussi l'équivalence est montrée de manière bijective, à partir d'une intéressante construction ECO sur les chemins de Dyck (section 2.3.2). À la section 2.4, d'autres ensembles de règles de succession sont traités en calculant leur série génératrice.

Dans la section 3.3 du **chapitre 3**, sont introduites les règles de succession négatives dans le but de définir l'opération de soustraction entre règle de succession et de déterminer d'inverse d'une règle de succession respectivement aux opérations de produit et de semiproduit introduites à la section 3.2. Dans la section 3.5 est traité le problème de déterminer des règles de succession associées à des récurrences linéaires. En particulier, à la section 3.5.1 sont introduites des règles de succession qui décrivent un grande classe de récurrence linéaire positives croissantes. La section 3.5.2 est consacrée à montrer comment il est possible de décrire des récurrences linéaire au moyen de règles de succession négative. En particulier, le cas des récurrences linéaires à deux termes est traité en détail.

Dans la seconde partie est présenté le résultat principal de la thèse, c'est-à-dire l'étude des relations entre la méthode ECO et les grammaires objets. Le **chapitre 4** est dédié à la démonstration qu'à une quelconque grammaire d'objets complète et non ambigue peut être associé un ECO-système suivant un paramètre linéaire. En particulier, dans la section 4.1 nous donnons les définitions principales sur les grammaires d'objets ainsi que quelques exemples et à la section 4.2 sont introduits les concepts de paramètres linéaires et q-linéaires. Pour démontrer le résultat, nous donnons une construction ECO pour une classe particulière d'arbres, en bijection avec les arbres de dérivation de G, qui sont à leur tour en bijection avec les objets de $\mathcal{O}$. Plus précisément, le cas des paramètres uniformes linéaires fait l'objet de la section 4.3.3, avant d'être étendu à la section 4.3.4 au cas des paramètres linéaires. Enfin à la section 4.4 est traité le cas plus général des grammaires multidimensionnelles, qui découle d'une construction ECO pour les arbres associés. Nous présentons ensuite une construction pour la classe des arbres associés aux polyominos convexes dirigés. Au **chapitre 5** on s'occupe des paramètres q-linéaires naturels des grammaires unidimensionnelles. En particulier à la section 5.1 est introduit le concept de paramètre q-naturel sur une grammaire G et ces paramètres sont transportés sur l'ECO-système associé à G. À la section 5.2 nous donnons des exemples et des applications. À la section 5.3 nous illustrons l'utilité d'avoir une construction ECO associées à une grammaire d'objets en montrant une construction ECO d'une classe de chemins qui s'étend simplement à une classe plus exotique, plus difficile à traiter directement avec les grammaires. Au **chapitre 6** nous nous intéressons au problème inverse, c'est-à-dire obtenir une grammaire à partir d'un ECO-système. Ce problème est résolu pour une classe particulière d'objets, les polyominos convexes. À la section 6.1, on détermine une construction ECO pour ces objets — jusqu'ici cette classe n'avait pas été énumérée par la méthode ECO. Aux sections 6.3 et 6.4, on détermine une décomposition algébrique des arbres de génération de la règle de succession associée. Ceci permet d'obtenir une grammaire pour la classe des polyominos convexe.

Contents

Chapter 1

Introduction

Combinatorics is an important branch of mathematics concerned in mathematical properties of discrete structures, as opposed to continuous ones. Structures of this kind often arise in theoretical computer science, making this field one of the many bridges between computer science and mathematics. Combinatorics had an increasing growth since the sixties, due to its relation with computer science and to the major impact that the latter had in our society. Today combinatorics is an active field with applications and interactions ranging from analysis of algorithms to statistical physics and bioinformatics for instance. Within combinatorics, enumerative and bijective combinatorics are more specifically dealing with the fundamental problems of counting structures in combinatorial classes and explaining the occurrences of recurring structures.

Enumerative combinatorics is one of the main subfield of combinatorics and is concerned with counting the number of elements of a finite class in an exact or approximate way. Various problems arising from different fields can be solved by analyzing them from a combinatorial point of view. Usually, these problems have the common feature to be represented by simple objects suitable to enumerative techniques of combinatorics. Given a class $\mathcal{O}$ of objects and a parameter p on this class, we focus on the set $\mathcal{O}_n$ of objects for which the value of the parameter, called the *size*, is equal to n, where n ranges over the set $\mathbb{N}$ of non negative integers. The parameter p is discriminating if, for each $n \in \mathbb{N}$, the number of objects of $\mathcal{O}_n$ is finite. Then, we ask for the cardinality a_n of the set $\mathcal{O}_n$ for each possible n. Enumerative combinatorics answers to this question.

Only in rare cases the answer will be a completely explicit closed formula for a_n, involving only well known functions, and free from summation symbols. However, a recurrence for a_n may be given in terms of previously calculated values a_k, thereby giving a simple procedure for calculating a_n for any $n \in \mathbb{N}$. Another approach is based on generating functions: whether we do not have a simple formula for a_n, we can hope to get one for the formal power series $f(x) = \sum_n a_n x^n$, which is called the generating function of the class $\mathcal{O}$ according to the parameter p. Notice that the n-th coefficient of the Taylor series of $f(x)$ is just the term a_n. In some cases, once that the generating function is known, we can apply standard techniques in order to obtain the required coefficients a_n (see for instance Goulden and Jackson [64] and Graham, Knuth, and Patashnik [65]). Otherwise we can obtain an asymptotic value of the coefficients through the analysis of the singularities in the generating function (Flajolet and Odlyzko [56] and Flajolet and Sedgewick [60]).

Methods of enumerations and their applications A first rough and empirical approach to the enumeration consists in calculating the first terms of a_n and then try to figure out the sequence. For instance, one can use the book from Sloane and Plouffe [92, 93] in order to compare the first numbers of the sequence with some known sequences and try to identify a_n. More advanced techniques (Brak and Guttmann [19] and Flajolet et al. [58]) start from the first terms of the sequence and find an algebraic or differential equation satisfied by the generating function of the sequence itself. A more common approach consists in looking for a construction of the studied class of objects and successively translating it into a recursive relation or an equation, usually called *functional equation*, satisfied by the generating function $f(x)$. The approach to enumeration of combinatorial objects by means of generating functions has been widely used in the last decades (see for instance Goulden and Jackson [64] and Wilf [100]).

The *Schützenberger's methodology*, also called *DSV* [90] is a method of enumeration by using algebraic languages. The idea is to estabilish a bijection between the objects and the words of an algebraic language such that the value of the parameter of the objects corresponds to the length of the words of the language. If the language is generated by an unambiguous context-free grammar, then it is possible to translate the productions of the grammar into a system of functional equations, whose solution is unique and algebraic and it is the generating function of the language (Schützenberger and Chomsky [24]). A variant of the DSV methodology are the operator grammars (Cori and Richard [29], Cori [28], and Chottin [25]). These grammars take in account some cases in which the language encoding the objects is not algebraic.

The theory of decomposable structure (Flajolet, Salvy, and Zimmermann [57, 59]), describes recursively the objects in terms of basic operations between them. These operations are directly translated into operations between the corresponding generating functions, cutting off the passage to words. A nice presentation of this theory appears in the book of Flajolet and Sedgewick [61]. A variant is the theory of species, introduced by Bergeron, Labelle and Leroux [11], which also follows the philosophy of decomposable structures. Basing on the idea of Joyal [69], they define an algebra on species of structures, where the operations between the species immediately reflect on the generating functions.

Finally, a very convenient formalization of the approach of decomposable structures was introduced by Dutour [44] with the concept of object grammars, allowing to describe objects using very general kinds of operations. One can also categorize object grammars as belonging to the domain of Universal Algebra and Magmas [47, 91].

A significantly different way of recursively describing objects appears in the ECO methodology, introduced by Barcucci, Del Lungo, Pergola, and Pinzani [6]. In the ECO method each object is obtained from a smaller object by making some local expansions. Often these local expansions are very regular and can be described in a simple way by a succession rule. In turn a succession rule can be translated into a functional equation for the generating function. It has been shown in the last years that this method is very effective on large number of combinatorial structures.

By comparing these two different recursive methods, on the one hand the object grammars, on the other hand the ECO method, we find that they have at least two important common applications: bijections and random generation. Indeed both approaches allow us to determine bijections between classes of different combinatorial objects, either when two object grammars are isomorphic (Dutour and Fédou [46]) or when two ECO constructions lead to the same growth mode (see Rinaldi [83]). The general method of random generation introduced by

Wilf [100] has been applied by Flajolet, Zimmermann, and Van Custem [62] for decomposable structures and also by Dutour and Fédou in [45] for object grammars. Roughly, the random generation of an n-sized object is realized by choosing randomly an integer $k < n$, and then randomly generating a k-sized and a $(n-1-k)$-sized object. On the other hand, a random ECO object corresponds to a random path in the generating tree, i.e. the tree associated with the ECO construction (Barcucci et al. [5]).

Another applications of both the ECO and the decomposable structure approach is in dealing with q-analogs. The notion of q-grammars, introduced by Delest and Fédou in [33], is based on the idea that coding the objects with the words of an algebraic language provides a structure on the objects themselves. By fitting the notion of attribute grammars [72], sometimes it is possible to describe non algebraic equations verified by the generating function of the class of objects, according to a further parameter represented by the indeterminate q. The resulting equations are some q-analogs of the original algebraic equations. There is no general method to solve these q-equations, but some particular cases have been treated (see for instance Bousquet-Mélou [13], Bousquet-Mélou and Fédou [16], and Prellberg and Brak [81]).

As already said, a common approach to determine a_n consists in computing the generating function $f(x)$. However, determining the generating function it is not always an easy task. Another approach is to establish a bijection between the studied class of objects and another one, simpler to count. In order to have consistent enumerative results, the bijection must preserve the size of the objects. A bijective approach also permits to understand better certain properties of the studied class and to relate them to the class in bijection with it. This kind of topics are part of *bijective combinatorics.*

1.1 Summary of the thesis

As already said, this thesis is about enumerative and bijective combinatorics. The fundamental trend of the thesis is the use of bijective transformations and the ECO method to solve combinatorial problems. After an introduction to the basic objects and methods we use, comes the first part of the thesis, where various questions about the expressiveness of the ECO method are adressed. In the second part the relations between object grammars and ECO method are studied.

A given power series can have representations by completely different succession rules: it is then of interest to get a combinatorial understanding of the corresponding diverse growth modes. We use this approach in **Chapter 2**. There, we give a bijective proof of the equivalence of an infinite set of rules for Ballot numbers by providing different ECO constructions for the same class of paths and according to the same parameter. At the end of the chapter we also introduce other infinite sets of equivalent succession rules.

A natural way to understand the expressiveness of succession rules is to consider the problem of finding combinatorial representations for the coefficients of any rational series, or equivalently for the sequence defined by any linear recurrence. In **Chapter 3** we show how a large class of linear recurrences with non negative terms can be represented by simple succession rules. Moreover, we show how any linear recurrence could be treated by means of signed succession rules, by providing an explicit solution for two terms linear recurrence. In this case we show that finite signed succession rules are enough to describe any two terms linear recurrence.

In **Chapter 4** we find the main contribution of this thesis. There we show how it is possible to give an ECO construction according to a linear parameter for all classes of combinatorial objects that can be generated by unambiguous and complete object grammars. This means that one can work with ECO method without the risk to miss anything that could be catched by object grammars. In **Chapter 5** the result of Chapter 4 is extended to natural q-parameters on unidimensional grammars.

Since it can also encode non algebraic structures, the ECO method is strictly more expressive than object grammars for linear parameters. However, one still wishes to get the grammar representation when possible, in order to exhibit algebraicness. In **Chapter 6** we extend a method of Fédou and Garcia in order to recover, in a quasi-automatic way, a grammar for the class of convex polyominoes from an ECO-system for this class. The point here is that the grammar for convex polyominoes is significantly more difficult to obtain directly than the ECO-system.

1.2 Some combinatorial structures

We start with the definition of some classical combinatorial structures. These structures will serve us as running examples during the all thesis.

1.2.1 Motzkin paths and their generalization

The word *path* is a generic term used to denote a sequence of points, $s_0, s_1, \ldots s_n$, in the plane $\mathbb{Z} \times \mathbb{Z}$. Paths play a fundamental role in Combinatorics. Goulden and Jackson [64] devoted a chapter of their book to "Combinatorics of Paths", including an exhaustive bibliography as well as enumerative results and bijections with other objects. In [54] Flajolet dealt with continued fractions from a combinatorial point of view, by giving a combinatorial interpretation of classical expansions in terms of paths. Roblet also worked on this topic by presenting a combinatorial theory to study Padé's approximations [84]. The concept of paths is essential in the combinatorial theory of orthogonal polynomials developed by Viennot [98]. Paths also appears in problems arising from Computer Science, such as the evaluation of algorithms on files (Flajolet et al. [55]).

A couple (s_i, s_{i+1}) is said to be a *step* of the path and the number of steps is called the *length* of the path. We shall concentrate on some families of directed paths that are related to classical sequences of numbers. Given $s_i = (x, y)$ then (s_i, s_{i+1}) is:

- an *east* or *horizontal* step if $s_{i+1} = (x + 1, y)$,
- a *k-horizontal* step if $s_{i+1} = (x + k, y)$,
- a *north-east* or a *rise* step if $s_{i+1} = (x + 1, y + 1)$,
- a *south-east* or a *fall* step if $s_{i+1} = (x + 1, y - 1)$.

For simplicity, a step (s_i, s_{i+1}) is often represented by a couple (k, l) where k (resp. l) is the difference between the abscissas (resp. the ordinates) of s_{i+1} and s_i. Therefore $(k, 0)$ denotes a k-horizontal step, $(1, 1)$ denotes a rise step, and $(-1, 1)$ denotes a fall step. Most often we find that it is more convenient to represent a path as a finite sequence, actually as a concatenation, of steps.

A *generalized Motzkin path* is a sequence of rise, fall and k-horizontal steps, running from $(0, 0)$ to $(n, 0)$ and remaining weakly above the x-axis. Generalized Motzkin paths have been extensively studied (see references in Sulanke [96, 95, 97], and also Pergola and Sulanke [48]). They include Dyck, Motzkin and Schröder paths, corresponding respectively to the cases $k = 0$, $k = 1$, and $k = 2$. A path remaining strictly above the x-axis is called *elevated*. An elevated Dyck path is shown in Figure 1.1.

A *k-coloured generalized Motzkin path* is a generalized Motzkin path for which the horizontal steps can have more than one colour. An example of a bicoloured Schröder path is shown in Figure 1.1. The bicoloured horizontal steps are represented by dashed and solid horizontal lines. Let P be a coloured generalized Motzkin path. As already said, the length of P is the abscissa of its end point. A *peak* (resp. *valley*) is a couple of consecutive rise and fall steps (resp. fall and rise). The *height* of a point of the path is its ordinate. The *height of* P is the ordinate of its higgest point. The *area* of P is usually defined as the sum of the final heights of its rise and horizontal steps (see Figure 1.2 for an example).

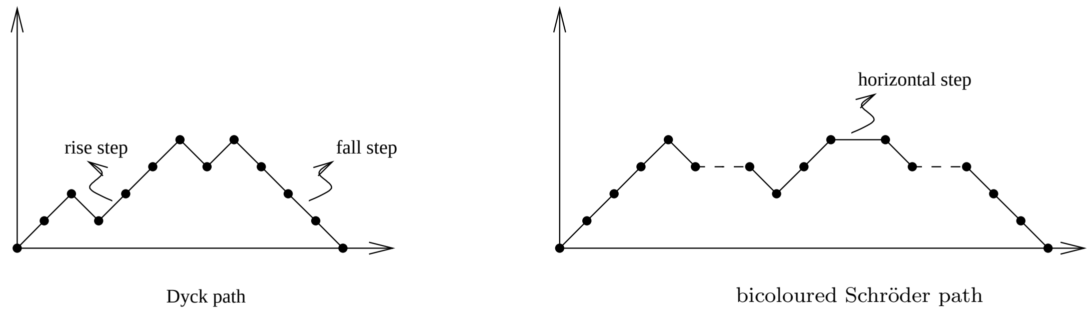

Figure 1.1: Two kinds of coloured generalized Motzkin paths.

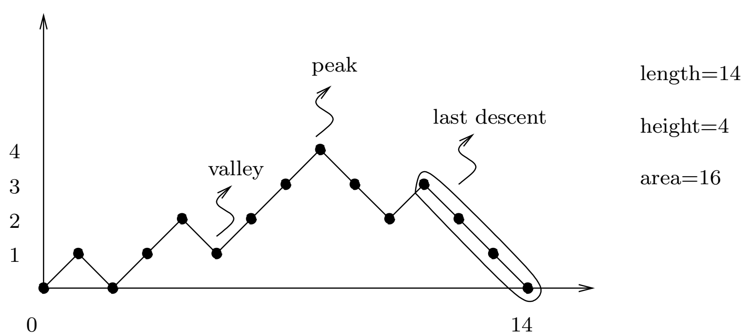

Figure 1.2: A Dyck path.

1.2.2 Polyominoes

A *cell* is defined as a unit square in the plane $\mathbb{Z}^2$. A *polyomino* is a finite connected union of cells without cut point. We consider the class of polyominoes defined up to translation. Polyominoes have been often studied in combinatorics. There are various problems related to them, like the problem of tilings of the plane, or of a rectangle, by polyominoes (Beauquier and M. Nivat [10]; Conway and Lagarias [27]), or the problem of covering a polyomino by using rectangles (Chaiken et al. [22]). The term *polyomino* is usually attributed to Golomb [63].

The number of cells of a polyomino is called the *area*, and the *perimeter* is the number of edges of its boundary (see Figure 1.3). The *height* and *width* of a polyomino are the number of its rows and columns, respectively. The intersection of a polyomino and an infinite vertical (resp. horizontal) sequence of connected cells is called a *column* (resp. *row*).

The general enumeration problem of polyominoes is difficult to solve and still open, though some asympotic results are known. For example, the number a_n of polyominoes having area n is known up to $n = 94$ (Redelmeier [82]), and in [70] Klarner and Rivest proved that $\lim_n \{a_n\}^{\frac{1}{n}} = \mu$, where $3.72 < \mu < 4.64$. In order to simplify the enumeration problem, various restricted classes of polyominoes were studied. These subclasses were defined with respect to certain notions of convexity or preferred directions of growth. A polyomino is called *vertically* (resp. *horizontally*) *convex* if its intersection with any vertical (resp. horizontal) line is connected. A polyomino both vertically and horizontally convex is said to be *convex*. The *semiperimeter* of a convex polyomino is equal to the sum of its height and width. In Figure 1.4

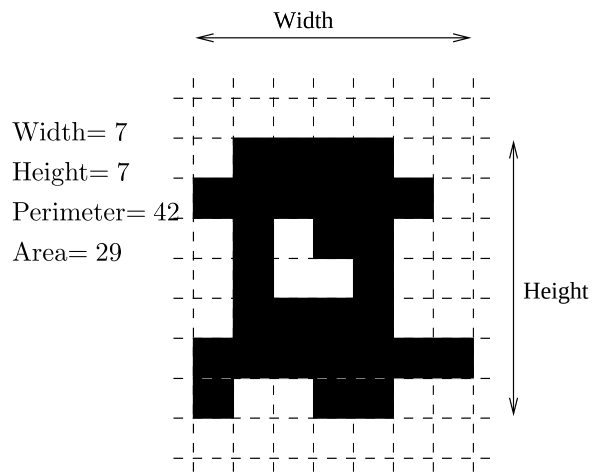

Figure 1.3: A polyomino.

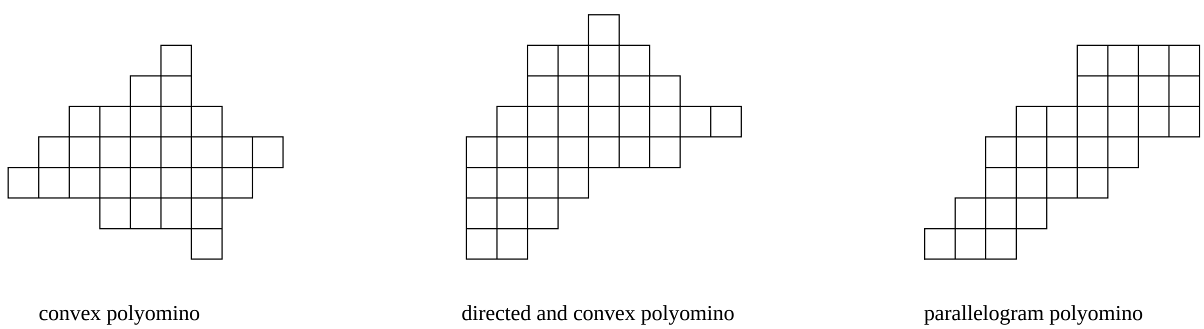

Figure 1.4: Different kinds of polyominoes.

is shown an example of a convex polyomino. A polyomino is *directed* if there is a cell, called the *source*, from which each other cell of the polyomino can be reached by means of a path having *north steps* $(0, 1)$ and *east steps* $(1, 0)$. A polyomino is said to be *directed convex* if it is both convex and directed. A *parallelogram polyomino* is the region lying between two paths made of north and east steps, that are disjoint except for their common end points. The class of parallelogram polyominoes is a subclass of directed convex polyominoes (see Figure 1.4). The results obtained for convex polyominoes are synthetically listed in the following tables adapted from [14], and we refer to Bousquet-Mélou [14] for a nice survey on the class of polyominoes.

1.2.3 Trees

The tree is a widely used structure, it appears in various disciplines and it is the basis for a great number of applications in Computer Science. Arithmetic expression evaluations is the classical example to indroduce binary trees in Computer Science. Trees are relevant for the analysis of algorithms, whether because they implicitly represent the structure of recursive

	Perimeter	Area
Parallelogram	**perimeter:** Levine 59, Polya 69. **perimeter, width and height:** Lin and Chang 88. **perimeter and site perimeter:** Delest et al. 87. **perimeter and different meshes:** Guttmann and Prellberg 93, Essam 93.	**area:** Klarner and Rivest 74. **area and width:** Bender 74, Delest and Fédou 93. **area and perimeter:** Brak and Guttmann 90. **area and width and height:** Ling and Tzeng 91, Bousquet-Mélou and Viennot 92, Fédou and Rouillon 95, Bousquet-Mélou 96, Prellberg and Brak 94. **area by means of continued fraction:** Flajolet 91, Bousquet-Mélou and Viennot 92, Roblet 94.

Table 1.1: The main results on parallelogram polyominoes

	Perimeter	Area
Directed convex	**perimeter, width and height:** Lin and Chang 88, Bousquet-Mélou 92, Bousquet-Mélou and Guttmann 96.	**area and width and height:** Bousquet-Mélou and Viennot 92, Bousquet-Mélou and Fédou 95, Bousquet-Mélou 96. **area and site perimeter:** Dubernard and Dutour 94.
Convex	**perimeter:** Delest and Viennot 84, Kim 88. **perimeter, width and height:** Guttmann and Enting (conjecture) 88, Lin and Chang 88, Gessel 90, Bousquet-Mélou and Guttmann 95.	**area and width and height:** Bousquet-Mélou and Fédou 95, Bousquet-Mélou 96. **symmetric classes:** Leroux and Robitaille 96.

Table 1.2: The main results on directed convex and convex polyominoes

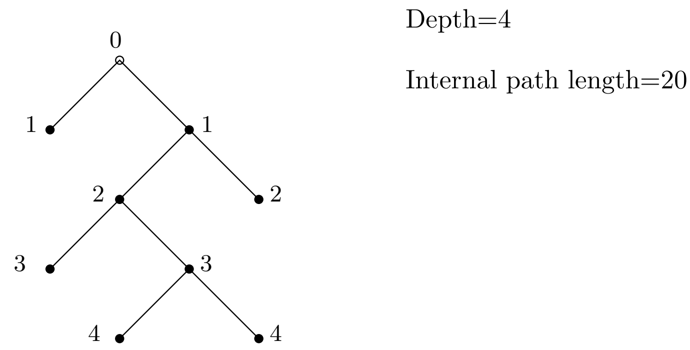

Figure 1.5: A complete binary tree.

programs or because they are explicitly involved in many basic algorithms. Since trees are widely studied, the literature on this subject is enormous (see for instance Knuth [71] and Goulden and Jackson [64]). For a survey of the main results on the enumeration of trees we refer to Flajolet and Sedgewick [60], where we can also find bibliographical references on this subject.

A *plane tree*, or equivalently an *unlabeled ordered tree* and briefly hereafter simply, a *tree*, is defined recursively: a tree consists of a particular node r, called the *root*, and of an ordered set of non-empty trees, $A_1, \dots, A_k$. A plane tree can thus be represented as a list $(r, A_1, \dots, A_k)$, where $A_1, \dots, A_k$ are themselves trees. The roots of the subtrees $A_1, \dots, A_k$ are said to be *sons* of r, and r is said to be their *father*; k is called the degree of r. The nodes without sons are called *leaves*. Let us consider a tree represented in the cartesian plane so that the root has ordinate 0 and each son has ordinate lower than that of its father. The *level* of a node of the tree is the opposite of its ordinate. The *length* of an edge is the difference between the level of the son and that of its father. If each edge has length 1, the level of a node becomes the number of edges between the root and the node. The *depth* of a tree is the maximum level of its nodes and the *internal path length* of a tree is the sum of the levels of its nodes. In agreement with these definitions, we remark that when illustrating a tree, the convention is to draw the tree with the root at the top (or at the left) and its sons located below (located to the right).

A *Schröder tree* is a plane tree without nodes of degree 1. As a particular case, the class of *complete binary trees*, i.e. trees with nodes of degree 0 or 2, is a subclass of Schröder trees. A complete binary tree is represented in Figure 1.5.

1.3 ECO method and succession rules

The *ECO method*, introduced by Barcucci, Del Lungo, Pergola, and Pinzani [6], is a method for Enumerating some classes of Combinatorial Objects. It provides a recursive construction of the objects according to their size, where by "size" we mean the value of a finite parameter defined on the objects. In the ECO method, each object is obtained from a smaller object by making some local expansions. If the recursive construction presents a certain regularity, then it can be encoded in a formal system called *succession rule*. Succession rules represent the intermediate step allowing to describe in a synthetic way the growth mode of the class of objects. Then, from a succession rule associated with an ECO construction for a class of objects, we can often obtain the generating function of the objects themselves. In conclusion, ECO method is a useful tool for the enumeration of combinatorial objects. In Pergola's PhD thesis [78] there are some important enumerative results obtained by means of ECO method. In [51], Ferrari presents an approach to the ECO method from an algebraic point of view.

1.3.1 ECO operators and generating trees

Let $\mathcal{O}$ be a class of combinatorial objects and let $p : \mathcal{O} \to \mathbb{N}$ be a *finite parameter* on $\mathcal{O}$, that is to say, a parameter p such that the set $\mathcal{O}_n = \{O \in \mathcal{O} : p(O) = n\}$ of objects of size n is finite. Let $\vartheta : \mathcal{O} \to 2^{\mathcal{O}}$ be an operator such that $\vartheta(\mathcal{O}_n) \subseteq 2^{\mathcal{O}_{n+1}}$. The operator ϑ describes how small objects produce larger ones. (For S a set, 2^S stands for the set of subsets of S.)

In order for the operator to define objects in a non-ambiguous way we introduce the following restriction.

Proposition 1.3.1. *If ϑ satisfies, for $n \geq 0$,*

1. *for each $O' \in \mathcal{O}_{n+1}$, there exists $O \in \mathcal{O}_n$ such that $O' \in \vartheta(O)$, and*

2. *for every $O, O' \in \mathcal{O}_n$, $\vartheta(O) \cap \vartheta(O') = \emptyset$ whenever $O \neq O'$,*

then the family of sets $\mathcal{F}_{n+1} = \{\vartheta(O) : O \in \mathcal{O}_n\}$ is a partition of $\mathcal{O}_{n+1}$.

Following the definition in [6], an operator ϑ, satisfying conditions 1 and 2 above, is said to be an *ECO operator*. Thus an ECO operator generates all the objects of $\mathcal{O}$ in such a way that each object $O' \in \mathcal{O}_{n+1}$ is obtained from a unique $O \in \mathcal{O}_n$. In fact we shall consider only ECO operators that make local expansions on the so-called *active sites* of the object. The construction performed by ϑ can be described by a *generating tree*, i.e., a rooted tree whose nodes correspond to the objects of $\mathcal{O}$. The root, placed at level 0 of the tree, is the object with minimum size, m. Objects with the same size lie at the same level and the sons of an object O are those produced by O through ϑ (see Barcucci et al. [6] and Chung et al. [26] for further informations on generating trees). Let $\{|\mathcal{O}_n|\}_n$ be the sequence determined by the number of objects with size n. Then $f_{\mathcal{O}}(x) = \sum_{n \geq m} |\mathcal{O}_n| x^n$ is its *generating function*.

Example 1.3.1. *Let $\mathcal{D}$ be the class of Dyck paths and $\mathcal{D}_n$ be the subclass of Dyck paths with semi-length n. For a Dyck path $D \in \mathcal{D}$, we denote by $\ell_d(D)$ the last sequence of fall steps of D, said also* last descent *of D. We define the set of active sites of D as the set $\mathbf{P}(D)$ of points on the last decent of D (or more precisely initial and final points of the steps of $\ell_d(D)$). Let us define an operator from $\mathcal{D}$ to $2^{\mathcal{D}}$ as follows: the image $\vartheta(D)$ of a Dyck path $D \in \mathcal{D}_n$ is the set of paths of $\mathcal{D}_{n+1}$ that can be obtained by adding a peak at one of the active sites of D.*

Let us check that ϑ is an ECO operator, that is, ϑ generates all the Dyck paths, and in a unique way. Indeed, given a non-empty Dyck path D' we recover the unique Dyck path D such that $D' \in \vartheta(D)$ by erasing the last peak of D'. In Figure 1.6 are represented the first levels of the generating tree of ϑ, where the circled points represent the active sites.

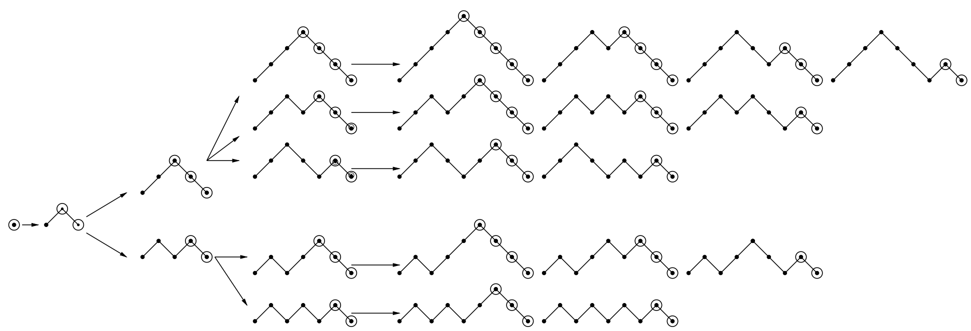

Figure 1.6: Classic generating tree for Dyck paths. (Paths at level 4 are displayed horizontally to save space.)

Example 1.3.2. *Let $\mathcal{S}$ be the class of Schröder paths and let $S \in \mathcal{S}$. Let $\ell_d(S)$ be the last sequence of fall and horizontal steps of S. The set of active sites of S is defined as the set $\mathbf{P}(D)$ of points on $\ell_d(S)$ (or more precisely initial or final points of the steps of $\ell_d(S)$). We define an ECO operator ϑ from $\mathcal{S}_n$, the class of Schröder paths with semi-length n, to the power set $2^{\mathcal{S}_{n+1}}$ of $\mathcal{S}_{n+1}$: the image $\vartheta(S)$ of a Schröder path S is the set of paths obtained by simultaneously inserting a rise step on any active site and a fall step at the end of S, or by inserting an horizontal step at the end of S. Again the fact that ϑ is an ECO operator is immediate: given S', the path S such that $S' \in \vartheta(S)$ is obtained by removing the last step of S' and, if it is a fall step, the last rise step. In Figure 1.7 are represented the first levels of the generating tree of ϑ.*

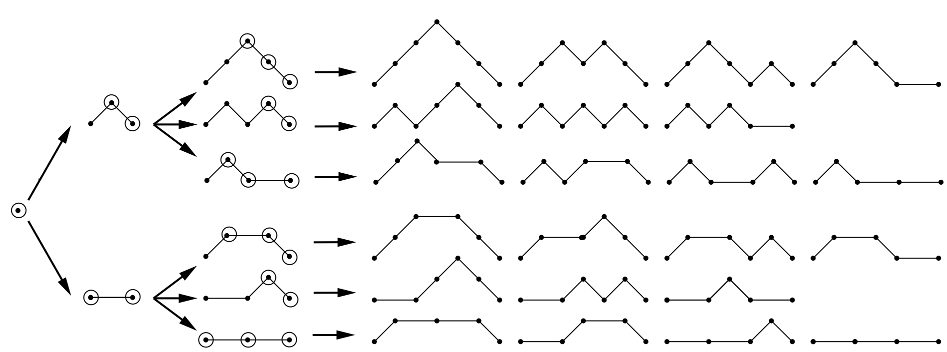

Figure 1.7: Classic generating tree for Schröder paths.

1.3.2 Succession rules

Succession rules were first introduced by Chung, Graham, Hoggat, and Kleimann [26] for studying Baxter permutations. Such formal systems were later used by West [99], and by

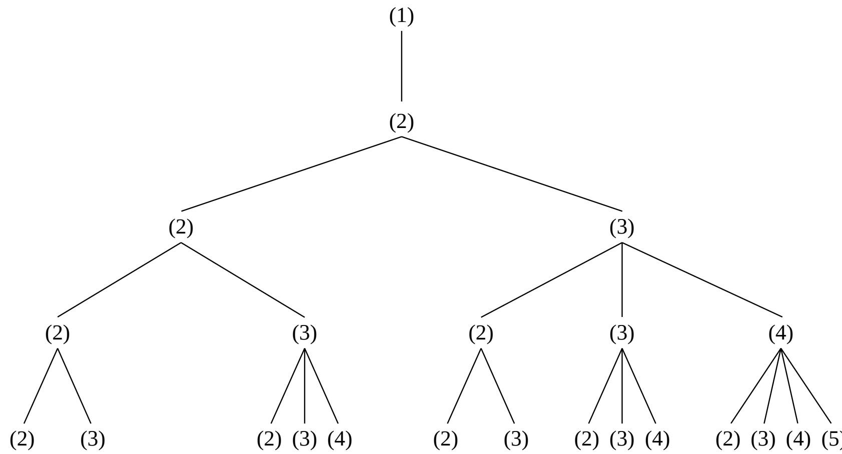

Figure 1.8: Generating tree for the Catalan succession rule.

Dulucq, Gire, and Guibert [43, 42, 41] for the enumeration of permutations with forbidden sequences. In Rinaldi's PhD thesis [83] we can find a complete survey on succession rules.

A *succession rule* Ω is a *system* $((a), \mathcal{P})$, consisting of an axiom (a) and a set $\mathcal{P}$ of *productions* or *rewriting rules* defined on a set of labels $M \subset \mathbb{N}^+$:

$$\Omega = \begin{cases} (a) \\ (k) \rightsquigarrow (e_1(k))(e_2(k)) \dots (e_k(k)), \quad \text{for all } k \in M, \end{cases} \tag{1.1}$$

where $a \in M$ is a fixed constant and the e_i are functions $M \to M$.

One of the main properties of a succession rule is the *consistency principle*, i.e. each label (k) must produce exactly k elements. A succession rule Ω induces, and is suitably represented by, a generating tree whose root is labeled by the axiom (a), and a node labeled (k) produces at the next level k sons labeled by $(e_1(k)), \dots, (e_k(k))$ respectively (which in turn will produce respectively $e_1(k), \dots, e_k(k)$ sons, etc.). The succession rule produces a sequence $\{f_n\}_n$ of positive integers, where f_n is the number of nodes at level n of the generating tree and its generating function is denoted $f_\Omega(x) = \sum_{n \geq 0} f_n x^n$.

Often an ECO operator ϑ can be encoded in a succession rule of the form (1.1), meaning that the object with minimum size has a sons and the k objects $O'_1, \dots, O'_k$, produced by an object O are such that O'_i will in turn produce $e_i(k)$ sons by ϑ, i.e. $|\vartheta(O'_i)| = e_i(k)$, $1 \leq i \leq k$. In that case there is an isomorphism between the generating tree of the ECO operator and that of the corresponding succession rule. Consequently we have $f_{\mathcal{O}}(x) = x^m f_\Omega(x)$, that reduces to $f_{\mathcal{O}}(x) = f_\Omega(x)$ when $m = 0$.

Let ϑ be an ECO operator for $\mathcal{O}$ according to p. Assume there is a succession rule Ω_ϑ associated with ϑ. The quadruple $\Sigma = (\mathcal{O}, p, \vartheta, \Omega_\vartheta)$ is called an *ECO-system*.

Example 1.3.3. *Let us consider again the classic ECO construction for Dyck paths, defined in Example 1.3.1. Let us suppose that k is the number of active sites of a path D. If the operator ϑ inserts a peak in the first active site (starting from the bottom) of D, then it obtains a new path D' with $|\mathbf{P}(D')| = 2$ active sites. On the other hand, if ϑ inserts a peak in the k-th active site of D, then it obtains a path D' with $|\mathbf{P}(D')| = k + 1$ active sites (see Figure 1.6). In general, a path with k active sites produces, through ϑ, k paths, respectively with* 2, 3, … ,

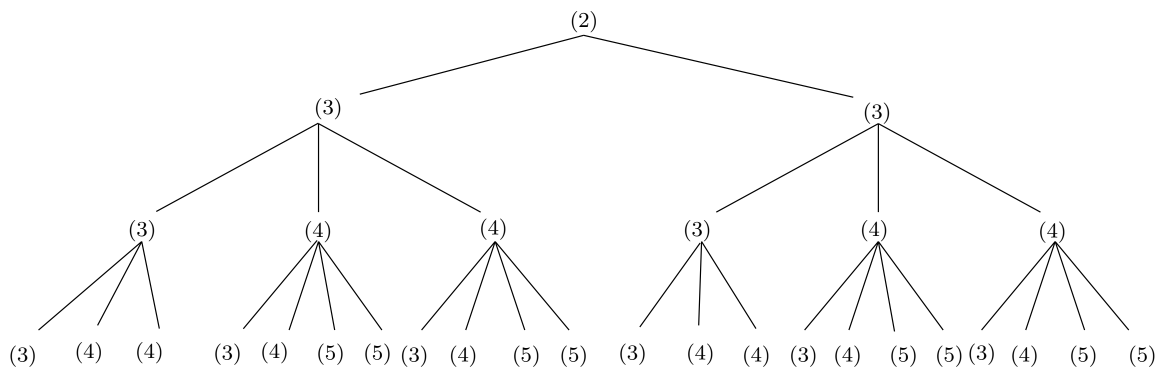

Figure 1.9: Generating tree for the Schröder succession rule.

$k+1$ active sites. Thus the rule associated with the classic ECO construction for Dyck paths is the classic rule for Catalan numbers:

$$\Gamma = \left\{ \begin{array}{l} (1) \\ (k) \rightsquigarrow (2)\dots(k)(k+1) \end{array} \right.$$

In Figure 1.8 are represented the first levels of the generating tree of Γ. This tree is clearly isomorphic to the generating tree of Figure 1.6.

Example 1.3.4. *Let us consider the classic ECO construction for Schröder paths, defined in Example 1.3.2. Let us take a Schröder path S with $k-1$ active sites. If the operator ϑ adds the rise step in the first active site of S, then it produces a path with two active sites. On the other hand, if ϑ adds the rise step in the $(k-1)$-th active site of S, then it obtains a path with k active sites. In general, a path with $k-1$ active sites produces $k-1$ paths, respectively with $2, 3, \dots\ k$ active sites, and a further path with k active sites, since ϑ can also inserts an horizontal step in the first active site of S (see Figure 1.7). We remark that, in this case, a path with $k-1$ active sites produces k new paths. Therefore the rule associated with ϑ is the classic rule for Schröder numbers:*

$$\Upsilon = \left\{ \begin{array}{l} (2) \\ (k) \rightsquigarrow (3)\dots(k)(k+1)^2, \end{array} \right.$$

where the power notation is used to express repetitions, that is $(k+1)^2$ stands for $(k+1)(k+1)$. The first levels of the generating tree of Υ are shown in Figure 1.9.

1.3.3 The generating function of a succession rule

Now, in order to illustrate the use of the ECO method for enumeration, and to introduce the kernel method, we compute the generating functions of Γ and Υ. Let

$$f_\Gamma(x) = \sum_{n\geq 0} f_n x^n$$

be the generating function of the succession rule Γ, where f_n is the number of nodes at level n of the generating tree. Let $f_{n,k}$ be the number of nodes at level n having label (k). Then we define

$$f_\Gamma(x,y) = \sum_{n\geq 0, k\geq 1} f_{n,k} x^n y^k.$$

Consequently $f_\Gamma(x,1) = f_\Gamma(x)$. From the succession rule Γ, we deduce that

$$f_\Gamma(x,y) = x^0 y + x \sum_{n\geq 0, k\geq 1} f_{n,k} x^n (y^2 + y^3 + \ldots + y^{k+1}). \tag{1.2}$$

Indeed, the $f_{n,k}$ nodes lying at level n and having label k, produce $kf_{n,k}$ nodes one level below, among which $f_{n,k}$ are labeled j, for $j = 2\ldots k+1$. From equation (1.2) we obtain

$$f_\Gamma(x,y) = y + x \sum_{n\geq 0, k\geq 1} f_{n,k} x^n \frac{y^2 - y^{k+2}}{1-y},$$

which leads to the following functional equation:

$$f_\Gamma(x,y)\left(1 + x\frac{y^2}{1-y}\right) = y + \frac{y^2}{1-y} f_\Gamma(x,1). \tag{1.3}$$

An essential ingredient to solve equation (1.3) and in general, to solve equations of that form, is the so-called *kernel method*. This method has been around since the 70's and it has been used in various combinatorial problems. For instance, we can find it in the work of Cori and Richard [29] on planar graphs, or in the Knuth's book [71] concerning the art of computer programming. An application of the kernel method in probabilities is in the paper of Fayolle et Iasnogorodski [49]. Recent applications and formalizations of this method can be found in the work of Banderier et al. [1], Banderier and Flajolet [2], Bousquet-Mélou [15], and Bousquet-Mélou and Petkovšek [18].

The kernel method consists in coupling the variables x and y in order to cancel the left-hand side of the functional equation (1.3) in a way that the coefficient of $f_\Gamma(x,y)$, called just "the kernel", is 0. One way to do this is to take y to be a solution $y(x)$ of the kernel equation $1 + x\frac{y^2}{1-y} = 0$. If $y(x)$ can be substituted into the right-hand side of the functional equation, then the value of $f_\Gamma(x,1)$ is obtained in the right-hand side.

Here the numerator of the kernel is a polynomial of degree 2, thus it has two roots:

$$\begin{aligned} y_0(x) = &\ \tfrac{1-\sqrt{1-4x}}{2x} = 1 + x + 2x^2 + O(x^3) \\ y_1(x) = &\ \tfrac{1+\sqrt{1-4x}}{2x} = x^{-1} - 1 - x - 2x^3 + O(x^3). \end{aligned}$$

Observe that only $y_0(x)$ can be substituted in equation (1.3). Indeed, because of the negative exponent of the first term of $y_1(x)$, $f_\Gamma(x, y_1(x))$ is not a well-defined power series. Thus, we substitute $y_0(x)$ in the equation $y + x\frac{y^2}{1-y} f_\Gamma(x,1) = 0$, and we obtain

$$f_\Gamma(x) = f_\Gamma(x,1) = \frac{y_0(x) - 1}{x y_0(x)} = \frac{1-\sqrt{1-4x}}{2x}$$

The generating function $f_\Gamma(x)$ defines the sequence 1, 1, 2, 5, 14, 42 of Catalan numbers (sequence A000108 in "The encyclopedia of integer sequences" [92]).

The calculation of $f_\Upsilon(x)$ is similar to that of $f_\Gamma(x)$. By maintaining the notations above we have

$$f_\Upsilon(x,y) = x^0y^2 + x \sum_{n\geq 0, k\geq 1} f_{n,k}x^n(y^3+y^4+\ldots+y^{k+1}+y^{k+1}), \tag{1.4}$$

from which we obtain

$$f_\Upsilon(x,y) = x^0y^2 + x \sum_{n\geq 0, k\geq 1} f_{n,k}x^n \frac{y^3-y^{k+2}}{1-y} + x \sum_{n\geq 0, k\geq 1} f_{n,k}x^n y^{k+1}.$$

After some steps we obtain

$$f_\Upsilon(x,y)\left(1+x\frac{y^2}{1-y}-xy\right) = y^2 + x\frac{y^3}{1-y}f_\Upsilon(x,1)$$

By applying the kernel method, we obtain

$$y_0(x)^2 + x\frac{y_0(x)^3}{1-y_0(x)}f_\Upsilon(x,1) = 0,$$

where $y_0(x) = \frac{1+x-\sqrt{1-6x+x^2}}{4x}$. Consequently,

$$f_\Gamma(x) = \frac{1-x-\sqrt{1-6x+x^2}}{2x},$$

defining the sequence 1, 2, 6, 22, 90, 394, $\ldots$ of Schröder numbers (sequence A006318 in [92]).

A succession rule is called *rational*, *algebraic* or *trascendental* according to its generating function type. In [1], Banderier et al. investigated the link between the structural properties of succession rules and the rationality, algebraicity, or trascendence of their corresponding generating functions. In particular, by using some criteria, they established the kind of generating function associated with several classes of succession rules.

We define a succession rule Ω to be *finite* if the number of labels in the productions is finite, that is, when $|M| < \infty$, in the notation of (1.1). In this particular case, the generating function is rational, as shown in [1], and sometimes has an interpretation as a regular language or other combinatorial structures (see Ferrari et al. [53], Rinaldi [83]).

A classical example of a finite succession rule is the one defining the Fibonacci numbers, $1, 1, 2, 3, 5, 8, 13, \ldots$ (sequence A000045 in [92])

$$\Omega_{\mathcal{F}} = \begin{cases} (1) \\ (1) \rightsquigarrow (2) \\ (2) \rightsquigarrow (1)(2), \end{cases} \tag{1.5}$$

whose generating function is $\frac{x}{1-x-x^2}$.

A succession rule has a *factorial form*, if a finite modification of the set $\{1, 2, \ldots, k\}$ is reachable from k. More formally, a factorial succession rule has the form:

$$\Omega = \begin{cases} (a) \\ (k) \rightsquigarrow (r_0)(r_0+1)\ldots(k-c-1)(k+d_1)(k+d_2)\ldots(k+d_m), \ k \geq r_0, \end{cases} \tag{1.6}$$

where $a \geq r_0 \geq 1$, $c \geq 0$ and the d_i are constants with $-c \leq d_1 \leq d_2 \leq \ldots \leq d_m$, and the consistency principle of succession rules is satisfied imposing that $r_0 + c = m$. The classic rule for Catalan numbers of Example 1.3.1 is factorial. In [1] Banderier et al. formalize the kernel method, and then apply it in order to find a solution to the functional equation arising from a factorial succession rule. Their main result states that a factorial succession rule has an algebraic generating function.

1.4 Object grammars

As already explained, *object grammars*, introduced by Dutour in [44], illustrate another recursive method for enumerating combinatorial objects, namely the approach of decomposable structures. An object grammar describes a class of combinatorial objects by means of terminal objects and operations applied to the objects, resulting in a tree-like decomposition. We give here a short introduction to object grammar, a more detailed account of which is given in Chapter 4

Definition 1.4.1. *Let $\mathbb{O}$ be a finite family of classes of objects. An* object operation *in $\mathbb{O}$ is a mapping $\phi : \mathcal{O}^1 \times \ldots \times \mathcal{O}^k \to \mathcal{O}$, where $\mathcal{O}, \mathcal{O}^i \in \mathbb{O}$, $i = 1, \ldots, k$. The domain and codomain of an object operation are respectively denoted as* dom *and* cod.

An object operation describes a way of building recursively an object of $\mathcal{O}$ from k objects belonging to $\mathcal{O}^1, \ldots, \mathcal{O}^k$.

Definition 1.4.2. *An* object grammar *is a quadruple $\langle \mathbb{O}, \mathbb{E}, \Phi, \mathcal{A} \rangle$ where:*

- *$\mathbb{O}$ is a finite family of classes of objects.*
- *$\mathbb{E} = \{\mathcal{E}_{\mathcal{O}}\}_{\mathcal{O} \in \mathbb{O}}$ is a finite family of finite subclasses of the classes belonging to $\mathbb{O}$. The objects of $\mathbb{E}$ are called* terminal objects.
- *Φ is a finite set of object operations in $\mathbb{O}$.*
- *$\mathcal{A}$ is a fixed class of $\mathbb{O}$, called the* axiom *of the grammar.*

The *dimension* of an object grammar is the cardinality of $\mathbb{O}$.

Definition 1.4.3. *Let $G = \langle \mathbb{O}, \mathbb{E}, \Phi, \mathcal{A} \rangle$ be an object grammar and let $\mathcal{O} \in \mathbb{O}$. A* derivation tree *of G on $\mathcal{O}$ is an ordered labelled tree T, recursively described as follows:*

- *if T is reduced to a leaf then the label is a terminal object belonging to $\mathcal{O}$,*

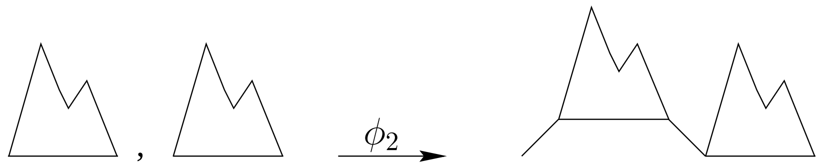

Figure 1.10: The object operation ϕ_2 of the grammar $G_{\mathcal{D}}$.

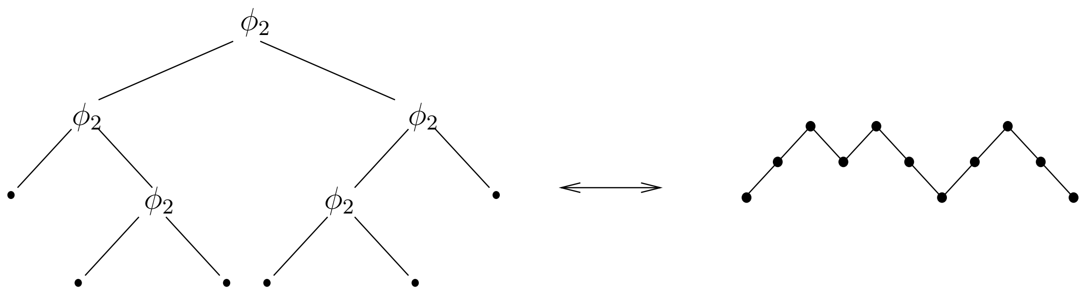

Figure 1.11: A derivation tree from the grammar $G_{\mathcal{D}}$ and the corresponding Dyck path.

- *if the root of T has k sons then its label is an object operation $\phi \in \Phi$,*

$$\phi : \mathcal{O}^1 \times \ldots \times \mathcal{O}^k \rightarrow \mathcal{O},$$

where $\mathcal{O}^i \in \mathbb{O}$ and the i-th son of the root is the root of a derivation tree on the class $\mathcal{O}^i$, $i = 1 \ldots k$.

Definition 1.4.4. *The valuation $ev(T)$ of a derivation tree T is an object defined as follows :*

- *if T is a single node labelled E, then $ev(T) = E$,*
- *otherwise, if the root of T is labelled $\phi \in \Phi$ and its k subtrees are $T_1, \ldots, T_k$, then $ev(T) = \phi(ev(T_1), \ldots, ev(T_k))$.*

Definition 1.4.5. *Let $G = \langle \mathbb{O}, \mathbb{E}, \Phi, \mathcal{A} \rangle$ be an object grammar. An object $O \in \mathcal{O}$ is said to be* generated *in G by $\mathcal{O}$ if there is a derivation tree T on $\mathcal{O}$ such that $ev(T) = O$.*

The class of objects generated in G by $\mathcal{A}$ is said to be the class generated by G.

Example 1.4.1. *Let $\mathcal{D}$ be the class of Dyck paths. The mapping ϕ_2 depicted in Figure 1.10 is a binary object operation on the class $\mathcal{D}$ of Dyck paths: it takes a pair of Dyck paths as its argument, adds a rise (resp. fall) step at the beginning (resp. end) of the first path and then appends the second path. The class $\mathcal{D}$ is obviously generated by the object grammar*

$$G_{\mathcal{D}} = \langle \mathcal{D}, \{\{.\}\}, \{\phi_2\} \rangle$$

where the terminal object is the Dyck path of zero length, commonly represented as a dot. Each Dyck path is then univocally associated with a derivation tree of $G_{\mathcal{D}}$ (see for instance Figure 1.11). Introducing the generating function $f_{\mathcal{D}}(x) = \sum_{n \geq 0} f_n x^n$ where f_n is the *number of Dyck path of semi length n, the object grammar immediately translates into an algebraic equation*

$$f_{\mathcal{D}}(x) = 1 + x f_{\mathcal{D}}(x)^2,$$

from which the generating function $f_{\mathcal{D}}(x) = \frac{1-\sqrt{1-4x}}{2x}$ is easily recovered.

Part I

ECO-systems and succession rules

Chapter 2

The equivalence problem for succession rules

The notion of succession rule provides a powerful tool for the enumeration of many classes of combinatorial objects. Therefore we are interested in problems related to such formal systems. An important one is the *equivalence problem*. The equivalence problem is that of determining if two different succession rules are equivalent. Two rules Ω_1 and Ω_2 are said to be *equivalent*, if they define the same numerical sequence, i.e.,

$$\Omega_1 \cong \Omega_2 \iff f_{\Omega_1}(x) = f_{\Omega_2}(x) \ .$$

For instance, as shown by Barcucci et al. in [7, 8], the following rules Υ', Υ'', and Υ''' are equivalent to the classic rule Υ for Schröder numbers (see Example 1.3.2), whose generating tree is represented in Figure 1.9.

$$\Upsilon = \begin{cases} (2) \\ (k) \rightsquigarrow (3)\dots(k)(k+1)^2, \end{cases}$$

$$\Upsilon' = \begin{cases} (2) \\ (2k) \rightsquigarrow (2)(4)^2\dots(2k)^2(2k+2), \end{cases}$$

$$\Upsilon'' = \begin{cases} (2) \\ (2) \rightsquigarrow (3)(3) \\ (2k-1) \rightsquigarrow (3)^2(5)^2\dots(2k-1)^2(2k+1), \end{cases}$$

$$\Upsilon''' = \begin{cases} (2) \\ (2^k) \rightsquigarrow (2)^{2^{k-1}}(4)^{2^{k-2}}(8)^{2^{k-3}}\dots(2^{k-1})^2(2^k)(2^{k+1}). \end{cases}$$

In general, as mentioned recently by M. Robson [85], the equivalence problem is not decidable. However there are classes of rules for which the problem is indeed decidable. In particular, the problem turns out to be decidable for the classes of finite and factorial succession rules, because the corresponding generating functions are respectively rational (see Rozenberg and Salomaa [86] or Banderier et al. [1]) and algebraic (Banderier et al. [1]). As illustrated by these two classes, a direct way of proving the equivalence of two rules is to calculate their generating functions. However, determining the generating function of a given succession rule is not always an easy task. Therefore, some recent papers have focused on the development of algebraic tools in order to study enumerative properties of succession rules, without computing the corresponding generating functions, by using a linear operator approach (Ferrari et al. [53]), or by using production matrices (Deutsch [36]). These methods allow to produce easily new equivalent bijective rules.

In this chapter we consider a different approach, that is the problem of showing the equivalence of two succession rules in a bijective way. The main advantage of this approach is that it gives a combinatorial understanding of the corresponding diverse growth modes. In other terms, two different rules can be proved equivalent by providing a combinatorial proof that they describe two different growth modes of a same class with respect to the same parameter. Such a bijective approach could also be useful when dealing with succession rules for which it is difficult to determine the generating functions. The main part of this chapter is devoted to show the equivalence of some sets of succession rules in this way, that is by providing different ECO constructions for the same combinatorial class according to the same parameter. At the end of the chapter we give other examples of equivalent succession rules, by proving their equivalence by means of generating functions.

The chapter is organized as follows. In Section 2.1 we recall the results on the equivalence problem for finite and factorial rules. We first show how the result for finite rules can be easily deduced from the language theory of *D0L*-systems. Then we recall the result on factorial succession rules and an extension of these. In Section 2.2 we mainly introduce new succession rules for Catalan and Schröder numbers and we show their equivalence by using ECO method. Finally we transport the ECO operators along some known bijections between paths and trees. In Section 2.3 we introduce an infinite set of rules for Ballot numbers and we show their equivalence by using the ECO method. Finally, in Section 2.4 we introduce other infinite sets of equivalent rules and we prove their equivalence by computing their generating functions. An account of the results of this chapter can be found in [20, 21].

2.1 Some decidable classes

2.1.1 Finite rules

The easy case of finite succession rules stems from the theory of *D0L systems*. We quickly give some notions concerning these systems. We refer to the books by Salomaa [87], Rozenberg and Salomaa [86], and Salomaa and Soittola [88] for a complete survey on this topic.

Let $\Sigma = \{a_1, \ldots, a_k\}$ an alphabet with k letters. The *length* of a word $w \in \Sigma^*$ is denoted $|w|$. The *empty word* is denoted ε. A *Deterministic 0-context L-system*, shortly a *D0L* system, is a triple

$$G = (\Sigma, h, w_0),$$

where h is an endomorphism defined on the monoid Σ^* with concatenation, and $w_0 \in \Sigma^*$ is called the *axiom*. The endomorphism h is *nonerasing* if $h(a_i) \neq \varepsilon$ for $i \in \{1, \ldots, k\}$. The *language* of G is defined by

$$L(G) = \{h^i(w_0) : i \geq 0\}.$$

The function $f_G : \mathbb{N} \to \mathbb{N}$ defined by $f_G(n) = |h^n(w_0)|$, $n \geq 0$, is the *growth function* of G, and the sequence $|h^n(w_0)|$, $n \geq 0$, is its *growth sequence*.

A *D0L system* is said to be *propagating* or, shortly, a *PD0L* system if h is nonerasing.

The *growth matrix* M associated to a *D0L* system G is defined by

$$M[i,j] = |h(a_j)|_{a_i},$$

where $|h(a_j)|_{a_i}$ is the number of occurrences of the letter a_i in $h(a_j)$. Let

$$\pi = (|w_0|_{a_1}, \dots |w_0|_{a_k}) \quad \text{and} \quad e = (1, \dots, 1)^t$$

where t stands for transpose. By induction on n we have that

$$f_G(n) = \pi M^n e. \tag{2.1}$$

Let $f_G(x)$ be the generating function of the growth sequence of G

$$f_G(x) = \sum_{n=0}^{\infty} f_G(n) x^n.$$

In view of equation (2.1), $f_G(x)$ can be expressed as the quotient of two polynomials, $Q(x)$ and $R(x)$, where

$$\begin{aligned} Q(x) &= \pi \left(\det(I - Mx)(I - Mx)^{-1} \right) e \\ R(x) &= \det(I - Mx). \end{aligned} \tag{2.2}$$

Let $G = (\Sigma, h, w_0)$ and $G' = (\Sigma', g, u_0)$ be two *D0L* systems and $f_G(x) = Q(x)/R(x)$ (*resp.* $f_{G'}(x) = Q'(x)/R'(x)$) be the generating function of G (*resp.* G'). Now, G and G' are *growth equivalent* if they have the same generating function, which amounts to check if two polynomials are equal

$$Q(x)R'(x) \text{ and } Q'(x)R(x).$$

However, the computation of the polynomials can be avoided, by checking the equality of the first few terms of $L(G)$ and $L(G')$. In fact one can see (Theorem 3.3 in [86]) that G and G' have the same generating function if and only if

$$|h^i(w_0)| = |g^i(u_0)| \quad \text{for } 0 \le i \le k + k' - 1,$$

where and k (resp. k') is the cardinality of Σ (resp. Σ').

We remark that any finite succession rule Ω can be viewed as a particular *PD0L* system where the alphabet Σ is the set of labels of Ω, h is defined by the productions of Ω, and $w_0 \in \Sigma$.

For instance, the rule (1.5) for Fibonacci numbers, defines a *PD0L* system F, where $\Sigma = \{1, 2\}$, $w_0 = 1$, and

$$\begin{array}{rcl} h(1) &=& 2 \\ h(2) &=& 12 \end{array} \qquad M = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}.$$

The words in the language of F are

$$1, 2, 12, 212, 12212, 21212212, 1221221212212, \dots,$$

and its growth sequence is obtained from the matrix associated to G by computing the generating function (2.2)

$$f_G(x) = (1 + x + 2\,x^2 + 3\,x^3 + 5\,x^4 + 8\,x^5 + 13\,x^6 + 21\,x^7 + O\left(x^8\right)).$$

From the previous assertions we have the following

Theorem 2.1.1. *The equivalence problem is decidable for the class of finite succession rules. More precisely two finite succession rules, Ω_1 and Ω_2, are equivalent if the first $k_1 + k_2$ terms of the two sequences they define coincide.*

For example, the finite rules

$$\Omega_1 = \begin{cases} (2) \\ (2) \rightsquigarrow (2)(3) \\ (3) \rightsquigarrow (2)(3)(3), \end{cases} \qquad \Omega_2 = \begin{cases} (2) \\ (1) \rightsquigarrow (2) \\ (2) \rightsquigarrow (1)(4) \\ (4) \rightsquigarrow (1)(2)(4)(4), \end{cases} \tag{2.3}$$

both define odd index Fibonacci numbers, $1, 2, 5, 13, 34, 89, \ldots$ (sequence M1439 in [92]). Their equivalence can be verified by comparing the first 5 terms of the defined sequences.

2.1.2 Factorial rules

As already mentioned, Banderier et al. [1] proved the following result:

Theorem 2.1.2 (Banderier et al.). *Factorial succession rules have algebraic generating functions.*

This result has been extended by Fédou and Garcia:

Theorem 2.1.3 (Fédou-Garcia). *Succession rules of the form*

$$(k) \rightsquigarrow (1)^{\alpha_{k-1}} \ldots (k-1)^{\alpha_1} (k)^{\lambda_0} \ldots (k+p)^{\lambda_p}$$

have an algebraic generating function when the sequence (α_i) is rational.

Corollary 2.1.1. *The equivalence problem is decidable for the rules of the previous kind.*

Proof. A classical result on the equality of algebraic generating functions in several commutative variables, shows that the equality is decidable (see Theorem IV. 5.1 by Salomaa and Soittola [88]) □

2.2 A proof of equivalence by ECO method

In this section we first present an infinite family of equivalent succession rules parameterized for a positive integer α. Then, we prove the equivalence thanks to a new ECO construction for Dyck paths. The construction is then extended to colored Schröder paths. In order to explore the properties of these new growth operators, we translate them in terms of binary trees and Schröder trees. These latter trees are enumerated by little Schröder numbers (Schröder [89]) according to the number of their leaves (Penaud et al. [77]).

The two families of rules that we will prove equivalent are, for any $\alpha \in \mathbb{N}^+$,

$$\Omega_\alpha = \begin{cases} (\alpha) \\ (\alpha) \rightsquigarrow (\alpha+1)^\alpha \\ (k) \rightsquigarrow (\alpha+1)(\alpha+2)\dots(k-1)(k)(k+1)^\alpha, \end{cases}$$

and

$$\Omega'_\alpha = \begin{cases} (\alpha) \\ (\alpha) \rightsquigarrow (\alpha)^{\alpha-1}(2\alpha) \\ (2k\alpha) \rightsquigarrow (\alpha)^{k\alpha}(2\alpha)^{\alpha-1}(4\alpha)^\alpha(6\alpha)^\alpha \dots (2(k-1)\alpha)^\alpha(2k\alpha)^\alpha(2(k+1)\alpha). \end{cases}$$

The rule Ω_α is related to the classic rules for Catalan and Schröder numbers associated with the classic ECO constructions described respectively in Examples 1.3.1 and 1.3.2. Indeed $\Omega_1 = \Gamma$ and $\Omega_2 = \Upsilon$. Thus, in order to prove the equivalence, we start by presenting two ECO operators constructing the classes of Dyck and Schröder paths according to the specializations $\alpha = 1$ and $\alpha = 2$ of Ω'_α:

$$\Omega'_1 = \begin{cases} (1) \\ (1) \rightsquigarrow (2) \\ (2k) \rightsquigarrow (1)^k(4)(6)\dots(2(k-1))(2k)(2(k+1)) \end{cases}$$

$$\Omega'_2 = \begin{cases} (2) \\ (2) \rightsquigarrow (2)(4) \\ (4k) \rightsquigarrow (2)^{2k}(4)(8)^2(12)^2\dots(4(k-1))^2(4k)^2(4(k+1)). \end{cases}$$

2.2.1 A new ECO construction for Dyck paths.

Let D be a Dyck path, it factors uniquely in blocks of elevated Dyck paths

$$D = D_1 D_2 \dots D_m.$$

D is said of *even type* (respectively *odd type*) if $m = 2j$ for some j (resp. $m = 2j+1$). Let $\mathbf{P}^0(D)$ be the set of points of the last descent $\ell_d(D)$ of D, excepting the point at level 0. The set of Dyck paths having semi-length n is denoted by $\mathcal{D}_n$, and the operator

$$\vartheta_{\mathcal{D}} : \mathcal{D}_n \longrightarrow 2^{\mathcal{D}_{n+1}}$$

is defined as follows:

D1. If D is of even type, then $\vartheta_{\mathcal{D}}(D)$ contains a single Dyck path, obtained by gluing a peak of height 1 at the end of D (see Figure 2.1(D1)). This corresponds to the production

$$(1) \rightsquigarrow (2).$$

(Recall that the label is the number of objects generated at next level.)

D2. If D is of odd type, then $\vartheta_{\mathcal{D}}(D)$ is the set of Dyck paths obtained from D by performing on any $A \in \mathbf{P}^0(D)$ one of the following actions:

a) insert a peak;

b) let A' be the leftmost point such that $A'A$ is a Dyck path; remove the sub-path $A'A$ from D, elevate it by 1, and glue it at the end of D (see Figure 2.1(D2)).

For a Dyck path with k active sites, the number of objects produced is $2k$, and more precisely these actions of $\vartheta_{\mathcal{D}}$ correspond to the production

$$(2k) \rightsquigarrow (1)^k(4)(6)\ldots(2(k-1))(2k)(2(k+1)).$$

As one can check, this is an ECO construction of Dyck paths. Moreover, it yields the succession rule Ω_1', thus proving its equivalence with Ω_1.

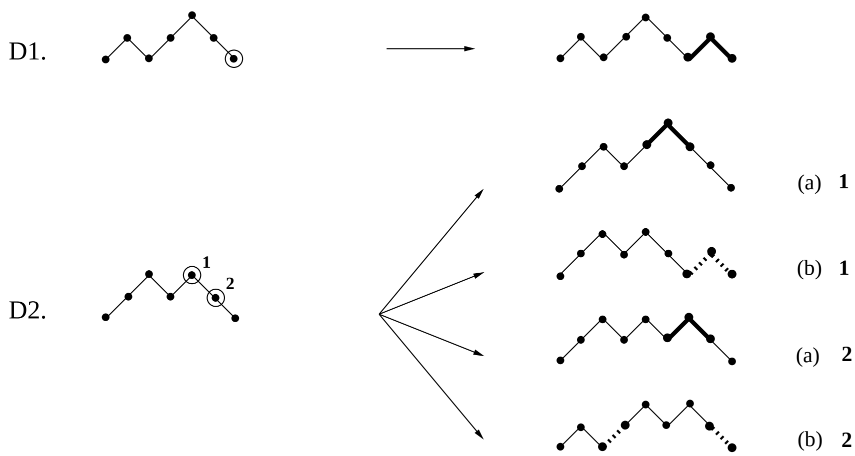

Figure 2.1: The construction for Dyck paths according to the rule Ω_1'.

2.2.2 Extension to Schröder and colored Schröder paths.

We give now a similar construction for Schröder paths. Each Schröder path S factors uniquely,

$$S = S_1 S_2 \ldots S_m,$$

where $S_i, 1 \leq i \leq m$, is either elevated or a horizontal step on the x-axis. The path S is said of *even type* (respectively *odd type*) if the number of elevated factors following the rightmost horizontal step on the x-axis is even (resp. odd). The *last descent* $\ell_d(S)$ of S is the last run of fall steps, and $\mathbf{P}^0(S)$ is the set of its points, excepting the last point.

The set of Schröder paths having semi-length n is denoted $\mathcal{S}_n$, and the operator

$$\vartheta_{\mathcal{S}} : \mathcal{S}_n \longrightarrow 2^{\mathcal{S}_{n+1}}$$

is defined by the following rules:

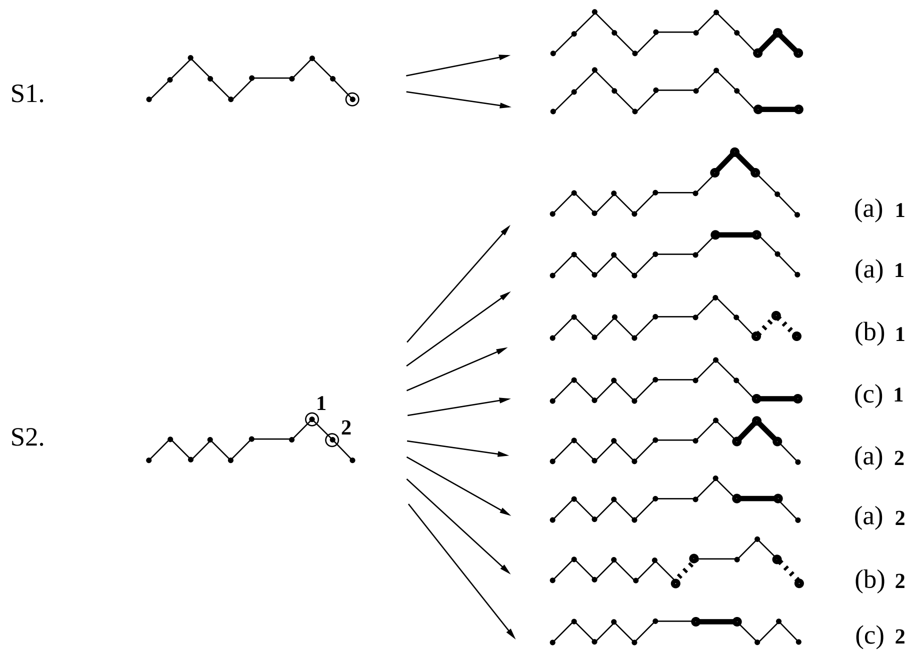

Figure 2.2: The construction for Schröder paths corresponding to the rule Ω_2'.

S1. If S is of even type, then $\vartheta_{\mathcal{S}}(S)$ contains two Schröder paths, obtained respectively by gluing at the end of S, either a peak of height 1, resulting in an odd type path, or a horizontal step, resulting in an even type path (Figure 2.2(S1)). This corresponds to the production $(2) \rightsquigarrow (2)(4)$.

S2. If S is of odd type, then $\vartheta_{\mathcal{S}}(S)$ is the set of paths obtained by performing one of the following actions on any point $A \in \mathbf{P}^0(S)$ (Figure 2.2(S2)):

a) insert a peak of height 1 or a horizontal step;

b) let A' be the leftmost point such that $A'A$ is a Schröder path. Then cut $A'A$, elevate it by 1, and glue it at the end of S;

c) let A'' be the first left point such that $A''A$ is a Schröder path; if $A''A$ is not empty, then replace it by a horizontal step and glue $A''A$ at the end of S; if $A''A$ is empty then glue a horizontal step at the end of S. In this way we obtain an even type path.

With $k-1$ the number of active sites, this corresponds to the production

$$(4k) \rightsquigarrow (2)^{2k}(4)(8)^2(12)^2 \dots (4(k-1))^2(4k)^2(4(k+1)).$$

One can check that this operator is indeed an ECO operator for Schröder paths and this proves the equivalence of Ω_2' and Ω_2.

The previous construction for Schröder paths, can be easily extended to Schröder α-colored paths by using α-colored horizontal steps. It leads to the succession rule $\Omega_{\alpha+1}'$, with $\alpha \geq 2$. For instance, when horizontal steps of two colors are used, we obtain Schröder bi-colored paths associated to the succession rule Ω_3'.

Moreover, if we use α-colored horizontal steps in the classic ECO construction for Schröder paths we obtain α-colored Schröder paths to which the rule $\Omega_{\alpha+1}$, $\alpha \geq 2$, is associated. So we have proved the equivalence between Ω_α and Ω'_α in a combinatorial way.

2.2.3 Transport of the new ECO constructions on trees

In this subsection we transport the operators $\vartheta_{\mathcal{D}}$ and $\vartheta_{\mathcal{S}}$ along a bijection. Then we provide a description of the new operators that is independent from the bijection. Let $\mathcal{B}$ and $\mathcal{T}$ respectively the classes of complete binary trees and Schröder trees (see Chapter 1). The nodes of a plane tree T can be totally ordered by means of the prefix traversal, and indexed increasingly by the integers, so that, given two nodes x_i and x_j,

$$x_i < x_j \iff i < j.$$

Accordingly, the maximum of two nodes is defined by

$$\max(x_i, x_j) = x_j \iff i < j.$$

Also, the total order allows to define notions like *first, last, successor, predecessor, etc.*, consequently, for every node p of T, we denote by (see Figures 2.4 and 2.6):

- $\ell_i(T), \ell_l(T), \ell_s(T)$ the last, respectively, internal node (i.e. not a leaf), leaf, internal sibling;
- $f(p)$ the set of leaves following p;
- father(p) the father of p;
- succ(p) the successor of p;

A common abuse of notation identifies a tree with the name of its root, and, consequently subtrees as nodes. The total order extends to the the class $\mathcal{F}$ of *forests*, whose objects are lists of trees, in the obvious way, making all the above definitions relevant for forests as well.

For convenience we denote the tree consisting of a single point by "$\bullet$", and define the "tree" and "raise" constructors

$$\text{tree}, \text{raise} : \mathcal{F} \longrightarrow \mathcal{T}$$

respectively, by

$$\text{tree}(T_1, T_2, \dots, T_m) = (\bullet, T_1, T_2, \dots, T_m),$$

and (see Figure 2.3),

$$\text{raise}(T_1, T_2, \dots, T_m) = \text{tree}(T_1, T_2, \dots, T_m, \bullet).$$

We use the short hand notation subs(T_1, T_2) to indicate the substitution of a subtree T_1 in place of a subtree T_2 ($T_2 \leftarrow T_1$). Moreover, we say that T is of *even type* (resp. *odd type*) if the length of its rightmost branch is even (resp. odd).

From now on, we consider the total order on two subclasses of plane trees, namely, the class $\mathcal{B}$ of complete binary trees and the class $\mathcal{T}$ of Schröder trees. The parameter p considered on these two classes of combinatorial objects is the number of leaves.

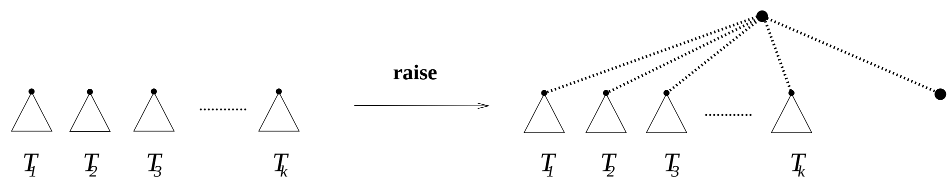

Figure 2.3: The raise constructor.

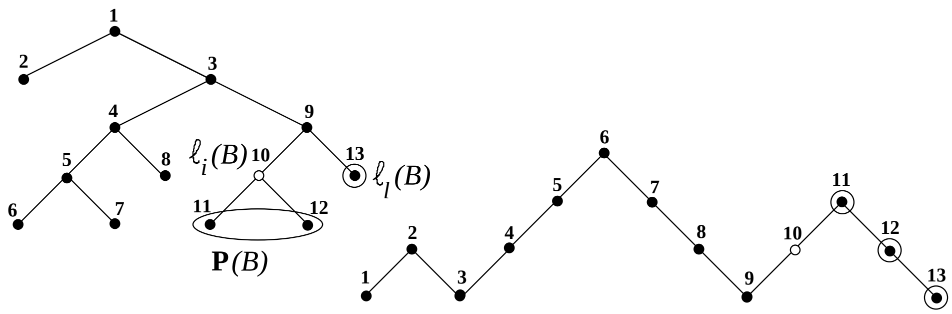

Figure 2.4: A complete binary tree B in $\mathcal{B}_7$, and the corresponding Dyck path.

There is a well-known bijection (see Figure 2.4) between Dyck paths and complete binary trees,

$$\Psi : \mathcal{D} \longrightarrow \mathcal{B}$$

(for instance, see the book from Stanley [94]). Let $\mathbf{x}$ and $\overline{\mathbf{x}}$ denote respectively rise and fall steps. A recursive definition of Ψ is then:

- the empty Dyck path corresponds to a leaf.
- a Dyck path of the form $\mathbf{x}D\overline{\mathbf{x}}D'$ with D and D' in $\mathcal{D}$ corresponds to the tree with a root, and left and right subtrees $\Psi(D)$ and $\Psi(D')$ respectively.

For $D \in \mathcal{D}$ and $B = \Psi(D)$, define

$$\mathbf{P}(B) = f(\ell_i(B)) \setminus \{\ell_l(B)\},$$

and observe that the number of elevated Dyck paths in D corresponds to the length of the right branch of B. Moreover, we have the underlying set bijection on nodes

$$\begin{aligned} f(\ell_i(B)) &= \Psi(\ell_d(D)); \\ \mathbf{P}(B) &= \Psi(\mathbf{P}^0(D)). \end{aligned}$$

These observations lead to an almost direct translation of the operator $\vartheta_{\mathcal{D}}$. Indeed, let $\mathcal{B}_n$ be the set of binary trees having n leaves, and let $B \in \mathcal{B}_n$, then the operator

$$\vartheta_{\mathcal{B}} : \mathcal{B}_n \longrightarrow 2^{\mathcal{B}_{n+1}}$$

is defined as follows (see Figure 2.5):

B1. if B is of even type then add two sons to $\ell_l(B)$, i.e. $\vartheta_{\mathcal{B}}(B) = \text{subs}(\text{raise}(\bullet), \ell_l(B))$;

B2. if B is of odd type then $\vartheta_{\mathcal{B}}(B)$ is the set of complete binary trees obtained by performing on any leaf $A \in \mathbf{P}(B)$ one of the following actions:

a) subs(raise($\bullet$), A);
b) let A' be the largest complete binary subtree of B such that $A = \ell_l(A')$; then, do subs(raise(A'), $\ell_l(B)$) and subs($\bullet$, A').

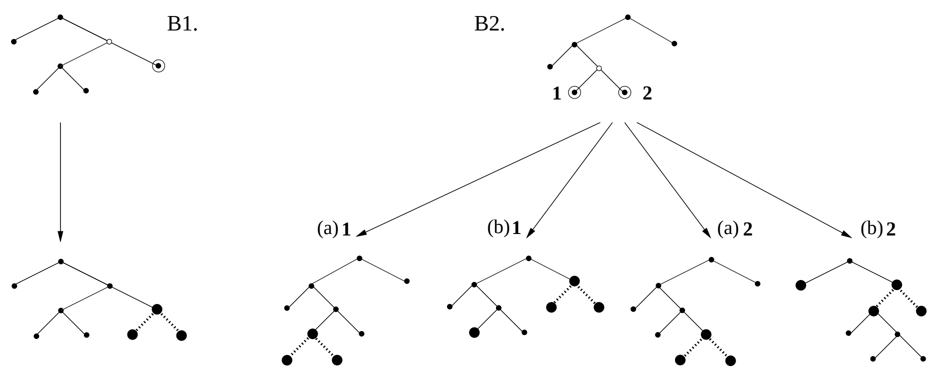

Figure 2.5: The construction for complete binary trees.

Clearly, $\vartheta_{\mathcal{D}}$ and $\vartheta_{\mathcal{B}}$ share the same succession rule Ω_1'.

We now turn to the case of Schröder trees. Let $\mathcal{S}'$ be the class of Schröder paths without horizontal steps at level 0, and let $\vartheta_{\mathcal{S}'}$ be the restriction of $\vartheta_{\mathcal{S}}$ to $\mathcal{S}'$. That is

$$\vartheta_{\mathcal{S}'}(\mathcal{S}_n') = \vartheta_{\mathcal{S}}(\mathcal{S}_n) \cap \mathcal{S}_{n+1}' \ , \ \forall n \geq 1.$$

As for Dyck paths, we show how to transport the operator $\vartheta_{\mathcal{S}'}$ along the bijection (Penaud et al. [77])

$$\Psi' : \mathcal{S}' \longrightarrow \mathcal{T}.$$

This bijection (see Figure 2.6) provides a simple interpretation of the required parameters. Indeed, a rise (resp. fall) step of S corresponds to a leftmost (resp. rightmost) sibling of T, and the horizontal steps of S correspond to the internal siblings of T, that is, those siblings strictly between the leftmost one and the rightmost one. The last run of fall steps $\ell_d(S)$ corresponds to, either the leaves following the last internal node $\ell_i(T)$, or, the last internal sibling $\ell_s(T)$ and its successors, whichever occurs the last. Therefore, define

$$z = \max(\text{succ}(\ell_i(T)), \ell_s(T)),$$

($z = 14$ in Figure 2.6), and set

$$\mathbf{P}(T) = \Psi'(\mathbf{P}(S)) = \{z\} \cup f(z) \setminus \{\ell_l(T)\}.$$

Observe that this generalizes the corresponding definition in the class $\mathcal{B}$.

Let $\mathcal{T}_n$ be the set of Schröder trees having n-leaves. The operator

$$\vartheta_{\mathcal{T}} : \mathcal{T}_n \longrightarrow 2^{\mathcal{T}_{n+1}}$$

is defined as follows (see Figure 2.7):

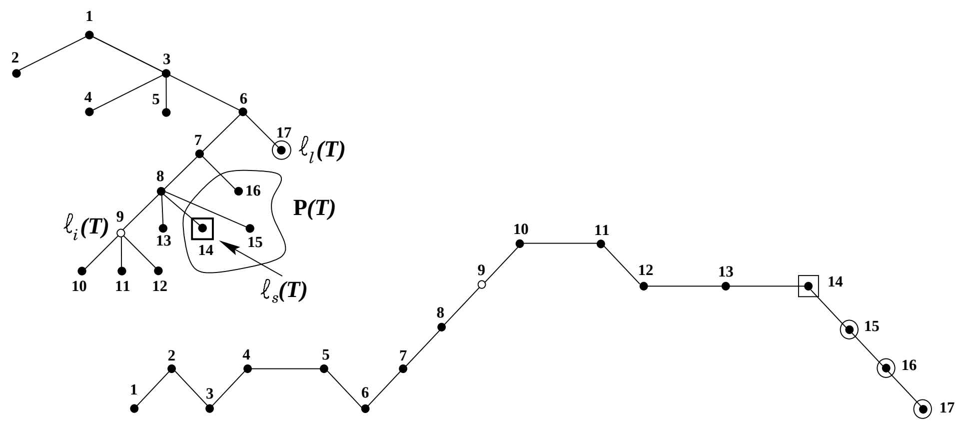

Figure 2.6: A Schröder tree and its corresponding path.

ST1. If T is of even type, then $\vartheta_{\mathcal{T}}(T) = \text{subs}(\text{raise}(\bullet), \ell_l(T))$ (see Figure 2.7(ST1)).

ST2. If T is of odd type, then $\vartheta_{\mathcal{T}}(T)$ is obtained by performing on any point $A \in \mathbf{P}(T)$ one of the the following actions (see Figure 2.7(ST2)):

a) $\text{subs}(\text{raise}(\bullet), A)$, or add a left brother to $succ(A)$;

b) let A' be the largest Schröder sub-forest of T, such that $A = \ell_l(A')$; then, do $\text{subs}(\text{raise}(A'), \ell_l(T))$ and $\text{subs}(\bullet, A')$;

c) if $A \neq z$, let A'' be the tree having father(A) for root; then, do $\text{subs}(A'', \ell_l(T))$, $\text{subs}(\bullet, A'')$, and add a right brother to A'' .

A careful comparison between the constructions associated to the operators $\vartheta_{\mathcal{T}}$ and $\vartheta_{\mathcal{S}}$ shows some differences. Indeed, since we are concerned with the restriction $\vartheta_{\mathcal{S}'}$, it was necessary to avoid the cases that generate a Schröder path with a horizontal step at level 0. This occurs precisely when the node z is treated. To conclude, we believe that the problem of characterizing the natural bijections between objects (allowing the translation of ECO-operators) is a problem that is worth investigating.

2.3 An infinite set of rules for Ballot numbers

In this section we give an infinite set of rules defining Catalan numbers. This set, denoted by $\Gamma^1_{\alpha,\beta}$, depends on two parameters α and β. The rule $\Gamma^1_{2,0}$, obtained for $\alpha = 2$ and $\beta = 0$, is equal to the rule Ω'_1 introduced in Section 2.2. Therefore, generalizing the ECO construction encoded by Ω'_1 (see Subsection 2.2.1), we provide a bijective proof that $\Gamma^1_{\alpha,\beta}$ is a set of equivalent rules defining Catalan numbers. Finally, we generalize our result to the set of Ballot numbers.

For $k, n \in \mathbb{N}$, let $a_{n,k}$, be the set of Ballot numbers, defined by the recurrence,

$$a_{1,1} = 1, \qquad a_{1,k} = 0, \text{ for } k \geq 2$$

$$a_{n+1,1} = \textstyle\sum_{j\geq 1} a_{n,j}, \quad a_{n+1,k} = \sum_{j\geq k-1} a_{n,j}, \text{ for } k \geq 2.$$

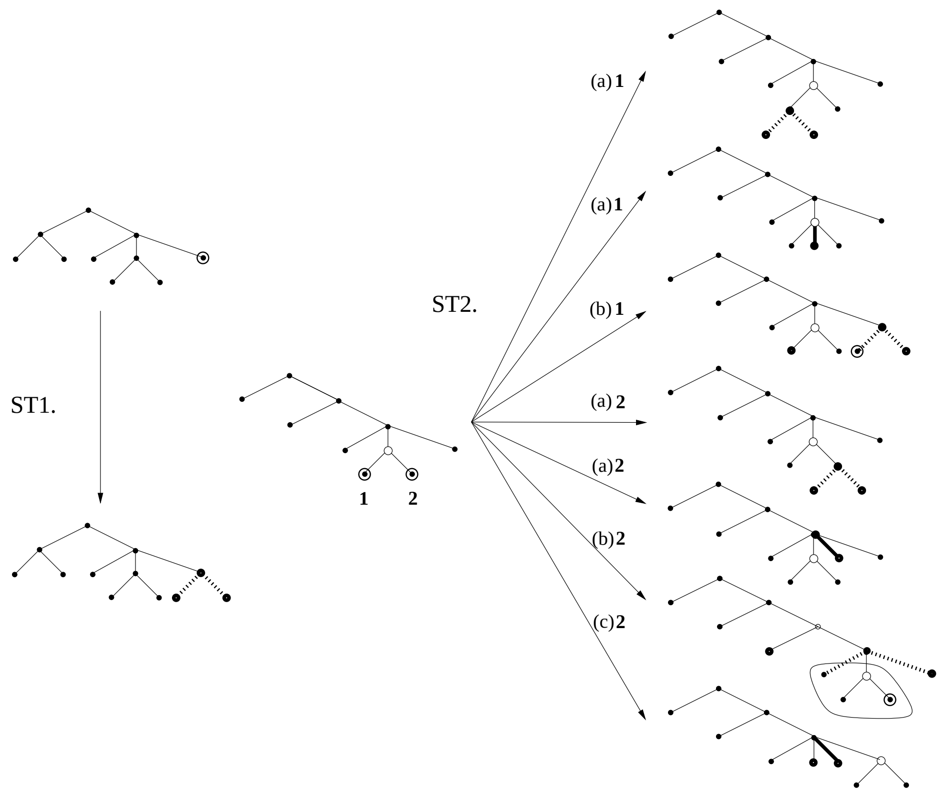

Figure 2.7: The construction for Schröder trees.

They can conveniently be displayed in a triangular array, sometimes known as the *Catalan triangle* shown in Table 2.1. For any positive integer h, the classic rule defining the sequence in the h-th column is the following:

$$\Gamma^h = \begin{cases} (h) \\ (k) \rightsquigarrow (2)(3)\dots(k)(k+1). \end{cases} \tag{2.4}$$

Please notice that for $h = 1$, we have the classic rule defining the Catalan numbers.

Let $h, \alpha \in \mathbb{N}^+$, and $\beta \in \mathbb{N}$. We first define the following set of rules depending on h, α, and β:

$$\Gamma^h_{\alpha,\beta} = \begin{cases} (h) \\ (1) \rightsquigarrow (2) \\ (2) \rightsquigarrow (2)(3) \\ \dots \qquad \dots \\ (\alpha+\beta-1) \rightsquigarrow (2)(3)\dots(\alpha+\beta) \\ (\alpha k+\beta) \rightsquigarrow (1)^k \dots (\alpha-1)^k(\alpha+1)\dots(\alpha+\beta) \\ \qquad\qquad (2\alpha+\beta)\dots((k+1)\alpha+\beta), k \geq 1. \end{cases} \tag{2.5}$$

If $h = 1$ we have

	1	2	3	4	5	6	7	...
1	1							...
2	1	1						...
3	2	2	1					...
4	5	5	3	1				...
5	14	14	9	4	1			...
6	42	42	28	14	5	1		...
7	132	132	90	48	20	6	1	...
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

Table 2.1: The Catalan triangle.

$$\Gamma^1_{\alpha,\beta} = \begin{cases} (1) \\ (1) \rightsquigarrow (2) \\ (2) \rightsquigarrow (2)(3) \\ \dots \quad\quad \dots \\ (\alpha+\beta-1) \rightsquigarrow (2)(3)\dots(\alpha+\beta) \\ (\alpha k+\beta) \rightsquigarrow (1)^k \dots (\alpha-1)^k(\alpha+1)\dots(\alpha+\beta) \\ \quad\quad\quad\quad (2\alpha+\beta)\dots((k+1)\alpha+\beta),\ k \geq 1. \end{cases} \tag{2.6}$$

In Subsection 2.3.2 we prove, bijectively, the equivalence of Γ^1 and $\Gamma^1_{\alpha,\beta}$, for any $\alpha \in \mathbb{N}^+$ and $\beta \in \mathbb{N}$. Therefore $\Gamma^1_{\alpha,\beta}$ can be viewed as a generalization of Γ^1 where the labels have been linearly combined according to the parameters α and β. Moreover we prove that, for $h \leq \alpha+\beta$, the rule $\Gamma^h_{\alpha,\beta}$ is equivalent to the rule Γ^h. As a consequence, we obtain that $\Gamma^h_{\alpha,\beta}$ defines the numbers $\{a_{n,h} : n \geq 0\}$, for any α and β such that $h \leq \alpha+\beta$. Thus we have an infinite set of succession rules defining Ballot numbers. For instance, the following rules define Catalan numbers:

$$\Gamma^1_{2,0} = \begin{cases} (1) \\ (1) \rightsquigarrow (2) \\ (2k) \rightsquigarrow (1)^k(4)(6)\dots(2k)(2k+2); \end{cases} \tag{2.7}$$

$$\Gamma^1_{3,1} = \begin{cases} (1) \\ (1) \rightsquigarrow (2) \\ (2) \rightsquigarrow (2)(3) \\ (3) \rightsquigarrow (2)(3)(4) \\ (3k+1) \rightsquigarrow (1)^k(2)^k(4)(7)\dots(3k+1)(3k+4). \end{cases} \tag{2.8}$$

2.3.1 Some further notations on Dyck paths.

Recall that $\mathbf{x}$ and $\overline{\mathbf{x}}$ denote respectively a rise and a fall step. The set $\mathcal{D}$ of Dyck paths can be seen as the subset of $\Sigma^* = \{\mathbf{x}, \overline{\mathbf{x}}\}^*$ generated by the grammar

$$D := \epsilon + \mathbf{x}D\overline{\mathbf{x}}D, \tag{2.9}$$

and we refer to paths as words, in which the notions of prefix, suffix have the usual meaning. The height of a point is denoted $h(P)$. An elevated Dyck path D is represented as $D = \mathbf{x}D'\overline{\mathbf{x}}$ with $D' \in \mathcal{D}$, and we denote the stripping operation by

$$D' = \mathrm{Top}(D).$$

Given two points P', P of a Dyck path D, the *factor* starting at P' and ending at P of the corresponding Dyck word is denoted $D[P'_x, P_x]$. By convention, $D[i,j] = \epsilon$ if $i \geq j$. The *insertion* of a word w in D at position i is defined by

$$\mathrm{insert}(D, w, i) = D[0, i] \cdot w \cdot D[i, 2n]\ .$$

The last sequence of fall steps $\ell_d(D)$ satisfies $\ell_d(D) = \overline{\mathbf{x}}^k$ for some $k \geq 1$, and $\mathbf{P}(D)$ is the set of its points. Finally, $|D|$ denotes the length of the word (number of steps of the path). From the grammar (2.9), one can easily deduce the properties summarized in the next statement.

Proposition 2.3.1. *Every non empty Dyck $D = u\mathbf{x}\overline{\mathbf{x}}^k, k \geq 1$, satisfies the conditions*
(a) $\exists m,\ D = D_1 D_2 \dots D_m$ *where* $\forall i, D_i$ *is elevated;*
(b) $D'' = u\overline{\mathbf{x}}^{k-1} \in \mathcal{D}$ *is such that* $D = \mathrm{insert}(D'', \mathbf{x}\overline{\mathbf{x}}, |D''| - (k-1))$*;*

2.3.2 An ECO operator for $\mathcal{D}$

We define an ECO operator $\vartheta : \mathcal{D}_n \to 2^{\mathcal{D}}_{n+1}$ generating Dyck paths according to their semi-length. The rule associated with ϑ is $\Gamma^1_{\alpha,\beta}$. The operator is defined separately on three subclasses $\mathcal{D}^0$, $\mathcal{D}^1$, and $\mathcal{D}^2$, such that $\{\mathcal{D}^0, \mathcal{D}^1, \mathcal{D}^2\}$ is a partition of $\mathcal{D}$. The description is inductive, we give directly the class to which belong the paths produced at one step. This is possible because the class of a path depends only on the way it is produced by ϑ.

The empty path ε is put in the class D^0, and the construction for each class is as follows.

[$D \in \mathcal{D}^0$] /*See Figure 2.8 for an example with $\alpha = 3, \beta = 2$, and $|\mathbf{P}(D)| = 4$.*/
In this class $\vartheta(D)$ is the set of paths produced by the classic construction:
• for any point $P \in \mathbf{P}(D)$ do
$D_{h(P)} \leftarrow \mathrm{insert}(D, \mathbf{x}\overline{\mathbf{x}}, P_x)$;
if $h(P) < \alpha + \beta - 2$ then $D_{h(P)} \in \mathcal{D}^0$ else $D_{h(P)} \in \mathcal{D}^1$. /*classifying*/

Remark. For each path in the class $\mathcal{D}^0$ we will have

$$D \in \mathcal{D}^0 \Longrightarrow h(\ell_d(D)) < \alpha + \beta - 1.$$

[$D \in \mathcal{D}^1$] /* See Figure 2.9 for an example with $\alpha = 3, \beta = 2$.*/
In this case there are two types of productions:
• for any point $P \in \mathbf{P}(D)$ such that $h(P) \geq \alpha - 1$ do

$D_{h(P)} \leftarrow \text{insert}(D, \mathbf{x\overline{x}}, P_x)$;
if $h(P) \geq \alpha + \beta - 2$ then $D_{h(P)} \in \mathcal{D}^1$ else $D_{h(P)} \in \mathcal{D}^0$.

- for any pair $P, Q \in \mathbf{P}(D)$ such that $h(P) \geq \alpha + \beta - 1$ and

$$0 \leq h(Q) \leq \alpha - 2 \text{ do}$$

let P' be the leftmost point of D such that $P'P \in \mathcal{D}$;
/* then $D = uD[P'_x, P_x]v$ with $uv \in \mathcal{D}$; */
$D_{h(P),h(Q)} \leftarrow \text{insert}(uv, \mathbf{x}D[P'_x, P_x]\overline{\mathbf{x}}, Q_x)$; $D_{h(P),h(Q)} \in \mathcal{D}^2$;
$\text{rank}(D_{h(P),h(Q)}) \leftarrow h(Q)$; /* ranking*/

Remark. Observe that the rank is defined only for paths in $\mathcal{D}^2$, and is equal to the height of the insertion point.

$\left[D \in \mathcal{D}^2\right]$ /* See Figure 2.10 for an example with $\alpha = 3, \beta = 0$. These paths have a rank.*/
In this case the classic construction applies again, but only to lower points:

- for any point $P \in \mathbf{P}(D)$ such that $h(P) \leq \text{rank}(D)$ do

$D_{h(P)} \leftarrow \text{insert}(D, \mathbf{x\overline{x}}, P_x)$;
if $\text{rank}(D) < \alpha + \beta - 2$ then $D_{h(P)} \in \mathcal{D}^0$ else $D_{h(P)} \in \mathcal{D}^1$.

In order to prove that the operator ϑ is well defined it is convenient to have a direct way to know the class of a Dyck path and, for paths of $\mathcal{D}^2$, to know their rank. In order to do this we introduce an evaluation $Val : \mathcal{D} \rightarrow \{0, 1, 2\} \times \mathbb{N}$. The first component stands for the class of the path and the second, which is used only for paths of $\mathcal{D}^2$, gives the corresponding rank.

An evaluation for Dyck paths

For $l, i, n \in \mathbb{N}$, a *Dyck path at level i* is the image of an ordinary Dyck path (at level 0) under the translation $(0, 0) \mapsto (l, i)$, running from (l, i) to $(l + 2n, i)$ and remaining weakly above the line $y = i$.

Let $\mathcal{D}(i)$ be the set of Dyck paths at level i, and $\mathbb{D} = \{\bigcup \mathcal{D}(i) : i \in \mathbb{N}\}$. By Proposition 2.3.1 (a), each path $D(i) \in \mathcal{D}(i)$ admits a unique decomposition in terms of elevated paths at level i,

$$D(i) = D(i, 1)D(i, 2) \ldots D(i, m),$$

where $D(i, j)$ denotes the j-th *component*, and $\#(D(i)) = m$ denotes the number of components. In this decomposition, $D''(i)$ denotes the rightmost factor having height less than $\alpha + \beta - i - 1$, while $D'(i)$ is the factor on the right of $D''(i)$. The evaluation is defined by $\text{Val}(D) = \overline{\text{Val}}(D, 0)$, where $\overline{\text{Val}}(D, i)$ is defined by the following algorithm, where the variables are defined by the notations above.

Algorithm $\overline{\text{Val}}(D, i)$;
$D - D''$ is the path obtained by removing D'' from D; $\overline{\text{Val}}(D, i)[1]$ refers to the first component of the evaluation.

if $D = \epsilon$ **then** return $(0, i)$

elseif $\#(D') = 0$ **then** there are three cases

if $\#(D'') > 1$ **then** return $(0, i)$

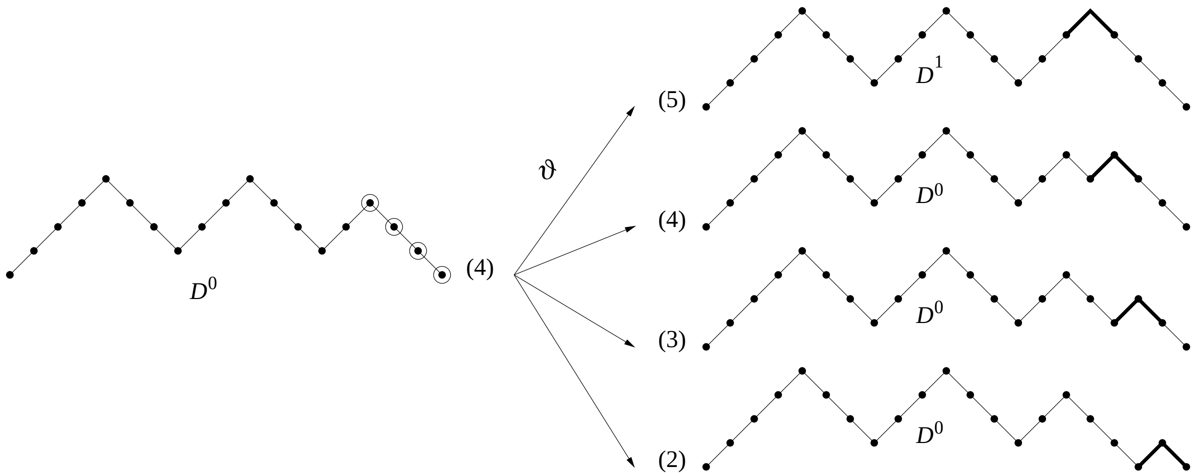

Figure 2.8: The operator ϑ applied to a path in $\mathcal{D}^0$. The belonging class is represented at the bottom of each path.

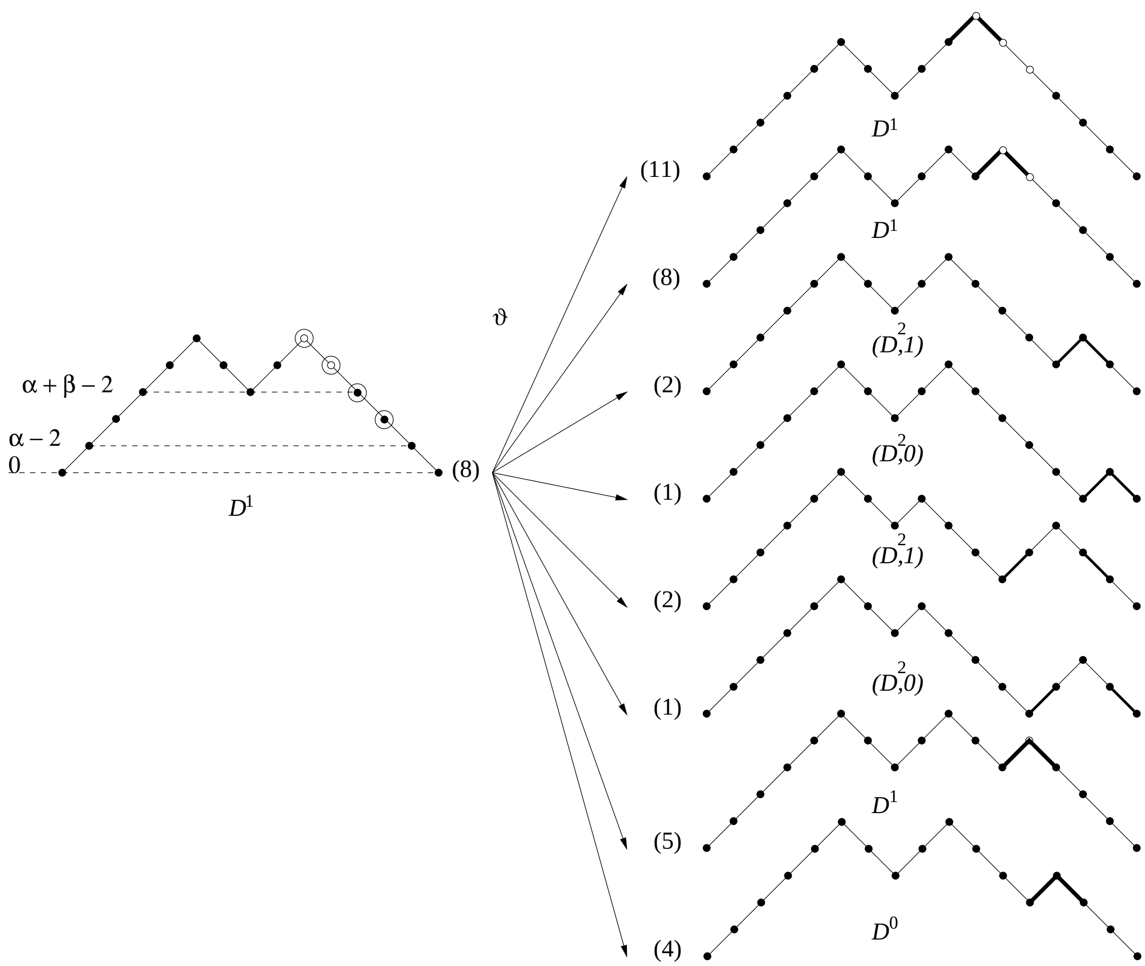

Figure 2.9: The operator ϑ applied to a path in $\mathcal{D}^1$.

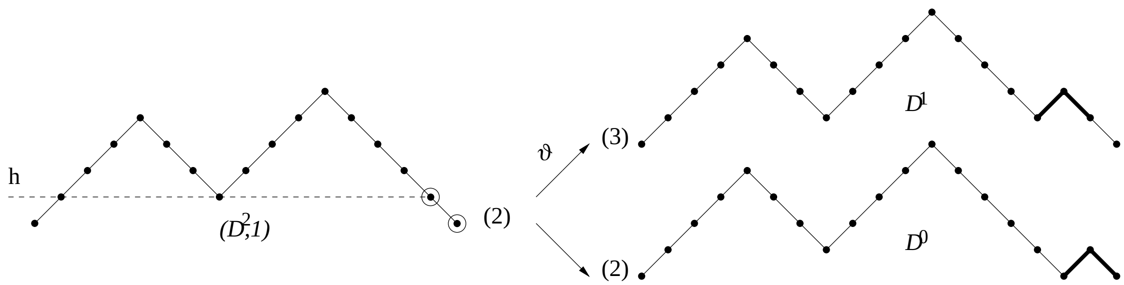

Figure 2.10: The operator ϑ applied to a path in $\mathcal{D}^2$.

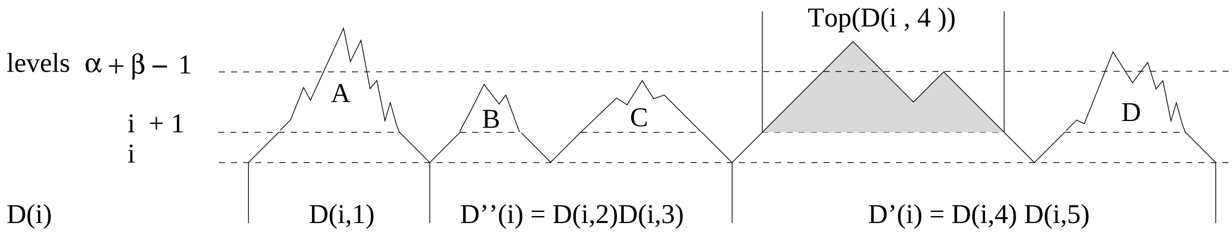

Figure 2.11: Decomposition of a Dyck path at level i and notations.

> **elseif** $\overline{\mathrm{Val}}(D - D'', i)[1] = 1$ **then** return $(2, i)$
>
> **else** return $(0, i)$

else let $D' = \prod_{j=1}^{m} D'(j)$; we have four cases:

> **if** $\overline{\mathrm{Val}}(\prod_{j=1}^{m-1} D'(j), i)[1] = 1$ **then** return $(2, i)$
>
> **elseif** $i < \alpha - 2$ **then** return $\overline{\mathrm{Val}}(\mathrm{Top}(D'_m), i + 1))$
>
> **elseif** $|\mathbf{P}(D'_m)| < \alpha + \beta - i$ **then** return $(0, i)$
>
> **else** return $(1, i)$.

end $\overline{\mathrm{Val}}(D, i)$**.**

An example of a computation of the evaluation of a Dyck path We provide a computation of the evaluation applied to the path in Figure 2.12 in the case where $\alpha = 3$ and $\beta = 2$. Let $D(0, 1)$, $D(0, 2)$, and $D(0, 3)$, for simplicity $D(1)$, $D(2)$, and $D(3)$, be the three components of the path D. In order to give a clear explanation of the calculus of $\overline{\mathrm{Val}}(D, 0)$ we first compute the values $\overline{\mathrm{Val}}(D(1), 0)$ and $\overline{\mathrm{Val}}(D(2), 0)$. The calculus of these values is necessary to obtain the final result.
$\overline{\mathrm{Val}}(\mathbf{D}(\mathbf{1}), \mathbf{0})$: this returns the value $\overline{\mathrm{Val}}(Top(D(1)), 1)$, since $\#(D'(1)) \neq 0$, $\#(D(1)) = 1$, and $i = 0 < 3 - 2$.

> $\overline{\mathrm{Val}}(\mathbf{Top}(\mathbf{D}(\mathbf{1})), \mathbf{1})$: this implies the calculus of $\overline{\mathrm{Val}}(Top(D(1))(1), 1)$.
>
> > $\overline{\mathrm{Val}}(\mathbf{Top}(\mathbf{D}(\mathbf{1}))(\mathbf{1}), \mathbf{1})$: this returns the value $(\mathbf{1}, \mathbf{1})$, since
> > #((Top(D(1))(1))=1, $i = 1 = 3 - 2$, and $|\mathbf{P}((Top(D(1))(1))| = = 4 = 3 + 2 - 1$.

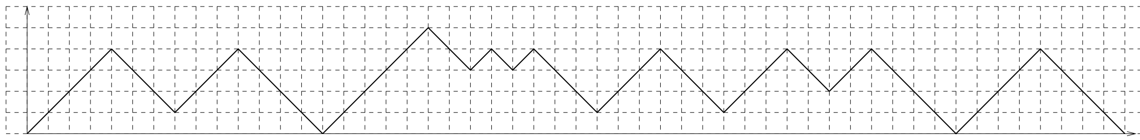

Figure 2.12: A Dyck path D with $\mathrm{Val}(D) = (2, 0)$.

Then $\overline{\mathrm{Val}}(Top(D(1)), 1)$ returns $(\mathbf{2}, \mathbf{1})$.

Consequently $\overline{\mathrm{Val}}(D(1), 0)$ returns $(\mathbf{2}, \mathbf{1})$.

$\overline{\mathrm{Val}}(\mathbf{D(2)}, \mathbf{0})$: this is equal to compute $\overline{\mathrm{Val}}(Top(D(2)), 1)$, since $\#(D'(2)) \neq 0$, $\#(D(2)) = 1$, and $i = 0 < 3 - 2$.

$\overline{\mathrm{Val}}(\mathbf{Top}(\mathbf{D}(\mathbf{2})), \mathbf{1})$: this implies the calculus of

$$\overline{\mathrm{Val}}(Top(D(2))(1)\ Top(D(2))(2), 1).$$

$\overline{\mathrm{Val}}(\mathbf{Top}(\mathbf{D}(\mathbf{2}))(\mathbf{1})\ \mathbf{Top}(\mathbf{D}(\mathbf{2}))(\mathbf{2}), \mathbf{1})$: this implies the calculus of $\overline{\mathrm{Val}}(Top(D(2))(1), 1)$.

$\overline{\mathrm{Val}}(\mathbf{Top}(\mathbf{D}(\mathbf{2}))(\mathbf{1}), \mathbf{1})$: this returns the value $(\mathbf{1}, \mathbf{1})$, since $\#((\mathrm{Top}(\mathrm{D}(2))(1))$=1, $i = 1 = 3-2$, and $|\mathbf{P}((Top(D(2))(1))| == 4 = 3+2-1$.

Then $\overline{\mathrm{Val}}(Top(D(2))(1)\ Top(D(2))(2), 1)$ returns the value $(\mathbf{2}, \mathbf{1})$.

Then $\overline{\mathrm{Val}}(Top(D(2)), 1)$ returns $(\mathbf{1}, \mathbf{1})$.

Consequently $\overline{\mathrm{Val}}(D(2), 0)$ returns $(\mathbf{1}, \mathbf{1})$.

$\overline{\mathrm{Val}}(\mathbf{D}, \mathbf{0})$: this implies to compute $\overline{\mathrm{Val}}(D(1)D(2), 0)$, since $\#(D') \neq 0$, $\#(D) = 3$, and $i = 0 < 3 - 2$.

$\overline{\mathrm{Val}}(\mathbf{D(1)D(2)}, \mathbf{0})$: this implies to compute $\overline{\mathrm{Val}}(D(1), 0)$

$\overline{\mathrm{Val}}(\mathbf{D(1)}, \mathbf{0})$: it returns the value $(\mathbf{2}, \mathbf{1})$.

Then $\overline{\mathrm{Val}}(D(1)D(2), 0)$ returns the value $\overline{\mathrm{Val}}(D(2), 0)$, that is $(\mathbf{1}, \mathbf{1})$.

Consequently $\overline{\mathrm{Val}}(D, 0)$ returns $(\mathbf{2}, \mathbf{0})$.

Finally, we constructed the class of Dyck paths according to the rule $\Gamma^1_{\alpha,\beta}$. Thus, we can enunciate the following.

Proposition 2.3.2. *Let p be the semi-length of a Dyck path, then*

$$\Sigma = (\mathcal{D}, p, \vartheta, \Gamma^1_{\alpha,\beta})$$

is an ECO-system.

Proof. We need to prove that the described construction generates all the Dyck paths i) and that we have a partition ii). This is achieved for both conditions by induction, according to the inductive definition of ϑ.

i) For any $D' \in \mathcal{D}_{n+1}$ there exists $D \in \mathcal{D}_n$, such that $D' \in \vartheta(D)$:

- if $D' \in \mathcal{D}^0 \cup \mathcal{D}^1$, then $D' = u\mathbf{x}\overline{\mathbf{x}}^k$ and Proposition 2.3.1(b) identifies the last peak of $D^{'}$ to be removed, i.e.
$$D = u\overline{\mathbf{x}}^{k-1} \in \mathcal{D}_n \ ,$$
such that $D' = \text{insert}(D, \mathbf{x}\overline{\mathbf{x}}, |D| - (k-1))$.
- if $D' \in \mathcal{D}^2$, from the evaluation we have
$$\text{Val}(D') = (2, |v|), \quad \text{and} \quad D' = uD''v \ ,$$
where $D'' \neq \epsilon$, and $\#(D'') \geq 2$. Then, Proposition 2.3.1(a) provides the factorization
$$D'' = D''_1 \dots D''_{m-1} D''_m \ ,$$
where $D''_m = \mathbf{x}\text{Top}(D''_m)\overline{\mathbf{x}}$. Let P_x be the position of the last peak of D''_{m-1}. Then,
$$D = \text{insert}(uD''_1 \dots D''_{m-1}v, \text{Top}(D''_m), P_x) \ .$$

ii) Let D and $D^{'} \in \mathcal{D}_n$, then $\vartheta(D) \cap \vartheta(D^{'}) = \emptyset$; when D and D' are such that ϑ performs the insertion of $\mathbf{x}\overline{\mathbf{x}}$ in their last descent, the result follows from the fact that, for each $P \in \mathbf{P}(D)$ and for each $P' \in \mathbf{P}(D')$, we have
$$\text{insert}(D, \mathbf{x}\overline{\mathbf{x}}, P_x) = \text{insert}(D', \mathbf{x}\overline{\mathbf{x}}, P'_x) \Longrightarrow D = D'.$$
When $\vartheta(D), \vartheta(D') \in \mathcal{D}^2$, we have two cases.

- $\text{Val}(\vartheta(D))[2] \neq \text{Val}(\vartheta(D'))[2] \implies \vartheta(D) \neq \vartheta(D')$.
- $\text{Val}(\vartheta(D))[2] = \text{Val}(\vartheta(D'))[2]$; if $\vartheta(D) = \vartheta(D')$ then the construction i) above shows that $D = D'$.

$\square$

We are now in a position to state our main result.

Proposition 2.3.3. *Let $\alpha \in \mathbb{N}^+$ and $\beta \in \mathbb{N}$, then $\Gamma^1_{\alpha,\beta} \cong \Gamma^1$.*

Proof. Both rules describe the class of Dyck paths according to the same parameter. $\square$

Corollary 2.3.1. *Let $h, \alpha \in \mathbb{N}^+$, $\beta \in \mathbb{N}$ and $h \leq \alpha + \beta$. We have $\Gamma^h_{\alpha,\beta} \cong \Gamma^h$.*

Proof. It is a direct consequence of Proposition 2.3.3. Indeed, the rules $\Gamma^h_{\alpha,\beta}$ and Γ^h both enumerate the class of Dyck paths beginning with h rise steps. $\square$

2.4 Other examples.

We give now some other examples of infinite sets of equivalent succession rules. In practice, the productions of these succession rules can be distinguished into two different types:

- the set of productions that behave like the original rule Ω;
- the set of productions that are obtained by linearly combining the productions of Ω.

Example 2.4.1. *An infinite set of rules defining Motzkin numbers.*

The sequence of Motzkin numbers, $1, 1, 2, 4, 9, 21, 51, 127, \ldots$ (sequence A001006 in [92]) is defined by the classic rule

$$\Delta = \begin{cases} (1) \\ (k) \rightsquigarrow (1)(2)\ldots(k-1)(k+1). \end{cases}$$

Let $M(x)$ be the generating function of the rule Δ. Recall that $M(x)$ satisfies the relation $M(x) = 1 + xM(x) + x^2M(x)^2$. Let us moreover denote $M_j(x)$ the generating function of the rule with the same production as Δ and axiom (j). Then $M_j(x) = M(x)(1 + xM(x))^{j-1}$.

An infinite class of succession rules can be derived from Δ, still defining Motzkin numbers

$$\begin{cases} (1) \\ (1) \rightsquigarrow (2) \\ (2) \rightsquigarrow (1)(3) \\ (3) \rightsquigarrow (1)(2)(4) \\ \vdots \\ (\alpha+\beta) \rightsquigarrow (1)(2)\ldots(\alpha+\beta-1)(\alpha+\beta+1) \\ \quad (\alpha k+\beta+1) \quad \rightsquigarrow (1)^k(2)^k\ldots(\alpha-1)^k(1)(\alpha+1)(\alpha+2)\ldots(\alpha+\beta) \\ \qquad\qquad\qquad\qquad (\alpha+\beta+1)(2\alpha+\beta+1)\ldots((k-1)\alpha+\beta+1)((k+1)\alpha+\beta+1). \end{cases}$$

This rule can be re-written in a simpler way as

$$\Delta_{\alpha,\beta} = \begin{cases} (1) \\ (k) \rightsquigarrow (1)(2)\ldots(k-1)(k+1) \qquad k \leq \alpha+\beta \\ (\alpha k+\beta+1) \rightsquigarrow (1)^k(2)^k\ldots(\alpha-1)^k(1)(\alpha+1)(\alpha+2)\ldots(\alpha+\beta) \\ \qquad (\alpha+\beta+1)(2\alpha+\beta+1)\ldots((k-1)\alpha+\beta+1)((k+1)\alpha+\beta+1). \end{cases}$$

We prove that $\Delta_{\alpha,\beta}$ is equivalent to Δ by computing its generating function, $f_{\Delta_{\alpha,\beta}}$. Let $f_j(x)$ be the generating function of the rule having the same productions as $\Delta_{\alpha,\beta}$ and axiom (j), then $f_{\Delta_{\alpha,\beta}} = f_1(x)$. By descending in the generating tree of $\Delta_{\alpha,\beta}$ up to the first occurrence of the label $\alpha+\beta+1$, we obtain the following system of equations:

$$\begin{aligned}
&f_1(x) = 1 + xf_2(x)\\
&f_2(x) = 1 + xf_1(x) + xf_3(x)\\
&\vdots\\
&f_{\alpha+\beta}(x) = 1 + xf_1(x) + xf_2(x) + \ldots + xf_{\alpha+\beta-1}(x) + xf_{\alpha+\beta+1}(x)\\
&f_{\alpha+\beta+1}(x) = \quad M(x) + (M(x)-1)(f_1(x) + f_2(x) + f_3(x) + \ldots + f_{\alpha-1}(x)) +\\
&\qquad\qquad + xM(x)f_1(x) + xM(x)(f_{\alpha+1}(x) + f_{\alpha+2}(x) + \ldots + f_{\alpha+\beta}(x)),
\end{aligned} \tag{2.10}$$

where the last equation, for the generating function of the rule

$$\begin{cases} (\alpha+\beta+1) & \\ (\alpha k+\beta+1) & \rightsquigarrow (1)^k(2)^k \ldots (\alpha-1)^k(1)(\alpha+1)(\alpha+2)\ldots(\alpha+\beta) \\ & \quad (\alpha+\beta+1)(2\alpha+\beta+1)\ldots((k-1)\alpha+\beta+1)((k+1)\alpha+\beta+1), \end{cases}$$

is obtained by decomposing its generating tree. Indeed, descending in this generating tree using only labels of the form $(i\alpha+\beta+1)$, for $i \in \{1, 2, \ldots, k-1, k+1\}$, we obtain the same generating tree as for the rule Δ. In particular, the term $M(x)$ is the generating function of nodes reached by a path in the generating trees having all nodes of the form $(i\alpha+\beta+1)$, for $i \in \{1, 2, \ldots, k-1, k+1\}$. But a path in the generating tree can also reach a first label not of the form $(i\alpha+\beta+1)$. We consider the different kind of subtrees that can be attached there. The term $xM(x)f_1(x)$ is the generating function of subtrees having the first node labeled (1), and the term $xM(x)(f_{\alpha+1}(x) + f_{\alpha+2}(x) + \ldots + f_{\alpha+\beta}(x))$ is the generating function of the subtrees with first node labeled (i), for $i \in \{\alpha+1, \ldots, \alpha+\beta\}$. Finally the term $(M(x)-1)(f_1(x) + f_2(x) + f_3(x) + \ldots + f_{\alpha-1}(x))$ is the generating function of the subtrees with first node labeled $(i)^k$, for $i \in \{1, \ldots, \alpha-1\}$: this follows from the fact that a node with label $\alpha i + \beta$ produces i labels of the same form at the next level.

Now, we need to prove that $f_1(x) = M(x)$. Observe that it is sufficient to check that the last equation of (2.10) is satisfied upon setting $f_j = M_j$ for $j \in \{1, \ldots, \alpha+\beta+1\}$, because the other $\alpha+\beta$ equation of (2.10) are identical to the equations induced by Δ for the M_j, $j = 1, \ldots, \alpha+\beta$. This verification is easy using the explicit values of the $M_j(x)$ in terms of $M(x)$ and the relation satisfied by this series.

Example 2.4.2. *An infinite set of rules for the number of m-ary trees.*

It is common knowledge that for any $m \geq 2$, the number of m-ary trees having n nodes is

$$\frac{1}{(m-1)n+1}\binom{mn}{n}.$$

A succession rule defining this sequence is determined in by Barcucci et al. [5]:

$$\Omega^m = \begin{cases} (m) \\ (k) \rightsquigarrow (m)(m+1)\ldots(k+m-1) \end{cases}$$

Let $T(x)$ be the generating function of the rule Ω^m. The series $T(x)$ counts m-ary trees according to the number of non root nodes and satisfies $T(x) = (1+xT(x))^m$. Let us moreover denote $T_j(x)$ the generating function of the rule with the same production as Ω^m and axiom (j). Then $T_j(x) = (1 + xT(x))^j$.
A class of succession rules equivalent to Ω^m is given by

$$\Omega^m_{\alpha,\beta}\left\{\begin{array}{ll} (m) & \\ (k) \rightsquigarrow (m)(m+1)\dots(k+m-1) & k \notin \{\alpha t+\beta \mid t \in \mathbb{N}^+\} \\ \quad(\alpha k+\beta) \rightsquigarrow & (1)^k(2)^k\dots(\alpha-1)^k(\alpha+m-1)(\alpha+m)\dots(\alpha+\beta+m-2) \\ & (\alpha m+\beta)(\alpha(m+1)+\beta)\dots(\alpha(k+m-1)+\beta) \end{array}\right.$$

with $\alpha \geq 1$, $\beta \geq 0$, and $\alpha + \beta \geq m$. We remark that the specialization $m = 2$, again yields the set of rules (2.5) defining $\Gamma^2_{\alpha,\beta}$.

Let $f_j(x)$ be the generating function of the rule having the same productions as $\Omega^m_{\alpha,\beta}$ and axiom (j). Let $f_{\alpha+\beta}$ be the generating function of the rule

$$\left\{\begin{array}{ll} (\alpha+\beta) & \\ (\alpha k+\beta) \rightsquigarrow & (1)^k(2)^k\dots(\alpha-1)^k(\alpha+m-1)(\alpha+m)\dots(\alpha+\beta+m-2) \\ & (\alpha m+\beta)(\alpha(m+1)+\beta)\dots(\alpha(k+m-1)+\beta). \end{array}\right.$$

Then using a similar decomposition as previously we have

$$\begin{aligned} f_{\alpha+\beta}(x) = \quad & (1+xT(x)) + xT(x)(f_1(x)+f_2(x)+\dots+f_{\alpha-1}(x)) + \\ & +x(1+xT(x))(f_{\alpha+m-1}(x)+f_{\alpha+m}(x)+\dots+f_{\alpha+\beta+m-2}(x)). \end{aligned} \tag{2.11}$$

Indeed, the term $(1 + xT(x))$ is the generating function of the nodes reached by a path with all labels of the form $(i\alpha + \beta)$, for $i \in \{1\} \cup \{m, m+1, \dots, m+k-1\}$. The subtrees attached to these nodes and having a root labeled (i), for $i \in \{\alpha+m-1, \dots, \alpha+\beta+m-2\}$ gives the term $x(1 + xT(x))(f_{\alpha+m-1}(x) + f_{\alpha+m}(x) + \dots + f_{\alpha+\beta+m-2}(x))$. Finally, using the same trick as in the previous example the term $xT(x)(f_1(x) + f_2(x) + f_3(x) + \dots + f_{\alpha-1}(x))$ corresponds to subtrees with root labeled $(i)^k$, for $i \in \{1, \dots, \alpha - 1\}$.

Now, we need to prove that $f_m(x) = T(x)$. As in the previous example, it is sufficient to check that the equation for $f_{\alpha+\beta}$ is satisfied upon setting $f_j = T_j$ for $j \in \{1, 2, \dots, \alpha+\beta+m-2\}$ $\cup\{\alpha + \beta\}$. This can be verified by using the explicit values of the $T_j(x)$ in terms of $T(x)$ and the relation $T(x) = (1 + xT(x))^m$.

Chapter 3

Rational succession rules

A natural problem is to find a catalogue of operations on succession rules and their effects on the generating functions, as shown by the work of Banderier et al. in [1]. Pergola, Pinzani and Rinaldi in [80], describe several operations on succession rules, such as sum, product or star operations and ask for subtraction and inversion operations on succession rules. Another similar approach is used by Merlini, Sprugnoli and Verri [75] in the context of Riordan arrays. More precisely, they use the equivalence between the notion of Riordan arrays and a particular subset of marked generating trees, to define an algebra on a subset of marked succession rules and their inverses with respect to Riordan array operations. Corteel [30] also defines some *eco-systèmes signés* for exponential generating functions. In this chapter we introduce operations on succession rules and we show how signed succession rules can be used to obtain an interpretation of the subtraction and inversion of succession rules.

We then turn to the main subject of the chapter, that is the problem of finding a combinatorial interpretation to coefficients of rational functions, or equivalently to sequences defined by linear recurrences. Other attempts were made to solve this problem using for instance rational languages. We show here that simple succession rules are enough to define a large class of such sequences with non decreasing positive values (Theorem 3.5.1). Moreover, we show how arbitrary sequences could be dealt with thanks to the concept of signed succession rules, by providing an explicit solution in the case of arbitrary linear recurrences with two terms (Theorem 3.5.2).

The chapter is organised as follows. We start by the definition of pseudo and colored succession rules, that are introduced mostly for notational convenience. In Section 3.2 we recall the usual operations on succession rules. Then we use the signed succession rules described in Section 3.3 in order to define the subtraction operation and also the product and semi-product inversion in Section 3.4. In Section 3.5 we deal with linear recurrences, first using standard rules for a class of positive recurrences. Then we show how to deal with arbitrary two term linear recurrences using signed succession rules. An account of some these results can be found in [37, 39].

3.1 Pseudo and colored succession rules

In this context it is convenient to use a more general type of rules that do not necessarily satisfy the consistency principle: a *pseudo succession rule* is a system consisting of an axiom and a set of productions, denoted by

$$\Omega = \begin{cases} (a) \\ (k) \rightsquigarrow (e_1(k))(e_2(k))\dots(e_{j(k)}(k)), \quad k \in M. \end{cases} \tag{3.1}$$

where M is the set of labels, $a \in M$, $j : M \to \mathbb{N}^+$ and the e_i are functions $M \to M$.

A succession rule Ω is said to be *colored* if there are at least two labels having the same value but different productions. More precisely, let $C = \{a, b, c, d, \dots\}$ be a finite set, called the set of colors of the rule. In this context the labels have the form $(k)_i$, $i \in C$, thus the rule is specified by a colored axiom $(a)_i$ and a set of productions having the form:

$$(k)_i \rightsquigarrow (e_{1,i}(k))_{c_1(k,i)} \ (e_{2,i}(k))_{c_2(k,i)} \dots (e_{k,i}(k))_{c_k(k,i)} ,$$

where the c_i are functions $\mathbb{N}^+ \times C \to C$ that indicate the color of produced objects.

If C is made of two elements, say $C = \{a, b\}$, then we have

$$(k)_a \rightsquigarrow (e_{1,a}(k))_{c_1(k,a)} \ (e_{2,a}(k))_{c_2(k,a)} \dots (e_{k,a}(k))_{c_k(k,a)} ,$$

$$(k)_b \rightsquigarrow (e_{1,b}(k))_{c_1(k,b)} \ (e_{2,b}(k))_{c_2(k,b)} \dots (e_{k,b}(k))_{c_k(k,b)} ,$$

where $c_i : \mathbb{N}^+ \times \{a, b\} \to \{a, b\}$. In such a case, for simplicity, we represent any node of the form $(j)_a$ (resp. $(j)_b$) simply by (j)(resp. $\overline{(j)}$).

Example 3.1.1. *The following* 3*-colored finite rule defines the numbers* 1*,* 2*,* 5*,* 11*,* 28*,* 68*,* $\dots$ *,*

$$\begin{cases} (2)_a \\ (2)_a \rightsquigarrow (2)_b(3)_c \\ (3)_a \rightsquigarrow (2)_b(2)_b(3)_a \\ (2)_b \rightsquigarrow (2)_b(3)_b \\ (3)_b \rightsquigarrow (2)_b(3)_a(3)_b \\ (3)_c \rightsquigarrow (2)_a(2)_a(2)_b, \end{cases}$$

and its generating function is $\frac{1-x-x^2-x^4}{1-3x+4x^3-4x^4+4x^5}$.

Example 3.1.2.

$$\begin{cases} \overline{(1)} \\ \overline{(1)} \rightsquigarrow (1) \\ (1) \rightsquigarrow (1)(2) \\ (k) \rightsquigarrow (1)(2)\dots(k+1), \end{cases}$$

is a colored pseudo succession rule for Catalan numbers and it is equivalent to Γ *(see Subsection 1.3.2). For sake of clarity, the production involving (1) is written explicitly.*

3.2 Some operations on succession rules

We first recall from Pergola, Pinzani and Rinaldi [80] the basic operations $\oplus$, $\otimes$ and $*$ on succession rules. For rules that come from an ECO-system, these operations correspond to natural operation on the classes of combinatorial objects: disjoint union, cartesian product, and sequences. Moreover we give an operation, the semi-product, corresponding to alternating sequences of objects of two types.

Let Ω and Ω' be the following succession rules,

$$\Omega = \begin{cases} (a) \\ (a) \rightsquigarrow (b_1)(b_2)\dots(b_a) \\ (k) \rightsquigarrow (e_1(k))\dots(e_k(k)) \end{cases} \tag{3.2}$$

$$\Omega' = \begin{cases} (a') \\ (a') \rightsquigarrow (b'_1)(b'_2)\dots(b'_{a'}) \\ (k) \rightsquigarrow (e'_1(k))\dots(e'_k(k)), \end{cases} \tag{3.3}$$

where the productions for the axiom are written explicitly.

Definition 3.2.1. *The operators* $\oplus$, $\otimes$ *and* $*$ *are defined by*

$$\Omega \oplus \Omega' = \begin{cases} \widetilde{(1)} & \\ \widetilde{(1)} & \rightsquigarrow \quad (b_1)(b_2)\dots(b_a)\overline{(b'_1)}\,\overline{(b'_2)}\dots\overline{(b'_{a'})} \\ (k) & \rightsquigarrow \quad (e_1(k))(e_2(k))\dots(e_k(k)) \\ \overline{(k)} & \rightsquigarrow \quad \overline{(e'_1(k))}\,\overline{(e'_2(k))}\dots\overline{(e'_k(k))} \end{cases}$$

$$\Omega \otimes \Omega' = \begin{cases} \widetilde{(1)} & \\ \widetilde{(1)} & \rightsquigarrow \quad (b_1)(b_2)\dots(b_a)\overline{(b'_1)}\,\overline{(b'_2)}\dots\overline{(b'_{a'})} \\ (k) & \rightsquigarrow \quad (e_1(k))(e_2(k))\dots(e_k(k))\overline{(b'_1)}\,\overline{(b'_2)}\dots\overline{(b'_{a'})} \\ \overline{(k)} & \rightsquigarrow \quad \overline{(e'_1(k))}\,\overline{(e'_2(k))}\dots\overline{(e'_k(k))} \end{cases}$$

$$\Omega^* = \begin{cases} \widetilde{(a)} & \\ \widetilde{(a)} & \rightsquigarrow \quad (b_1)(b_2)\dots(b_a) \\ (k) & \rightsquigarrow \quad (e_1(k))(e_2(k))\dots(e_k(k))(b_1)(b_2)\dots(b_a). \end{cases}$$

Similarly, the operation $\boxtimes$ *is defined by*

$$\Omega \boxtimes \Omega' = \begin{cases} \widetilde{(1)} & \\ \widetilde{(1)} & \rightsquigarrow \quad (b_1)(b_2)\dots(b_a)\overline{(b'_1)}\,\overline{(b'_2)}\dots\overline{(b'_{a'})} \\ (k) & \rightsquigarrow \quad (e_1(k))(e_2(k))\dots(e_k(k))\overline{(b'_1)}\,\overline{(b'_2)}\dots\overline{(b'_{a'})} \\ \overline{(k)} & \rightsquigarrow \quad \overline{(e'_1(k))}\,\overline{(e'_2(k))}\dots\overline{(e'_k(k))}(b_1)(b_2)\dots(b_a). \end{cases}$$

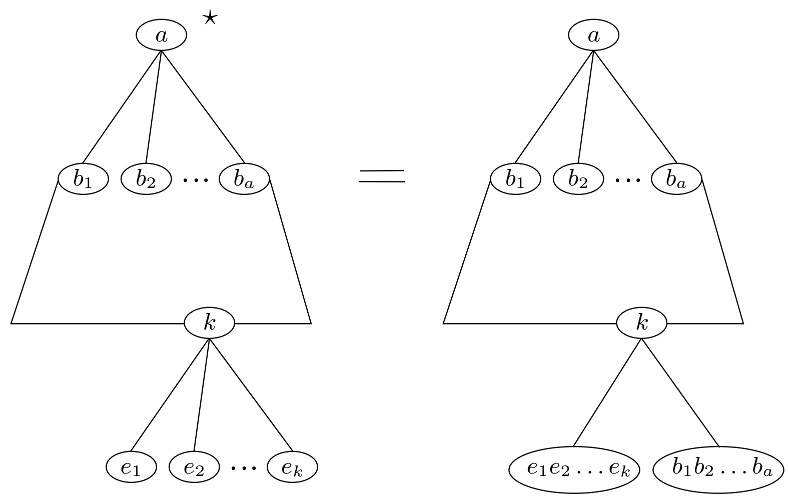

Figure 3.1: Star operation.

Remark 3.2.1. *The rules $\Omega \oplus \Omega'$, $\Omega \otimes \Omega'$, Ω^*, and $\Omega \boxtimes \Omega'$ are colored pseudo-succession rules and can be easily transformed into equivalent colored succession rules by shifting the labels.*

The operations $\oplus$, $\otimes$, $\star$, and $\boxtimes$ are easily translated into the algebra of generating functions as follows.

Proposition 3.2.1. *Given two succession rules Ω and Ω', the generating functions of $\Omega \oplus \Omega'$, $\Omega \otimes \Omega'$, Ω^*, and $\Omega \boxtimes \Omega'$ satisfy:*

$$\begin{aligned} f_{\Omega\oplus\Omega'}(x) &= f_\Omega(x) + f_{\Omega'}(x) - 1 \,, \\ f_{\Omega\otimes\Omega'}(x) &= f_\Omega(x) f_{\Omega'}(x) \,, \\ f_{\Omega^*}(x) &= \frac{1}{1-(f_\Omega(x)-1)} \,, \\ f_{\Omega\boxtimes\Omega'}(x) &= \frac{f_\Omega(x) f_{\Omega'}(x)}{f_\Omega(x) + f_{\Omega'}(x) - f_\Omega(x) f_{\Omega'}(x)} \,. \end{aligned}$$

Example 3.2.1. *Recall that a Dyck path is a sequence of rise and fall steps, running from $(0,0)$ to $(2n,0)$, and remaining weakly above the x-axis. A* Grand Dyck path *is a sequence of rise and fall steps, running from $(0,0)$ to $(2n,0)$. There is a natural bijection between $\mathcal{D} \boxtimes \mathcal{D}$ and the class of Grand Dyck paths. Let Γ be the classical succession rule for $\mathcal{D}$ (see Example 1.3.1), then the succession rule for the class of Grand Dyck paths is,*

$$\Gamma \boxtimes \Gamma = \begin{cases} (\widetilde{2}) & \\ (\widetilde{2}) & \rightsquigarrow \quad (3)(\overline{3}) \\ (k) & \rightsquigarrow \quad (3)\dots(k+1)(\overline{3}) \\ (\overline{k}) & \rightsquigarrow \quad (\overline{3})\dots(\overline{k+1})(3) \end{cases}$$

and is clearly equivalent to the following one,

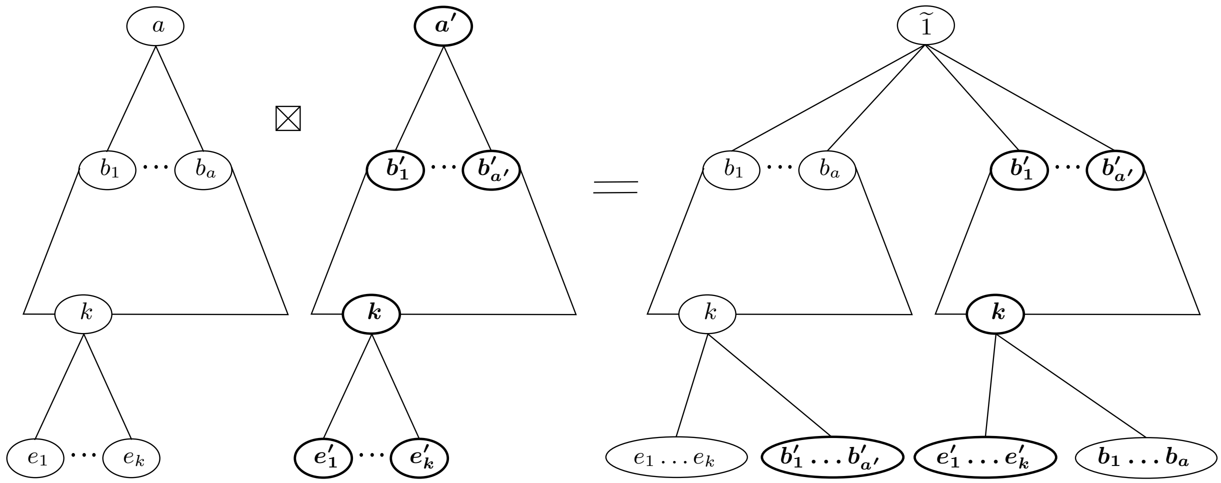

Figure 3.2: The alternating product ⊠ of two succession rules.

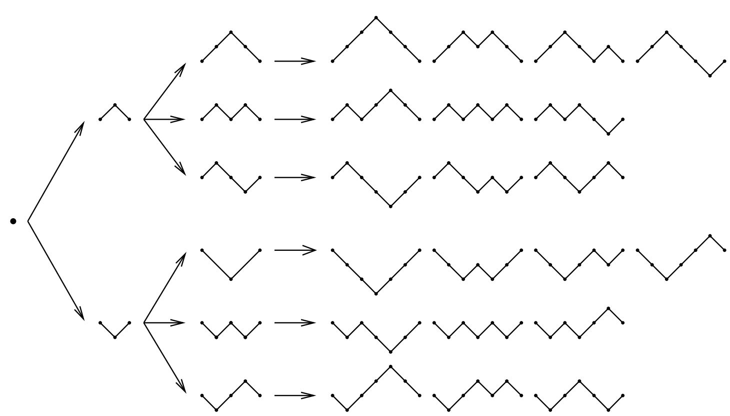

Figure 3.3: Semi-product of two Dyck paths.

$$\Gamma \boxtimes \Gamma \equiv \begin{cases} (2) \\ (k) \rightsquigarrow (3)(3)\dots(k+1). \end{cases}$$

The ECO-construction for the semi-product of two Dyck paths, and also for Grand Dyck paths, is shown in Figure 3.3. The generating function for Dyck paths is $f(x) = \frac{1-\sqrt{1-4x}}{2x}$, *and we obtain the well known generating function of Grand Dyck paths* $\frac{1}{\sqrt{1-4x}}$.

3.3 Signed succession rules

Let us consider *signed succession rules*, that is succession rules having positive or negative labels. By interpreting negative labels in the generating tree as negative objects which can annihilate some equivalent positive objects, we get signed succession rules as defined in [76, 75].

This concept allows us to interpret the inverse of a rule with respect to the Cauchy product of succession rules, and gives a combinatorial interpretation for the generating function $\frac{1}{f(x)}$.

Definition 3.3.1. *A signed succession rule is a succession rule on* $\mathbb{Z}$*,*

$$\Omega = \begin{cases} (a) \\ (a) \rightsquigarrow (b_1)\dots(b_a) \\ (k) \rightsquigarrow (e_1(k))\dots(e_k(k)) \end{cases}$$

where $k \in \mathbb{N}^+$*,* a *is a constant in* $\mathbb{N}^+$*,* $b_i \in \mathbb{Z}$ *for* $i = 1\dots a$*, and the* e_j *are functions* $\mathbb{N} \to \mathbb{Z}$*. The production for* $(-k)$ *is taken to be the opposite of the* (k) *rule, that is,*

$$(-k) \rightsquigarrow (-e_1(k))\dots(-e_k(k)).$$

Remark Definition 3.3.1 fits with the definition of Merlini, Sprugnoli and Verri [75] or Corteel [30] definition of a signed succession rule, even if our notations are slightly different. We choose to mark negative productions by negative integers. For instance our signed rule $(3) \rightsquigarrow (2)(-3)^2$ corresponds, in their notation, to the rule $(3) \rightsquigarrow (2)(3)^{-2}$. Our notation allows us to define proper signed succession rules, in which the exponents really correspond to multiplicities.

Let Ω be a signed succession rule. We denote by $\sigma(v) \in \{+,-\}$ and $l(v)$ respectively the sign and the level of a given node v. The function $f_\Omega(x) = \sum_v \sigma(v)x^{l(v)}$ is the generating function associated to Ω.

Example 3.3.1. *In Figure 3.4 there are the first levels of the generating tree of the following signed succession rule:*

$$\begin{cases} (1) \\ (1) \rightsquigarrow (-2) \\ (k) \rightsquigarrow (2)\dots(k+1)(-2), \end{cases} \tag{3.4}$$

we can easily check that the first numbers of the sequence defined by this rule are 1*,* -1*,* -1*, and* -2*.*

Definition 3.3.2. *For a signed succession rule* Ω*, we denote by* $|\Omega|$ *the following rule,*

$$|\Omega| = \begin{cases} (a) \\ (a) \rightsquigarrow (|b_1|)\dots(|b_a|) \\ (k) \rightsquigarrow (|e_1(k)|)\dots(|e_k(k)|) \end{cases}$$

Given a signed succession rule Ω, and an ECO-system $(\|\mathcal{O}\|, p, \vartheta_{\|\mathcal{O}\|}, |\Omega|)$, it is quite natural to define the set of signed objects $\mathcal{O}$ which is described by Ω. A *signed object* u of $\mathcal{O}$ is a couple $u = (\sigma(u), \|u\|)$, where $\sigma(u) \in \{+,-\}$ is the sign of u and $\|u\|$ is the corresponding object in the non signed class $\|\mathcal{O}\|$. Then, each path in the signed generating tree defines a non signed object in the corresponding non signed generating tree, by forgetting the signs. And the sign of a signed object is the sign of the last node in the signed path. The generating function of a class $\mathcal{O}$ of signed objects is $f_{\mathcal{O}(x)} = 1 + \sum_{u\neq\varepsilon} \sigma(u)x^{p(\|u\|)}$.

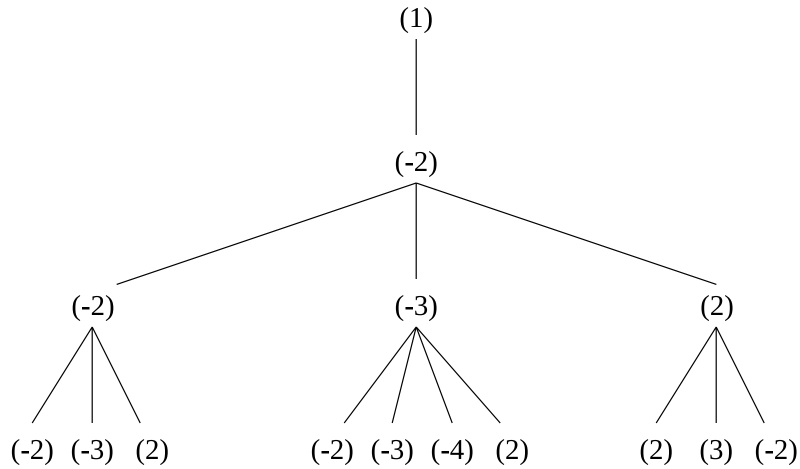

Figure 3.4: The first levels of the generating tree of the rule of Example 3.3.1.

As a first application of the definition of signed succession rules, we define the subtraction operator, answering a question of Pergola et al. [80]. In the last section we have defined the sum $\oplus$ between two succession rules and this definition can be easily extended to signed rules. In order to define the subtraction operator on succession rules, we introduce the unary operator $-$, which is related to the signed succession rules.

Definition 3.3.3. *Let Ω be a succession rule, then $-\Omega$ is defined by,*

$$-\Omega = \begin{cases} \widetilde{(a)} \\ \widetilde{(a)} \rightsquigarrow (-b_1)\dots(-b_a) \\ (k) \rightsquigarrow (e_1(k))\dots(e_k(k)) \end{cases}$$

The generating function of $-\Omega$ clearly satisfies $f_{-\Omega} = 1 - (f_\Omega - 1)$.

Definition 3.3.4. *Let Ω and Ω' be two successions rules, then the binary operator $\ominus$ is defined by $\Omega \ominus \Omega' = \Omega \oplus (-(\Omega'))$.*

Lemma 3.3.1. *If Ω and Ω' are two successions rules, then $\Omega \ominus \Omega'$ corresponds to the following signed succession rule and its generating function satisfies $f_{\Omega\ominus\Omega'} = f_\Omega - f_{\Omega'} + 1$,*

$$\Omega \ominus \Omega' = \begin{cases} \widetilde{(a+a')} \\ \widetilde{(a+a')} \rightsquigarrow (b_1)\dots(b_a)(\overline{-b'_1})\dots(\overline{-b_{a'}}) \\ (k) \rightsquigarrow (e_1(k))\dots(e_k(k)) \\ (\overline{k}) \rightsquigarrow (\overline{e'_1(k)})\dots(\overline{e'_k(k)}) \end{cases}$$

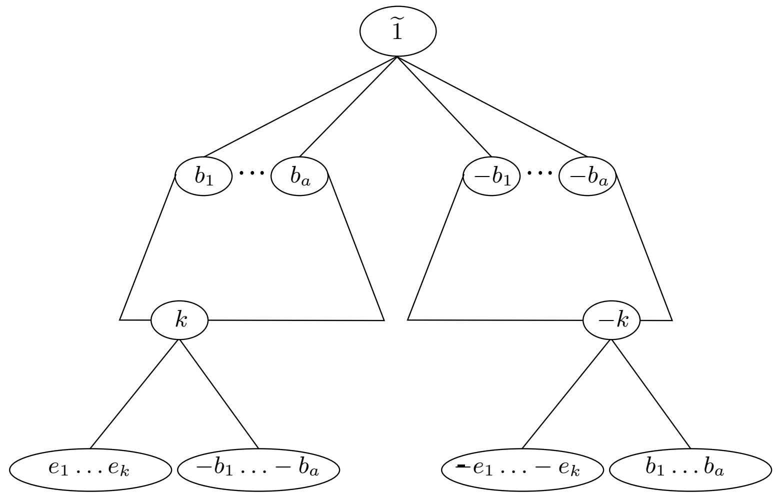

Figure 3.5: The generating tree associated to $\Omega \otimes \Omega^{-\otimes}$.

3.4 Inversion of succession rules

We are now able to give pseudo-succession rules for the inverse of a rule with respect to $\otimes$ and $\boxtimes$ operations, which are deduced by combining the unary operators $-$ and $*$. Let us define

$$\Omega^{-\otimes} = \begin{cases} \widetilde{(a)} \\ \widetilde{(a)} \rightsquigarrow (-b_1)\dots(-b_a) \\ (k) \rightsquigarrow (e_1(k))\dots(e_k(k))(-b_1)\dots(-b_a) \end{cases}$$

$$\Omega^{-\boxtimes} = \begin{cases} \widetilde{(a)} \\ \widetilde{(a)} \rightsquigarrow (-b_1)\dots(-b_a) \\ (k) \rightsquigarrow (e_1(k))\dots(e_k(k))(-b_1)^2\dots(-b_a)^2. \end{cases}$$

Theorem 3.4.1. *The pseudo-succession rules $\Omega^{-\otimes}$ and $\Omega^{-\boxtimes}$ are inverses of Ω, such that $f_{\Omega\otimes\Omega^{-\otimes}} = 1$ and $f_{\Omega\boxtimes\Omega^{-\boxtimes}} = 1$.*

Proof. We remark that these rules can be written $\Omega^{-\otimes} = (-\Omega)^*$ and $\Omega^{-\boxtimes} = ((-\Omega)^*)^*$, so that their generating functions satisfy $f_{\Omega^{-\otimes}\otimes\Omega}(x) = 1$ and $f_{\Omega^{-\boxtimes}\boxtimes\Omega}(x) = 1$. Another (bijective) proof can be given as shown in Figure 3.5 for the operation $\otimes$, by defining a natural involution on the paths of this generating tree. At level 1, each b_i can be sent on the corresponding $-b_i$. At greater levels, a path from b_i to e_j (resp. $-b_j$) can be sent on the corresponding path from $-b_i$ to $-e_j$ (resp. b_j), which concludes the proof. A similar involution can be found for proving $f_{\Omega\boxtimes\Omega^{-\boxtimes}} = 1$. $\square$

Example 3.4.1. *Inverse of Fibonacci and Catalan, with respect to the operators $\otimes$ and $\boxtimes$:*

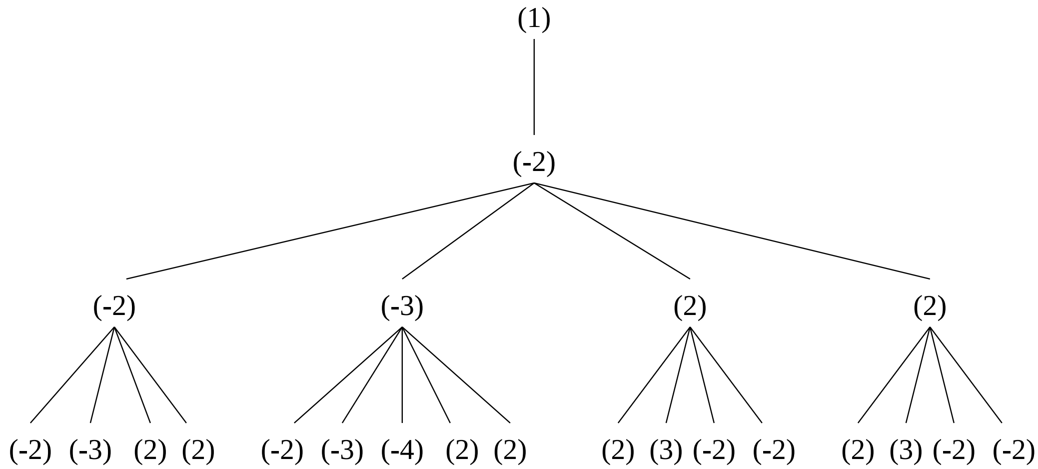

Figure 3.6: The generating tree associated to $\Gamma^{-\boxtimes}$

Fibonacci $\Psi^{-\otimes} = \begin{cases} \widetilde{(1)} \\ \widetilde{(1)} \rightsquigarrow (-2) \\ (1) \rightsquigarrow (2)(-2) \\ (2) \rightsquigarrow (1)(2)(-2) \end{cases}$ $\quad \Psi^{-\boxtimes} = \begin{cases} \widetilde{(1)} \\ \widetilde{(1)} \rightsquigarrow (-2) \\ (1) \rightsquigarrow (2)(-2)^2 \\ (2) \rightsquigarrow (1)(2)(-2)^2 \end{cases}$

Catalan $\Gamma^{-\otimes} = \begin{cases} (1) \\ (1) \rightsquigarrow (-2) \\ (k) \rightsquigarrow (2)\dots(k+1)(-2) \end{cases}$ $\quad \Gamma^{-\boxtimes} = \begin{cases} (1) \\ (1) \rightsquigarrow (-2) \\ (k) \rightsquigarrow (2)\dots(k+1)(-2)^2 \end{cases}$

The first levels of the generating trees associated with $\Gamma^{-\otimes}$ *and* $\Gamma^{-\boxtimes}$ *are represented respectively in Figure 3.4 and in Figure 3.6.*

3.5 Rational succession rules

The subject of this section are rational succession rules. We first propose a simple tool to pass from a rather general family of linear recurrence relations defining non-decreasing sequences of positive integers, to rational succession rules defining the same sequences. Secondly we use signed succession rules in order to deal with two term recurrences defining any sequence of numbers. Related results on succession rules with rational generating functions can be found in Banderier et al. in [1].

Our technique provides interesting combinatorial interpretations (in terms of generating trees) for sequences that are defined by a linear recurrence relation, using an approach different from that in [4] and [9]. In particular, as an application of our method, we give a simple solution to a problem proposed by Jim Propp on the mailing list "domino" (1999), where he asked for the combinatorial interpretation of the sequence $1, 1, 1, 2, 3, 7, 11, 26, \ldots$ (sequence A005246 in [92]) defined by the linear recurrence relation:

$$\begin{cases} f_0 = 1, \quad f_1 = 1, \quad f_2 = 1, \quad f_3 = 2 \\ f_n = 4f_{n-2} - f_{n-4}. \end{cases}$$

3.5.1 Some positive two term linear recurrences.

Consider a sequence of non-decreasing positive integers $(f_n)_{n\geq 0}$ satisfying a two term linear recurrence of the form:

$$f_n = h_1 f_{n-1} + h_2 f_{n-2}, \qquad h_1, h_2 \in \mathbb{Z}.$$

As far as we know it is an open problem to give combinatorial interpretations to arbitrary sequences of this type. As shown by the following proposition succession rules allow to solve easily this problem for a large class of these recurrences.

Proposition 3.5.1. *Assume $h_1 > 0$, and $h_1 + h_2 > 0$. Then the succession rule*

$$\Omega = \begin{cases} (s_0) \\ (k) \rightsquigarrow (1)^{k-1}\,(\phi(k))\,, \end{cases}$$

with $\phi(k) = (h_1 - 1)k + h_2 + 1$, defines the sequence $(f_n)_{n\geq 2}$ with initial conditions $f_0 = 1$, $f_1 = s_0 \in \mathbb{N}^+$.

Proof. Observe that the conditions imposed on h_1 and h_2, ensure the positivity of the labels of Ω. Let f'_n denote the number of nodes at level n of the generating tree. We will check that f'_n satisfies the same recurrence as f_n. Trivially, we have $f'_0 = 1$ and $f'_1 = s_0$. Let $k_1, k_2, \ldots k_{f'_{n-2}}$ be the labels at level $n-2$ of the generating tree of Ω. Let us consider the f'_{n-1} labels at level $n-1$. On the one hand their sum is equal to f'_n according to the consistency principle. On the other hand, according to the definition of the rule their sum is equal to

$$k_1 + k_2 + \ldots + k_{f'_{n-2}} - f'_{n-2} + (h_1 - 1)(k_1 + k_2 + \ldots + k_{f'_{n-2}}) + (h_2 + 1)f'_{n-2}.$$

Consequently, using again the consistency principle we have

$$f'_n = f'_{n-1} - f'_{n-2} + (h_1-1)f'_{n-1} + (h_2+1)f'_{n-2} = h_1 f'_{n-1} + h_2 f'_{n-2} \quad n \geq 2.$$

□

Observe that a similar kind of succession rules had already been shown to have rational generating function by Banderier et al. [1]. More generally, assuming that there exists a positive integer c such that:

(i.) if $c \leq s_0$ then $1 \leq c \leq h_1$ and $(h_1 - c)c + h_2 > 0$;

(ii.) if $c > s_0$ then $s_0 < c \leq h_1$ and $(h_1 - c)s_0 + h_2 + c > s_0$,

the sequence f_n is also given by the succession rule:

$$\Omega_2 = \begin{cases} (s_0) \\ (k) \rightsquigarrow (c)^{k-1}\,(\phi(k)) \end{cases}$$

where $c, s_0 \in \mathbb{N}^+$, $\phi(k) = (h_1 - c)k + h_2 + c$. Again the conditions on h_1 and h_2 are required for the positivity of the labels.

Example 3.5.1. *Let us consider the recurrence*

$$f_0 = 1, \quad f_1 = 2 \quad f_n = 3f_{n-1} - f_{n-2},$$

defining odd-index Fibonacci numbers $1, 2, 5, 13, 34, 89, 233, \ldots$ *(sequence A001519 in [93]). A succession rule defining this sequence is*

$$\begin{cases} (2) \\ (k) \rightsquigarrow (2)^{k-1}(k+1). \end{cases} \tag{3.5}$$

3.5.2 Linear recurrences with more than two terms.

In this paragraph we consider linear recurrences defining non-decreasing sequences of positive integers with the following form:

$$f_n = h_1 f_{n-1} + h_2 f_{n-2} + \ldots + h_m f_{n-m}, \text{ where } h_i \in \mathbb{Z},$$

with $m \geq 3$.

We now extend the statement of Proposition 3.5.1. In order to do that let us fix two integers c_0 and s_0, and define for $\ell = 1, \ldots, m-2$

$$\begin{cases} c_\ell = (h_1 - c_0)c_{\ell-1} + \sum_{i=2}^{\ell+1} h_i + c_0 \\[2ex] s_\ell = (h_1 - c_0)s_{\ell-1} + \sum_{i=2}^{\ell+1} h_i + c_0 \end{cases}$$

then we consider the rule

$$\Omega_m = \begin{cases} (s_0) \\ (s_\ell) \rightsquigarrow (c_0)^{s_\ell - 1}(s_{\ell+1}) \text{ for } \ell = 0, \ldots, m-3 \\ (c_\ell) \rightsquigarrow (c_0)^{c_\ell - 1}(c_{\ell+1}) \text{ for } \ell = 0, \ldots, m-3 \\ (k) \rightsquigarrow (c_0)^{k-1}\,(\phi(k)) \end{cases}$$

where

$$\phi(k) = (h_1 - c_0)k + \sum_{i=2}^{m} h_i + c_0,$$

and we add colors if necessary to distinguish ambiguous labels.

For this rule to be well defined with positive labels, a sufficient condition is that the constants c_ℓ and s_ℓ be positive, $c_0 \leq h_1$, $\phi(c_{m-2}) > \min(c_{m-2}, s_{m-2})$ and $\phi(s_{m-2}) > \min(c_{m-2}, s_{m-2})$.

Theorem 3.5.1. *Assume that there exists c_0 such that $h_1, \ldots, h_m$ and s_0 satisfy the previous conditions. Then the succession rule Ω_m gives the non-decreasing positive sequence satisfying the recurrence relation:*

$$f_n = h_1 f_{n-1} + h_2 f_{n-2} + \ldots + h_m f_{n-m},$$

with initial conditions $f_i = 0$, for $i = -m+2, \ldots, -1$, $f_0 = 1$, and $f_1 = s_0$.

In particular, if $h_1 > 0$, $h_1 + h_2 > 0$, $\ldots$, $h_1 + \ldots + h_m > 0$ then the result is valid for any $1 \leq c_0 \leq h_1$. Taking $c_0 = h_1$ gives moreover a finite rule.

Proof. Observe that the number of labels c_ℓ at level n of the generating tree of the rule Ω_m is $f'_{n-\ell} - f'_{n-\ell-1}$, for all $\ell = 0, \ldots, m-2$, where f'_n denote the total number of labels at level n. The proof is then a variation on the proof of Proposition 3.5.1. □

Example 3.5.2.

i) The sequence $(f_n)_{n \geq 0}$ satisfying the recurrence relation:

$$f_n = 3f_{n-1} - 2f_{n-2} + f_{n-3},$$

with $f_1 = 0$, $f_0 = 1$, $f_1 = 2$, is defined by the succession rule:

$$\left\{ \begin{array}{l} (2) \\ (1) \rightsquigarrow (1) \\ (2) \rightsquigarrow (1)(3) \\ (k) \rightsquigarrow (1)^{k-1}(2k) \quad k \geq 3. \end{array} \right.$$

ii) NSW numbers (sequence A002315 in [92]) are defined by the recurrence relation:

$$f_n = 6f_{n-1} - f_{n-2}, \qquad f_0 = 1, \ f_1 = 7.$$

These numbers count the total area under elevated Schröder paths [79, 12]. According to Theorem 3.5.1, the succession rule defining these numbers is:

$$\left\{ \begin{array}{l} (7) \\ (k) \rightsquigarrow (1)^{k-1}(5k) \end{array} \right.$$

iii) Self-avoiding walks of length n, contained in the strip $\{0,1\} \times [-\infty, \infty]$, are counted by the sequence $\{f_n\}$ that satisfies a linear recurrence relation [101]:

$$\begin{array}{l} f_0 = 1,\ f_1 = 3,\ f_2 = 6,\ f_3 = 12,\ f_4 = 20,\ f_5 = 36,\ f_6 = 58,\ f_7 = 100, \\ f_n = f_{n-1} + 3f_{n-2} + 2f_{n-3} - 3f_{n-4} + f_{n-5} + f_{n-6} \quad n > 7. \end{array} \tag{3.6}$$

For simplicity we change the initial conditions into:

$$\begin{array}{l} f_{-i} = 0, \quad i = 1 \dots 5 \\ f_0 = 1. \end{array}$$

Then the succession rule obtained applying Theorem 3.5.1 is:

$$\begin{cases} (1) \\ (1) \rightsquigarrow (4) \\ (3) \rightsquigarrow (1)^2(\overline{4}) \\ (4) \rightsquigarrow (1)^3(6) \\ (\overline{4}) \rightsquigarrow (1)^3(5) \\ (5) \rightsquigarrow (1)^4(5) \\ (6) \rightsquigarrow (1)^5(3). \end{cases}$$

We remark that a rule defining the original number sequence can be simply obtained by adding some other productions, in order to satisfy the initial conditions.

Example 3.5.3. *It is also possible in some cases to deal with recurrences having $h_1 = 0$, along the similar lines as above. For instance here is a succession rule for the number sequence $1, 2, 3, 7, 11, 26, \dots$, defined at the beginning of Section 3.5.*

$$\begin{cases} (2) \\ (2) \rightsquigarrow (1)(2) \\ (1) \rightsquigarrow (4) \\ (4) \rightsquigarrow (1)^3(\overline{1}) \\ (3) \rightsquigarrow (1)^2(\overline{1}) \\ (\overline{1}) \rightsquigarrow (3). \end{cases}$$

3.5.3 Succession rules with negative labels.

Theorem 3.5.1 clearly does not involve the whole set of rational generating functions. If we want to treat the whole set of rational generating functions we have to allow labels of the rules to contain negative values. Under this hypothesis, we find again the concept of signed succession rules, introduced in Section 3.3. A signed succession rule defines a sequence of integer numbers $(f_n)_{n \geq 0}$, where the term f_n is given by the number of positive labels minus the number of negative labels at level n of the generating tree. Now, we want to investigate the relationship between linear recurrences and signed succession rules by applying the same tools that we used for recurrences with positive terms. Hereafter we present the main results concerning linear recurrences with two terms.

Let us consider a two term linear recurrence $f_n = h_1 f_{n-1} + h_2 f_{n-2}$, with $h_1, h_2 \in \mathbb{Z}$. and initial conditions $f_1 = s_1$, $f_2 = s_2$ in $\mathbb{Z}$. To this sequence we associate the rule

$$\Omega_2^{\varepsilon} = \begin{cases} \overline{(|s_1|+2)} & \\ \overline{(|s_1|+2)} \rightsquigarrow (\varepsilon_{s_1} 1)^{|s_1|}(p)(-q) & \\ (k) \rightsquigarrow (\varepsilon_1 c)^{k-1}\,(\varepsilon_2 \phi(k)) & \text{for } k > 0, \\ (-k) \rightsquigarrow (-\varepsilon_1 c)^{k-1}\,(-\varepsilon_2 \phi(k)) & \end{cases}$$

where ε_{s_1} is the sign of s_1, p, q are positive integers such that $p - q = \varepsilon_1 s_2 - \varepsilon_1 \varepsilon_2 s_1$,

$$\phi(k) = (\varepsilon_2 h_1 - \varepsilon_1 \varepsilon_2 (c-1) - 1)k + 2 - 2\varepsilon_1\varepsilon_2 + \varepsilon_1\varepsilon_2(h_2 + c) + (\varepsilon_1 - \varepsilon_2)h_1,$$

and c, ε_1, and ε_2 satisfy the following conditions (that imply $\phi(k) > 0$ for $k \geq 1$):

(i.) $\varepsilon_2 h_1 - \varepsilon_1\varepsilon_2(c-1) - 1 \geq 0$

(ii.) $\phi(1) = -\varepsilon_1\varepsilon_2 + \varepsilon_1\varepsilon_2 h_2 + \varepsilon_1 h_1 + 1 > 0$

Observe that given $h_1 \neq 0$, h_2, s_1, and s_2 it is always possible to find c, ε_1, and ε_2: in particular, if we take ε_1 of the sign of h_1, then we can set $c = \varepsilon_1 h_1 - \varepsilon_1\varepsilon_2 + 1$ to satisfy the first condition, and the second condition reduces to $c + \varepsilon_1\varepsilon_2 h_2 > 0$, which can be satisfied by choosing ε_2 of the sign of $\varepsilon_1 h_2$ since $c \geq 1$.

Theorem 3.5.2. *The succession rule Ω_2^{ε} defines the sequence $(f_n)_{n\geq 1}$, satisfying the linear recurrence $f_n = h_1 f_{n-1} + h_2 f_{n-2}$, with initial terms $f_1 = s_1$, $f_2 = s_2$, and $h_1 \neq 0$, h_2 in $\mathbb{Z}$.*

Proof. The axiom produces, at level 1 of the generating tree, $|s_1| + 1$ nodes with labels of the sign of s_1 and 1 of the other sign. Therefore $f_1 = s_1$. At the second level the condition on $p - q$ and the general production gives $f_2 = s_2$.

Let us suppose to have l positive labels and m negative ones at level $n-2$ of the generating tree, for $n \geq 2$. Thus by definition of negative succession rules, $l - m = f_{n-2}$. Let us denote by k_1, $k_2, \ldots k_l$ the positive labels, and by $l_1\, l_2, \ldots l_m$ the negative ones. A node with a positive label (k_i), produces $(\varepsilon_1 c)^{k_i - 1}(\varepsilon_2\phi(k_i))$, for $i = 1 \ldots l$. On the other hand, a node with negative labels, (l_j), produces $(-\varepsilon_1 c)^{l_j-1}(-\varepsilon_2\phi(l_j))$, for $j = 1 \ldots m$. Consequently, at level $n-1$, we have the following labels

$$\begin{array}{ll} (\varepsilon_1 c)^{k_i-1}(\varepsilon_2\phi(k_i)) & \text{for } i = 1 \ldots l \\ (-\varepsilon_1 c)^{l_j-1}(-\varepsilon_2\phi(l_j)) & \text{for } j = 1 \ldots m \end{array} \tag{3.7}$$

Since $\phi(k)$ is positive, we know the sign of all these labels. Therefore the number of nodes at level $n-1$ of the generating tree is given by

$$\begin{aligned} f_{n-1} &= \varepsilon_1 \textstyle\sum_{j=1}^{l}(k_i - 1) + \varepsilon_2 l - \varepsilon_1 \sum_{j=1}^{m}(l_j - 1) - \varepsilon_2 m \\ &= \varepsilon_1 \textstyle\sum_{j=1}^{l}(k_i - 1) - \varepsilon_1 \sum_{j=1}^{m}(l_j - 1) + \varepsilon_2 f_{n-2}. \end{aligned} \tag{3.8}$$

Therefore

$$\varepsilon_1(f_{n-1} - \varepsilon_2 f_{n-2}) = \sum_{i=1}^{l}(k_i - 1) - \sum_{i=1}^{m}(l_i - 1)$$

Now, the productions of the labels at level $n-1$ are (3.7)

$$\begin{array}{rcll}
(\varepsilon_1 c) & \rightsquigarrow & (c)^{c-1}(\varepsilon_2\varepsilon_1\phi(c)), & \sum_{i=1}^{l}(k_i - 1) \text{ times}, \\
(\varepsilon_2\phi(k_i)) & \rightsquigarrow & (\varepsilon_2\varepsilon_1 c)^{\phi(k_i)-1}(\phi(\phi(k_i))), & \text{for } i = 1, \dots, l, \\
(-\varepsilon_1 c) & \rightsquigarrow & (-c)^{c-1}(-\varepsilon_2\varepsilon_1\phi(c)), & \sum_{i=1}^{m}(l_i - 1) \text{ times}, \\
(-\varepsilon_2\phi(l_i)) & \rightsquigarrow & (-\varepsilon_2\varepsilon_1 c)^{\phi(l_i)-1}(-\phi(\phi(l_i))), & \text{for } i = 1, \dots, m.
\end{array}$$

Hence

$$\begin{aligned}
f_n = \ & \left(\textstyle\sum_{j=1}^{l}(k_i - 1) - \sum_{j=1}^{m}(l_j - 1)\right)((c-1) + \varepsilon_1\varepsilon_2) + \ f_{n-2} + \\
& + \varepsilon_1\varepsilon_2 \textstyle\sum_{j=1}^{l}(\phi(k_i) - 1) - \varepsilon_1\varepsilon_2 \sum_{j=1}^{m}(\phi(l_j) - 1).
\end{aligned}$$

By using equation (3.8) we obtain

$$f_n = \varepsilon_1(f_{n-1} - \varepsilon_2 f_{n-2})(c - 1 + \varepsilon_1\varepsilon_2) + f_{n-2} + \varepsilon_1\varepsilon_2\sum_{j=1}^{l}(\phi(k_i) - 1) - \varepsilon_1\varepsilon_2\sum_{j=1}^{m}(\phi(l_j) - 1),$$

which is equal to

$$\begin{aligned}
f_n \ & = (f_{n-1} - \varepsilon_2 f_{n-2})(\varepsilon_1(c-1) + \varepsilon_2) + f_{n-2}(1 - \varepsilon_1\varepsilon_2) + \\
& + \varepsilon_1\varepsilon_2(\varepsilon_2 h_1 - \varepsilon_1\varepsilon_2(c-1) - 1)(\textstyle\sum_{j=1}^{l}(k_i - 1) - \sum_{j=1}^{m}(l_j - 1)) + \\
& + \varepsilon_1\varepsilon_2(l - m)(-\varepsilon_1\varepsilon_2 + \varepsilon_1\varepsilon_2 h_2 + \varepsilon_1 h_1 + 1).
\end{aligned}$$

Consequently,

$$\begin{aligned}
f_n \ & = (f_{n-1} - \varepsilon_2 f_{n-2})(\varepsilon_1(c-1) + \varepsilon_2 + \varepsilon_2(\varepsilon_2 h_1 - \varepsilon_1\varepsilon_2(c-1) - 1)) \\
& + f_{n-2}(1 - \varepsilon_1\varepsilon_2) + \varepsilon_1\varepsilon_2 f_{n-2}(-\varepsilon_1\varepsilon_2 + \varepsilon_1\varepsilon_2 h_2 + \varepsilon_1 h_1 + 1) \\
& = f_{n-1}h_1 + f_{n-2}h_2.
\end{aligned}$$

$\square$

Example 3.5.4. *Let us consider the number sequence* $1, 2, -10, 22, -26, -10,$ $134, \dots,$ *defined by the recurrence relation:*

$$\begin{aligned}
& f_0 = 1, \ f_1 = 2, \\
& f_n = -3f_{n-1} - 4f_{n-2} \quad n > 1.
\end{aligned}$$

A succession rule defining this sequence is obtained from rule Ω_2^{ε} *with* $c = 1$, $\varepsilon_1 = 1$, *and* $\varepsilon_2 = -1$. *In this case, since* $f_0 = 1$, *we can take* $s_1 = 2$ *and* $s_2 = -10$. *Thus we obtain the followin rule:*

$$\left\{\begin{array}{l} (4) \\ (k) \rightsquigarrow (1)^{k-1}(-2k-1) \\ (-k) \rightsquigarrow (-1)^{k-1}(2k+1). \end{array}\right.$$

Example 3.5.5. *Odd-index Fibonacci numbers with alternating sign,* $1, -2, 5,$ $-13, 34, -89, \dots,$ *are defined by the recurrence relation,*

$$\begin{array}{l} f_0 = 1,\ f_1 = -2, \\ f_n = -3f_{n-1} - f_{n-2} \quad n > 1. \end{array}$$

A succession rule defining such a sequence is obtained from Ω_2^{ε} *where* $c = 1$, $\varepsilon_1 = -1$, *and* $\varepsilon_2 = -1$,

$$\left\{\begin{array}{l} (4) \\ (4) \rightsquigarrow (-1)^2(1)(-4) \\ (k) \rightsquigarrow (-1)^{k-1}(-2k) \\ (-k) \rightsquigarrow (1)^{k-1}(2k). \end{array}\right.$$

We point out that the rule (3.5.5) is very similar to (3.5), which defines odd-index Fibonacci numbers.

Part II

ECO method and Object grammars

Chapter 4

From Object Grammars to ECO-systems

This chapter will investigate the relationship between object grammars and ECO method. As already said, the ECO method constructs each object from a smaller one by making some local expansions. On the other hand, object grammars describe the objects by decomposing them. These recursive methods have many common applications, as for instance enumeration of objects, bijections, and uniform random generation. We are interested in comparing them. Roughly speaking, an object grammar and an ECO-system are considered "equivalent" when they both define a construction for the same class $\mathcal{O}$ according to the same parameter p. Therefore there are two natural questions:

1. Is it possible to obtain an ECO-system from an object grammar?
2. Is it possible to obtain an object grammar from an ECO-system?

Here we examine Question 1., that is, the problem of passing from an object grammar to an ECO-system (Question 2. will be discussed in Chapter 6). The main contribution of this chapter is a method to get, from a complete and unambiguous object grammar G for a class $\mathcal{O}$, an ECO-system for $\mathcal{O}$, according to some linear parameter. The result is obtained by providing an ECO construction for a class of trees, called *α-trees*, in bijection with the class of derivation trees of G, that is in turn in bijection with the class $\mathcal{O}$. We first prove it for unidimensional object grammars (Theorem 4.3.1). Then we extend it to multidimensional object grammars (Theorem 4.4.1). Pratically this means that a class described by a complete and unambiguous object grammar can be also described by an ECO-system according to a linear parameter.

The chapter is organized as follows. In Section 4.1 we recall the main definitions concerning object grammars. In Section 4.2 we introduce the concept of linear parameters and q-parameters on object grammars. In Section 4.3 we show how to obtain an ECO-system starting from an unidimensional, unambiguous, and complete object grammar according to a linear parameter. In particular we first deal with the easier case of uniform linear parameters (Theorem 4.3.2) and then we deal with linear parameters (Theorem 4.3.3). In Section 4.4 we extend the result obtained for unidimensional grammars to multidimensional ones (Theorem 4.4.1). In Section 4.5 we explicitly construct the ECO-system associated with the multidimensional grammar for directed convex polyominoes. An account of the results of this chapter can be found in [38].

4.1 Object Grammars

We recall some basic definitions for object grammars and then we give some examples. One can refer to Dutour [44] for more details.

4.1.1 Definitions

Definition 4.1.1. *Let $\mathbb{O}$ be a finite family of classes of objects. An* object operation *in $\mathbb{O}$ is a mapping $\phi : \mathcal{O}^1 \times \ldots \times \mathcal{O}^k \to \mathcal{O}$, where $\mathcal{O}, \mathcal{O}^i \in \mathbb{O}$, $i = 1, \ldots, k$. The domain and codomain of an object operation are respectively denoted as* dom *and* cod*.*

An object operation describes a way of building recursively an object of $\mathcal{O}$ from k objects belonging to $\mathcal{O}^1, \ldots, \mathcal{O}^k$.

Definition 4.1.2. *An* object grammar *is a quadruple $\langle \mathbb{O}, \mathbb{E}, \Phi, \mathcal{A} \rangle$ where :*

- *$\mathbb{O}$ is a finite family of classes of objects.*
- *$\mathbb{E} = \{\mathcal{E}_\mathcal{O}\}_{\mathcal{O} \in \mathbb{O}}$ is a finite family of finite subclasses of the classes belonging to $\mathbb{O}$. The objects of $\mathbb{E}$ are called* terminal objects*.*
- *Φ is a finite set of object operations in $\mathbb{O}$.*
- *$\mathcal{A}$ is a fixed class of $\mathbb{O}$, called the* axiom *of the grammar.*

The *dimension* of an object grammar is the cardinality of $\mathbb{O}$.

Definition 4.1.3. *Let $G = \langle \mathbb{O}, \mathbb{E}, \Phi, \mathcal{A} \rangle$ be an object grammar and let $\mathcal{O} \in \mathbb{O}$. A* derivation tree *of G on $\mathcal{O}$ is an ordered labelled tree T, recursively described as follows :*

- *if T is reduced to a leaf then the label is a terminal object belonging to $\mathcal{O}$,*
- *if the root of T has k sons then its label is an object operation $\phi \in \Phi$,*

$$\phi : \mathcal{O}^1 \times \ldots \times \mathcal{O}^k \to \mathcal{O},$$

  where $\mathcal{O}^i \in \mathbb{O}$ and such that the i-th son of the root is the root of a derivation tree on the class $\mathcal{O}^i$, $i = 1 \ldots k$.

Definition 4.1.4. *The evaluation $ev(T)$ of a derivation tree T is an object defined as follows :*

- *if T is a single node labelled E, then $ev(T) = E$,*
- *otherwise, if the root of T is labelled $\phi \in \Phi$ and its k subtrees are $T_1 \ldots T_k$, then $ev(T) = \phi(ev(T_1), \ldots, ev(T_k))$.*

Definition 4.1.5. *Let $G = \langle \mathbb{O}, \mathbb{E}, \Phi, \mathcal{A} \rangle$ be an object grammar. An object $O \in \mathcal{O}$ is said to be* generated *in G by $\mathcal{O}$ if there is a derivation tree T on $\mathcal{O}$ such that $ev(T) = O$.*

The class of objects generated in G by $\mathcal{A}$ is said to be the class generated by G.

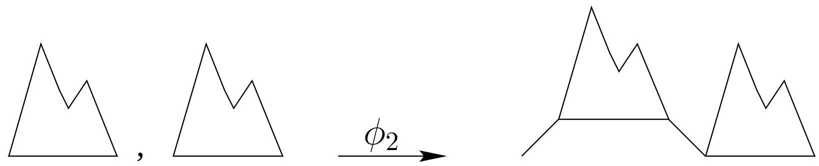

Figure 4.1: The object operation ϕ_2 of the grammar $G_{\mathcal{D}}$.

Definition 4.1.6. *An object grammar G is said to be* complete *if, for all $\mathcal{O} \in \mathbb{O}$, the class of objects generated in G by $\mathcal{O}$ is equal to $\mathcal{O}$.*

Definition 4.1.7. *An object grammar G is said to be* unambiguous *if every object generated by G admits at most one derivation tree.*

If $\mathcal{T}_G$ denotes the class of derivation trees of a grammar G, then the following statement trivially holds:

Proposition 4.1.1. *If G is a complete and unambiguous object grammar generating the class $\mathcal{O}$, then the function $ev : \mathcal{T}_G \to \mathcal{O}$ is a bijection.*

A complete and unambiguous object grammar $G = \langle \mathbb{O}, \mathbb{E}, \Phi, \mathcal{A} \rangle$ of dimension m can be also written as a system Σ of equations:

$$\Sigma = \{\mathcal{O}^1 = P_1(\mathcal{O}^1, \dots, \mathcal{O}^m), \dots, \mathcal{O}^m = P_m(\mathcal{O}^1, \dots, \mathcal{O}^m)\},$$

where for all $i = 1 \dots m$

$$P_i(\mathcal{O}^1, \dots, \mathcal{O}^m) = \sum_{e_i \in \mathcal{E}_{\mathcal{O}^i}} e_i + \sum_{cod(\Phi) = \mathcal{O}^i} \Phi(\mathcal{O}^{i_{1,\Phi}}, \dots, \mathcal{O}^{i_{k,\Phi}}).$$

From now on, we will often use the term *grammar* in place of *object grammar*. Moreover we will omit the axiom in the case of unidimensional grammars. Object grammars are most often described by pictures, as in the following examples.

4.1.2 Some examples of object grammars

Dyck paths. Let $\mathcal{D}$ be the class of Dyck paths. The mapping ϕ_2 depicted in Figure 4.1 is a binary object operation on the class $\mathcal{D}$ of Dyck paths: it takes a pair of Dyck paths as its argument, adds a rise (resp. fall) step at the beginning (resp. end) of the first path and then appends the second path. The class $\mathcal{D}$ is generated by the unidimensional, complete, and unambiguous object grammar

$$G_{\mathcal{D}} = \langle \mathcal{D}, \{\{.\}\}, \{\phi_2\} \rangle$$

where the terminal object is the Dyck path of zero length, commonly represented as a dot. Each Dyck path is then univocally associated with a derivation tree of $G_{\mathcal{D}}$ (see for instance Figure 4.2).

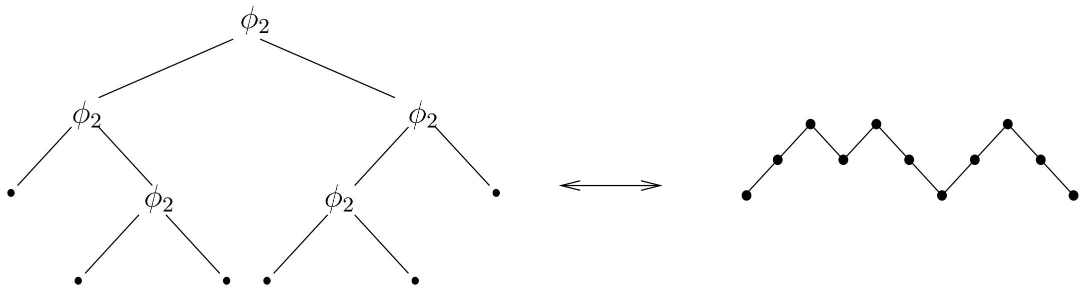

Figure 4.2: A derivation tree from the grammar $G_{\mathcal{D}}$ and the corresponding Dyck path.

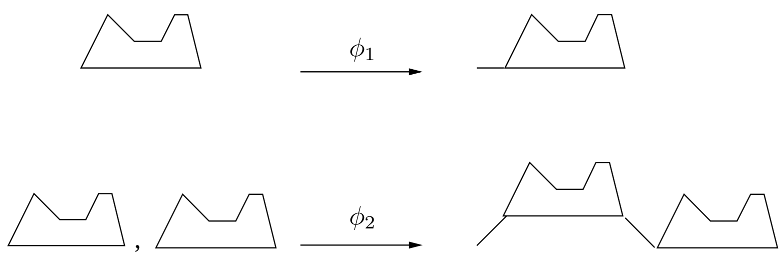

Figure 4.3: The object operations, ϕ_1, ϕ_2 of the grammar $G_{\mathcal{M}}$.

Motzkin paths. Let $\mathcal{M}$ be the class of Motzkin paths. The mappings ϕ_1 and ϕ_2, illustrated in Figure 4.3, are the object operations on $\mathcal{M}$. The first is unary while the second is binary:

- operation ϕ_1 adds an horizontal step at the beginning of the Motzkin path;
- operation ϕ_2 takes a pair of Motzkin paths as its argument, adds a rise (resp. fall) step at the beginning (resp. end) of the first path and then appends the second path.

The class $\mathcal{M}$ is generated by the unidimensional, complete, and unambiguous object grammar

$$G_{\mathcal{M}} = \langle \{\mathcal{M}\}, \{\{.\}\}, \{\phi_1, \phi_2\}\rangle$$

where the terminal object is the path of zero length, represented as a dot.

Parallelogram polyominoes. Let $\mathcal{P}$ be the set of parallelogram polyominoes. The mappings ϕ_1^1, ϕ_1^2, and ϕ_2, illustrated in Figure 4.4, are the object operations on $\mathcal{P}$:

- operation ϕ_1^1 adds a cell at the left of the lowest cell of the first column of the polyomino;
- operation ϕ_1^2 adds a cell at the bottom of every column of the polyomino;
- operation ϕ_2 applies ϕ_1^2 to the first polyomino and then glues the right side of the top cell of the last column of the first one to the left side bottom cell of the first column of the second one.

The unidimensional, complete, and unambiguous grammar

$$G_{\mathcal{P}} = \left\langle \{\mathcal{P}\}, \{\{\square\}\}, \{\phi_1^1, \phi_1^2, \phi_2\}\right\rangle$$

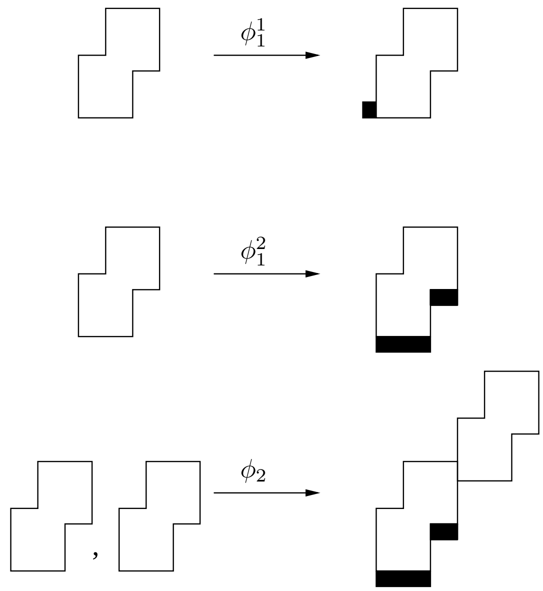

Figure 4.4: The object operations ϕ_1^1, ϕ_1^2, ϕ_2 of the grammar $G_{\mathcal{P}}$.

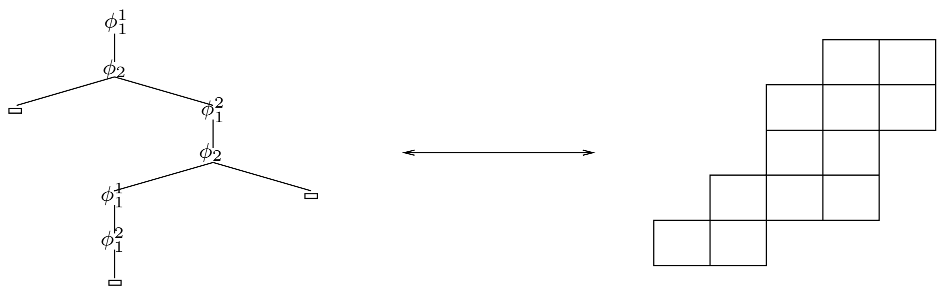

Figure 4.5: A polyomino and its derivation tree according to $G_{\mathcal{P}}$.

generates the class $\mathcal{P}$, where $\square$ denotes the single-cell polyomino. Figure 4.5 illustrates a derivation tree of $G_{\mathcal{P}}$ and its corresponding polyomino.

Directed convex polyominoes Let $\mathcal{P}_{cd}$ be the class of directed convex polyominoes, let $\mathcal{P}_{cdm}$ be the class of directed convex polyominoes with one marked cell in their last column, and let $\mathcal{P}$ be the class of parallelogram polyominoes. Then $\mathcal{P}_{cd}$ is generated by the 3-dimensional grammar

$$G_{\mathcal{P}_{cd}} =< \{\mathcal{P}_{cd}, \mathcal{P}_{cdm}, \mathcal{P}\}, \{\{\square\}, \emptyset, \{\square\}\}, \{\theta_1^1, \theta_1^2, \theta_2, \psi_1^1, \psi_1^2, \psi_2, \phi_1^1, \phi_1^2, \phi_2\}, \mathcal{P}_{cd} >,$$

represented in Figure 4.6 as a system of graphical equations. The operations θ_1^1 and ψ_1^2 are defined in the same manner as ϕ_1^2, and θ_2 and ψ_2 are similar to ϕ_2, where ϕ_1^2 and ϕ_2 were previously described for the grammar $G_{\mathcal{P}}$ of parallelogram polyominoes. The operation θ_1^2 takes a polyomino in $\mathcal{P}_{cdm}$ and it glues a new cell to the right of the marked cell in the

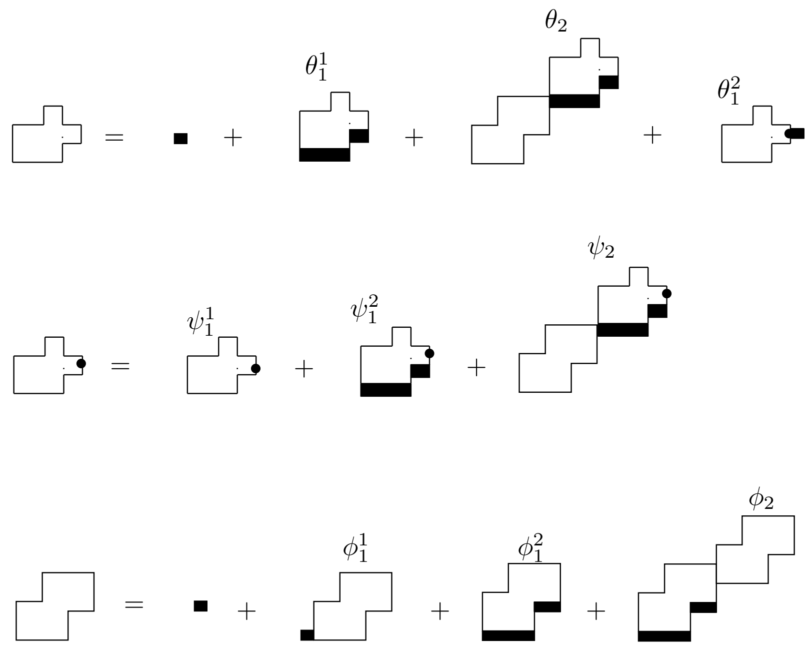

Figure 4.6: The grammar $G_{\mathcal{P}_{cd}}$ for directed convex polyominoes.

polyomino. Finally ψ_1^1 takes a polyomino in $\mathcal{P}_{cd}$ and marks the bottom cell of the last column.

4.2 Linear parameters and q-parameters

Definition 4.2.1. *Let $\mathcal{O}, \mathcal{O}^1, \dots \mathcal{O}^k$ be some classes of combinatorial objects and p a finite parameter on $\mathcal{O}, \mathcal{O}^1, \dots, \mathcal{O}^k$. Let ϕ be an object operation such that $dom(\phi) = \mathcal{O}^1 \times \dots \times \mathcal{O}^k$ and $cod(\phi) = \mathcal{O}$. Then p is said to be a* linear parameter *with respect to ϕ if*

$$p\left(\phi(O_1, \dots, O_k)\right) = \sum_{i=1}^{k} p(O_i) + p(\phi), \tag{4.1}$$

where $(O_1, \dots, O_k) \in dom(\phi)$ and $p(\phi) \in \mathbb{N}$ is a constant.

Definition 4.2.2. *Let $G = \langle \mathbb{O}, \mathbb{E}, \Phi, \mathcal{A} \rangle$ be an object grammar. A parameter p is said to be G-linear if, for all $\mathcal{O} \in \mathbb{O}$, p is linear with respect to each operation $\phi \in \Phi$ such that $cod(\phi) = \mathcal{O}$.*

A G-linear parameter p is called *uniform* if,

$$\forall \phi \in \Phi, \quad p(\phi) = 1$$

and,

$$\forall e \in \bigcup\nolimits_{\mathcal{O} \in \mathbb{O}} \mathcal{E}_{\mathcal{O}}, \quad p(e) = 1.$$

Lemma 4.2.1. *Let p be a G-linear parameter on G. Then, for any object O generated by G with derivation tree T (so that $ev(T) = O$), we have*

$$p(O) = \sum_{x \in T} p(\phi_x), \tag{4.2}$$

where, for x a node of T, ϕ_x is its label.

Proof. The proof can be achieved by recursion on equation (4.1) defining p. $\square$

Example 4.2.1. *Let $G_{\mathcal{D}}$ be the grammar for Dyck paths (see 4.1.2), then the length l of a path is $G_{\mathcal{D}}$-linear. Indeed for every $D_1, D_2 \in \mathcal{D}$,*

$$l(\phi_2(D_1, D_2)) = l(D_1) + l(D_2) + 2.$$

We take here $l(\phi_2) = 2$ since ϕ_2 adds two steps to (D_1, D_2), and $l(.) = 0$ since the empty path has zero length.

Example 4.2.2. *Let $G_{\mathcal{M}}$ be the grammar for Motzkin paths (see 4.1.2), then the length l of a path is $G_{\mathcal{M}}$-linear. Indeed for every $M_1, M_2 \in \mathcal{M}$,*

$$\begin{aligned} l(\phi_1(M_1)) &= l(M_1) + 1 \\ l(\phi_2(M_1, M_2)) &= l(M_1) + l(M_2) + 2 \end{aligned}$$

since ϕ_1 adds one step to M_1 and ϕ_2 adds two steps to (M_1, M_2).

Example 4.2.3. *Let $G_{\mathcal{P}}$ be the grammar for parallelogram polyominoes (see 4.1.2), then the perimeter p of a polyomino is $G_{\mathcal{P}}$-linear. Indeed for every $P_1, P_2 \in \mathcal{P}$ we have*

$$\begin{aligned} p(\phi_1^1(P_1)) &= p(P_1) + 2 \\ p(\phi_1^2(P_1)) &= p(P_1) + 2 \\ p(\phi_2(P_1, P_2)) &= p(P_1) + p(P_2). \end{aligned}$$

In [44] Dutour proved that linear parameters lead to algebraic generating functions for the class $\mathcal{O}$.

Definition 4.2.3. *Let $\mathcal{O}, \mathcal{O}^1, \ldots \mathcal{O}^k$ be some classes of combinatorial objects and q be a parameter on $\mathcal{O}, \mathcal{O}^1, \ldots, \mathcal{O}^k$. Let ϕ be an object operation such that $dom(\phi) = \mathcal{O}^1 \times \ldots \times \mathcal{O}^k$ and $cod(\phi) = \mathcal{O}$. Then q is a q-*parameter *with respect to ϕ if, for all $(O_1, \ldots, O_k) \in dom(\phi)$,*

$$q\left(\phi(O_1, \ldots, O_k)\right) = \sum_{i=1}^{k} q(O_i) + \sum_{i=1}^{k} q_i(\phi) t(O_i) + q(\phi), \tag{4.3}$$

where the $q_i(\phi) \in \mathbb{N}$ for $i = 1 \ldots k$, and $q(\phi) \in \mathbb{N}$ are constants, and t is a parameter on $\mathcal{O}^1, \ldots, \mathcal{O}^k$.

Definition 4.2.4. *Let $G = \langle \mathbb{O}, \mathbb{E}, \Phi, \mathcal{A} \rangle$ be an object grammar. A parameter q is said to be G-q-linear if, for all $\mathcal{O} \in \mathbb{O}$, q is a q-parameter with respect to each operation $\phi \in \Phi$ such that $cod(\phi) = \mathcal{O}$.*

Lemma 4.2.2. *Let q be a G-q-linear and t be a parameter. Then, for any object O generated by G with derivation tree T, we have*

$$q(O) = \sum_{x \in T} \left(\sum_{i=1}^{k(x)} q_i(\phi_x) t(ev(T_{i,x})) + q(\phi_x) \right), \tag{4.4}$$

where, for x a node of T, ϕ_x is its label, and the $T_{i,x}$, $i = 1 \ldots k(x)$, are the subtrees attached to x.

Proof. The proof can be achieved by recursion on equation (4.3) defining q. □

Remark 4.2.1. *If the parameter t is G-linear, we can apply (4.2) to (4.4) and then obtain the following:*

$$q(O) = \sum_{x \in T} \left(\sum_{i=1}^{k(x)} q_i(\phi_x) \sum_{y \in T_{i,x}} t(\phi_y) + q(\phi_x) \right). \tag{4.5}$$

Example 4.2.4. *Let $G_{\mathcal{P}}$ be the grammar for parallelogram polyominoes, then the area a of a polyomino is a q-parameter on $\mathcal{P}$ with respect to the operations ϕ_1^2 and ϕ_2. Indeed for every $P_1, P_2 \in \mathcal{P}$ we have*

$$
\begin{aligned}
a(\phi_1^1(P_1)) &= a(P_1) + 1 \\
a(\phi_1^2(P_1)) &= a(P_1) + nc(P_1) \\
a(\phi_2(P_1, P_2)) &= a(P_1) + a(P_2) + nc(P_1),
\end{aligned}
$$

where the parameter nc counts the number of columns of a polyomino.

From now on, we will only deal with complete and unambiguous object grammars.

4.3 Object grammars and ECO method

In this section we introduce a generalization of succession rules that allows us to prove the following theorem:

Theorem 4.3.1. *Any unidimensional, complete, and unambiguous object grammar with a linear parameter can be represented by an ECO-system with bounded jumps.*

For this part we thus assume that the grammar G is unidimensional. In order to prove the theorem we shall introduce a generic representation of derivation trees in terms of *weighted α-trees* and of a well chosen parameter p' on this class. Then we shall determine an ECO-system describing the growth of this class of trees according to p'.

4.3.1 Jumping succession rules

Recently Ferrari, Pergola, Pinzani, and Rinaldi [52] extended ECO method by allowing the ECO operator to generate objects of different sizes, greater than n, from any object of size n. This naturally leads to the concept of *jumping succession rules.*

Let us fix m integers, $0 \leq i_1 < i_2 \ldots < i_m$. These integers will represent the possible jump lengths. We consider a growth operator of the form $\vartheta : \mathcal{O} \to 2^{\mathcal{O}}$ such that $\vartheta(\mathcal{O}_n) \subseteq 2^{\cup_{j=1}^m \mathcal{O}_{n+i_j}}$, where $n \in \mathbb{N}$. Proposition 1.3.1 generalizes as follows:

Proposition 4.3.1. *If ϑ satisfies, for $n \geq 0$,*

1. *for each $O' \in \mathcal{O}_n$ there exists $O \in \cup_{j=1}^m \mathcal{O}_{n-i_j}$ such that $O' \in \vartheta(O)$, and*

2. *for every $O, O' \in \mathcal{O}$, $\vartheta(O) \cap \vartheta(O') = \emptyset$, whenever $O \neq O'$,*

then the family of sets $\mathcal{F} = \left\{\vartheta(O) : O \in \cup_{j=1}^m \mathcal{O}_{n-i_j}\right\} \cap 2^{\mathcal{O}_n}$ is a partition of $\mathcal{O}_n$.

In this case we call ϑ an ECO operator with jumps. The operator ϑ can be described by means of a generating tree with edges of various lengths, $i_1, \ldots, i_m$. This leads us to the definition of *jumping succession rule* of the form:

$$\Omega = \begin{cases} (a) & \\ (k) & \overset{i_1}{\rightsquigarrow} \quad (e_1^1(k))(e_2^1(k)) \ldots (e_{k_1}^1(k)) \\ & \vdots \\ & \overset{i_m}{\rightsquigarrow} \quad (e_1^m(k))(e_2^m(k)) \ldots (e_{k_m}^m(k)), \end{cases} \tag{4.6}$$

where $k \in \mathbb{N}^+$, a is a constant in $\mathbb{N}^+$, the e_i^j are functions $\mathbb{N}^+ \to \mathbb{N}^+$, and we assume that $k_1 + k_2 + \ldots + k_m = k$ to fit with the consistency principle. The growth of the objects by means of ϑ is described by Ω if, for every object O such that $|\vartheta(O)| = k$ and for all $j = 1 \ldots m$, there are exactly k_j objects $O'_1, \ldots, O'_{k_j}$ that belong to $\vartheta(O)$, have size $|O| + i_j$, and verify $|\vartheta(O'_l)| = e_l^j(k)$ for $1 \leq l \leq k_j$. A succession rule in the form of (1.1) is then a jumping succession rule with $m = 1$ and $i_1 = 1$. Like for succession rules, we denote $\{f_n\}_n$ the sequence defined by a jumping succession rule, where f_n is the number of nodes at level n of the generating tree.

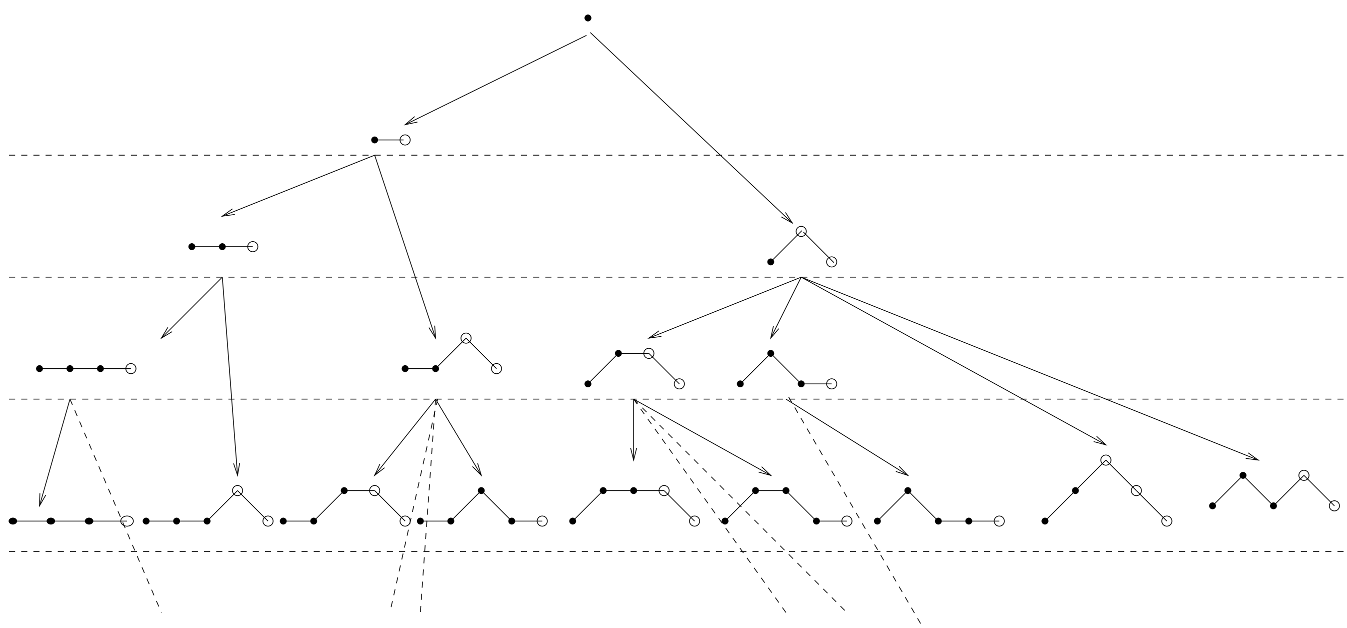

Figure 4.7: The first levels of ϑ

Example 4.3.1. *Let $\mathcal{M}$ be the class of Motzkin paths, and let $M \in \mathcal{M}$. Let us denote by $\ell_d(M)$ the last sequence of fall steps of M. Then $\vartheta(M)$ is the set of Motzkin paths that are obtained*

- *by adding a peak on a point of $\ell_d(M)$; then the length increases by 2;*

- *by adding an horizontal step on a point of $\ell_d(M)$; then the length increases by 1.*

Let $P(M)$ the set of points of $\ell_d(M)$. In particular $|\vartheta(M)| = 2|P(M)|$. Figure 4.7 represents the first levels of the generating tree of ϑ. Replacing each object M by a label $|\vartheta(M)|$, this generating tree is encoded by the jumping succession rule

$$\Omega = \begin{cases} (2) & \\ (2k) & \overset{1}{\rightsquigarrow} (2)(4)\dots(2k) \\ & \overset{2}{\rightsquigarrow} (4)(6)\dots(2k+2). \end{cases}$$

The first levels of the generating tree of the rule are represented in Figure 4.8. This rule defines the sequence $1, 1, 2, 4, 9, 21, \dots$ of Motzkin numbers and is equivalent to the rule of Example 2.4.1, whose generating function is $\frac{1-x-\sqrt{1-2x-3x^2}}{2x^2}$.

4.3.2 Generic derivation trees

Let $\mathcal{O}$ be the class generated by G, Φ the set of operations of the grammar, and p a linear parameter on $\mathcal{O}$. As usual, let $\mathcal{O}_n$ denote the subset of objects of size n, that is $\mathcal{O}_n = \{O \in \mathcal{O} : p(O) = n\}$.

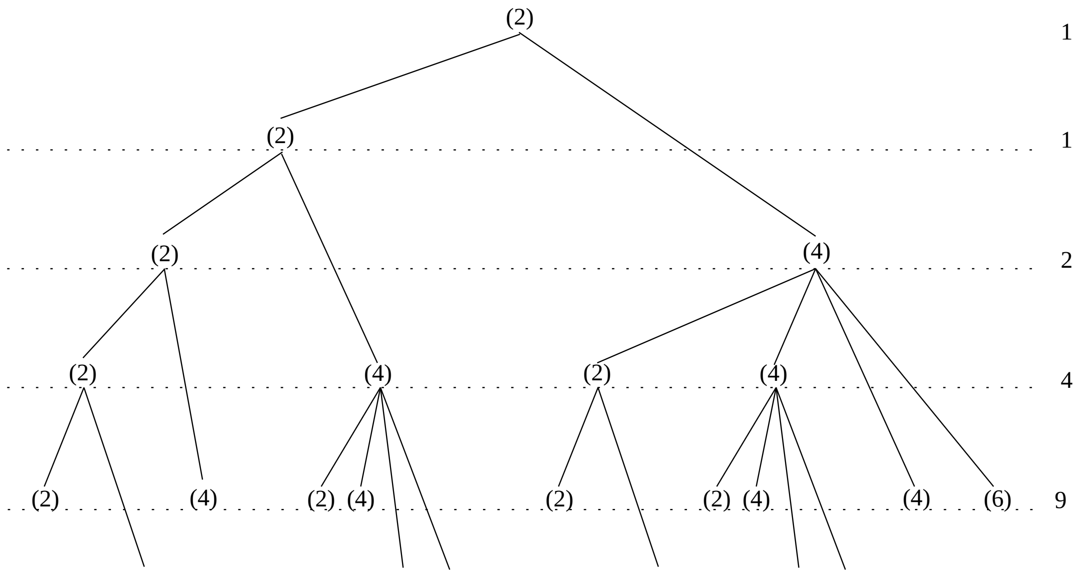

Figure 4.8: The first levels of the generating tree associated with the rule in Example 4.3.1.

For a fixed positive integer d, let Φ_j, $0 < j \leq d$, denote the subset of Φ with operations of degree j. Let Φ_0 denote the set of terminal objects of the grammar G. In order to give a uniform presentation of the ECO construction associated to a grammar, we introduce a generic model of derivation tree.

Definition 4.3.1. *Let d be a fixed non-negative integer and*

$$\alpha = (\alpha_0, \alpha_1, \ldots, \alpha_d),\ \alpha_i \in \mathbb{N}.$$

An α-tree *is a labelled tree with nodes of degree at most d and such that each node of degree j has a color $i \in \{1, 2, \ldots, \alpha_j\}$. Given a weight $w_{ij} \in \mathbb{N}$ for each node of color i and degree j, the associated* weighted α-tree *has labels of the form (i, w_{ij}) on nodes of degree j.*

There is a simple bijection from $\mathcal{T}_G$, the set of derivation trees of G, to $\mathcal{T}_G^{w\alpha}$, the set of weighted α-trees, where $\alpha = (|\Phi_0|, |\Phi_1|, \ldots, |\Phi_d|)$ and $w_{ij} = p(\phi_j^i)$ for all $i = 1 \ldots |\Phi_j|$, $j = 0 \ldots d$. For any $T \in \mathcal{T}_G$, the corresponding tree is obtained by replacing each label ϕ_j^i in the tree T by the label $(i, p(\phi_j^i))$, and vice versa.

From the previous statements one can easily adapt Proposition 4.1.1 to the class $\mathcal{T}_G^{w\alpha}$:

Proposition 4.3.2. *If G is a complete, unambiguous, and unidimensional object grammar generating the class $\mathcal{O}$, then the function $ev : \mathcal{T}_G^{w\alpha} \to \mathcal{O}$ is a bijection.*

Definition 4.3.2. *Let $T \in \mathcal{T}_G^{w\alpha}$. For any $x \in T$ denote $l(x) = (co(x), w(x))$ the label of x. Then $p'(T) = \sum_{x \in T} w(x)$.*

Then, from Proposition 4.3.2, Definition 4.3.2, and Lemma 4.2.1 we have the following:

Lemma 4.3.1. *Let $T \in \mathcal{T}_G^{w\alpha}$, then $p'(T) = p(ev(T))$.*

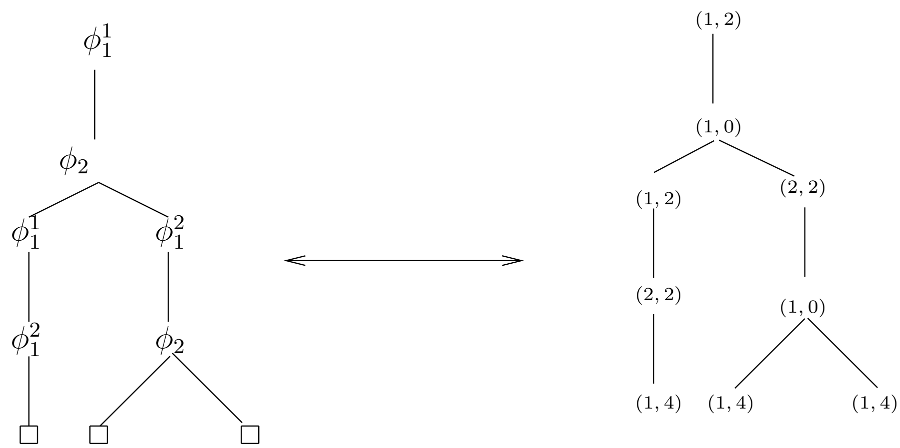

Figure 4.9: A derivation tree of the grammar $G_{\mathcal{P}}$ and the corresponding weighted α-tree.

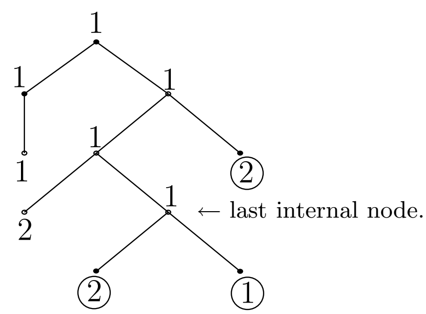

Figure 4.10: The active sites in a $(2, 1, 1)$-tree.

Example 4.3.2. *Figure 4.9 shows a derivation tree of the grammar $G_{\mathcal{P}}$ of Subsection 4.1.2 and the corresponding weighted α-tree, where the weight is defined according to the perimeter, with $p(\phi_1^1) = p(\phi_1^2) = 2$ and $p(\phi_2) = 0$. The sum p' of the weights is then equal to* 20.

In order to complete our program, we must determine an ECO construction for the class $\mathcal{T}_G^{w\alpha}$ according to p'. For this purpose we need to extend slightly the class $\mathcal{T}_G^{w\alpha}$ including the empty tree of size 0, denoted by ε, that will correspond to the root of the generating tree associated with the ECO construction. This root produces the initial α_0 leaves in the generating tree. We remark that, by extending the class $\mathcal{T}_G^{w\alpha}$ to include ε, the generating function of such a class increases by one. We will first show the ECO construction in the simpler case of uniform parameters, and then we will extend it to the general case of linear parameters.

4.3.3 The uniform parameter case

For the case where the parameter is uniform, given an object grammar and the resulting $\mathcal{T}_G$, the problem reduces to determining an ECO-system for the associated class $\mathcal{T}_G^{\alpha}$ of unweighted α-trees according to p', which in this case becomes the number of nodes of a tree in

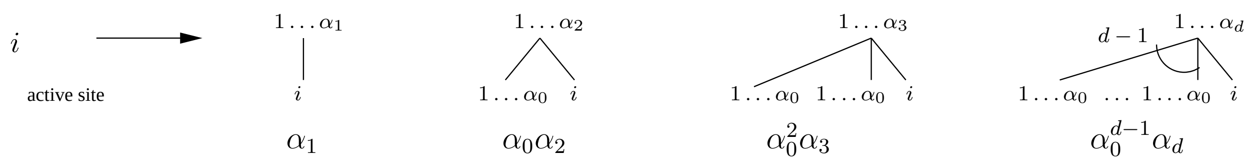

Figure 4.11: A graphical representation of the possible actions of the operator ϑ_1 on an active site of color i.

$\mathcal{T}_G^\alpha$. Observe that these trees are unweighted since the parameter on the objects is uniform. In order to do this we generalize a construction that was proposed in [6] for plane trees.

For any $T \in \mathcal{T}_G^\alpha$, consider all leaves following the last internal node in the preorder traversal, and call them the *active sites* of T. In Figure 4.10 we have marked the active sites of a $(2,1,1)$-tree. Let $\mathcal{T}_n^\alpha$ be the set of trees in $\mathcal{T}_G^\alpha$ having exactly n nodes. Let ϑ_1 denote the operator from $\mathcal{T}_n^\alpha$ to $2^{\cup_{i=1}^d \mathcal{T}_{n+i}^\alpha}$, performing the following transformations on T:

i) if T is the empty tree, then ϑ_1 produces α_0 trees, namely for each color $i = 1, \ldots, \alpha_0$, the tree reduced to a leaf with color i.

ii) otherwise, for each active site A of T with color i, $1 \le i \le \alpha_0$, and for each $j \in \{1 \ldots d\}$, ϑ_1 produces, j levels below, a tree with j new sons attached to A: the rightmost is colored i, and the remaining $j-1$ can be colored $1, \ldots, \alpha_0$. At this stage A is an internal node of degree j and can be colored in α_j ways (see Figure 4.11). The number of trees generated by ϑ_1, through this transformation, is then equal to $\sum_{j=1}^d \alpha_0^{j-1}\alpha_j$.

In Figure 4.12 has been developed, through ϑ_1, the last active site (the first leaf in preorder traversal) of the tree in Figure 4.10. Let $c = \sum_{j=1}^d \alpha_0^{j-1}\alpha_j$. Let us suppose that T has k active sites. Then, from the construction, ϑ_1 produces kc trees, among which $(\alpha_0^{j-1}\alpha_j)k$ lie j levels below in the generating tree, for $j = 1 \ldots d$.

Theorem 4.3.2. *The system $\Sigma = (\mathcal{T}_G^\alpha, p', \vartheta_1, \Omega_1)$ is an ECO-system, where:*

$$\Omega_1 = \begin{cases} (\alpha_0) & & & & & \\ (\alpha_0) & \overset{1}{\rightsquigarrow} & (c)^{\alpha_0} & & & \\ (kc) & \overset{1}{\rightsquigarrow} & (c)^{\alpha_1} & \ldots & ((k-1)c)^{\alpha_1} & (kc)^{\alpha_1} \\ & \overset{2}{\rightsquigarrow} & (2c)^{\alpha_0\alpha_2} & \ldots & (kc)^{\alpha_0\alpha_2} & ((k+1)c)^{\alpha_0\alpha_2} \\ & \vdots & \vdots & \vdots & \vdots & \vdots \\ & \overset{d-1}{\rightsquigarrow} & ((d-1)c)^{\alpha_0^{d-2}\alpha_{d-1}} & \ldots & ((k+d-3)c)^{\alpha_0^{d-2}\alpha_{d-1}} & ((k+d-2)c)^{\alpha_0^{d-2}\alpha_{d-1}} \\ & \overset{d}{\rightsquigarrow} & (dc)^{\alpha_0^{d-1}\alpha_d} & \ldots & ((k+d-2)c)^{\alpha_0^{d-1}\alpha_d} & ((k+d-1)c)^{\alpha_0^{d-1}\alpha_d} \end{cases}$$

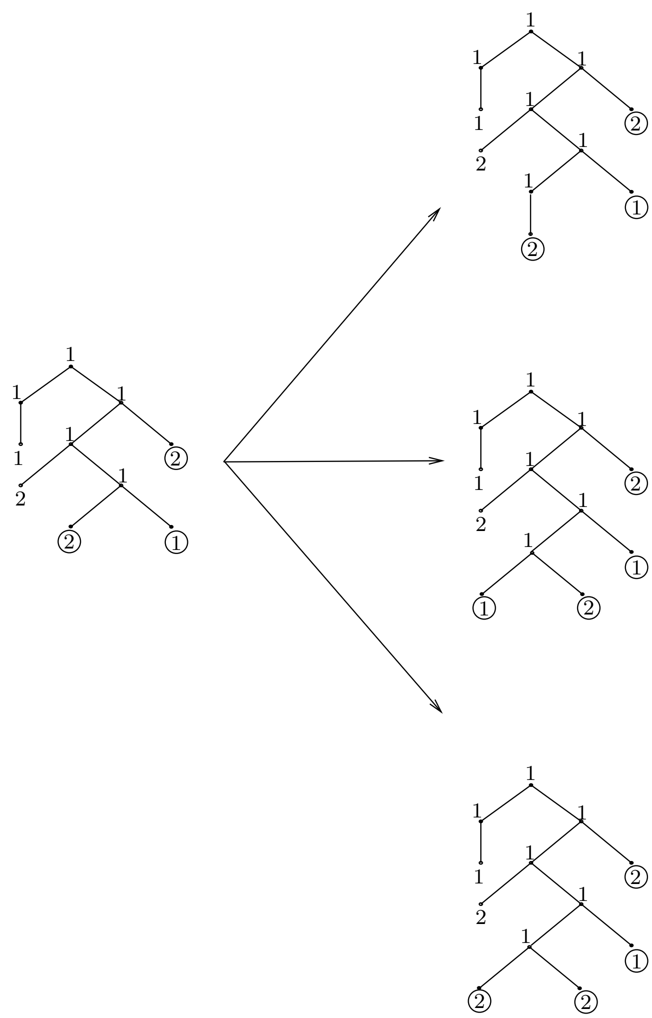

Figure 4.12: The trees obtained from a $(2, 1, 1)$-tree by developing one of its active sites.

Proof. The operator ϑ_1 satisfies the conditions of Proposition 4.3.1.

1. *for each $T' \in \mathcal{T}_n^\alpha$ there is a $T \in \cup_{j=1}^m \mathcal{T}_{n-j}^\alpha$ such that $T' \in \vartheta_1(T)$.* To see this let l be the last internal node of T' in the preorder traversal; T is the tree obtained from T' where the subtree having l as root is replaced with a leaf having the same color as the rightmost son of l.

2. *for each $T \in \mathcal{T}_n^\alpha$, $T' \in \mathcal{T}_m^\alpha$ such that $T \neq T'$, $\vartheta_1(T) \cap \vartheta_1(T') = \emptyset$.* This is easily deduced from the construction. $\square$

Now we will consider the generating function of Ω_1. We recall that $f_{\Omega_1} = \sum_{n\geq 0} f_n x^n$, where n denotes the level in the generating tree of Ω_1 and f_n is the number of labels at that level. Let $g_{\Omega_1'}(x) = \sum_{n\geq 0} g_n x^n$ denote the generating function of

$$\Omega_1' = \left\{ \begin{array}{llllll} (1) & & & & & \\ & & & & & \\ (k) & \stackrel{1}{\rightsquigarrow} & (1)^{\alpha_1} & \dots & (k-1)^{\alpha_1} & (k)^{\alpha_1} \\ & \stackrel{2}{\rightsquigarrow} & (2)^{\alpha_0\alpha_2} & \dots & (k)^{\alpha_0\alpha_2} & (k+1)^{\alpha_0\alpha_2} \\ & \vdots & \vdots & \vdots & \vdots & \vdots \\ & \stackrel{d-1}{\rightsquigarrow} & (d-1)^{\alpha_0^{d-2}\alpha_{d-1}} & \dots & (k+d-3)^{\alpha_0^{d-2}\alpha_{d-1}} & (k+d-2)^{\alpha_0^{d-2}\alpha_{d-1}} \\ & \stackrel{d}{\rightsquigarrow} & (d)^{\alpha_0^{d-1}\alpha_d} & \dots & (k+d-2)^{\alpha_0^{d-1}\alpha_d} & (k+d-1)^{\alpha_0^{d-1}\alpha_d}, \end{array} \right.$$

the succession rule obtained from Ω_1 by eliminating the constant c from each label and by choosing (1) as axiom. Changing the axiom of Ω_1 into (1) corresponds to descend one level in the generating tree of Ω_1. Thus

$$f_{\Omega_1}(x) = 1 + x\alpha_0 g_{\Omega_1'}(x). \tag{4.7}$$

Now let $g_{n,k}$ be the number of nodes at level n having label k, and

$$g_{\Omega_1'}(x,y) = \sum_{n\geq 0, k\geq 1} g_{n,k}x^n y^k.$$

Thus $g_{\Omega_1'}(x) = g_{\Omega_1'}(x,1)$. From rule Ω_1', a node labelled (k) produces, j levels below, $(\alpha_0^{j-1}\alpha_j)k$ nodes, for $j = 1, \dots, d$. Among these, $\alpha_0^{j-1}\alpha_j$ are labelled (k'), for $k' = j \dots k+j-1$. Then we have the following:

$$\begin{aligned} g_{\Omega_1'}(x,y) &= y + \sum_{n\geq 0, k\geq 1} g_{n,k}x^n \sum_{j=1}^{d} \alpha_0^{j-1}\alpha_j \; x^j \; (y^j + y^{j+1} + \dots + y^{k+j-1}) \\ &= y + \sum_{n\geq 0, k\geq 1} g_{n,k}x^n \sum_{j=1}^{d} \alpha_0^{j-1}\alpha_j \; x^j \; \frac{y^j - y^{k+j}}{1-y}. \end{aligned}$$

Consequently,

$$\begin{aligned} g_{\Omega_1'}(x,y) \;\; = \;\; & y \; + \; \alpha_1 x \textstyle\sum_{n\geq 0, k\geq 1} g_{n,k}x^n \; \frac{y-y^{k+1}}{1-y} \; + \; \alpha_0\alpha_2 x^2 \textstyle\sum_{n\geq 0, k\geq 1} g_{n,k}x^n \; \frac{y^2-y^{k+2}}{1-y} \; + \\ & \dots + \; \alpha_0^{d-1}\alpha_d x^d \textstyle\sum_{n\geq 0, k\geq 1} g_{n,k}x^n \; \frac{y^d-y^{k+d}}{1-y}. \end{aligned}$$

Then $g_{\Omega_1'}(x,y)$ satisfies the recurrence

$$g_{\Omega_1'}(x,y) = y + \left(\alpha_1 x\frac{y}{1-y} + \alpha_0\alpha_2 x^2\frac{y^2}{1-y} + \dots + \alpha_0^{d-1}\alpha_d x^d \; \frac{y^d}{1-y}\right)\left(g_{\Omega_1'}(x,1) - g_{\Omega_1'}(x,y)\right).$$

Let us introduce $h(x,y)$ so that

$$h(x,y) = \left(1 - y + \alpha_1 xy + \alpha_0\alpha_2 x^2y^2 + \dots + \alpha_0^{d-1}\alpha_d x^d y^d\right),$$

then

$$g_{\Omega_1'}(x,y)h(x,y) = y(1-y) + g_{\Omega_1'}(x,1)(h(x,y) - 1 + y).$$

Using the kernel method we have

$$g_{\Omega_1'}(x,1) = y_0(x), \tag{4.8}$$

where $y_0(x)$ is the unique formal power series satisfying $h(x,y)=0$, that is to say

$$y = 1 + \alpha_1 xy + \alpha_0\alpha_2 x^2y^2 + \ldots + \alpha_0^{d-1}\alpha_d x^d y^d.$$

Then from (4.7) and (4.8) we obtain

$$f_{\Omega_1}(x) = 1 + x\alpha_0 y_0.$$

Example 4.3.3. *Let us consider the class of* $(1,2,2)$*-trees. Then* c*, defined as* $\sum_{j=1}^{2}\alpha_0^{j-1}\alpha_i$*, is equal to* 4*. Consequently the rule* Ω_1 *for the class of* $(1,2,2)$*-trees, extended with the empty tree* ε*, is*

$$\Omega_1 = \begin{cases} (1) & \\ & \\ (1) & \overset{1}{\rightsquigarrow} \quad (4) \\ & \\ (4k) & \overset{1}{\rightsquigarrow} \quad (4)^2 \ldots (4(k-1))^2(4k)^2 \\ & \overset{2}{\rightsquigarrow} \quad (8)^2 \ldots (4k)^2(4(k+1))^2. \end{cases}$$

Here $h(x,y) = 1 - y + 2xy + 2x^2y^2$ *and the generating function is* $1 + \frac{1-2x-\sqrt{((2x-1)^2-8x^2)}}{4x}$.

Example 4.3.4. *The rule* Ω_1 *for the class of* $(1,0,1)$*-trees, extended with* ε*, is*

$$\Omega_1 = \begin{cases} (1) & \\ & \\ (1) & \overset{1}{\rightsquigarrow} \quad (1) \\ & \\ (k) & \overset{2}{\rightsquigarrow} \quad (2)\ldots(k)(k+1). \end{cases}$$

Here $h(x,y) = 1 - y + x^2y^2$ *and the generating function is* $1 + \frac{1-\sqrt{1-4x^2}}{2x}$.

Example 4.3.5. *The rule* Ω_1 *for the class of* $(1,1,1)$*-trees, extended with* ε*, is*

$$\Omega_1 = \begin{cases} (1) & \\ & \\ (1) & \overset{1}{\rightsquigarrow} \quad (2) \\ & \\ (2k) & \overset{1}{\rightsquigarrow} \quad (2)\ldots(2(k-1))(2k) \\ & \overset{2}{\rightsquigarrow} \quad (4)\ldots(2k)(2(k+1)). \end{cases}$$

Here $h(x,y) = 1 - y + xy + x^2y^2$ *and the generating function is* $1 + \frac{1-x-\sqrt{-3x^2-2x+1}}{2x}$.

4.3.4 The linear parameter case.

In this subsection we extend the statement of Theorem 4.3.2 to the case of linear parameters. Recall that $p'(T) = \sum_{x \in T} w(x)$, so that in general adding a node changes the parameter by $w(x)$ instead of 1. We adapt the construction of the uniform case by "playing" on the jumps of the associated generating tree: we define an operator ϑ_2 in the same way as the operator ϑ_1 of Subsection 4.3.3, but when the new operator attaches j new sons to an active site A, the resulting tree is produced at a level that depends on the exact colors of the nodes. More precisely, A becomes an internal node of degree j, with label $(i_j, w_{i_j j})$, where $i_j \in \{1 \dots \alpha_j\}$; the rightmost son of A receives the same color of A, and the other $j-1$ sons are colored $(i_{0,t}, w_{i_{0,t}0})$, where $i_{0,t} \in \{1 \dots \alpha_0\}$ for $t = 1 \dots j-1$. Then, the jump produced in the generating tree has length $w_{i_j j} + \sum_{t=1}^{j-1} w_{i_{0,t}0}$, or, in terms of the $p(\phi^{i_j})$, $p(\phi_j^{i_j}) + \sum_{t=1}^{j-1} p(\phi_0^{i_{0,t}})$. The result is the following succession rule where, for clarity, we keep writing the jumps in terms of $p(\phi_j^i)$ instead of w_{ij}.

$$\Omega_2 = \left\{ \begin{array}{llllll} (\alpha_0) & & & & & \\ (\alpha_0) & \xrightarrow{p(\phi_0^{i_0})} & (c) & & & i_0 = 1 \dots \alpha_0 \\ (kc) & \xrightarrow{p(\phi_1^{i_1})} & (c) \; \dots & ((k-1)c) & (kc) & i_1 = 1 \dots \alpha_1 \\ & \xrightarrow{p(\phi_2^{i_2}) + p(\phi_0^{i_{0,1}})} & (2c) \; \dots & (kc) & ((k+1)c) & \left\{ \begin{array}{l} i_2 = 1 \dots \alpha_2 \\ i_{0,1} = 1 \dots \alpha_0 \end{array} \right. \\ & \vdots & \vdots \quad \vdots & \vdots & \vdots & \\ & \xrightarrow{p(\phi_j^{i_j}) + \sum_{t=1}^{j-1} p(\phi_0^{i_{0,t}})} & (jc) \; \dots & ((k+j-2)c) & ((k+j-1)c) & \left\{ \begin{array}{l} i_j = 1 \dots \alpha_j \\ i_{0,t} = 1 \dots \alpha_0 \end{array} \right. \\ & \vdots & \vdots \quad \vdots & \vdots & \vdots & \\ & \xrightarrow{p(\phi_d^{i_d}) + \sum_{t=1}^{d-1} p(\phi_0^{i_{0,t}})} & (dc) \; \dots & ((k+d-2)c) & ((k+d-1)c) & \left\{ \begin{array}{l} i_d = 1 \dots \alpha_d \\ i_{0,t} = 1 \dots \alpha_0 \end{array} \right. \end{array} \right.$$

As a consequence, the following theorem generalizes Theorem 4.3.2:

Theorem 4.3.3. *The system $\Sigma = (\mathcal{T}_G^{w\alpha}, p', \vartheta_2, \Omega_2)$ is an ECO-system.*

The calculus of f_{Ω_2}, the generating function of Ω_2, is analogous, through more complicated, to that in Subsection 4.3.3. We have

$$f_{\Omega_2}(x) = 1 + \sum_{i_0=1}^{\alpha_0} x^{p(\phi_0^{i_0})} g_{\Omega_2'}(x),$$

where Ω_2' is obtained from Ω_2 by eliminating the constant c from each label and by choosing (1) as axiom. Now, by using the same arguments as for Ω_1' (see Subsection 4.3.3), we obtain

$$g_{\Omega_2'}(x,y) = y + \sum_{n \geq 0, k \geq 1} g_{n,k} x^n \left(\sum_{i_1} x^{p(\phi_1^{i_1})} \frac{y - y^{k+1}}{1-y} + \sum_{i_2} \sum_{i_0} x^{(p(\phi_2^{i_2}) + p(\phi_0^{i_{0,1}}))} \frac{y^2 - y^{k+2}}{1-y} + \right.$$

$$+\ldots+\sum_{i_d}\sum_{i_{0,1}i_{0,2}\ldots i_{0,d-1}} x^{(p(\phi_d^{i_d})+\sum_{t=1}^{d-1}p(\phi_0^{i_{0,t}}))}\,\frac{y^d-y^{k+d}}{1-y}\Bigg),$$

where $i_j=1\ldots\alpha_j$, and $i_{0,t}=1\ldots\alpha_0$. Thus

$$g_{\Omega_2'}(x,y)=y+\Bigg(\sum_{i_1}x^{p(\phi_1^{i_1})}\,\frac{y}{1-y}+\sum_{i_2}\sum_{i_0}x^{(p(\phi_2^{i_2})+p(\phi_0^{i_{0,1}}))}\,\frac{y^2}{1-y}+$$

$$+\ldots+\sum_{i_d}\sum_{i_{0,1}i_{0,2}\ldots i_{0,d-1}} x^{(p(\phi_d^{i_d})+\sum_{t=1}^{d-1}p(\phi_0^{i_{0,t}}))}\,\frac{y^d}{1-y}\Bigg)(g_{\Omega_2'}(x)-g_{\Omega_2'}(x,y)),$$

Consequently, by applying the kernel method, we obtain

$$f_{\Omega_2}(x)=1+\sum_{i=1}^{\alpha_0}x^{p(\phi_0^i)}y_0(x),$$

where $y_0(x)$ is the unique formal power series satisfying the equation

$$y=1+\sum_{j=1}^{d}\sum_{i_j}\sum_{i_{0,1},\ldots,i_{0,j-1}}x^{p(\phi_j^{i_j})+\sum_{t=1}^{j-1}p(\phi_0^{i_{0,t}})}y^j,$$

where $i_j=1\ldots\alpha_j$, and $i_{0,t}=1\ldots\alpha_0$.

As a particular case, if $p(\phi_j^{i_j})=\begin{cases}1 & \text{if} \quad j\geq 1\\ 0 & \text{if} \quad j=0\end{cases}$ then the rules Ω_2 becomes a simple succession rule

$$\Omega_3=\begin{cases}(\alpha_0) & & \\ (\alpha_0) & \overset{0}{\rightsquigarrow} & (c)^{\alpha_0} \\ (kc) & \overset{1}{\rightsquigarrow} & \begin{array}{llll}(c)^{\alpha_1} & \ldots & ((k-1)c)^{\alpha_1} & (kc)^{\alpha_1}\\ (2c)^{\alpha_0\alpha_2} & \ldots & (kc)^{\alpha_0\alpha_2} & ((k+1)c)^{\alpha_0\alpha_2}\\ \vdots & \vdots & \vdots & \vdots\\ (dc)^{\alpha_0^{d-1}\alpha_d} & \ldots & ((k+d-2)c)^{\alpha_0^{d-1}\alpha_d} & ((k+d-1)c)^{\alpha_0^{d-1}\alpha_d}.\end{array}\end{cases}$$

whose generating function is $f_{\Omega_3}(x)=1+y_0(x)$, where $y_0(x)$ satisfies

$$y=1+\sum_{j=1}^{d}\alpha_0^{j-1}\alpha^j xy^j.$$

We remark that, in rule Ω_3, the notation $(\alpha_0)\overset{0}{\rightsquigarrow}(c)^{\alpha_0}$ means that at level 0 of the generating tree we have a node labelled by (α_0) and we have α_0 nodes labelled by (c), each of which produces c nodes at level 1.

The conclusion of this section is that the rule Ω_2 defines the sequence $|\mathcal{O}_0|+1, \{|\mathcal{O}_n|\}_{n\geq 1}$ when p is a linear parameter. We remark that the number of objects with size 0 is increased by one because of the empty tree we add in order to construct the class $\mathcal{T}_G^{w\alpha}$. The bijection between the class of weighted α-trees and the class of objects produced by the object grammar, $\mathcal{O}$, allows us to translate the ECO construction from the first class to the second one.

Example 4.3.6. *Let $G_{\mathcal{D}}$ be the grammar for Dyck paths defined in Subsection 4.1.2 and p be the semi-length of a Dyck path. Then the class of derivation trees of the grammar is in bijection with the class of weighted $(1,0,1)$-trees with $w_{10}=0$ and $w_{12}=1$. Indeed, the grammar G_D has only one terminal object, with semi-length 0, and one operation ϕ_2 of degree 2, such that $p(\phi_2)=1$ (see Example 4.2.1). The operator ϑ_2 determines a construction for the weighted $(1,0,1)$-trees according to p' and, consequently, it determines a construction for $\mathcal{D}$ according to p. Figure 4.13 shows the first levels of these constructions. The empty tree corresponds to an object of size 0 in the class of Dyck paths and it is still represented by ε. This construction, determined by ϑ_2, leads to the following succession rule:*

$$\Omega_2^D = \begin{cases} (1) & \\ (1) \stackrel{0}{\rightsquigarrow} (1) & \\ (k) \stackrel{1}{\rightsquigarrow} (2)(3)\dots(k+1). & \end{cases}$$

The generating function of Ω_2^D is $1+\frac{1-\sqrt{1-4x}}{2x}$, where $\frac{1-\sqrt{1-4x}}{2x}$ defines the sequence of Catalan numbers.

Example 4.3.7. *Let $G_{\mathcal{M}}$ be the grammar for Motzkin paths defined in Subsection 4.1.2 and p be the length of a Motzkin path, then $p(.)=0$, $p(\phi_1)=1$, and $p(\phi_2)=2$ (see Example 4.2.2). Therefore the class of derivation trees of the grammar $G_{\mathcal{M}}$ is in bijection with the class of weighted $(1,1,1)$-trees with $w_{10}=0$, $w_{11}=1$, and $w_{12}=2$. The operator ϑ_2 determines a construction the class of weighted $(1,1,1)$-trees according to p'. The first levels of the construction are depicted in Figure 4.14 and the corresponding succession rule is*

$$\Omega_2^M = \begin{cases} (1) & \\ (1) \stackrel{0}{\rightsquigarrow} (2) & \\ (2k) \stackrel{1}{\rightsquigarrow} (2)(4)\dots(2k) & \\ \qquad\; \stackrel{2}{\rightsquigarrow} (4)(6)\dots(2k+2), & \end{cases}$$

already introduced in Example 4.3.1. The generating function of Ω_2^M is $1+\frac{1-x-\sqrt{1-2x-3x^2}}{2x^2}$ where $\frac{1-x-\sqrt{1-2x-3x^2}}{2x^2}$ defines the sequence of Motzkin numbers.

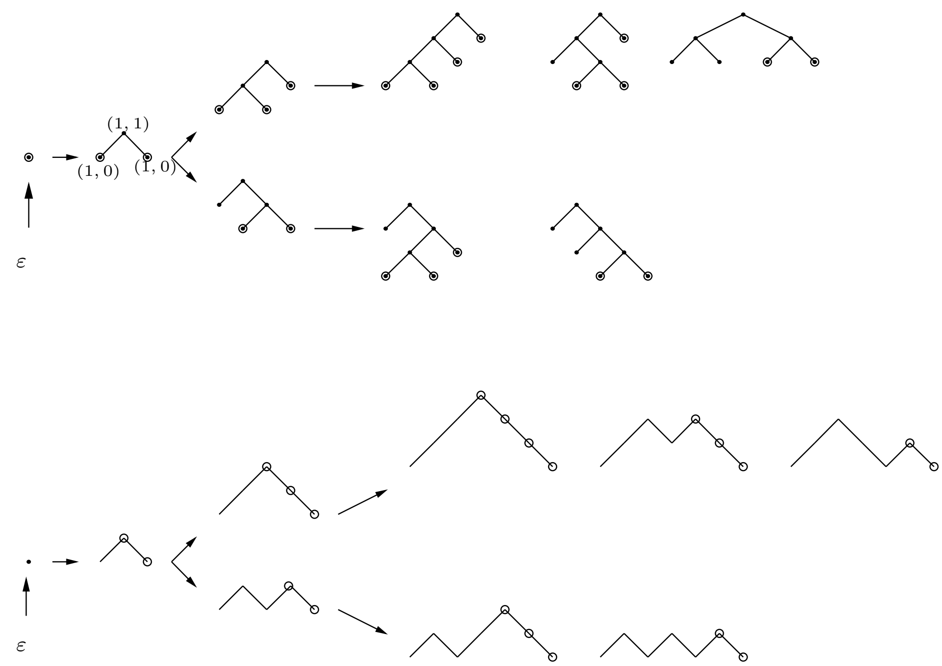

Figure 4.13: The constructions for complete binary trees and for Dyck paths. To simplify, the labels are depicted only on the first tree. The circles on the objects represent their active sites.

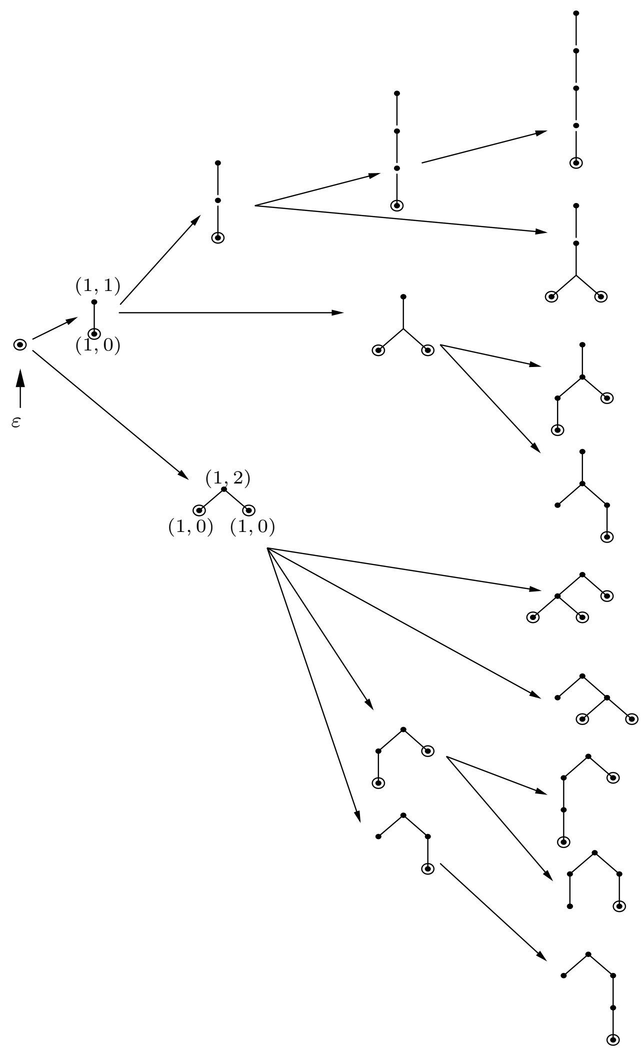

Figure 4.14: The construction for the class of $(1, 1, 1)$-trees.

4.4 The multidimensional case

In this context we need to introduce rules with multiple labels. Similar rules have been already treated by Guibert [66] for describing permutations with forbidden sequences, and by Guibert, Pergola, Pinzani [67] to deal with vexillary involutions. A *succession rule with multiple labels* Ω is a *system* $((\vec{a}), \mathcal{P})$, consisting of an axiom $(\vec{a})$ and a set $\mathcal{P}$ of *productions* or *rewriting rules* defined on a set of labels $M^m \subset \mathbb{N}^m$:

$$\Omega = \begin{cases} (\vec{a}) \\ (\vec{k}) \rightsquigarrow (e_1(\vec{k}))(e_2(\vec{k}))\dots(e_{d(\vec{k})}(\vec{k})), \quad \text{for all } \vec{k} \in M^m, \end{cases} \tag{4.9}$$

where $\vec{a} = (a_1, \dots, a_m) \in M^m$ is a fixed constant, the e_i are functions $M^m \to M^m$, and d is a function $M^m \to \mathbb{N}^+$ giving the number of elements produced by a label $(\vec{k})$. (These are therefore analogs of pseudo-succession rules. In this context a natural extension of the consistency principle of succession rules could be that $d(\vec{k}) = k_1 + \cdots + k_m$, but it is simpler not to impose it.)

Let $G = \langle \{\mathcal{O}^1, \mathcal{O}^2, \dots, \mathcal{O}^m\}, \mathbb{E}, \Phi, \mathcal{A} \rangle$ be a grammar with dimension $m > 1$, and p be a G-linear parameter on G. As for the unidimensional case we introduce a class of *weighted* α-trees and a finite parameter p' on it, such that the number of trees with size n is equal to $|\mathcal{A}|_n$. Finally we determine an ECO-system describing the growth for this class of trees according to p'.

Passing to the multidimensional case, the definition of *weighted* α-trees becomes slightly different with respect to that in Section 4.3, due to the fact that we are dealing with trees with nodes belonging to m different classes. From now on, without loss of generality, we suppose that $\mathcal{A} = \mathcal{O}_1$.

Definition 4.4.1. *Let $m \in \mathbb{N}^+$ and $M = \{1, \dots, m\}$. Let us fix*

$$\alpha : \alpha_w^i \in \mathbb{N}, \text{ with } i \in M \text{ and } w \in M^*.$$

A weighted *α-tree is a labeled tree whose nodes belong to m different classes. Each node of class i with l sons of respective classes $w_1, \dots, w_l$ has a color*

$$k \in \{(i, 1, u_w^i(1)), (i, 2, u_w^i(2)), \dots (i, \alpha_w^i, u_w^i(\alpha_w^i))\},$$

where $w = w_1 \dots w_l$, and where $u_w^i(j)$ is the weight associated to the j-th possible color, for $j = 1, \dots, \alpha_w^i$. In particular such a node can exists only if $\alpha_w^i > 0$.

Let $i \in M$ and $w \in M^*$. We denote Φ_w^i, the subset of Φ with operations going from $\mathcal{O}^{w_1} \times \mathcal{O}^{w_2} \times \cdots \times \mathcal{O}^{w_l}$ to $\mathcal{O}^i$, w_j being the j-th letter of w and l being its length. Let $\mathcal{T}_G$ be the set of derivation trees of G. There is a simple bijection from $\mathcal{T}_G$ to $\mathcal{T}_G^{w\alpha}$, the set of weighted α-trees where $\alpha_w^i = |\Phi_w^i|$ for $i \in M$ and $w \in M^*$, and $u_w^i(j) = p(\phi_w^i(j))$ for all $j = 1 \dots |\Phi_w^i|$. For any $T \in \mathcal{T}_G$, the corresponding tree T' is obtained by replacing each label $\phi_w^i(j)$ in the tree T by the label $(i, j, p\left(\phi_w^i(j)\right))$, and vice versa. In particular we have that the root of T' is obtained by replacing the label $\phi_w^1(j)$ of the root of T by the label $(1, j, p\left(\phi_w^1(j)\right))$, and vice versa. From the previous assertions we can extend the statements of Subsection 4.3.2:

Proposition 4.4.1. *If G is a complete and unambiguous object grammar generating the class $\mathcal{O}$, then the function $ev : \mathcal{T}_G^{w\alpha} \to \mathcal{O}$ is a bijection.*

Definition 4.4.2. *Let $T \in \mathcal{T}_G^{w\alpha}$. For any $x \in T$ denote $l(x) = (cl(x), co(x), u(x))$ the label of x. We define*

$$p'(T) = \sum_{x \in T} u(x).$$

Then, from Proposition 4.4.1, Definition 4.4.2, and Lemma 4.2.1 we obtain the following

Lemma 4.4.1. *Let $T \in \mathcal{T}_G^{w\alpha}$, then $p'(T) = p(ev(T))$.*

In order to construct the new class $\mathcal{T}_G^{w\alpha}$ according to p', we define a new ECO operator ϑ. This operator is a generalization of the operator ϑ_2 introduced in Subsection 4.3.4. To this purpose, we make the simplifying assumption that for all $i \in \{1, \dots, m\}$, there is exactly one terminal object of class i. In other terms, we assume $\alpha_\varepsilon^i = 1$ for all $i \in \{1, \dots, m\}$. This assumption could be raised but this would require more complicated notations (see however the discussion at the end of the section).

In all the following definitions the preorder is implicitly assumed on nodes of a tree. Let T_1 denote the tree reduced to one leaf of class 1, which will be our initial tree. Let $T \in \mathcal{T}_G^{w\alpha}$. In order to give a new definition of the active sites of T, we introduce the concept of *dominant internal node*. We recall that the root vertex of T is of class 1. We define inductively the sequence of dominant internal nodes of type $i = 1, \dots, m$ of T: let $\ell_0(T)$ denote the root of T.

- For each $i = 1, \dots, m$, the dominant internal node $\ell_i(T)$ of class i is the last internal node of class i in the subtree rooted at $\ell_{i-1}(T)$, if there is such an internal node. Otherwise $\ell_i(T) = \ell_{i-1}(T)$.

From the notion of dominant internal node we can define the set of active sites of class i:

- The set $L_i(T)$ of active sites of class i consists of all the leaves that are after $\ell_i(T)$ in the subtree rooted at $\ell_{i-1}(T)$. Observe that $|L_1(T_1)| = 1$ and $|L_i(T_1)| = 0$ for $i = 2, \dots, m$.

Let $L(T) = L_1(T) \cup \ldots \cup L_m(T)$ be the set of active sites of T. The operator ϑ applied to a tree T produces new trees for each element of the set L. More precisely for each subtree t_{ijw} made of one internal node of class $i \in \{1, \dots, m\}$ with degree type $w \neq \varepsilon$, color $j \in \{1, \dots, \alpha_w^i\}$, and leaves of class given by w, there is an operator ϑ_{ijw} making the following operations on T:

- for any active site $A \in L_i$ of T, $\vartheta_{ijw}(T, A)$ is the tree with A replaced by t_{ijw}.

The image of the tree T by the operator ϑ is defined to be the set of all possible trees obtained in this way. More precisely:

$$\vartheta(T) \;=\; \{\vartheta_{ijw}(T, A) \mid i \in \{1, \dots, m\}, w \in \{1, \dots, m\}^*, j \in \{1, \dots, \alpha_w^i\}, A \in L_i\}.$$

The rule corresponding to the operator ϑ is then easily written by considering the effect of the operator ϑ_{ijw} on the number of active sites of each class: for a tree T, let $k_i(T) = |L_i(T)|$ be the number of active sites of class i of T, and call $\vec{k}(T) = (k_1(T), \dots, k_m(T))$ the identity of T. Let us moreover denote by $|w|_i$ the number of leaves of class i in w. The following lemma follows from the definition of ϑ_{ijw}.

Lemma 4.4.2. *Assume $T' = \vartheta_{ijw}(T, A)$ where $i \in \{1, \ldots, m\}$, $A \in L_i(T)$ is the l-th active site of $L_i(T)$, and $j \in \{1, \ldots, \alpha^i_w\}$. Then*

$$\vec{k}(T') = (k_1(T) + |w|_1, \ldots, k_{i-1}(T) + |w|_{i-1},\ l - 1 + |w|_i,\ |w|_{i+1}, \ldots, |w|_m).$$

For simplicity we use the following abbreviation for the last formula.

$$\vec{k}(T') = (k_1(T), \ldots, k_{i-1}(T), l-1, \vec{0}) + w.$$

In view of the definition of ϑ, we are led to define the following succession rule

$$\Omega = \begin{cases} (1, 0, \ldots, 0) & \\ (k_1, \ldots, k_m) \overset{p(ijw)}{\rightsquigarrow} & ((k_1, \ldots, k_{i-1}, 0, \vec{0}) + w), ((k_1, \ldots, k_{i-1}, 1, \vec{0}) + w), \ldots \\ & \ldots, ((k_1, \ldots, k_{i-1}, k_i, \vec{0}) + w) \quad \text{for all } i, j, w \end{cases}$$

where $p(ijw)$ is the weight $p'(t_{ijw})$ of the tree t_{ijw} minus the weight of the leaf of color i, and there is one production for each possible tree t_{ijw}.

Theorem 4.4.1. *The system $\Sigma = (\mathcal{T}^\alpha_G, p', \vartheta, \Omega)$ is an ECO-system representing the multidimensional grammar G.*

Proof. We must verify that the operator ϑ is an ECO operator. In order to prove the result we make the following observations: for a tree $T' \in \mathcal{T}^{w\alpha}_G$ we look for the larger index i such that $\ell_i(T') \neq \ell_{i-1}(T')$ and we consider $\ell_i(T')$. Observe that the sons of $\ell_i(T')$ are all leaves. Indeed, if they were internal nodes, then $\ell_i(T')$ could not be the node under consideration. Now, we want to show that for each $T' \in \mathcal{T}^{w\alpha}_G$ there is one tree $T \in \mathcal{T}^{w\alpha}_G$ such that $T' \in \vartheta(T)$. This is done by looking for the larger index i such that $\ell_i(T') \neq \ell_{i-1}(T')$ and replacing the corresponding subtree with the only leaf of class i. Observe that $\ell_i(T')$, that was the dominant internal node of type i in T', is now an active site of class i, in the new tree T. It remains to prove that there is only one tree T having T' as image. This is easily deduced by construction. $\square$

The assumption that there is exactly one leaf of each color could be raised at the price of more complicated notations. For instance, leaves of other colors could be added by creating for each new color a new class with only leaves (thus increasing m by one). On the other hand the assumption that there are leaves of each class can always be dealt with by adding fake leaves: upon marking them in the generating functions, the contribution of generation trees without fake leaves would then be easily recovered. The case of directed convex polyominoes of next section illustrates explicitely another possible approach, without fake leaves, for a case with $\alpha^2_\varepsilon = 0$.

4.5 A multidimensional case: directed convex polyominoes

Now we focus on the class of weighted α-trees associated with the grammar for directed convex polyominoes introduced in Subsection 4.1.2. In view of what we said above, we slightly change the notations for such a grammar and we obtain the following

$$G_{\mathcal{P}} =< \{\mathcal{P}^1, \mathcal{P}^2, \mathcal{P}^3\}, \{\{\phi^1_\varepsilon\}, \emptyset, \{\phi^3_\varepsilon\}\}, \{\phi^1_1, \phi^1_{31}, \phi^1_2, \phi^2_1, \phi^2_2, \phi^2_{32}, \phi^3_3(1), \phi^3_3(2), \phi^3_{33}\}, \mathcal{P}^1 >,$$

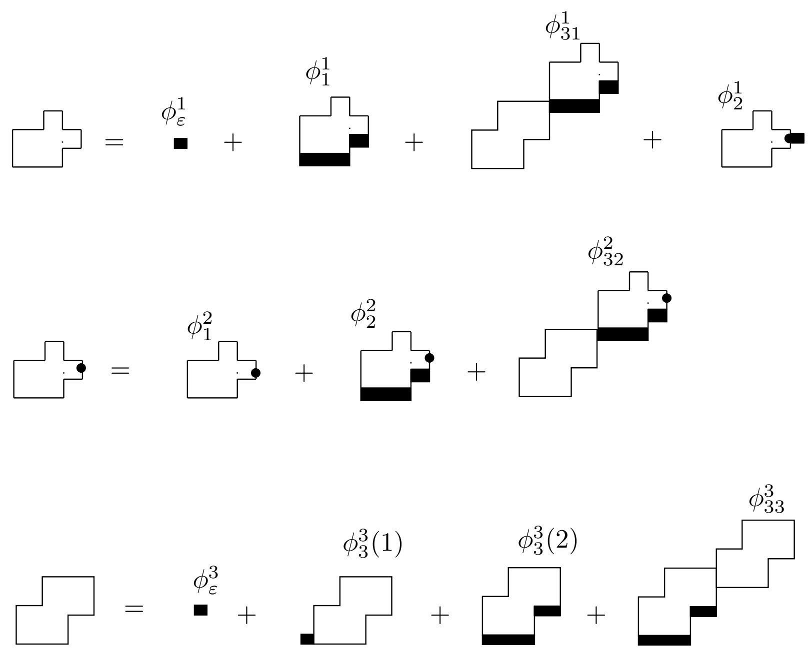

Figure 4.15: The grammar for convex directed polyominoes.

where $\mathcal{P}^1$ is the class of directed convex polyominoes, $\mathcal{P}^2$ is the class of directed convex polyominoes with one marked cell in their last column, and $\mathcal{P}^3$ is the class of parallelogram polyominoes. The operations of the grammar are depicted in Figure 4.15. Thus we have

$$\begin{array}{llll} |\Phi^1_\varepsilon| = 1 & |\Phi^1_1| = 1 & |\Phi^1_{31}| = 1 & |\Phi^1_2| = 1 \\ |\Phi^2_1| = 1 & |\Phi^2_2| = 1 & |\Phi^2_{32}| = 1 & \\ |\Phi^3_\varepsilon| = 1 & |\Phi^3_3| = 2 & |\Phi^3_{33}| = 1 & \end{array} \tag{4.10}$$

and, for each other set Φ^i_w, with $i \in \{1, 2, 3\}$ and $w \in \{1, 2, 3\}^*$, $|\Phi^i_w| = 0$. Let us take the semi-perimeter as a parameter p on the classes $\mathcal{P}^1$, $\mathcal{P}^2$, and $\mathcal{P}^3$. Then p is a $G_{\mathcal{P}}$-linear parameter, where

$$\begin{array}{llll} p(\phi^1_\varepsilon) = 2 & p(\phi^1_1) = 1 & p(\phi^1_2) = 1 & p(\phi^1_{31}) = 0 \\ & p(\phi^2_1) = 0 & p(\phi^2_2) = 1 & p(\phi^2_{32}) = 0 \\ p(\phi^3_\varepsilon) = 2 & p(\phi^3_3(1)) = 1 & p(\phi^3_3(2)) = 1 & p(\phi^3_{33}) = 0. \end{array} \tag{4.11}$$

From the previous observations we deduce that, the class $\mathcal{T}^{w\alpha}_{G_{\mathcal{P}}}$ of weighted α-trees associated with the grammar $G_{\mathcal{P}}$, has nodes with 3 different classes, and root of class 1. Moreover

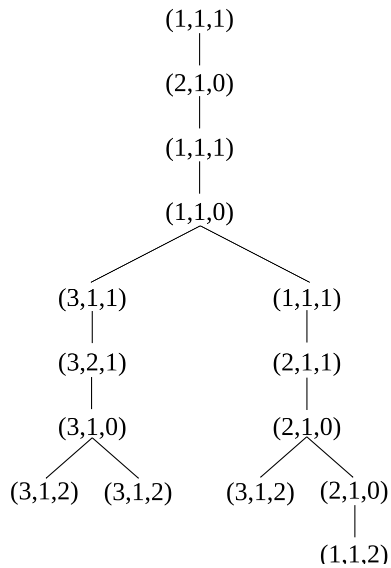

Figure 4.16: A tree belonging to $\mathcal{T}^{w\alpha}_{G_\mathcal{P}}$.

$$
\begin{array}{llll}
\alpha^1_\varepsilon = 1 & \alpha^1_1 = 1 & \alpha^1_{31} = 1 & \alpha^1_2 = 1 \\
\alpha^2_1 = 1 & \alpha^2_2 = 1 & \alpha^2_{32} = 1 & \\
\alpha^3_\varepsilon = 1 & \alpha^3_3 = 2 & \alpha^3_{33} = 1, &
\end{array}
\tag{4.12}
$$

with $\alpha^i_w = 0$ for each other $i \in \{1,2,3\}$ and $w \in \{1,2,3\}^*$. In the columns of table (4.13) we represent the labels that a node can have depending on the classes of its sons.

ε	1	2	3	31	32	33
$(1,1,2)$	$(1,1,1)$	$(1,1,1)$	$(3,1,1)$	$(1,1,0)$	$(2,1,0)$	$(3,1,0)$
$(3,1,2)$	$(2,1,0)$	$(2,1,1)$	$(3,2,1)$			

(4.13)

For instance a node with one son of class 1 can be labeled $(1,1,1)$ or $(2,1,0)$. In Figure 4.16 is represented a tree of the class $\mathcal{T}^{w\alpha}_{G_\mathcal{P}}$.

Now, we want to determine an ECO construction for the class $\mathcal{T}^{w\alpha}_{G_\mathcal{P}}$ according to the parameter p' (see Definition 4.4.2). Let us denote l_i the last internal node of a tree in preorder traversal. We shall distinguish five subsets of $\mathcal{T}^{w\alpha}_{G_\mathcal{P}}$ depending on $f(l_i)$, the number of leaves following l_i, and depending on the class of l_i. Observe that by definition, l_i is not a leaf.

i) $f(l_i) = 1$: Observe that l_i cannot be of class 3. Consequently the only leaf following l_i is of class 1 (therefore it is equal to $(1,1,2)$).

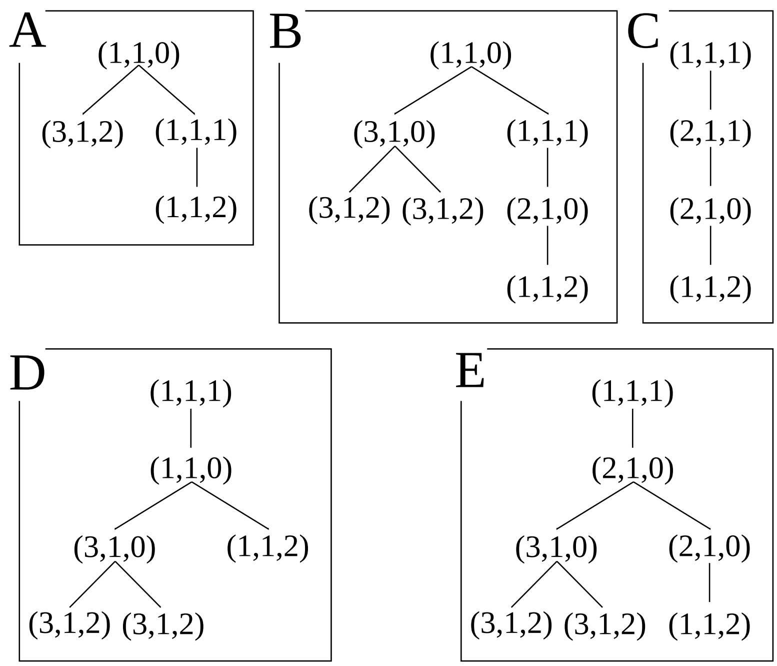

Figure 4.17: Some trees belonging to $\mathcal{T}^{w\alpha}_{G_{\mathcal{P}}}$.

- l_i is of class 1 with one son of class 1 ($l_i = (1,1,1)$): this set of trees is denoted A. Observe that the other type of node of class 1 with one son cannot appear as last internal node since its son is of class 2.
- l_i is of class 2 with one son of class 1 ($l_i = (2,1,0)$): Observe that the other type of node of class 2 with one son cannot appear as last internal node since its son is of class 2.
  - the father of l_i is of class 1 with one son of class 2 (therefore it is equal to $(1,1,1)$): this set of trees is denoted B. Observe that the other types of node of class 1 cannot be the father of the last internal node.
  - the father of l_i is of class 2 and it has one son of class 2 (therefore it is equal to $(2,1,1)$): this set of trees is denoted C.
  - the father of l_i is of class 2 and it has two sons of classes 32 (therefore it is equal to $(2,1,0)$): this set of trees is denoted E.
    Observe that the last type of a node of class 2 cannot appear as the father of l_i since its son is of class 1. Moreover, observe that the father of l_i cannot be of class 3.

ii) $f(l_i) > 1$: this set of trees is denoted D. Observe that the last leaf following l_i is of class 1, and all the other ones are of class 3. Moreover, observe that l_i can be of class 1 or 3.

In Figure 4.17 there is an example of tree for each set described above. Observe that the trivial tree $(1,1,2)$ does not belong to the classes defined above. From the previous assertions we deduce that

Lemma 4.5.1. *The family of sets* $\{A, B, C, D, E\} \cup \{(1,1,2)\}$ *is a partition of* $\mathcal{T}^{w\alpha}_{G_{\mathcal{P}}}$.

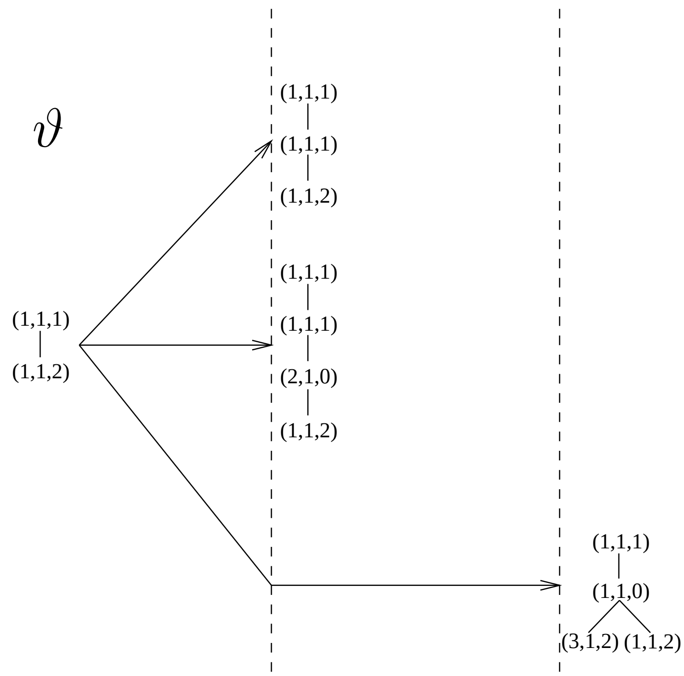

Figure 4.18: The operator ϑ on a tree belonging to A.

4.5.1 An ECO construction for the generic derivation trees of the grammar for directed convex polyominoes

Let $\mathcal{T}_n^\alpha$ be the set of trees belonging to $\mathcal{T}_{G_\mathcal{P}}^{w\alpha}$ with $p' = n$, and ϑ an operator from $\mathcal{T}_n^\alpha$ to $2^{\mathcal{T}_{n+1}^\alpha \cup \mathcal{T}_{n+2}^\alpha}$, performing the following transformations on $T \in \mathcal{T}_n^\alpha$:

a) T is the leaf $(1,1,2)$ or T belongs to A (see Figure 4.18). Let us call L the only leaf following l_i. If $T = (1,1,2)$ then $L = (1,1,2)$. Then we have the following:

   i) ϑ produces a tree with 1 new son, with label $(1,1,2)$, attached to L. Thus L becomes an internal node with a son of class 1 and it is labeled $(1,1,1)$. The new tree belongs to A and the parameter p' increases by one.

   ii) ϑ substitutes L with the tree $((1,1,1),((2,1,0),(1,1,2)))$. The new tree obtained through ϑ belongs to B, and the parameter p' increases by one.

   iii) ϑ produces a tree with 2 new sons attached to L. The label of the left son is $(3,1,2)$ and that of the right one is $(1,1,2)$. Thus L becomes an internal node with two sons (of classes 31) and it is labeled $(1,1,0)$. The new tree belongs to D and the parameter p' increases by two.

b) T belongs to B (see Figure 4.19). Then ϑ makes the operations i), ii), iii) of point a), obtaining three new trees belonging to A, B, and D. Moreover it makes a further operation on l_i:

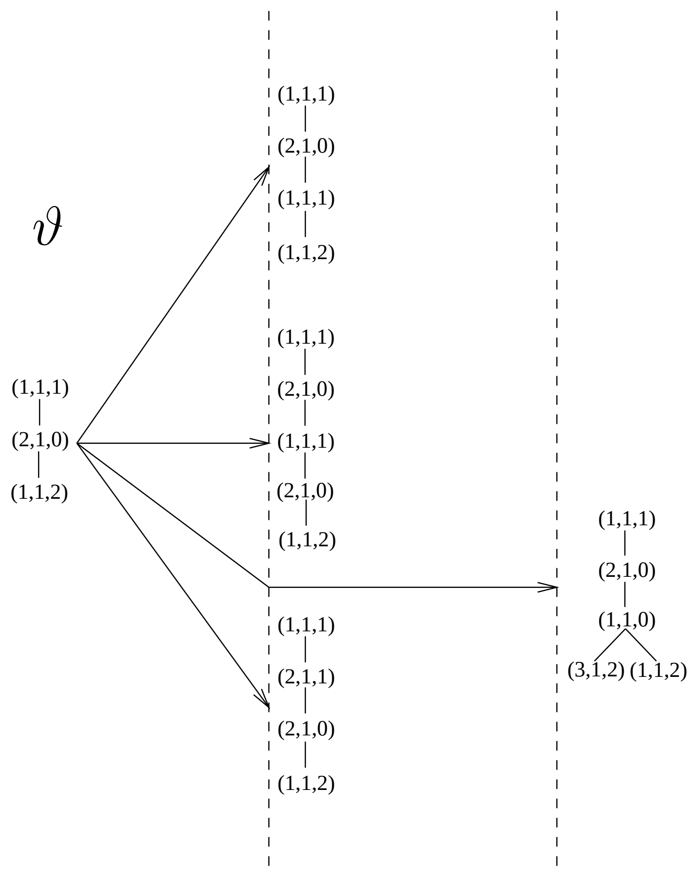

Figure 4.19: The operator ϑ on a tree belonging to B.

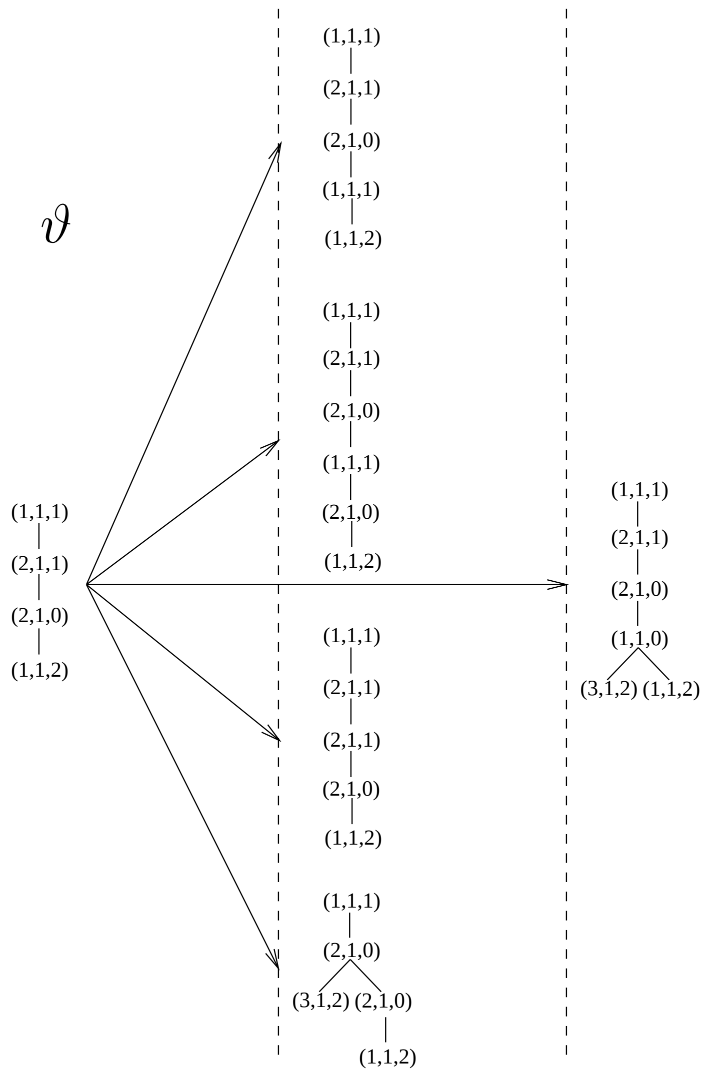

Figure 4.20: The operator ϑ on a tree belonging to C.

iv) ϑ substitutes l_i with the tree $((2,1,1),(2,1,0))$. The new tree obtained through ϑ belongs to C and the parameter p' increases by one.

c) T belongs to C (see Figure 4.20). Then ϑ makes the operations i), ii), iii), iv) described in point a) and in point b). The trees obtained belong to the classes A, B, C, and D. Moreover ϑ makes a further operation on the father F of l_i:

v) it attaches a left son labeled $(3,1,2)$ to F. Thus F becomes a node with two sons (of classes 32) and it has label $(2,1,0)$. The tree obtained belongs to E and the parameter p' increases by one.

d) T belongs to D (see Figure 4.21). Then ϑ makes operations i), ii), iii) on the last leaf following l_i, obtaining three trees belonging to A, B, and D. Moreover it makes a further operation on each leaf L following l_i, except for the last one:

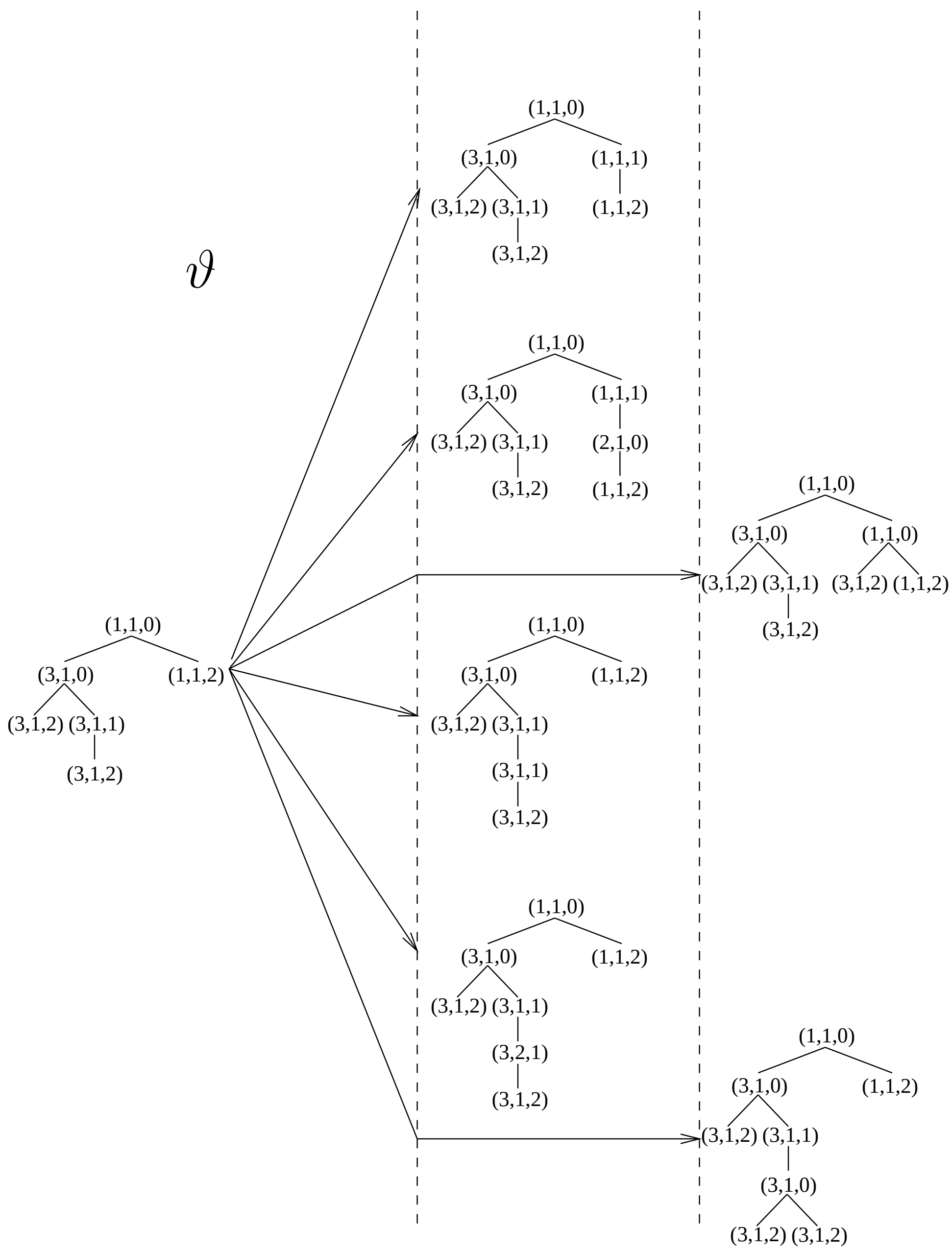

Figure 4.21: The operator ϑ on a tree belonging to D.

vi) for $i \in \{1,2\}$, ϑ attaches i new sons labeled $(3,1,2)$ to L. At this stage L is an internal node with i sons of class 3. If $i=1$ then L can be labeled in 2 ways ($(3,1,1)$ or $(3,2,1)$) otherwise it is labeled $(3,1,0)$.

The new trees obtained still belong to D and the parameter p' increases by one when $i=1$, otherwise it increases by two. To conclude, let $k+1$ be the number of leaves following l_i. Thus ϑ produces 3 trees of type A, B, and D, among which that of type D is produced 2 levels below. Moreover ϑ produces $2k$ trees of type D lying 1 level below, and k trees of type D lying 2 levels below.

e) T belongs to E (see Figure 4.22). Then ϑ makes operations i), ii), iii), and iv) on the only leaf following l_i, obtaining trees belonging to A, B, C, and D. Moreover it makes operation vi) on each leaf following the last but one internal node, except for the last one. In this case, the new trees obtained still belong to E. Finally, if $k+1$ is the number of leaves following the last but one internal node in preorder traversal, then ϑ produces 4 trees of type A, B, C, and D, among which that of type D is produced 2 levels below. Moreover ϑ produces $2k$ trees of type E lying 1 level below, and k trees of type E lying 2 levels below.

In Figure 4.23 are represented the first levels of the generating tree of the operator ϑ. For simplicity at level 3 has been developed only one tree.

Theorem 4.5.1. *The system* $\Sigma = \left(\mathcal{T}^{\alpha}_{G_{\mathcal{P}}}, p', \vartheta, \Omega\right)$ *is an ECO-system, where:*

$$\Omega = \begin{cases} (3)_a \\[2ex] (3)_a \overset{1}{\rightsquigarrow} (3)_a(4)_b \\ \qquad \overset{2}{\rightsquigarrow} (6)_d \\[2ex] (4)_b \overset{1}{\rightsquigarrow} (3)_a(4)_b(5)_c \\ \qquad \overset{2}{\rightsquigarrow} (6)_d \\[2ex] (5)_c \overset{1}{\rightsquigarrow} (3)_a(4)_b(5)_c(7)_e \\ \qquad \overset{2}{\rightsquigarrow} (6)_d \\[2ex] (3k+3)_d \quad \overset{1}{\rightsquigarrow} (3)_a(4)_b \\ \qquad\qquad\quad \overset{2}{\rightsquigarrow} (6)_d \\ \qquad\qquad\quad \overset{1}{\rightsquigarrow} (6)^2_d \ldots (3k+3)^2_d \\ \qquad\qquad\quad \overset{2}{\rightsquigarrow} (6+3)_d \ldots (3(k+1)+3)_d \\[2ex] (3k+4)_e \quad \overset{1}{\rightsquigarrow} (3)_a(4)_b(5)_c \\ \qquad\qquad\quad \overset{2}{\rightsquigarrow} (6)_d \\ \qquad\qquad\quad \overset{1}{\rightsquigarrow} (3+4)^2_e \ldots (3k+4)^2_e \\ \qquad\qquad\quad \overset{2}{\rightsquigarrow} (6+4)_e \ldots (3(k+1)+4))_e \end{cases}$$

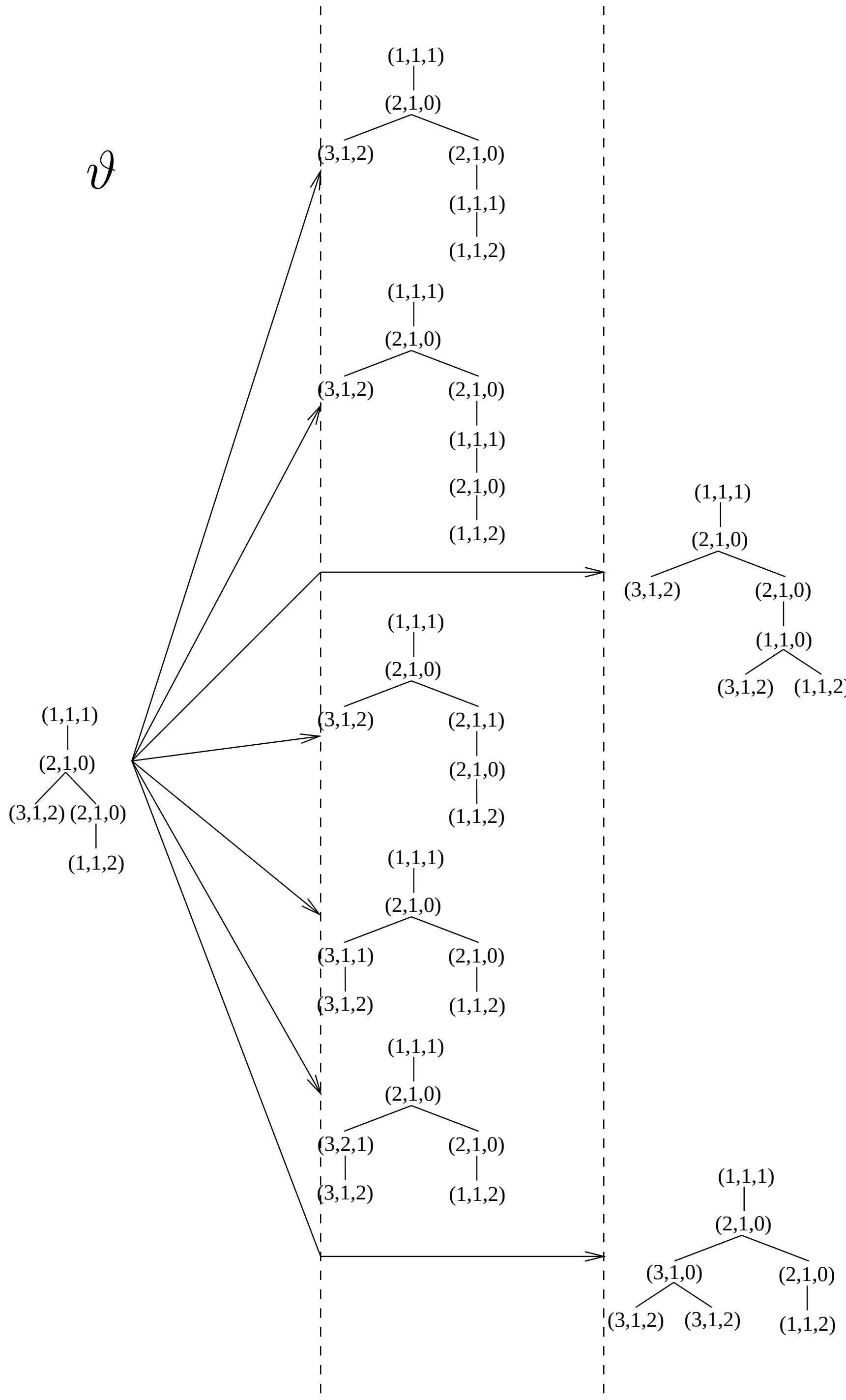

Figure 4.22: The operator ϑ on a tree belonging to E.

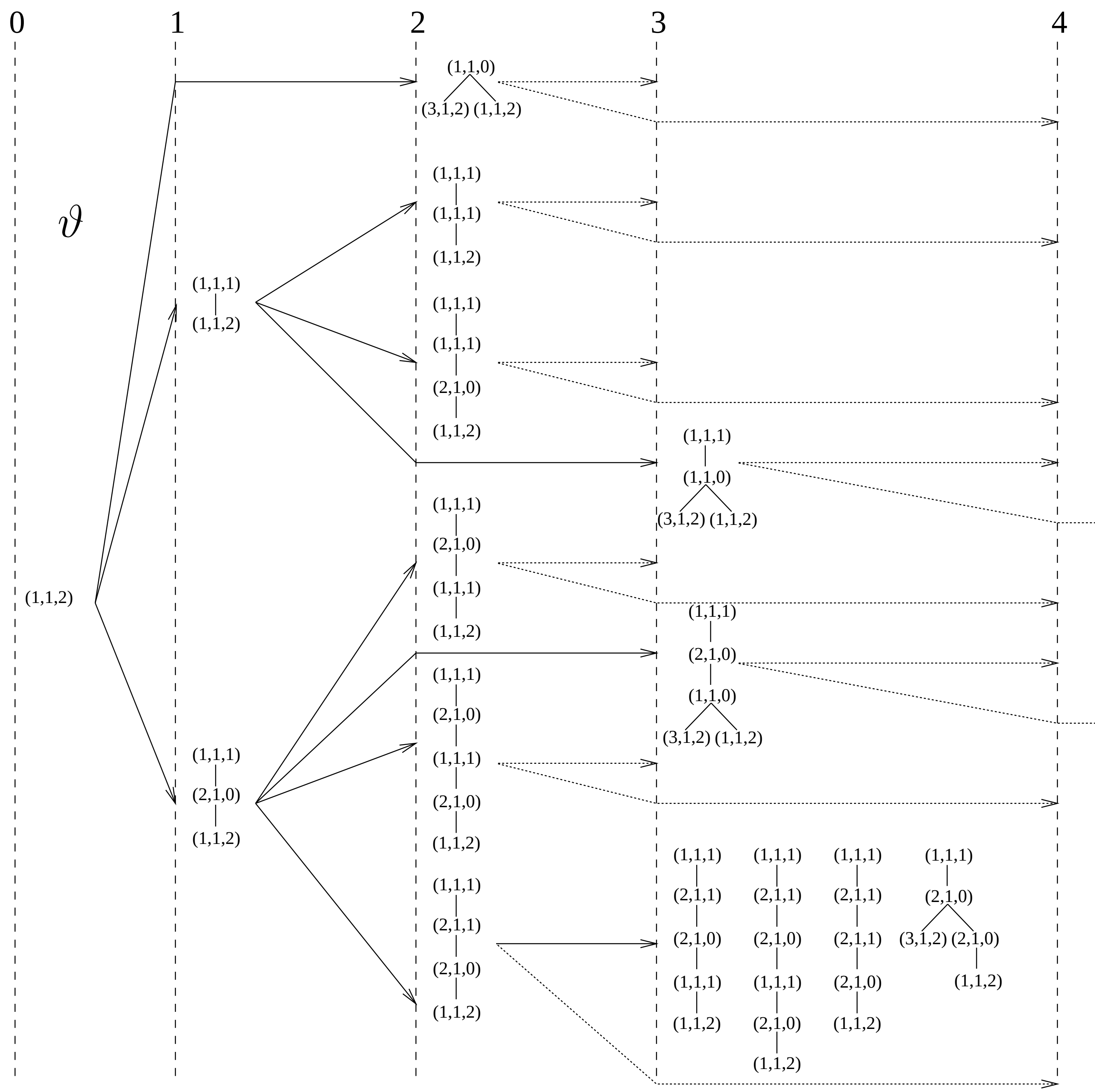

Figure 4.23: The first levels of the generating tree of ϑ.

Proof. The operator ϑ satisfies the conditions of Proposition 4.3.1.

1. *for each* $T' \in \mathcal{T}_n^\alpha$ *there is* $T \in \cup_{j=1}^2 \mathcal{T}_{n-j}^\alpha$ *such that* $T' \in \vartheta(T)$. Let l_i be the last internal node of T'. We distinguish the following cases:
   a) T' belongs to A. Then T is obtained by replacing the subtree (of T') of root l_i with the leaf $(1,1,2)$.
   b) T' belongs to B. Then T is obtained by replacing the subtree whose root is the father of l_i, with the leaf $(1,1,2)$.
   c) T' belongs to C. Then T is obtained by replacing the subtree whose root is the father of l_i with the tree $((2,1,0)(1,1,2))$.
   d) T' belongs to D. Then we must distinguish two cases:
      - l_i is a node of class 1 with two sons, i.e. $(1,1,0)$. Then T is obtained by replacing the subtree of root l_i with the leaf $(1,1,2)$.
      - l_i is a node of class 3. Then T is obtained by replacing the subtree of root l_i with the leaf $(3,1,2)$.
   e) T' belongs to E. Then we must distinguish two cases:
      - the last but one internal node is a node of class 2 with two sons, i.e. $(2,1,0)$. Then T is obtained by replacing the subtree whose root is the last but one internal node, with the tree $((2,1,1),((2,1,0)(1,1,2)))$.
      - the last but one internal node is a node of class 3. Then T is obtained by replacing the subtree whose root is the last but one internal node, with the leaf $(3,1,2)$.
2. *for each* $T \in \mathcal{T}_n^\alpha$, $T' \in \mathcal{T}_m^\alpha$ *such that* $T \neq T'$, $\vartheta(T) \cap \vartheta(T') = \emptyset$. This can be easily deduced from the construction.

□

Remark 4.5.1. *In this case we do not need to introduce multiple labels in the style of Section 4.4. Indeed, there are only active sites of class* 1 *and* 3*, and we know that there is only one active site of class* 1 *and it is the last leaf following the last internal node in preorder traversal.*

Finally, we bijectively proved that the growth of the class of $\mathcal{T}_{G_\mathcal{P}}^{w\alpha}$ according to p' is described by Ω. In order to give an analytical proof of this fact we calculate the generating function $f_\Omega(x) = \sum_{n\geq 0} f_n x^n$ of Ω. Let $f_B(x)$, $f_C(x)$, $f_D(x)$, and $f_E(x)$ be the generating functions of the rules with the same productions of Ω but with different axioms, respectively $(4)_b$, $(5)_c$, $(6)_d$, and $(7)_e$. Let us moreover denote by $f_A(x)$ the generating function of $f_\Omega(x)$. From rule Ω we obtain the following equations:

$$
\begin{aligned}
&f_A(x) = 1 + xf_A(x) + xf_B(x) + x^2 f_D(x) \\
&f_B(x) = 1 + xf_A(x) + xf_B(x) + xf_C(x) + x^2 f_D(x) \\
&f_C(x) = 1 + xf_A(x) + xf_B(x) + xf_C(x) + x^2 f_D(x) + xf_E(x) \\
&f_D(x) = f_{D_d}(x) + xf_{D_d}(x)(f_A(x) + f_B(x)) \\
&f_E(x) = f_{E_e}(x) + xf_{E_e}(x)(f_A(x) + f_B(x) + f_C(x)) + x^2 f_{E_e}(x) f_D(x),
\end{aligned}
\tag{4.14}
$$

where $f_{D_d}(x)$ and $f_{E_e}(x)$ are the generating functions of the rules

$$\Omega_d = \begin{cases} (3k+3)_d & \overset{2}{\rightsquigarrow} (6)_d \\ & \overset{1}{\rightsquigarrow} (6)_d^2 \dots (3k+3)_d^2 \\ & \overset{2}{\rightsquigarrow} (6+3)_d \dots (3(k+1)+3)_d \end{cases}$$

$$\Omega_e = \begin{cases} (3k+4)_e & \overset{1}{\rightsquigarrow} (3+4)_e^2 \dots (3k+4)_e^2 \\ & \overset{2}{\rightsquigarrow} (6+4)_e \dots (3(k+1)+4))_e, \end{cases}$$

describing two subsets of the nodes of rule Ω, respectively the subset of nodes with labels of type d and that one of nodes with label of type e. Now let f_{dn} (*resp.* f_{en}) be the number of nodes at level n of the generating tree of Ω_d(*resp.* Ω_e) and let $f_{dn,k}$ (*resp.* $f_{en,k}$) be the number of nodes at level n having label k. Then

$$f_{D_d}(x) = \sum_{n\geq 0} f_{dn}x^n \text{ and } f_{D_d}(x,y) = \sum_{n\geq 0, k\geq 1} f_{dn,k}x^n y^k,$$

and

$$f_{E_e}(x) = \sum_{n\geq 0} f_{en}x^n \text{ and } f_{E_e}(x,y) = \sum_{n\geq 0, k\geq 1} f_{en,k}x^n y^k.$$

Thus we have $f_{D_d}(x) = f_{D_d}(x,1)$, where $f_{D_d}(x,y)$ satisfies the following recurrence

$$f_{D_d}(x,y) = y + \frac{2xy + x^2y}{1-y} f_{D_d}(x,1) - \frac{2xy + x^2y^2}{1-y} f_{D_d}(x,y).$$

By using kernel method we obtain

$$f_{D_d}(x,1) = \frac{y_0(x) - 1}{2x + x^2} \tag{4.15}$$

where

$$y_0(x) = \frac{1 - 2x - \sqrt{1-4x}}{2x^2}.$$

A similar result holds for $f_{E_e}(x,1)$,

$$f_{E_e}(x,y) = y + \frac{2xy + x^2y^2}{1-y} f_{E_e}(x,1) - \frac{2xy + x^2y^2}{1-y} f_{D_d}(x,y).$$

By applying kernel method we obtain that

$$f_{D_d}(x,1) = \frac{y_0(x) - 1}{2x + x^2 y}. \tag{4.16}$$

Thus, by solving the system (4.14), we have

$$f_{\Omega(x)} = f_A(x) = \frac{1}{\sqrt{1-4x}},$$

and

$$f_{\mathcal{P}} = x^2 f_{\Omega(x)} = \frac{x^2}{\sqrt{1-4x}},$$

where $f_{\mathcal{P}}$ is the well-known generating function for directed convex polyominoes according to the semi-perimeter.

A variant of the ECO construction

The ECO construction for the class $\mathcal{T}_G^{w\alpha}$ is a natural extension of the construction for the multidimensional case (see Section 4.4). Indeed each leaf is developed by adding 1 or 2 sons to it, following the general idea introduced for the multidimensional case. Nevertheless, the operator ϑ makes further operations on the trees of $\mathcal{T}_G^{w\alpha}$ (see operations ii), iv), v) in Subsection 4.5.1), due to the absence of the terminal object of class 2. In particular, the operation v) transforms a node of degree one in a node of degree two, by adding a left son to it. This operation does not produce a jump in the corresponding succession rule, since the parameter p' increases only by one. However, nodes of degree two and of class 1 or 3 are obtained by adding two sons on a leaf. This operation produces a jump of length two on the corresponding succession rule (see operations iii) and vi)). Hereafter, we present a slight variant of the construction described in Subsection 4.5.1. The main idea is to produce each node of degree two by adding a left son to a node of degree one, as in v). The new construction leads to a succession rule without jumps. In order to describe the new ECO construction, we need to partition the sets D and E introduced in Section 4.5:

- D_1 is the subset of D with trees having the last internal node of degree one and label $(3,1,1)$. D_2 is the complement of D_1.
- E_1 is the subset of E with trees having the last but one internal node of degree one and label $(3,1,1)$. E_2 is the complement of E_1.

Let ϑ' be an operator from $\mathcal{T}_n^\alpha$ to $2^{\mathcal{T}_{n+1}^\alpha}$, performing the following transformations on $T \in \mathcal{T}_n^\alpha$:

a) If T belongs to A or $T = (1,1,2)$, then ϑ makes the operations i), ii) described in Subsection 4.5.1, obtaining two trees belonging to A and B. Moreover, if $T \neq (1,1,2)$, it makes a further operation on l_i:

   1. it attaches a left son labeled $(3,1,2)$ to l_i. Thus l_i becomes a node with two sons (of classes 31) and it has label $(1,1,0)$. The new tree belongs to D_2 and the parameter p' increases by one.

b) If T belongs to B then ϑ makes the operations i), ii), iv), obtaining trees belonging to A, B, and C.

c) If T belongs to C then ϑ makes the operations i), ii), iv), and v), obtaining four trees belonging to A, B, C, and E_2.

d_1) If T belongs to D_1 then ϑ makes the operations i), and ii) on the last leaf following the last internal node. The new trees belong respectively to A and B. Moreover, ϑ makes a further operation on each leaf L following the last internal node, except for the last one:

   2. it attaches 1 son labeled $(3,1,2)$ to L. At this stage L is an internal node with 1 son of class 3. Thus L can be labeled $(3,1,1)$ or $(3,2,1)$. In the first case, the new tree belongs to D_1, in the second one it belongs to D_2.

   Finally, ϑ makes another operation on the last internal node of T:

3. it attaches a left son labeled $(3,1,2)$ to l_i. Thus l_i becomes a node with two sons (of classes 33) and it has label $(3,1,0)$. The new tree belongs to D_2 and the parameter p' increases by one.

Let k be the number of leaves following the last internal node of T, except for the last one. Observe that the operator ϑ produces two trees belonging to A and B, k trees belonging to D_1, and $k+1$ trees belonging to D_2.

d_2 If T belongs to D_2 then ϑ makes the operations i), ii), and 2., obtaining trees belonging to A, B, D_1, and D_2. In this case, ϑ produces two trees belonging to A and B, k trees belonging to D_1, and k trees belonging to D_2.

e_1 If T belongs to E_1 then ϑ makes the operations i), ii), iv) on the last leaf following the last but one internal node of T. The new trees belong respectively to A, B, and C . Moreover ϑ makes the operation 2. for each leaf following the last but one internal node, except for the last one. Let k be the number of these leaves, then ϑ produces k trees belonging to E_1 and k trees belonging to E_2. Finally it makes the operation 3. on the last but one internal node, obtaining a tree belonging to E_2.

e_2 If T belongs to E_2 then ϑ makes the operations i), ii), iv), and 2.. Thus it obtains three trees belonging to A, B, and C, k trees belonging to E_1, and k trees belonging to E_2.

From the previous observation we can deduce that the ECO construction by means of θ' can be encoded by the following succession rule without jumps:

$$\Omega' = \begin{cases} (2)_a \\ (2)_a \rightsquigarrow (3)_a(3)_b \\ (3)_a \rightsquigarrow (3)_a(3)_b(4)_{d_2} \\ (3)_b \rightsquigarrow (3)_a(3)_b(4)_c \\ (4)_c \rightsquigarrow (3)_a(3)_b(4)_c(5)_{e_2} \\ (2k+3)_{d_1} \rightsquigarrow (3)_a(3)_b(4)_{d_2}(5)_{d_1} \dots (2k+2)_{d_2}(2k+3)_{d_1}(2k+4)_{d_2} \\ (2k+2)_{d_2} \rightsquigarrow (3)_a(3)_b(4)_{d_2}(5)_{d_1} \dots (2k+2)_{d_2}(2k+3)_{d_1} \\ (2k+4)_{e_1} \rightsquigarrow (3)_a(3)_b(4)_c(5)_{e_2}(6)_{e_1} \dots (2k+3)_{e_2}(2k+4)_{e_1}(2k+5)_{e_2} \\ (2k+3)_{e_2} \rightsquigarrow (3)_a(3)_b(4)_c(5)_{e_2}(6)_{e_1} \dots (2k+3)_{e_2}(2k+4)_{e_1}. \end{cases}$$

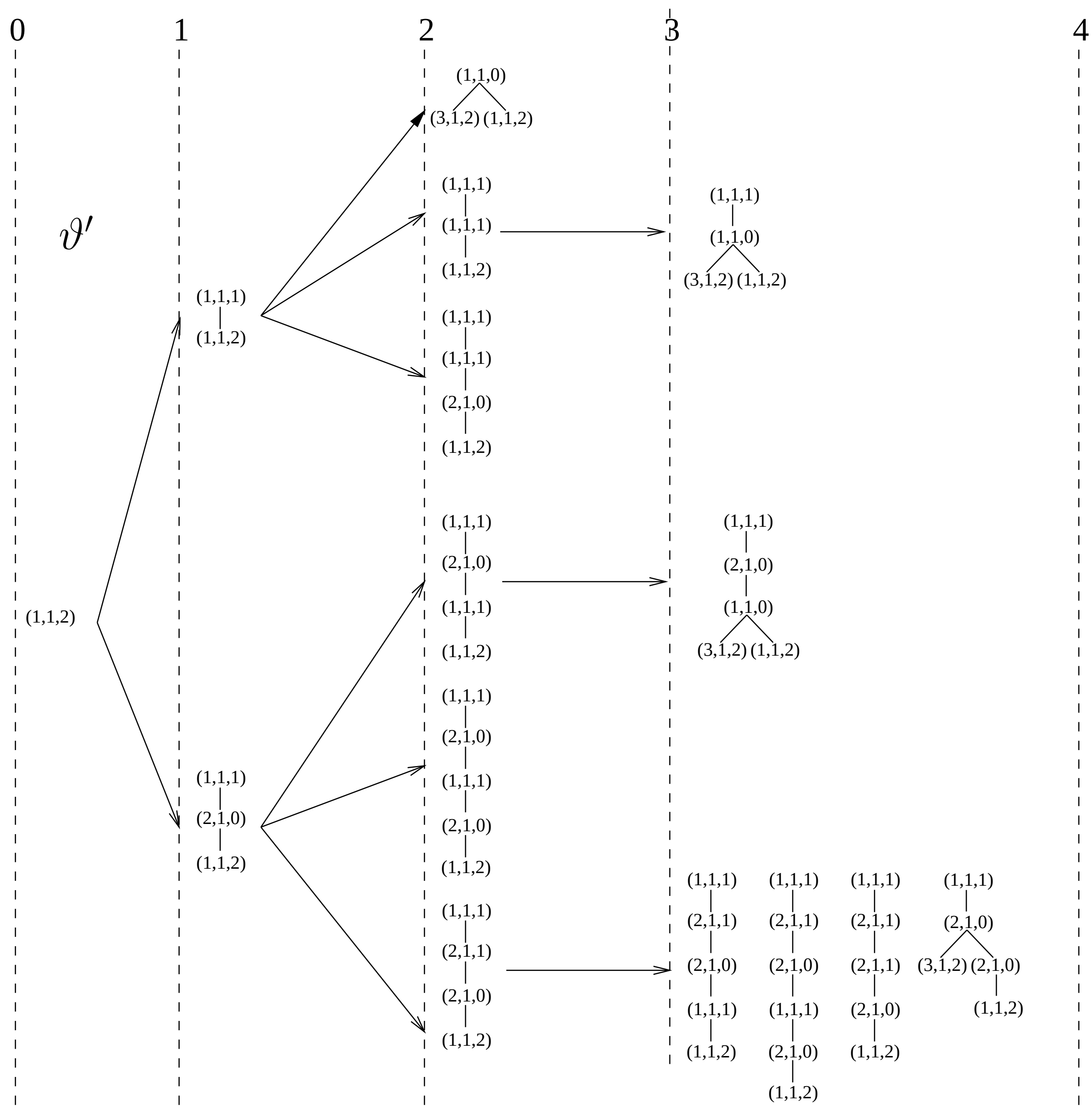

Figure 4.24: The first levels of the generating tree of ϑ.

Chapter 5

Extension to q-parameters

Now we turn to the enumeration with respect to q-parameters, that is with respect to area or generalizations thereof. The problem of counting objects according to q-parameters was first considered by MacMahon [74] (see for instance Delest et al. [32], Denise and Simion [35], Bousquet-Mélou [13], Duchon [40] for a more recent work). Here we deal with q-parameters in the context of object grammars and of the ECO method. A contribute on q-enumeration by using object grammars has been given by Dutour in [44], where she presents an extension of the method introduced by Prellberg and Brak in [81] for solving some particular q-equations. Moreover she provides some examples of q-enumeration by using object grammars and apply this method to solve the q-equations arising from these. A further contribute on q-enumeration in the context of object grammars has been given by Barcucci, Del Lungo, Fédou, and Pinzani in [3]. A different approach has been proposed by Barcucci, Del Lungo, Pergola, and Pinzani in [7], where they use ECO method in order to give a combinatorial interpretation of some q-analogs of Schröder numbers.

The main result of Chapter 4 is a transfer theorem from object grammars to ECO-systems according to linear parameters. In this chapter we shall consider q-parameters in the case of unidimensional grammars. In particular, given an unidimensional grammar G, we define a class of q-parameters on G, called *natural* q-parameters, and we transport them in the corresponding ECO-system. In order to do it we introduce the concept of parametrized succession rules. The functional equations arising from these rules are solved by using the following lemma, introduced in [13] by Bousquet-Mélou. The lemma is obtained by iteration of the initial equation. We denote by $f(s)$ any series of the form $f(s,t,x,y,q)$.

Lemma 5.0.2 (Bousquet-Mélou). *Let $\mathbb{R}[[s,t,x,y,q]]$ be the algebra of formal power series in the variables s,t,x,y,q with real coefficients. Let $\mathcal{A}$ be the sub-algebra of $\mathbb{R}[[s,t,x,y,q]]$ such that the series converge for $s=1$. Given $A(s,t,x,y,q)$ a formal power series in $\mathcal{A}$, we suppose that:*

$$A(s) = xe(s) + xf(s)A(1) + xg(s)A(sq),$$

where $e(s), f(s), g(s)$ are some given power series in $\mathcal{A}$. Then

$$A(s) = \frac{E(s) + E(1)F(s) - E(s)F(1)}{1 - F(1)},$$

where

$$E(s) = \sum_{n\geq 0} x^{n+1} g(s)g(sq)\dots g(sq^{n-1})e(sq^n)$$

and

$$F(s) = \sum_{n\geq 0} x^{n+1} g(s)g(sq)\dots g(sq^{n-1})f(sq^n).$$

The chapter is organised as follows. We start by giving the definition of parametrized succession rules. In Section 5.2 we introduce the concept of natural q-parameters on an object grammar. Then we show how to transport this kind of parameters from an unidimensional object grammar to an ECO-system. In Section 5.3 we present some examples and applications. In Section 5.4 we give an example of the usefulness of knowing an ECO construction associated with an object grammar. In particular, we show how such a construction for a particular class of paths is extendable to another class more difficult to deal with.

5.1 Parametrized succession rules

Let $\Sigma = (\mathcal{O}, p, \vartheta, \Omega_\vartheta)$ be an ECO system. Let us consider other parameters $p_1, \dots, p_h$, $h \in \mathbb{N}^+$, on the class $\mathcal{O}$. A *parametrized succession rule* describes the variations of such parameters through ϑ. It has the following form:

$$\Omega = \begin{cases} (a, a_1, \dots, a_h) & \\ (k, p_1, \dots, p_h) \rightsquigarrow & (e_1(k), t_1^1(k, p_1, \dots, p_h), \dots, t_1^h(k, p_1, \dots, p_h)) \\ & (e_2(k), t_2^1(k, p_1, \dots, p_h), \dots, t_2^h(k, p_1, \dots, p_h)) \\ & \vdots \\ & (e_k(k), t_k^1(k, p_1, \dots, p_h), \dots, t_k^h(k, p_1, \dots, p_h)), \\ & \text{for } k \in M, \ \ (p_1, \dots, p_h) \in \mathbb{N}^h, \end{cases} \tag{5.1}$$

where the set of labels is $M \subseteq \mathbb{N}^+$, a is a constant in M, and the t_i^j are functions $M \times \mathbb{N}^h \to \mathbb{N}$, for all i, j.

Example 5.1.1. *Let $\mathcal{D}$ be the class of Dyck paths and a be the area of a path defined as in Section 1. Let ϑ be the operator of Example 1.3.1. In Figure 5.1 is represented the variation of the area when the operator acts on the active site at level i. We can encode such a variation*

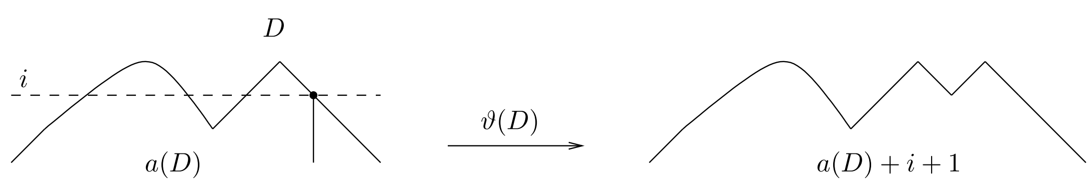

Figure 5.1: The variation of the area through the operator ϑ.

in the parametrized succession rule

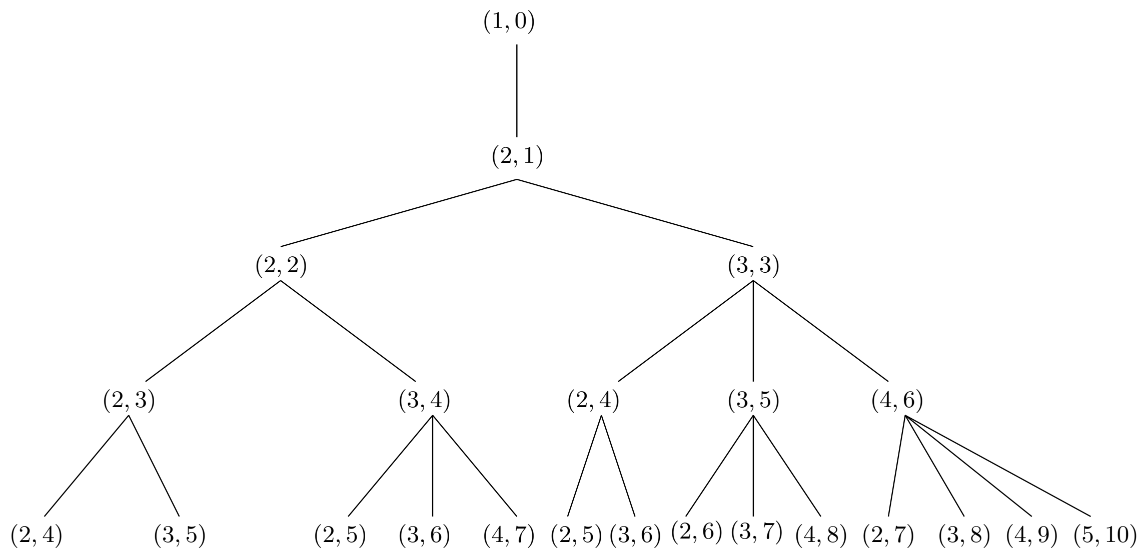

Figure 5.2: The first levels of Γ'.

$$\Gamma' = \begin{cases} (1,0) \\ (1,0) \rightsquigarrow (2,1) \\ (k,a) \rightsquigarrow (2,a+1)\dots(k,a+k-1)(k+1,a+k) \end{cases}$$

whose first levels of the generating tree are depicted in Figure 5.2.

5.2 Natural q-parameters on an object grammar

We focus on a particular type of q-parameter that we call *natural* q-parameter. Then we transport this kind of parameter from unidimensional object grammars to ECO-systems.

Definition 5.2.1. *Let $G = \langle \mathbb{O}, \mathbb{E}, \Phi, \mathcal{A} \rangle$ be an object grammar. Let $\mathcal{O}, \mathcal{O}^i \in \mathbb{O}$ for $i = 1 \dots k$ and q be a parameter on $\mathcal{O}, \mathcal{O}^1, \dots, \mathcal{O}^k$. Let $\phi \in \Phi$ be an object operation such that $dom(\phi) = \mathcal{O}^1 \times \dots \times \mathcal{O}^k$ and $cod(\phi) = \mathcal{O}$. Then q is a* natural q-parameter *with respect to ϕ if, for all $(O_1, \dots, O_k) \in dom(\phi)$,*

$$q\left(\phi(O_1, \dots, O_k)\right) = \sum_{i=1}^{k} q(O_i) + \sum_{i=1}^{k} (k-i)t(O_i) + q(\phi),$$

where $i \in \mathbb{N}^+$, $q(\phi) \in \mathbb{N}$, and t is a G-linear parameter on the grammar G.

Let G be an unidimensional object grammar, p be a G-linear parameter, and q be a natural G-q-linear parameter with associated G-linear parameter t. Here, in order to deal with q-parameters, we naturally extend the Definition 4.3.1 of weighted α-trees, by considering trees with labels of the form $(i, w_{ij}, w'_{ij}, w''_{ij})$. Then we extend the bijection in Subsection 4.3.2, between derivation trees and weighted α-trees, by taking $w'_{ij} = q(\phi^i_j)$ and $w''_{ij} = t(\phi^i_j)$, for $i = 1 \dots |\Phi_j|$ and $j = 0 \dots d$.

Definition 5.2.2. *Let $T \in \mathcal{T}_G^{w\alpha}$. For any $x \in T$ denote $l(x) = (co(x), w(x), w'(x), w''(x))$ the label of x. Then*

$$q'(T) = \sum_{x \in T} \left(\sum_{i=1}^{k(x)} (k(x) - i) \sum_{y \in T_{i,x}} w''(y) + w'(x) \right)$$

and

$$t'(T) = \sum_{x \in T} w''(x).$$

From Definition 5.2.2 and from (4.5) we obtain the following Lemma:

Lemma 5.2.1. *Let $T \in \mathcal{T}_G^{w\alpha}$, then $q'(T) = q(ev(T))$.*

Let us consider the ECO operator ϑ_2 defined in Subsection 4.3.4 for the class of weighted α-trees associated with G. We want to give a parametrized succession rule describing the variation of the parameter q through ϑ_2. Let T' be the tree obtained through ϑ_2 by adding a tree S on the l-th active site A of T. Then we have the following lemma:

Lemma 5.2.2.

$$q'(T') - q'(T) = (l-1)(t'(S) - w''(A)) + q(S) - w'(A)$$

Proof. Let b be the branch from the root of T to the father of its l-th active site A. Because of the definition of ϑ_2, only the subtrees of T having their root in b change through ϑ_2. Observe that b is also the branch from the root of T' to the father of the root of S. Thus we have

$$\begin{aligned} q'(T') - q'(T) &= \textstyle\sum_{x \in b \cup S} \left(\sum_{i=1}^{k(x)} (k(x) - i) \sum_{y \in T'_{i,x}} w''(y) + w'(x) \right) - \\ &\quad \textstyle\sum_{x \in b \cup \{A\}} \left(\sum_{i=1}^{k(x)} (k(x) - i) \sum_{y \in T_{i,x}} w''(y) + w'(x) \right) \\ &= \textstyle\sum_{x \in b} \sum_{i=1}^{k(x)} (k(x) - i) \left(\sum_{y \in T'_{i,x}} w''(y) \; - \; \sum_{y \in T_{i,x}} w''(y) \right) + q'(S) - w'(A). \end{aligned}$$

For x in b, let $i(x)$ be the index of the subtree $T_{i,x}$ that contains A. Since only the subtrees containing A change, we have $T'_{i,x} = T_{i,x}$ for $i \neq i(x)$ and $T'_{i(x),x} = T_{i(x),x} \setminus \{A\} \cup S$. Hence

$$\begin{aligned} q'(T') - q'(T) &= \sum_{x \in b} (k(x) - i(x)) \left(\sum_{y \in T'_{i(x),x}} w''(y) \; - \; \sum_{y \in T_{i(x),x}} w''(y) \right) + q'(S) - w'(A), \\ &= \sum_{x \in b} (k(x) - i(x)) \left(\sum_{y \in S} w''(y) \; - \; w''(A) \right) + q'(S) - w'(A). \end{aligned}$$

Since A is an active site, the sons of x with index larger than $i(x)$ are then active sites: the number of active sites attached to x is $k(x) - i(x)$. Moreover A is the l-th active site, hence

$$\begin{aligned} q'(T') - q'(T) &= (l-1) \left(\sum_{y \in S} w''(y) \; - \; w''(A) \right) + q'(S) - w'(A) \\ &= (l-1)(t'(S) - w''(A)) + q'(S) - w'(A). \end{aligned}$$

Thus we have that

$$q'(T') = q'(T) + (l-1)(t'(S) - w''(A)) + q'(S) - w'(A). \tag{5.2}$$

□

Now, in the application ϑ_2, S is a tree made of a root and j leaves, the rightmost of these being A, for $j = 1 \dots d$. More precisely, S is associated to a derivation tree $(\phi_j^{i_j}, (\phi_0^{i_{0,1}}, \dots, \phi_0^{i_{0,j-1}}, A))$. Therefore, in terms of the ϕ, equation (5.2) becomes

$$q'(T') = q'(T) + (l-1)\left(t(\phi_j^{i_j}) + \sum_{r=1}^{j-1} t(\phi_0^{i_{0,r}})\right) + \sum_{r=1}^{j-1}(j-r)t(\phi_0^{i_{0,r}}) + q(\phi_j^{i_j}) + \sum_{r=1}^{j-1} q(\phi_0^{i_{0,r}}). \tag{5.3}$$

Let us denote

$$R(S) = t(\phi_j^{i_j}) + \sum_{r=1}^{j-1} t(\phi_0^{i_{0,r}}) \text{ and } Q(S) = \sum_{r=1}^{j-1}(j-r)t(\phi_0^{i_{0,r}}) + q(\phi_j^{i_j}) + \sum_{r=1}^{j-1} q(\phi_0^{i_{0,r}}),$$

then

$$q'(T') = q'(T) + (l-1)R(S) + Q(S).$$

Now we have the variation of the parameter q when the operator ϑ_2 attaches j leaves on the l-th active site. Therefore we can extend the succession rule Ω_2 (see Subsection 4.3.4) to a parametrized succession rule depending also on the parameter q. Observe that the axiom of Ω_2 corresponds to the empty tree, therefore in this case q is equal to 0. Then we obtain the following:

$$\Omega_2^q = \begin{cases} (\alpha_0, 0) \\ (\alpha_0, 0) \overset{p(\phi_0^{i_0})}{\rightsquigarrow} (c, q(\phi_0^{i_0})) \\ (kc, q) \overset{P(S)}{\rightsquigarrow} (jc,\, q + Q(S))\,((j+1)c,\, q + Q(S) + R(S)) \dots ((k+j-1)c,\, q + Q(S) + (k-1)R(S)) \end{cases}$$

where there is a production for each tree S corresponding to a derivation tree $(\phi_j^{i_j}, (\phi_0^{i_{0,1}}, \dots, \phi_0^{i_{0,j-1}}, A))$ with $j = 1 \dots d$, $i_j = 1 \dots \alpha_j$, and $i_{0,r} = 1 \dots \alpha_0$, and where

$$\begin{aligned} P(S) &= p(\phi_j^{i_j}) + \textstyle\sum_{r=1}^{j-1} p(\phi_0^{i_{0,r}}), \\ R(S) &= t(\phi_j^{i_j}) + \textstyle\sum_{r=1}^{j-1} t(\phi_0^{i_{0,r}}), \text{ and} \\ Q(S) &= \textstyle\sum_{r=1}^{j-1}(j-r)t(\phi_0^{i_{0,r}}) + q(\phi_j^{i_j}) + \sum_{r=1}^{j-1} q(\phi_0^{i_{0,r}}). \end{aligned}$$

5.3 Examples and applications.

5.3.1 The h-coloured plane trees.

An h-coloured plane tree has h possible colours on its leaves, with $h \in \mathbb{N}^+$. An exception is the tree made up by one node, which is not coloured. Let $\mathcal{T}_h$ be the class of h-coloured plane

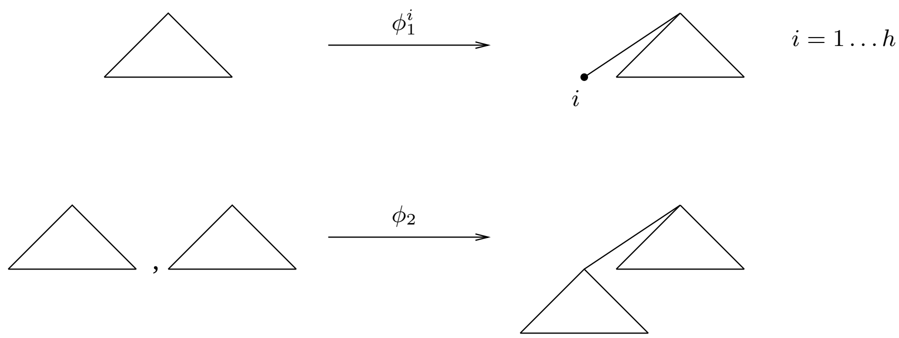

Figure 5.3: The operations, ϕ_1^i, ϕ_2 of the grammar $G_{\mathcal{T}_h}$.

trees. We want to enumerate $\mathcal{T}_h$ according to the internal path length, defined in Section 1. The mappings ϕ_1^i and ϕ_2, illustrated in Figure 5.3, are object operations on $\mathcal{T}_h$. For $i = 1 \ldots h$ we have:

- operation ϕ_1^i takes a plane tree and it attaches a i-coloured leaf as the leftmost son of the root of such a tree;
- operation ϕ_2 takes a pair of plane trees as its argument, it attaches the first one as the leftmost son of the root of the second one.

The class $\mathcal{T}_h$ is generated by the unidimensional object grammar

$$G_{\mathcal{T}_h} = \left\langle \{\mathcal{T}_h\},\ \{\{.\}\}, \{\phi_1^i,\ \phi_2\} \right\rangle$$

where $i = 1 \ldots h$ and the terminal object is the tree with only one node. Let n denote the number of nodes of a given h-coloured tree and s denote its internal path length. Then s is a natural q-parameter with respect to the operation ϕ_2. Indeed, given $T_1, T_2 \in \mathcal{T}_h$, we have the following:

$$\begin{array}{lll} n(.) = 1 & n(\phi_1^i(T_1)) = n(T_1) + 1 & n(\phi_2(T_1, T_2)) = n(T_1) + n(T_2) \\ s(.) = 0 & s(\phi_1^i(T_1)) = s(T_1) + 1 & s(\phi_2(T_1, T_2)) = s(T_1) + s(T_2) + n(T_1). \end{array} \tag{5.4}$$

The class of $(1, h, 1)$-trees is the class of weighted α-trees associated with $\mathcal{T}_h$. Now, from the general rule Ω_2^q we obtain the parametrized succession rule for the $(1, h, 1)$-trees. In this case we have

$$n(\phi_0) = 1 \quad n(\phi_1^i) = 1 \quad n(\phi_2) = 0$$

$$s(\phi_0) = 0 \quad s(\phi_1^i) = 1 \quad s(\phi_2) = 0,$$

and $t = n$. Consequently, in the notation of the previous section, $P(S) = 1$, $Q(S) = 1$ and $R(S) = 1$ for $j = 1, 2$. Moreover $\alpha_0 = 1$ and $c = h + 1$, so that we obtain the following parametrized succession rule:

$$\Omega_s = \begin{cases} (1,0) & \\ (1,0) & \overset{1}{\rightsquigarrow} \quad (h+1,0) \\ ((h+1)k,s) & \overset{1}{\rightsquigarrow} \quad (h+1,s+1)^h((h+1)2,s+2)^h \dots ((h+1)k,s+k)^h \\ & \overset{1}{\rightsquigarrow} \quad ((h+1)2,s+1)((h+1)3,s+2)\dots((h+1)(k+1),s+k) \end{cases} \tag{5.5}$$

and the system $\Sigma = \left(\mathcal{T}^{w\alpha}_{G_{\mathcal{T}_h}}, n, \vartheta_2, \Omega_s\right)$ is an ECO-system.

Let L be the set of nodes of the generating tree associated with Ω'_s, where

$$\Omega'_s = \begin{cases} (h+1,0) & \\ ((h+1)k,s) & \overset{1}{\rightsquigarrow} \quad (h+1,s+1)^h((h+1)2),s+2)^h \dots ((h+1)k,s+k)^h \\ & \overset{1}{\rightsquigarrow} \quad ((h+1)2,s+1)((h+1)3,s+2)\dots((h+1)(k+1),s+k). \end{cases} \tag{5.6}$$

Given $v \in L$, we denote $n(v)$ the level of v in the generating tree. Then the generating function of the rule Ω_s with respect to n, k and s is

$$f_{\Omega_s}(x,t,q) = 1 + x f_{\Omega'_s}(x,t,q),$$

where

$$f_{\Omega'_s}(x,t,q) = \sum_{v \in L} x^{n(v)} t^{k(v)} q^{s(v)},$$

with $t = y^{h+1}$ (for consistency with the definition of the generating function of a succession rule). From Ω'_s we obtain

$$\begin{aligned} f_{\Omega'_s}(x,t,q) &= xt + xh \textstyle\sum_{v\in L} x^{n(v)} \sum_{i=1}^{k(v)} t^i q^{s(v)+i} + \frac{x}{q} \sum_{v\in L} x^{s(v)} \sum_{i=2}^{k(v)+1} t^i q^{s(v)+i} \\ &= xt + \tfrac{xhtq}{1-tq}\left(f_{\Omega'_s}(x,1,q) - f_{\Omega'_s}(x,tq,q)\right) + \tfrac{xt^2q}{1-tq}\left(f_{\Omega'_s}(x,1,q) - f_{\Omega'_s}(x,tq,q)\right) \\ &= xt + x\tfrac{htq+t^2q}{1-tq} f_{\Omega'_s}(x,1,q) - x\tfrac{htq+t^2q}{1-tq} f_{\Omega'_s}(x,tq,q). \end{aligned}$$

Let

$$e(t) = t \qquad f(t) = \frac{htq+t^2q}{1-tq} \qquad g(t) = -\frac{htq+t^2q}{1-tq},$$

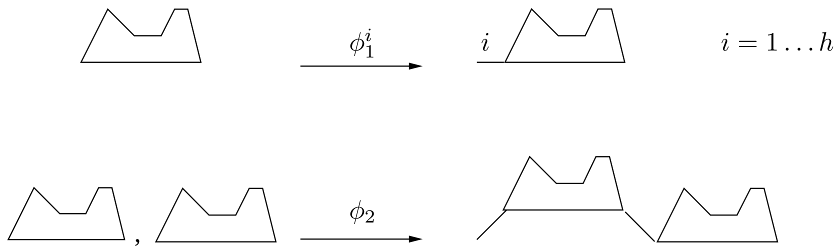

Figure 5.4: The operations, ϕ_1^i, ϕ_2 of the grammar $G_{\mathcal{M}_h}$.

by applying lemma 5.0.2 we obtain

$$f_{\Omega'_s}(x,1,q) = \frac{E_1(x,1,q)}{E_0(x,1,q)},$$

where

$$E_1(x,1,q) = \sum_{n\geq 1} (-1)^{n-1} \frac{x^n}{(q)_{n-1}} q^{\frac{(n-1)(n+2)}{2}} \prod_{k=0}^{n-2} (h+q^k)$$

and

$$E_0(x,1,q) = \sum_{n\geq 0} (-1)^{n} \frac{x^n}{(q)_{n}} q^{\frac{n(n-1)}{2}} q^n \prod_{k=0}^{n-1} (h+q^k).$$

5.3.2 The h-coloured Motzkin paths.

An h-coloured Motzkin path is a coloured Motkin path with h possible colours on its horizontal step, with $h \in \mathbb{N}$. Let $\mathcal{M}_h$ be the class of h-coloured Motzkin paths. We want to enumerate $\mathcal{M}_h$ according to the area, defined as the sum of the ordinates of each endpoint of a rise or horizontal step. The mappings ϕ_1^i and ϕ_2, illustrated in Figure 5.4, are object operations on $\mathcal{M}_h$. For $i = 1 \dots h$ we have:

- operation ϕ_1^i adds a i-coloured horizontal step at the begining of a path;
- operation ϕ_2 takes a pair of paths as its argument, adds a rise (resp. fall) step at the beginning (resp. end) of the first path and then appends the second path.

The class $\mathcal{M}_h$ is generated by the unidimensional object grammar

$$G_{\mathcal{M}_h} = \left\langle \{\mathcal{M}_h\},\ \{\{.\}\}, \{\phi_1^i,\ \phi_2\} \right\rangle$$

where $i = 1 \dots h$ and the terminal object is the path of zero length, commonly represented as a dot. Let us denote l the length of an h-coloured Motzkin path, t its number of rise and horizontal steps, and a its area. Then a is a natural q-parameter with respect to the operation ϕ_2. Indeed, given $M_1, M_2 \in \mathcal{M}_h$, the following relations hold:

$$
\begin{aligned}
&l(.) = 0 \quad l(\phi_1^i(M_1)) = l(M_1) + 1 \quad l(\phi_2(M_1, M_2)) = l(M_1) + l(M_2) + 2 \\
&a(.) = 0 \;\; a(\phi_1^i(M_1)) = a(M_1) \;\; a(\phi_2(M_1, M_2))) = a(M_1) + a(M_2) + t(M_1) + 1 \\
&t(.) = 0 \;\; t(\phi_1^i(M_1)) = t(M_1) + 1 \;\; t(\phi_2(M_1, M_2)) = t(M_1) + t(M_2) + 1.
\end{aligned}
\tag{5.7}
$$

The class of $(1, h, 1)$-trees is the class of weighted α-trees associated with $\mathcal{M}_h$. Now, from the general rule Ω_2^q we obtain the parametrized succession rule for the $(1, h, 1)$-trees. In this case we have,

$$
\begin{aligned}
&l(\phi_0) = 0 \quad l(\phi_1^i) = 1 \quad l(\phi_2) = 2 \\
&a(\phi_0) = 0 \quad a(\phi_1^i) = 0 \quad a(\phi_2) = 1, \\
&t(\phi_0) = 0 \quad t(\phi_1^i) = 1 \quad t(\phi_2) = 1.
\end{aligned}
$$

Consequently, in the notation of the previous section, we have $P(S) = j$, $Q(S) = j - 1$ and $R(S) = 1$ for $j = 1, 2$. Moreover $\alpha_0 = 1$ and $c = h + 1$, so that we obtain the following parametrized succession rule:

$$
\Omega_a = \begin{cases}
(1, 0) & \\
(1, 0) & \overset{0}{\rightsquigarrow} \; (h + 1, 0) \\
((h + 1)k, a) & \overset{1}{\rightsquigarrow} \; (h + 1, a)^h((h + 1)2), a + 1)^h \dots ((h + 1)k, a + k - 1)^h \\
& \overset{2}{\rightsquigarrow} \; ((h + 1)2, a + 1)((h + 1)3, a + 2) \dots ((h + 1)(k + 1), a + k)
\end{cases}
\tag{5.8}
$$

and the system $\Sigma = \left(\mathcal{T}^{w\alpha}_{G_{\mathcal{M}_h}}, l, \vartheta_2, \Omega_a\right)$ is an ECO-system.

Let L be the set of nodes of the generating tree associated with Ω_a', where

$$
\Omega_a' = \begin{cases}
(h + 1, 0) & \\
((h + 1)k, a) & \overset{1}{\rightsquigarrow} \; (h + 1, a)^h((h + 1)2), a + 1)^h \dots ((h + 1)k, a + k - 1)^h \\
& \overset{2}{\rightsquigarrow} \; ((h + 1)2, a + 1)((h + 1)3, a + 2) \dots ((h + 1)(k + 1), a + k).
\end{cases}
\tag{5.9}
$$

Given $v \in L$, we denote $l(v)$ the level of v in the generating tree. Then the generating function of Ω_a with respect to l, k, and a is

$$
f_{\Omega_a}(x, t, q) = 1 + f_{\Omega_a'}(x, t, q),
$$

where

$$f_{\Omega'_a}(x,t,q) = \sum_{v \in L} x^{l(v)} t^{k(v)} q^{a(v)},$$

with $t = y^{h+1}$ (again for consistency with the definition). From Ω'_a we obtain

$$\begin{aligned} f_{\Omega'_a}(x,t,q) &= t + xh \textstyle\sum_{v \in L} x^{l(v)} \sum_{i=1}^{k(v)} t^i q^{a(v)+i-1} + x^2 \sum_{v \in L} x^{l(v)} \sum_{i=2}^{k(v)+1} t^i q^{a(v)+i-1} \\ &= t + \tfrac{xht}{1-tq}\left(f_{\Omega'_a}(x,1,q) - f_{\Omega'_a}(x,tq,q)\right) + \tfrac{x^2t^2q}{1-tq}\left(f_{\Omega'_a}(x,1,q) - f_{\Omega'_a}(x,tq,q)\right) \\ &= t + x\tfrac{ht+xt^2q}{1-tq} f_{\Omega'_a}(x,1,q) - x\tfrac{ht+xt^2q}{1-tq} f_{\Omega'_a}(x,tq,q). \end{aligned}$$

We can apply Lemma 5.0.2, where

$$e(t) = \frac{t}{x} \qquad f(t) = \frac{ht + xt^2q}{1-tq} \qquad g(t) = -\frac{ht+xt^2q}{1-tq}$$

From Lemma 5.0.2 we have

$$f_{\Omega'_a}(x,1,q) = \frac{E(x,1,q)}{1 - F(x,1,q)}$$

Let us denote $E_0(x,1,q) = E(x,1,q)$ and $E_1(x,1,q) = 1 - F(x,1,q)$, then

$$f_{\Omega'_a}(x,1,q) = \frac{E_1(x,1,q)}{E_0(x,1,q)},$$

where

$$E_1(x,1,q) = \sum_{n \geq 0} (-1)^n \frac{x^n}{(q)_n} q^n \prod_{k=0}^{n-1} (hq^k + xq^{2k+1})$$

and

$$E_0(x,1,q) = \sum_{n \geq 0} (-1)^n \frac{x^n}{(q)_n} \prod_{k=0}^{n-1} (hq^k + xq^{2k+1}).$$

5.3.3 The area under m-Dyck paths

A m-Dyck path is a path with steps $(1,m)$ and $(1,-1)$, going from $(0,0)$ to $((m+1)l, 0)$, and remaining weakly above the x-axis. Let $\mathcal{D}_m$ be the set of m-Dyck paths. We want again to enumerate these paths according to the area, defined as the sum of the ordinates of the endpoints of the rise steps. The mapping ϕ_{m+1} is an object operation on $\mathcal{D}_m$: it takes $m+1$ m-Dyck paths, adds a rise step at the beginning of the first path, and attaches at its end an alternating sequence of down steps and paths. See Figure 5.5 for the case $m = 3$.

The class $\mathcal{D}_m$ is generated by the unidimensional object grammar

$$G_{\mathcal{D}_m} = \langle \{\mathcal{D}_m\},\ \{\{.\}\}, \{\phi_{m+1}\} \rangle$$

where the terminal object is the path of zero length, commonly represented as a dot. Let us denote $(m+1)l$ the length of an m-Dyck path, and a its area. Then a is a natural q-parameter

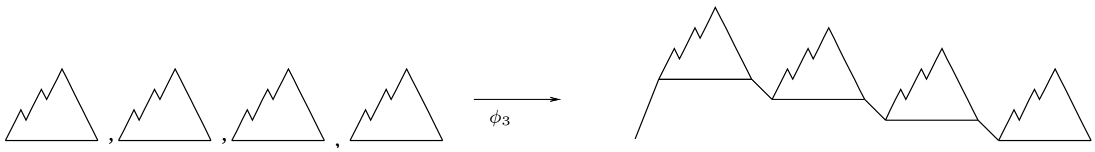

Figure 5.5: The object operation ϕ_3.

with respect to the operation ϕ_{m+1}. Indeed, given $d_1, \ldots, d_{m+1} \in \mathcal{D}_m$, the following relations hold:

$$\begin{array}{lrcl} l(.) = 0 & l(\phi_{m+1}(d_1, \ldots, d_{m+1}))) & = & l(d_1) + \cdots + l(d_{m+1}) + 1 \\ a(.) = 0 & a(\phi_{m+1}(d_1, \ldots, d_{m+1}))) & = & a(d_1) + \cdots + a(d_{m+1}) + \\ & & & +ml(d_1) + (m-1)l(d_2) + \ldots + l(d_m) \ + \ m. \end{array} \tag{5.10}$$

The class of $(1, \underbrace{0, \ldots, 0}_{m}, 1)$-trees is the class of weighted α-trees associated with the grammar $G_{\mathcal{D}_m}$. Now, from the general rule Ω_2^q we obtain the parametrized succession rule for the $(1, \underbrace{0, \ldots, 0}_{m}, 1)$-trees. In this case we have

$$l(\phi_0) = 0 \quad l(\phi_{m+1}) = 1$$

$$a(\phi_0) = 0 \quad a(\phi_{m+1}) = m,$$

and $t = l$. Consequently, in the notation of the previous section, $P(S) = 1$, $Q(S) = m$ and $R(S) = 1$ for $j = m + 1$. Moreover, $\alpha_0 = 1$ and $c = 1$, and we obtain the following parametrized succession rule:

$$\Omega_m = \begin{cases} (1,0) & \\ (1,0) \overset{0}{\rightsquigarrow} (1,0) & \\ (k,a) \overset{1}{\rightsquigarrow} (m+1, a+m)(m+2, a+m+1)\ldots(m+k, a+m+k-1) \end{cases} \tag{5.11}$$

Then the generating function of Ω_m with respect to l, k, and a is

$$f_{\Omega_m}(x, y, q) = 1 + f_{\Omega'_m}(x, y, q),$$

where

$$f_{\Omega'_m}(x, y, q) = \sum_{v \in L} x^{l(v)} y^{k(v)} q^{a(v)}.$$

From Ω'_m we obtain

$$\begin{aligned} f_{\Omega'_m}(x,y,q) &= y + x\sum_{v\in L} x^{l(v)}\sum_{i=1}^{k(v)} y^i q^{a(v)+m+i-1} \\ &= y + \tfrac{xyq^m}{1-yq}\left(f_{\Omega'_m}(x,1,q) - f_{\Omega'_m}(x,yq,q)\right) \\ &= y + x\tfrac{yq^m}{1-yq} f_{\Omega'_m}(x,1,q) - x\tfrac{yq^m}{1-yq} f_{\Omega'_m}(x,yq,q). \end{aligned}$$

We can apply Lemma 5.0.2, where

$$e(y) = \frac{y}{x} \qquad f(y) = \frac{yq^m}{1-yq} \qquad g(y) = -\frac{yq^m}{1-yq}$$

From Lemma 5.0.2 we have

$$f_{\Omega'_m}(x,1,q) = \frac{E(x,1,q)}{1-F(x,1,q)}$$

Let us denote $E_0(x,1,q) = E(x,1,q)$ and $E_1(x,1,q) = 1 - F(x,1,q)$, then

$$f_{\Omega'_m}(x,1,q) = \frac{E_1(x,1,q)}{E_0(x,1,q)},$$

where

$$E_1(x,1,q) = \sum_{n\geq 0}(-1)^n \frac{x^n}{(q)_n} q^{mn} \prod_{k=0}^{n-1}(q^{mk})$$

and

$$E_0(x,1,q) = \sum_{n\geq 0}(-1)^n \frac{x^n}{(q)_n} \prod_{k=0}^{n-1}(q^{mk}).$$

5.4 The area under $+3-2$ paths

Let us consider the following variation $\mathcal{D}^j_{2,3}$ of the class of Dyck paths: a path in $\mathcal{D}^j_{2,3}$ is a sequence of $(1,3)$ and $(1,-2)$ steps, going from $(0,0)$ to (n,j) and remaining weakly above the x-axis. The generating function of this class of paths according to the length was obtained by Labelle and Yeh [73] and Duchon [40] using grammar decompositions. These decompositions generalize the classic one for m-Dyck paths but they are more difficult to obtain. Here we see that it is useful to know the ECO construction associated with an object grammar. Indeed the ECO construction of m-Dyck paths is easily extendable and, as we will see, it allows to find the generating function with respect to the area.

For simplicity we denote $\mathcal{D}^j$ the class $\mathcal{D}^j_{2,3}$. In order to give an ECO construction for the class $\mathcal{D}^{0,1} = \mathcal{D}^0 \bigcup \mathcal{D}^1$, we can extend the previous construction for m-Dyck paths: the operator ϑ_m adds a peak on each point of the last descent of each Dyck path.

Let $\mathcal{D}^{0,1}_n$ be the set of paths in $\mathcal{D}^{0,1}$ having lenght n, and ψ an operator from $\mathcal{D}^{0,1}_n$ to $2^{\mathcal{D}^{0,1}_{n+2} \bigcup \mathcal{D}^{0,1}_{n+3}}$ performing the following transformations on $D \in \mathcal{D}^{0,1}_n$:

If $D \in \mathcal{D}^0_n$, ψ adds the path $(1,3)(1,-2)$ on each point of the last descent of D. Then the paths obtained belong to $\mathcal{D}^1_{n+2}$.

If $D \in \mathcal{D}_n^1$, ψ adds the path $(1,3)(1,-2)(1,-2)$ on each point of the last descent of D. Then the paths obtained belong to $\mathcal{D}_{n+3}^0$.

Let $D \in \mathcal{D}_n^{0,1}$ with k points on its last descent. Let a be the area of D, defined as the sum of the ordinates of each $(1,3)$ step. We can easily verify that:

$$\text{if} \quad D \in \mathcal{D}_n^0 \quad \text{then} \quad a(\psi(D)) = a(D) + 3 + 2(j-1), \quad \text{for} \quad j \in \{1, \ldots, k\},$$

$$\text{if} \quad D \in \mathcal{D}_n^1 \quad \text{then} \quad a(\psi(D)) = a(D) + 4 + 2(j-1), \quad \text{for} \quad j \in \{1, \ldots, k\}.$$

Consequently the parametrized succession rule associated to ψ is:

$$\Omega_\psi = \begin{cases} (1,0) \\ \\ (k,a) \overset{2}{\rightsquigarrow} \overline{(2,a+3)}\,\overline{(3,a+3+2)} \ldots \overline{(k+1,a+3+2(k-1))} \\ \\ \overline{(k,a)} \overset{3}{\rightsquigarrow} (3,a+4)(4,a+4+2) \ldots (k+2,a+4+2(k-1)) \end{cases} \tag{5.12}$$

Let L be the set of nodes of the generating tree associated with Ω_ψ. Then $\{L_1, L_2\}$ is a partition of L, where L_1 is the subset of L with overlined labels and L_2 is the remaining subset. Let us denote $l(v)$ the level of a node v in the generating tree. The generating function $f_{\Omega_\psi}(x,y,q)$ associated with Ω_ψ according to l, k, and a is:

$$f_{\Omega_\psi}(x,y,q) = \sum_{v \in L} x^{l(v)} y^{k(v)} q^{a(v)}.$$

Then

$$f_{\Omega_\psi}(x,y,q) = f_1(x,y,q) + f_2(x,y,q),$$

where

$$f_1(x,y,q) = \sum_{v \in L_1} x^{l(v)} y^{k(v)} q^{a(v)}$$

and

$$f_2(x,y,q) = \sum_{v \in L_2} x^{l(v)} y^{k(v)} q^{a(v)}.$$

From Ω_ψ we obtain

$$\begin{aligned} f_1(x,y,q) = & \quad y + x^3 \tfrac{y^3 q^4}{1-yq^2} f_2(x,1,q) - x^3 \tfrac{y^3 q^4}{1-yq^2} f_2(x,yq^2,q), \\ f_2(x,y,q) = & \quad x^2 \tfrac{y^2 q^3}{1-yq^2} f_1(x,1,q) - x^2 \tfrac{y^2 q^3}{1-yq^2} f_1(x,yq^2,q). \end{aligned}$$

In order to solve such equations we slightly extend lemma 5.0.2. The idea is the same but we iterate both $f_1(x,y,q)$ and $f_2(x,y,q)$. Let us denote $f(y)$ any series of the form $f(x,y,q)$ and

$$a(y) = y \quad b(y) = x^3 \tfrac{y^3 q^4}{1-yq^2} \quad c(y) = -x^3 \tfrac{y^3 q^4}{1-yq^2},$$

$$b'(y) = x^2 \tfrac{y^2 q^3}{1-yq^2} \quad c'(y) = -x^2 \tfrac{y^2 q^3}{1-yq^2}.$$

By iteration we obtain

$$\begin{aligned} f_1(y) = &\quad E_{11}(y) f_1(1) + E_{12}(y) f_2(1) + H_1(y), \\ f_2(y) = &\quad E_{21}(y) f_1(1) + E_{22}(y) f_2(1) + H_2(y), \end{aligned} \tag{5.13}$$

where

$$\begin{aligned} E_{11}(y) = &\quad \textstyle\sum_{n\geq 0} \prod_{k=0}^{n-1} c(yq^{4k}) c'(yq^{4k+2}) c(yq^{4n}) b'(yq^{4n+2}) \\ E_{12}(y) = &\quad \textstyle\sum_{n\geq 0} \prod_{k=0}^{n-1} c(yq^{4k}) c'(yq^{4k+2}) b(yq^{4n}) \\ H_1(y) = &\quad \textstyle\sum_{n\geq 0} \prod_{k=0}^{n-1} c(yq^{4k}) c'(yq^{4k+2}) a(yq^{4n}) \end{aligned}$$

and

$$\begin{aligned} E_{21}(y) = &\quad \textstyle\sum_{n\geq 0} \prod_{k=0}^{n-1} c'(yq^{4k}) c(yq^{4k+2}) b'(yq^{4n}) \\ E_{22}(y) = &\quad \textstyle\sum_{n\geq 0} \prod_{k=0}^{n-1} c'(yq^{4k}) c(yq^{4k+2}) c'(yq^{4n}) b(yq^{4n+2}) \\ H_2(y) = &\quad \textstyle\sum_{n\geq 0} \prod_{k=0}^{n-1} c'(yq^{4k}) c(yq^{4k+2}) c'(yq^{4n}) a(yq^{4n+2}). \end{aligned}$$

From (5.13) we finally obtain:

Proposition 5.4.1. *The generating function $f_1(1) = f_1(x,1,q)$ of $\mathcal{D}^0$ and $f_2(1) = f_2(x,1,q)$ of $\mathcal{D}^1$ are respectively*

$$\begin{aligned} f_1(1) = &\quad \tfrac{E_{12}(1) H_2(1) + H_1(1)(1-E_{22}(1))}{(1-E_{11}(1))(1-E_{22}(1)) - E_{12}(1) E_{21}(1)} \\ f_2(1) = &\quad \tfrac{E_{21}(1) H_1(1) + H_2(1)(1-E_{11}(1))}{(1-E_{22}(1))(1-E_{11}(1)) - E_{12}(1) E_{21}(1)}, \end{aligned}$$

where

$$\begin{aligned}
E_{11}(1) = &\quad -\textstyle\sum_{n\geq 0} x^{5(n+1)} \prod_{k=0}^{n} \frac{q^{20k+11}}{(1-q^{4k+2})(1-q^{4k+4})} \\
E_{12}(1) = &\quad \textstyle\sum_{n\geq 0} x^{5n+3} \frac{q^{12n+4}}{1-q^{4n+2}} \prod_{k=0}^{n-1} \frac{q^{20k+11}}{(1-q^{4k+2})(1-q^{4k+4})} \\
H_{1}(1) = &\quad \textstyle\sum_{n\geq 0} x^{5n} q^{4n} \prod_{k=0}^{n-1} \frac{q^{20k+11}}{(1-q^{4k+2})(1-q^{4k+4})}
\end{aligned}$$

and

$$\begin{aligned}
E_{21}(1) = &\quad \textstyle\sum_{n\geq 0} x^{5n+2} \frac{q^{8n+3}}{1-q^{4n+2}} \prod_{k=0}^{n-1} \frac{q^{20k+13}}{(1-q^{4k+2})(1-q^{4k+4})} \\
E_{22}(1) = &\quad -\textstyle\sum_{n\geq 0} x^{5(n+1)} \prod_{k=0}^{n} \frac{q^{20k+13}}{(1-q^{4k+2}(1-q^{4k+4})} \\
H_{2}(1) = &\quad -\textstyle\sum_{n\geq 0} x^{5n+2} \frac{q^{12n+5}}{1-q^{4n+2}} \prod_{k=0}^{n-1} \frac{q^{20k+13}}{(1-q^{4k+2})(1-q^{4k+4})}.
\end{aligned}$$

Chapter 6

From an ECO-system to an object grammar: convex polyominoes

In this chapter, we consider the problem of passing from an ECO-system to an object grammar. Fédou and Garcia [50] already considered this problem and provided a grammar-like decomposition for a particular class of succession rules, in order to show that their generating functions are algebraic. The succession rules that they consider are very general but have only one type of production. We extend here their approach to deal with a case of succession rules with several types of production: the class of convex polyominoes. This class of polyominoes is algebraic [34] and a direct grammar-like decomposition has been obtained before by [45]. However this decomposition contains a lot of productions and is much more difficult to obtain directly than the ECO-system. The advantage of our method is that the grammar is obtained in a quasi-automatic way from the ECO-system.

The general strategy that we apply is the following. Let T be the generating tree of a succession rule Ω. To each path p of T, going from the root to a node at level n, we associate a word w of lenght $n+1$ made of the labels of the nodes of p. Then, to Ω corresponds a noncommutative formal power series S_Ω constituted of the sum over words w, with multiplicities given by the number of associated paths. Then we determine a recursive decomposition of S_Ω by decomposing the words associated with Ω. More precisely, we look whether these words contain the label of the root of T or the labels of the roots of trees that are in turn decomposable. When the answer is positive we provide a decomposition of the words in these terms. Finally, from S_Ω, we obtain an algebraic system of equations by taking its commutative image.

The chapter is organized as follows. We first determine an ECO construction for convex polyominoes according to their semi-perimeter. We easily pass from the ECO construction to the associated succession rule Ω. We then apply the decomposition and determine a system of functional equations leading to the generating function for convex polyominoes. An account of the result of this chapter can be found in [31].

6.1 An ECO operator for the class of convex polyominoes

The number of convex polyominoes with respect to the semi-perimeter was first determined by Delest and Viennot [34]. In the recent years the result had been re-estabilished using different analytical strategies (Bousquet-Mélou [13], Chang and Lin [23], Guttmann and Enting [68]),

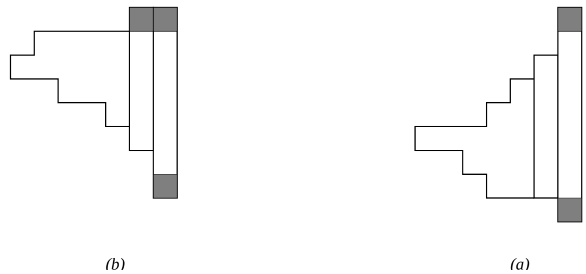

Figure 6.1: Convex polyominoes in $\mathcal{C}_b$, (b), $\mathcal{C}_a$, (a).

or bijective proofs (Bousquet-Mélou and Guttmann [17]). In this section we present an ECO approach to the enumeration of the set of convex polyominoes according to the semi-perimeter. We first partition the set of convex polyominoes $\mathcal{C}$ into four classes, denoted by $\mathcal{C}_b$, $\mathcal{C}_a$, $\mathcal{C}_r$, and $\mathcal{C}_g$:

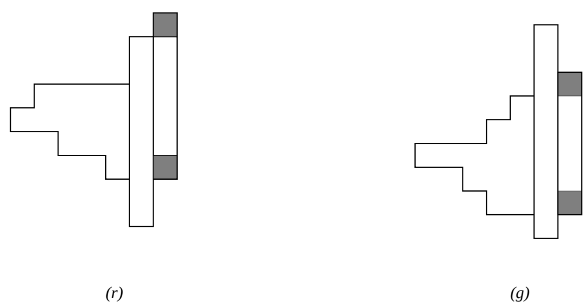

Figure 6.2: Convex polyominoes in $\mathcal{C}_r$, (r), and $\mathcal{C}_g$, (g).

i) $\mathcal{C}_b$ is the set of convex polyominoes having at least two columns and such that (Figure 6.1, (b)):

1. The uppermost cell of the rightmost column has the maximal ordinate among all the cells of the polyomino, and it is the same ordinate as the uppermost cell of the column on its left.
2. The lowest cell of the rightmost column has the minimal ordinate among all the cells of the polyomino.

ii) $\mathcal{C}_a$ is the set of convex polyominoes not in $\mathcal{C}_b$, and such that (see Figure 6.1, (a)):

1. The uppermost cell of the rightmost column has the maximal ordinate among all the cells of the polyomino.
2. The lowest cell of the rightmost column has the minimal ordinate among all the cells of the polyomino.

Observe that, according to this definition, all convex polyominoes made only of one column lie in the class $\mathcal{C}_a$.

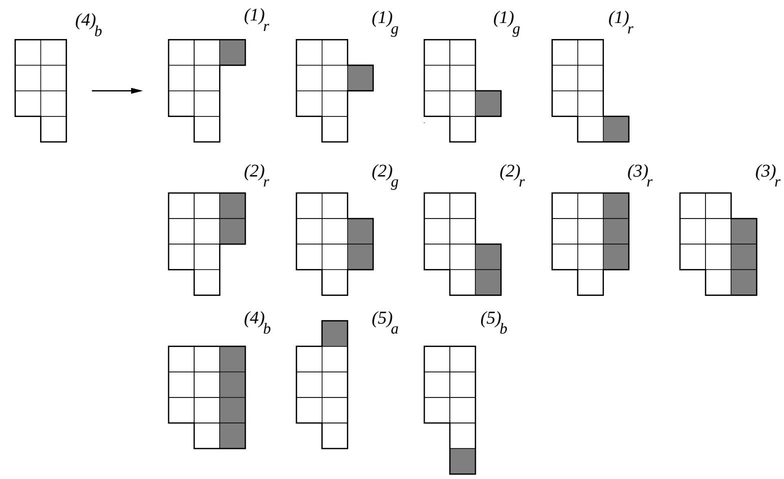

Figure 6.3: The ECO operator for the class $\mathcal{C}_b$.

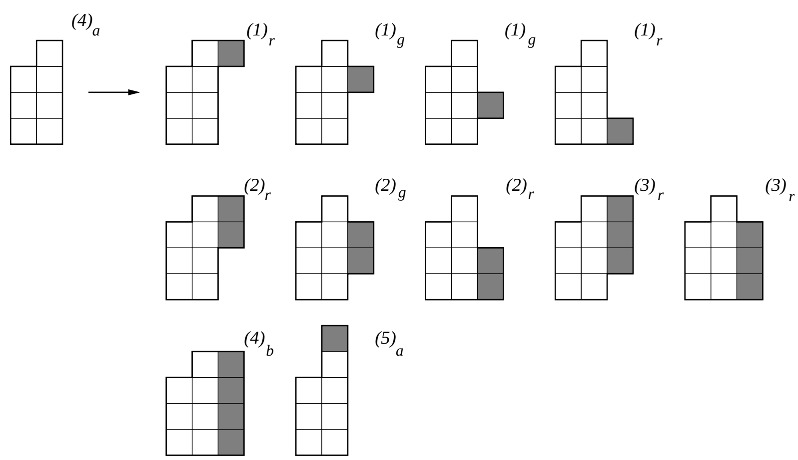

Figure 6.4: The ECO operator for the class $\mathcal{C}_a$.

iii) $\mathcal{C}_r$ is the set of convex polyominoes where only one among the lowest and the uppermost cells of the rightmost column has minimal (resp. maximal) ordinate among all the cells of the polyomino (see Figure 6.2, (r)).

iv) $\mathcal{C}_g$ is the set of remaining convex polyominoes (see Figure 6.2, (g)).

The ECO operator, namely ϑ, performs local expansions on the rightmost column of any polyomino of semi-perimeter $n+2$, producing a set of polyominoes of semi-perimeter $n+3$. More precisely, the operator ϑ performs the following set of expansions on any convex polyomino P, with semi-perimeter $n+2$ and k cells in the rightmost column:

- for any $i = 1, \dots, k$ the operator ϑ glues a column of length i to the rightmost column of P; this can be done in $k-i+1$ possible ways. Therefore this operation produces $1+2+\ldots+k$ polyominoes with semi-perimeter $n+3$.

Moreover, the operator performs some other transformations on convex polyominoes, according to the belonging class:

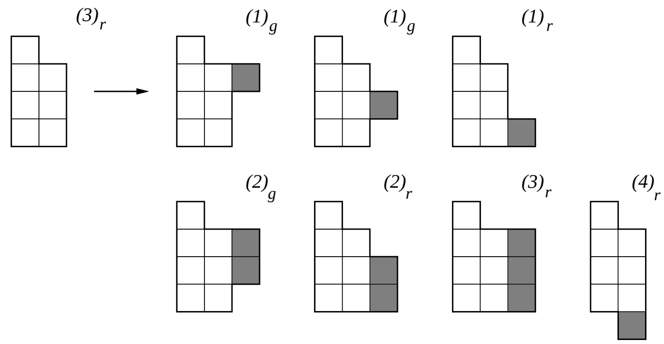

Figure 6.5: The ECO operator for the class $\mathcal{C}_r$.

- if $P \in \mathcal{C}_b$, then the operator ϑ produces two more polyominoes, one by gluing a cell onto the top of the rightmost column of P, and another by gluing a cell on the bottom of the rightmost column of P (Figure 6.3).
- if $P \in \mathcal{C}_a$, then the operator ϑ produces one polyomino by gluing a cell onto the top of the rightmost column of P (Figure 6.4).
- if $P \in \mathcal{C}_r$, then:

  if the uppermost cell of the rightmost column of P has the maximal ordinate, the operator ϑ glues a cell onto the top of that column ;

  else, the operator ϑ glues a cell on the bottom of the rightmost column of P (Figure 6.5).

The construction for polyominoes in $\mathcal{C}_g$ requires no additive expansions, and it is graphically explained in Figure 6.6.

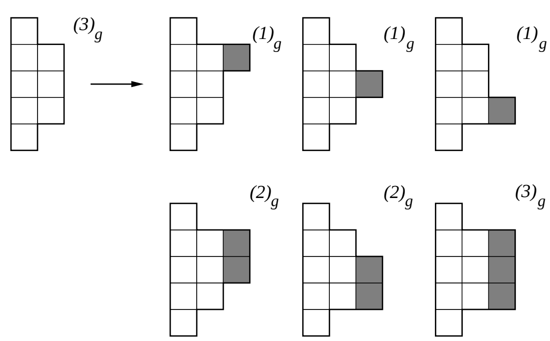

Figure 6.6: The ECO operator for the class $\mathcal{C}_g$.

The next step consists in translating the previous construction into a set of equations whose solution is the generating function for convex polyominoes. To realize this purpose, we first encode the ECO operator in a succession rule.

Basically, a polyomino in $\mathcal{C}_i$, $i \in \{a, b, g, r\}$ with k cells in the rightmost column is labelled $(k)_i$. Let us take as an example, the polyomino in Figure 6.5, with label $(3)_r$; according to the figure, the performance of the ECO operator on the polyomino can be sketched by the production:

$$(3)_r \rightsquigarrow (1)_g\ (1)_g\ (1)_r\ (2)_g\ (2)_r\ (3)_r\ (4)_r.$$

In a similar fashion, the performance of the ECO operator on a polyomino can be sketched by the following succession rule:

$$\Omega \begin{cases} (1)_a \\ (k)_g \rightsquigarrow \prod_{j=1}^{k}(j)_g^{k-j+1} \\ (k)_r \rightsquigarrow \prod_{j=1}^{k-1}(j)_g^{k-j}\ \prod_{j=1}^{k+1}(j)_r \\ (k)_a \rightsquigarrow \prod_{j=1}^{k-2}(j)_g^{k-j-1}\ \prod_{j=1}^{k-1}(j)_r^2\ (k)_b\ (k+1)_a \\ (k)_b \rightsquigarrow \prod_{j=1}^{k-2}(j)_g^{k-j-1}\ \prod_{j=1}^{k-1}(j)_r^2\ (k)_b\ (k+1)_a\ (k+1)_b, \end{cases} \tag{6.1}$$

As an example, for $k = 1, 2, 3$ we have the following productions of Ω:

$$\begin{array}{ll} (1)_a \rightsquigarrow (1)_b(2)_a & (1)_b \rightsquigarrow (1)_b(2)_a(2)_b \\ (2)_a \rightsquigarrow (1)_r(1)_r(2)_b(3)_a & (2)_b \rightsquigarrow (1)_r(1)_r(2)_b(3)_a(3)_b \\ (3)_a \rightsquigarrow (1)_g(1)_r(1)_r(2)_r(2)_r(3)_b(4)_a & (3)_b \rightsquigarrow (1)_g(1)_r(1)_r(2)_r(2)_r(3)_b(4)_a(4)_b. \\ \dots \quad \dots \quad \dots & \dots \quad \dots \quad \dots \\ (1)_r \rightsquigarrow (1)_r(2)_r & (1)_g \rightsquigarrow (1)_g \\ (2)_r \rightsquigarrow (1)_g(1)_r(2)_r(3)_r & (2)_g \rightsquigarrow (1)_g(1)_g(2)_g \\ (3)_r \rightsquigarrow (1)_g(1)_g(1)_r(2)_g(2)_r(3)_r(4)_r & (3)_g \rightsquigarrow (1)_g(1)_g(1)_g(2)_g(2)_g(3)_g \\ \dots \quad \dots \quad \dots & \dots \quad \dots \quad \dots \end{array}$$

Remark 6.1.1. *The rule Ω does not satisfy the consistency principle; we have chosen this representation for Ω in order to have simpler notations.*

Figure 6.7 depicts the first levels of the generating tree of the rule (6.1).

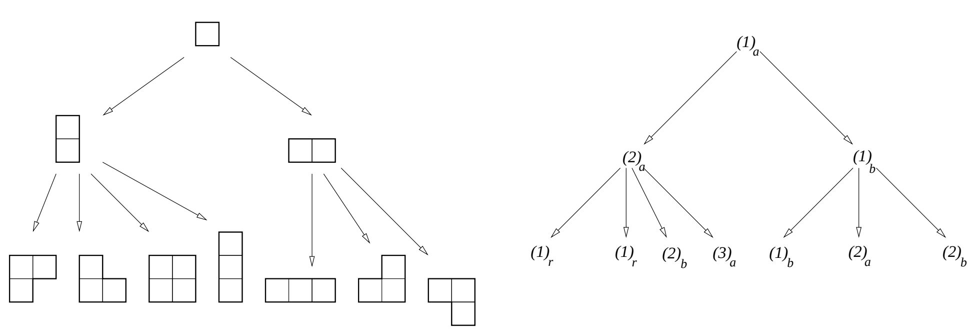

Figure 6.7: (a) The first levels of the generating tree of the ECO operator ϑ; (b) the first levels of the generating tree of Ω.

Proposition 6.1.1. *Let p denote the semi-perimeter of a convex polyomino, then $\Sigma = (\mathcal{C}, p, \vartheta, \Omega)$ is an ECO-system.*

Proof. The operator ϑ satisfies the conditions of Proposition 1.3.1: for each polyomino P' of size $n+3$ there is only one polyomino P, of size $n+2$, generating the first one through ϑ. Let us denote $R(P')$ the rightmost column of P'. Then we have:

i) $P' \in C_a$: then P is obtained by removing the uppermost cell of $R(P')$ and it belongs to C_a or C_b.

ii) $P' \in C_b$: let i, j be respectively the number of cells of $R(P')$ and the number of cells of the column on the left of $R(P')$. Then we have the following cases:

- $i = j$: then P is obtained by removing $R(P')$ and it belongs to C_a or C_b.
- $i \neq j$: then P is obtained by removing the cell at the bottom of $R(P')$ and it belongs to C_b.

iii) $P' \in C_r$:

- the lowest (resp. uppermost) cell of $R(P')$, with minimal (resp. maximal) ordinate, has the same ordinate of the lowest (resp. uppermost) cell of the column on the left of $R(P')$: then P is obtained by removing $R(P')$ and it can belong to C_a, C_b, or C_r.
- otherwise P is obtained by removing the lowest (resp. uppermost) cell of $R(P')$ with minimal (resp. maximal) ordinate and it belongs to C_r.

iv) $P' \in C_g$: then P is obtained by removing $R(P')$ and it belongs to C_g.

It remains to prove that there is only one P having P' as image. This can be easily deduced from the construction. □

6.2 The decomposition approach

Our aim is to determine the generating function of the rule Ω by extending the approach introduced in [50]: we treat succession rules by means of noncommutative formal power series.

Each convex polyomino is univocally identified by a node N of the generating tree of the rule Ω, and this node can be encoded by a word in the infinite alphabet $\Sigma = \{(i)_a, (j)_b, (h)_g, (l)_r : i, j, h, l \in \mathbb{N}^+\}$. Such a word is naturally defined by the sequence of labels of the nodes starting from the root and ending at N. As an example, the polyomino depicted in Figure 6.8 is encoded by the word $(1)_a(1)_b(2)_b(3)_a(3)_b(4)_a(5)_a(3)_r(2)_g(2)_g$.

Naturally, due to the form of the productions of the rule Ω, some convex polyominoes have the same word representation. For example the word $(1)_a(2)_a(1)_r$ represents two polyominoes of size 4, as the reader can easily verify in Figure 6.7.

Formally, let L_Ω be the set of words, over Σ, beginning by $(1)_a$ and satisfying the productions of Ω. Each word w of L_Ω corresponds to at least one path in the generating tree of Ω. We denote by S_Ω the noncommutative formal power series:

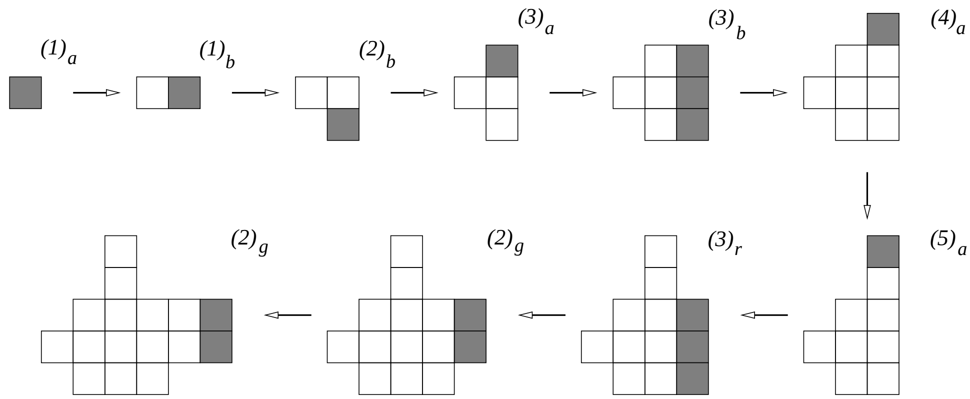

Figure 6.8: The ECO construction of a convex polyomino and the corresponding word.

$$S_\Omega = \sum_{w \in L_\Omega} m(w)w,$$

where $m(w)$ is the number of paths corresponding to w in the generating tree of Ω. The generating function of the noncommutative formal power series S_Ω is

$$S_\Omega(x) = \sum_{n>0} f_n x^n,$$

where:

$$f_n = \sum_{w \in L_\Omega \ |w|=n} m(w).$$

By construction we have that

$$S_\Omega(x) = x f_\Omega(x).$$

For example, we have

$$S_\Omega = (1)_a + (1)_a(1)_b + (1)_a(2)_a + (1)_a(1)_b(1)_b + (1)_a(1)_b(2)_a + (1)_a(1)_b(2)_b +$$

$$2 \cdot (1)_a(2)_a(1)_r + (1)_a(2)_a(2)_b + (1)_a(2)_a(3)_a + \dots$$

$$f_\Omega(x) = 1 + 2x + 7x^2 + 28x^3 + 122x^4 + \dots$$

We work on the series S_Ω using the standard operations on noncommutative formal power series, in particular, for any positive integer n, and $(i)_j \in \Sigma$:

$$nS_\Omega = \sum_{w \in L_\Omega} (nm(w))w,$$

$$(i)_j S_\Omega = \sum_{w \in L_\Omega} m(w)(i)_j w.$$

We introduce the operation $\oplus$: for any word $u = (i_1)_{j_1}(i_2)_{j_2}\dots(i_k)_{j_k}$ of L_Ω, $u^\oplus$ will denote the word:

$$(i_1+1)_{j_1}(i_2+1)_{j_2}\dots(i_k+1)_{j_k}.$$

For example $((1)_a(2)_a(1)_r)^\oplus = (2)_a(3)_a(2)_r$. We call $L_\Omega^\oplus$ the set of the words $u^\oplus$, with $u \in L_\Omega$. Moreover

$$S_\Omega^\oplus = \sum_{w\in L_\Omega} (m(w))w^\oplus.$$

It is clear that $S_\Omega^\oplus$ and S_Ω have the same generating function.

Generally speaking, a noncommutative formal power series S_δ can be associated with any succession rule δ in a completely analogous way.

Catalan succession rule. To fully understand the heart of the matter, we start presenting an example. Let us consider the pseudo-succession rule defining Catalan numbers:

$$\Gamma \begin{cases} (1) \\ \\ (k) \rightsquigarrow (1)(2)\dots(k+1), \end{cases} \tag{6.2}$$

Let $C = S_\Gamma$ be the noncommutative formal power series associated with the language L_Γ of the words of Γ. In practice:

$$C = (1) + (1)(1) + (1)(2) + (1)(1)(1) + (1)(1)(2) + (1)(2)(1) + (1)(2)(2) + (1)(2)(3) + \dots$$

Let us prove that:

$$C = (1) + (1)\,C + (1)\,C^\oplus + (1)\,C^\oplus\,C. \tag{6.3}$$

Indeed, a word $w \in L_\Gamma$ has one of the following forms:

- $|w| = 1$.
- $w = (1)v$ where :

  1) v begins with (1). Then the contribution of this set of words is

  $$\sum_w m(w)w = (1)\sum_{v\in L_\Gamma} m(v)v = (1)C.$$

  2) $v = (2)(u_1)\dots(u_k)$, with $u_i > 1$, for $i \in \{1,\dots,k\}$. Then

  $$\sum_w m(w)w = (1)\sum_{v\in L_\Gamma^\oplus} m(v)v = (1)C^\oplus.$$

3) $v = (2)(u_1)\dots(u_k)w_2$, where $u_i > 1$, for $i \in \{1,\dots,k\}$, and w_2 begins with (1). Then

$$\sum_w m(w)w = (1)\sum_{v\in L_\Gamma^{\oplus}L_\Gamma} m(v)v = (1)C^{\oplus}C.$$

Remark 6.2.1. *C can be viewed as an object grammar for the class of the words of the generating tree of Ω. Such a grammar has one terminal object, two unary operations, and one binary operation. Therefore it is isomorphic to the grammar for parallelogram polyominoes introduced in Subsection 4.1.2. This is not surprising since parallelogram polyominoes are enumerated by Catalan numbers according to their semi-perimeter.*

By taking the commutative image of C, we immediately derive a functional equation satisfied by the generating function $C(x)$ of C:

$$C(x) = x + x\,C(x) + x\,C(x) + x\,C(x)^2. \tag{6.4}$$

Consequently $C(x) = \frac{1-2x-\sqrt{1-4x}}{2x}$.

6.3 The decomposition for a partial rule: direct convex polyominoes

We first present hereafter a detailed description of the calculus of the generating function for a succession rule less complex than (6.1). Let us consider the succession rule Ω' with axiom $(1)_r$, and whose productions are the same as Ω, i.e. those defined in (6.1). In practice:

$$\Omega' \begin{cases} (1)_r \\ (k)_g \rightsquigarrow \prod_{j=1}^{k}(j)_g^{k-j+1} \\ (k)_r \rightsquigarrow \prod_{j=1}^{k-1}(j)_g^{k-j}\ \prod_{j=1}^{k+1}(j)_r. \end{cases} \tag{6.5}$$

The rule Ω' encodes the ECO construction for directed convex polyominoes, obtained by restricting the construction of Section 6.1 to the classes C_r and C_g. Then Ω' defines central binomial coefficients, $\binom{2n}{n}$. Our aim, in this paragraph, is to give a proof of this fact by computing the generating function of Ω'. This will remarkably simplify the complete calculus of $f_\Omega(x)$.

As usual, let us denote $L_{\Omega'}$ the set of the words produced by Ω', and

$$R = S_{\Omega'} = \sum_{w\in L_{\Omega'}} m(w)w.$$

The main theorem is preceded by two technical lemmas.

Lemma 6.3.1. *In the succession rule Ω', the label $(k_2)_{j_2}$ is produced by $(k_1)_{j_1}$ if and only if $(k_2-1)_{j_2}$ is produced by $(k_1-1)_{j_1}$, with $k_1, k_2 > 1$.*

One can easily verify the above lemma by observing the productions of Ω'. $\square$

Lemma 6.3.2. *Let* $L_P = \{u = (2)_r(u_2)_{j_2} \dots (u_k)_{j_k} \,|\, u_i > 1\,,$ *for* $i \in \{2, \dots k\},$ *and* $(1)_r u \in L_{\Omega'}\}$. *Then* $L_P = L^{\oplus}_{\Omega'}$.

Proof. ($\Rightarrow$) Let $u = (2)_r(u_2)_{j_2} \dots (u_k)_{j_k} \in L_P$. By definition of $L^{\oplus}_{\Omega'}$, the result can be achieved by proving that $u^{\ominus} = (1)_r(u_2 - 1)_{j_2} \dots (u_k - 1)_{j_k} \in L_{\Omega'}$. We proceed by induction on the length of $u^{\ominus}$.

Base: if $|u^{\ominus}| = 1$ the result immediately follows;

Step $n \to n+1$**:** let $u = (2)_r(u_2)_{j_2} \dots (u_n)_{j_n}(u_{n+1})_{j_{n+1}} \in L_P$. By inductive hypothesis, the word $(1)_r(u_2 - 1)_{j_2} \dots (u_n - 1)_{j_n}$ belongs to $L_{\Omega'}$. By Lemma 6.3.1, the label $(u_{n+1} - 1)_{j_{n+1}}$ is produced by the label $(u_n - 1)_{j_n}$ according the productions of the rule Ω'. Consequently $u^{\ominus} \in L_{\Omega'}$.

($\Leftarrow$) The result can be achieved again by induction. $\square$

Theorem 6.3.1. *The noncommutative formal power series* R *can be decomposed into the following sum:*

$$(1)_r + (1)_r R + (1)_r R^{\oplus} + (1)_r C^{\oplus} R + (1)_r P^{\oplus} G + (1)_r Q^{\oplus} G, \tag{6.6}$$

where

$$\begin{aligned} C = &\ \ (1)_r + (1)_r C + (1)_r C^{\oplus} + (1)_r C^{\oplus}\, C \\ G = &\ \ (1)_g + (1)_g G \\ P = &\ \ (1)_r + (1)_r P + (1)_r C^{\oplus} P + (1)_r P^{\oplus} + (1)_r C^{\oplus} \\ Q = &\ \ (1)_r Q + (1)_r C^{\oplus} Q + 2(1)_r P^{\oplus} G + 2(1)_r Q^{\oplus} G + \\ &\ \ (1)_r Q^{\oplus} \ + \ (1)_r (R - C)^{\oplus}. \end{aligned} \tag{6.7}$$

Proof. In order to let the reader have a better comprehension of the role of each term of the sum in (6.6), we depict in Figure 6.9 the first levels of the generating tree of Ω'.

Let w be a word of $L_{\Omega'}$. The general idea of the proof is to decompose the tree at the first return to label (1). We have the following cases:

-$|w| = 1$, then $w = (1)_r$.

- $|w| > 1$ then $w = (1)_r v$, and we distinguish the following cases:

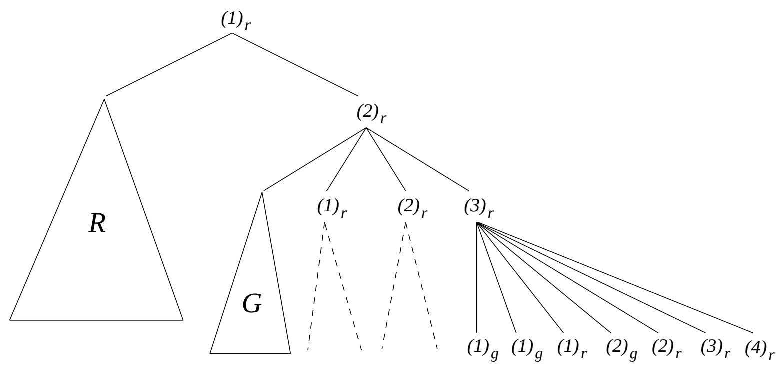

Figure 6.9: The first levels of the generating tree of Ω'.

1) v begins with $(1)_r$. The set of words in $L_{\Omega'}$ having the form $w = (1)_r v$ is then equal to $(1)_r L_{\Omega'}$. Consequently

$$\sum_{w=(1)_r v} m(w)w = (1)_r R.$$

For example, the word $(1)_r(1)_r(1)_r(2)_r(1)_r(1)_r(2)_r(3)_r(1)_r$ is a term of $(1)_r R$.

2) v begins with $(2)_r$. Four cases are possible:

a) $v \in L_P$, where L_P is defined in Lemma 6.3.2. Lemma 6.3.2 holds that $L_P = L^{\oplus}_{\Omega'}$. Consequently

$$\sum_{w=(1)_r v,\ v\in L_P} m(w)w = (1)_r \sum_{v\in L_P} m(v)v = (1)_r R^{\oplus},$$

For example, the word $(1)_r(2)_r(3)_r(4)_r(5)_r(3)_g(3)_g(2)_g$ is a term of $(1)_r R^{\oplus}$.

b) $v = (2)_r(u_2)_r \dots (u_k)_r(1)_r w_2$, with $u_i > 1$ for $i \in \{1, \dots k\}$.
One can easily verify that the language of words having the form $(2)_r(u_2)_r \dots (u_k)_r$, $u_i > 1$, coincides with $L^{\oplus}_{\Gamma}$, where L_Γ is the language of Catalan words. Consequently, summing over the set of such words w leads to:

$$\sum_w m(w)w = (1)_r \sum_{v\in L^{\oplus}_{\Gamma} L_{\Omega'}} m(v)v = (1)_r C^{\oplus} R.$$

The word $(1)_r(2)_r(3)_r(2)_r(1)_r(2)_r(3)_r(2)_g(2)_g(1)_g$ is a term of $(1)_r C^{\oplus} R$.

c) $v = (2)_r(u_2)_r \dots (u_k)_r g_1$, where $u_i > 1$, for $i \in \{2, \dots, k\}$, and $g_1 \in L_G = \{(1)_g, (1)_g(1)_g, (1)_g(1)_g(1)_g, (1)_g(1)_g(1)_g(1)_g, \dots\} = (1)_g^+$. Using the considerations in step *b*), the sum over all the words of this type leads to

$$\sum_w m(w)w = (1)_r \sum_{v\in L^{\oplus}_{\Gamma} L_G} m(v)v. \tag{6.8}$$

In this case, for any word v, the value $m(v)$ depends on u_k: more precisely, according to the productions of Ω', if $u_k = j$,

$$m(v) = (j-1) \cdot m\left((2)_r (u_2)_r \dots (j)_r\right) \cdot m(g_1). \tag{6.9}$$

Let us denote by $L_{\Gamma(i)}$ the language of the words of L_Γ ending with a label $(i)_r$. Using equations (6.8),(6.9) the sum over the words w of such form leads to:

$$\sum_w m(w)w = (1)_r \sum_{j\geq 2} \left((j-1) \cdot \sum_{v \in L_{\Gamma^{\oplus}_{(j-1)}}} m(v)v \right) \sum_{v\in L_G} m(v)v.$$

Let

$$C_{(j-1)} = \sum_{v\in L_{\Gamma(j-1)}} m(v)v \text{ and } G = \sum_{v \in L_G} m(v)v,$$

then (6.8) becomes

$$\sum_w m(w)w = (1)_r \sum_{j\geq 2} (j-1) C^{\oplus}_{(j-1)} G = (1)_r P^{\oplus} G,$$

where $P = \sum_{j\geq 2}(j-1)C_{(j-1)}$. The word

$$(1)_r(2)_r(3)_r(4)_r(2)_r(3)_r(1)_g(1)_g(1)_g$$

is an example of a term of $(1)_r P^{\oplus} G$.

d) $v = (2)_r (u_2)_{j_2} \dots (u_k)_{j_k} g_1$, with $u_i > 1$, for $i \in \{2, \dots, k\}$, $j_k = g$, and $g_1 \in L_G$. Considerations analogous to those in step *c)* show that the sum over all words w of this type leads to:

$$\sum_w m(w)w = (1)_r Q^{\oplus} G,$$

where

$$Q = \sum_{j\geq 2} j\ R_{(j-1)_g} \text{ and } R_{(j-1)_g} = \sum_{v\in L_{\Omega'(j-1)_g}} m(v)v,$$

$L'_{\Omega(j-1)_g}$ being the set of words belonging to $L_{\Omega'}$ and ending with $(j-1)_g$. The word $(1)_r(2)_r(3)_r(4)_r(2)_r(3)_r(2)_g(2)_g(1)_g(1)_g$ is an example of a term of $(1)_r Q^{\oplus} G$.

It is easy to verify that the decomposition (6.6) takes into account all the words that satisfy the succession rule Ω'.

To conclude the proof we must verify that the noncommutative formal power series C, G, P, and Q satisfy the system of equations (6.7). The statement is obvious for C and G. Below, we will prove that P satisfies:

$$P = (1)_r + (1)_r P + (1)_r C^{\oplus} P + (1)_r P^{\oplus} + (1)_r C^{\oplus}.$$

We first recall that $P = \sum_{j\geq 2}(j-1)C_{(j-1)}$. In view of the noncommutative equation holding for C we deduce that:

$$\begin{aligned} C_{(i)} = &\quad (1)_r\, C_{(i)} + (1)_r\, C^{\oplus}_{(i-1)} + (1)_r\, C^{\oplus}\, C_{(i)} \qquad \text{for } i > 1, \\ C_{(1)} = &\quad (1)_r + (1)_r\, C_{(1)} + (1)_r\, C^{\oplus}\, C_{(1)}. \end{aligned} \tag{6.10}$$

Consequently

$$\begin{aligned} P &= C_{(1)} + \textstyle\sum_{j\geq 3}(j-1)C_{(j-1)} \\ &= (1)_r + (1)_r\, C_{(1)} + (1)_r\, C^{\oplus}\, C_{(1)} + (1)_r \textstyle\sum_{j\geq 3}(j-1)\, C_{(j-1)} + \\ &\quad + (1)_r \textstyle\sum_{j\geq 3}(j-1)\, C^{\oplus}_{(j-2)} + (1)_r \textstyle\sum_{j\geq 3}(j-1)\, C^{\oplus}\, C_{(j-1)}. \end{aligned}$$

By performing some algebraic manipulations we obtain

$$\begin{aligned} P &= (1)_r + (1)_r P + (1)_r C^{\oplus} P + (1)_r \textstyle\sum_{j\geq 3}(j-1)\, C^{\oplus}_{(j-2)} \\ &= (1)_r + (1)_r P + (1)_r C^{\oplus} P + (1)_r P^{\oplus} + (1)_r C^{\oplus}. \end{aligned}$$

A similar proof holds for Q. Let us recall that $Q = \sum_{j\geq 2} j\, R_{(j-1)_g}$. From the noncommutative equation holding for R we deduce that:

$$R_g = (1)_r R_g + (1)_r R_g^{\oplus} + (1)_r C^{\oplus} R_g + (1)_r P^{\oplus} G + (1)_r Q^{\oplus} G, \tag{6.11}$$

consequently

$$\begin{aligned} R_{(i)_g} = &\quad (1)_r\, R_{(i)_g} + (1)_r\, R^{\oplus}_{(i-1)_g} + (1)_r C^{\oplus} R_{(i)_g} \qquad \text{for } i > 1, \\ R_{(1)_g} = &\quad (1)_r R_{(1)_g} + (1)_r\, C^{\oplus}\, R_{(1)_g} + (1)_r P^{\oplus} G + (1)_r Q^{\oplus} G. \end{aligned} \tag{6.12}$$

Then

$$\begin{aligned} Q &= 2\, R_{(1)_g} + \textstyle\sum_{j\geq 3} j\, R_{(j-1)_g} \\ &= 2(1)_r R_{(1)_g} + 2(1)_r\, C^{\oplus}\, R_{(1)_g} + 2(1)_r P^{\oplus} G + 2(1)_r Q^{\oplus} G + \\ &\quad + (1)_r \textstyle\sum_{j\geq 3} j\, R_{(j-1)_g} + (1)_r \textstyle\sum_{j\geq 3} j R^{\oplus}_{(j-2)_g} + (1)_r C^{\oplus} \textstyle\sum_{j\geq 3} j R_{(j-1)_g}. \end{aligned}$$

By performing some algebraic manipulations we obtain that

$$\begin{aligned} Q &= (1)_r\, Q + (1)_r\, C^{\oplus}\, Q + 2(1)_r P^{\oplus} G + 2(1)_r Q^{\oplus} G + (1)_r \textstyle\sum_{j\geq 3} j R^{\oplus}_{(j-2)_g} \\ &= (1)_r\, Q + (1)_r\, C^{\oplus}\, Q + 2(1)_r P^{\oplus} G + 2(1)_r Q^{\oplus} G + (1)_r \textstyle\sum_{j\geq 2} j R^{\oplus}_{(j-1)_g} + (1)_r \textstyle\sum_{j\geq 2} R^{\oplus}_{(j-1)_g} \\ &= (1)_r\, Q + (1)_r\, C^{\oplus}\, Q + 2(1)_r P^{\oplus} G + 2(1)_r Q^{\oplus} G + (1)_r\, Q^{\oplus} + (1)_r\, (R-C)^{\oplus}. \end{aligned}$$

$\square$

From Theorem 6.3.1 we immediately derive a system of functional equations,

$$\begin{aligned}
R(x) = \quad & x + x\ R(x) + x\ R(x) + x\ C(x)\ R(x) + x\ P(x)\ G(x) + \\
& x\ Q(x)\ G(x) \\
C(x) = \quad & x + x\ C(x) + x\ C(x) + x\ C^2(x) \\
G(x) = \quad & x + x\ G(x) \\
P(x) = \quad & x + x\ P(x) + x\ C(x)\ P(x) + x\ P(x) + x\ C(x) \\
Q(x) = \quad & x\ Q(x) + x\ C(x)\ Q(x) + 2x\ P(x)\ G(x) + 2x\ Q(x)\ G(x) + \\
& xQ(x)\ +\ x(R(x) - C(x))
\end{aligned} \tag{6.13}$$

In this system, $C(x) = \frac{1-2x-\sqrt{1-4x}}{2x}$, $G(x) = \frac{x}{1-x}$, and all other equations are linear in these two. Solving the system we get

$$R(x) = \frac{x}{\sqrt{1-4x}}$$

and the generating function of directed convex polyominoes is

$$f_P(x) = x^2 f_{\Omega'}(x) = xR(x) = \frac{x^2}{\sqrt{1-4x}}.$$

6.4 The decomposition for the complete rule Ω

Let A be the noncommutative formal power series associated with the succession rule Ω, i.e.

$$A = S_\Omega = \sum_{w \in L_\Omega} m(w)w.$$

Using the same strategies as in the previous case we manage to determine a decomposition for the series A, and then translate it into a system of equations. Let us call Ω'' the succession rule having the same productions of Ω and starting with the axiom $(1)_b$,

$$\Omega'' \begin{cases} (1)_b \\ (k)_g \rightsquigarrow \prod_{j=1}^{k} (j)_g^{k-j+1} \\ (k)_r \rightsquigarrow \prod_{j=1}^{k-1} (j)_g^{k-j}\ \prod_{j=1}^{k+1} (j)_r \\ (k)_a \rightsquigarrow \prod_{j=1}^{k-2} (j)_g^{k-j-1}\ \prod_{j=1}^{k-1} (j)_r^2\ (k)_b\ (k+1)_a \\ (k)_b \rightsquigarrow \prod_{j=1}^{k-2} (j)_g^{k-j-1}\ \prod_{j=1}^{k-1} (j)_r^2\ (k)_b\ (k+1)_a\ (k+1)_b, \end{cases} \tag{6.14}$$

We denote

$$B = \sum_{w \in L_{\Omega''}} m(w)w.$$

For S a subset $\{a, b, r, g\}$, we define $L_{\Omega(i)_S} = \cup_{q \in S} L_{\Omega(i)_q}$ (resp. $L_{\Omega''(i)_S} = \cup_{q \in S} L_{\Omega''(i)_q}$), where $L_{\Omega(i)_q}$ (resp. $L_{\Omega''(i)_q}$) denotes the sets of words of L_Ω (resp. $L_{\Omega''}$) ending with $(i)_q$, and $A_{(i)_S}$ (resp. $B_{(i)_S}$) denotes the corresponding noncommutative formal power series. Similarly we define $L_{\Omega S} = \cup_{q \in S} L_{\Omega q}$ (resp. $L_{\Omega'' S} = \cup_{q \in S} L_{\Omega'' q}$), where $L_{\Omega q}$ (resp. $L_{\Omega'' q}$) denotes the sets of words of L_Ω (resp. $L_{\Omega''}$) ending with color q, and A_S (resp. B_S) denotes the corresponding formal power series.

Lemma 6.4.1. *In the succession rules Ω and Ω'', the label $(k_2)_{j_2}$ is produced by $(k_1)_{j_1}$ if and only if $(k_2-1)_{j_2}$ is produced by $(k_1-1)_{j_1}$, with $k_1, k_2 > 1$.*

One can easily verify the above lemma by observing the productions of Ω and Ω''. □

Lemma 6.4.2. *Let $L_Q = \{u = (2)_a (u_2)_{j_2} \dots (u_k)_{j_k} \,|\, u_i > 1\,,$ for $i \in \{2, \dots k\}$, and $(1)_a u \in L_\Omega\}$. Then $L_Q = L_\Omega^\oplus$.*

Moreover, let $L_R = \{u = (2)_b (u_2)_{j_2} \dots (u_k)_{j_k} \,|\, u_i > 1\,,$ for $i \in \{2, \dots k\}$, and $(1)_b u \in L_{\Omega''}\}$. Then $L_R = L_{\Omega''}^\oplus$.

Proof. The proof is identical to that of Lemma 6.3.2. □

Theorem 6.4.1. *The noncommutative formal power series A can be decomposed into the following sum:*

$$\begin{aligned} A = \;& (1)_a + (1)_a\, B + (1)_a\, A^\oplus + 2(1)_a\, A^\oplus_{a,b}\, R + (1)_a\, A^\oplus_r\, R + \\ & (1)_a\, P^\oplus_A\, G + (1)_a\, Q^\oplus_A\, G + \\ & (1)_a\, S^\oplus_A\, G, \end{aligned} \tag{6.15}$$

where

$$\begin{aligned} B = \;& \xi(A) + (1)_b\, B^\oplus + 2(1)_b\, B^\oplus_{a,b}\, R + (1)_b\, B^\oplus_r\, R + \\ & (1)_b\, P^\oplus_B G + (1)_b\, Q^\oplus_B G + \\ & (1)_b\, S^\oplus_B G, \end{aligned} \tag{6.16}$$

$$
\begin{aligned}
A_{a,b} =\ & (1)_a + (1)_a\ B_{a,b} + (1)_a\ A^{\oplus}_{a,b} \\
B_{a,b} =\ & \xi(A_{a,b}) + (1)_b\ B^{\oplus}_{a,b} \\
A_r =\ & (1)_a\ B_r + (1)_a\ A^{\oplus}_r + 2(1)_a\ A^{\oplus}_{a,b}\ C + (1)_a\ A^{\oplus}_r\ C \\
B_r =\ & \xi(A_r) + (1)_b\ B^{\oplus}_r + 2(1)_b\ B^{\oplus}_{a,b}\ C + (1)_b\ B^{\oplus}_r\ C, \\
A_g =\ & (1)_a\ B_g + (1)_a\ A^{\oplus}_g + 2(1)_a\ A^{\oplus}_{a,b}\ R_g + (1)_a\ A^{\oplus}_r\ R_g + \\
& (1)_a\ P^{\oplus}_A\ G + (1)_a\ Q^{\oplus}_A\ G + (1)_a\ S^{\oplus}_A\ G \\
B_g =\ & \xi(A_g) + (1)_b\ B^{\oplus}_g + 2(1)_b\ B^{\oplus}_{a,b}\ R_g + (1)_b\ B^{\oplus}_r\ R_g + \\
& (1)_b\ P^{\oplus}_B\ G + (1)_b\ Q^{\oplus}_B\ G + (1)_b\ S^{\oplus}_B\ G,
\end{aligned} \tag{6.17}
$$

$$
\begin{aligned}
P_A =\ & (1)_a\ P_B + (1)_a\ P^{\oplus}_A + (1)_a\ A^{\oplus}_{a,b} \\
P_B =\ & \xi(P_A) + (1)_b\ P^{\oplus}_B + (1)_b\ B^{\oplus}_{a,b} \\
Q_A =\ & (1)_a\ Q_B + 2(1)_a\ A^{\oplus}_{a,b}\ P + (1)_a\ A^{\oplus}_r\ P + (1)_a\ Q^{\oplus}_A + (1)_a\ A^{\oplus}_r \\
Q_B =\ & \xi(Q_A) + 2(1)_b\ B^{\oplus}_{a,b}\ P + (1)_b\ B^{\oplus}_r\ P + (1)_b\ Q^{\oplus}_B + (1)_b\ B^{\oplus}_r \\
S_A =\ & (1)_a\ S_B + 2(1)_a\ A^{\oplus}_{a,b}\ Q + (1)_a\ A^{\oplus}_r\ Q + 2(1)_a\ P^{\oplus}_A\ G\ + 2(1)_a\ Q^{\oplus}_A\ G + \\
& 2(1)_a\ S^{\oplus}_A\ G\ + (1)_a\ S^{\oplus}_A + (1)_a\ A^{\oplus}_g \\
S_B =\ & \xi(S_A) + 2(1)_b\ B^{\oplus}_{a,b}\ Q + (1)_b\ B^{\oplus}_r\ Q + 2(1)_b\ P^{\oplus}_B\ G + 2(1)_b\ Q^{\oplus}_B\ G + \\
& 2(1)_b\ S^{\oplus}_B\ G + (1)_b\ S^{\oplus}_B + (1)_b\ B^{\oplus}_g,
\end{aligned} \tag{6.18}
$$

where R, G, P, Q, and R_g are the equations introduced for the rule Ω', and $\xi(A)$ is the formal power series obtained from A by replacing, in each term, the first occurrence of $(1)_a$ with $(1)_b$.

Proof. We first compute the equations for A and B. In Figure 6.10 are depicted the first levels of the generating trees of the rules Ω and Ω''.

The equation for A Let w be a word of L_Ω. Then we have the following cases:

-$|w| = 1$, then $w = (1)_a$.

- $|w| > 1$ then $w = (1)_a v$, and we distinguish the following cases:

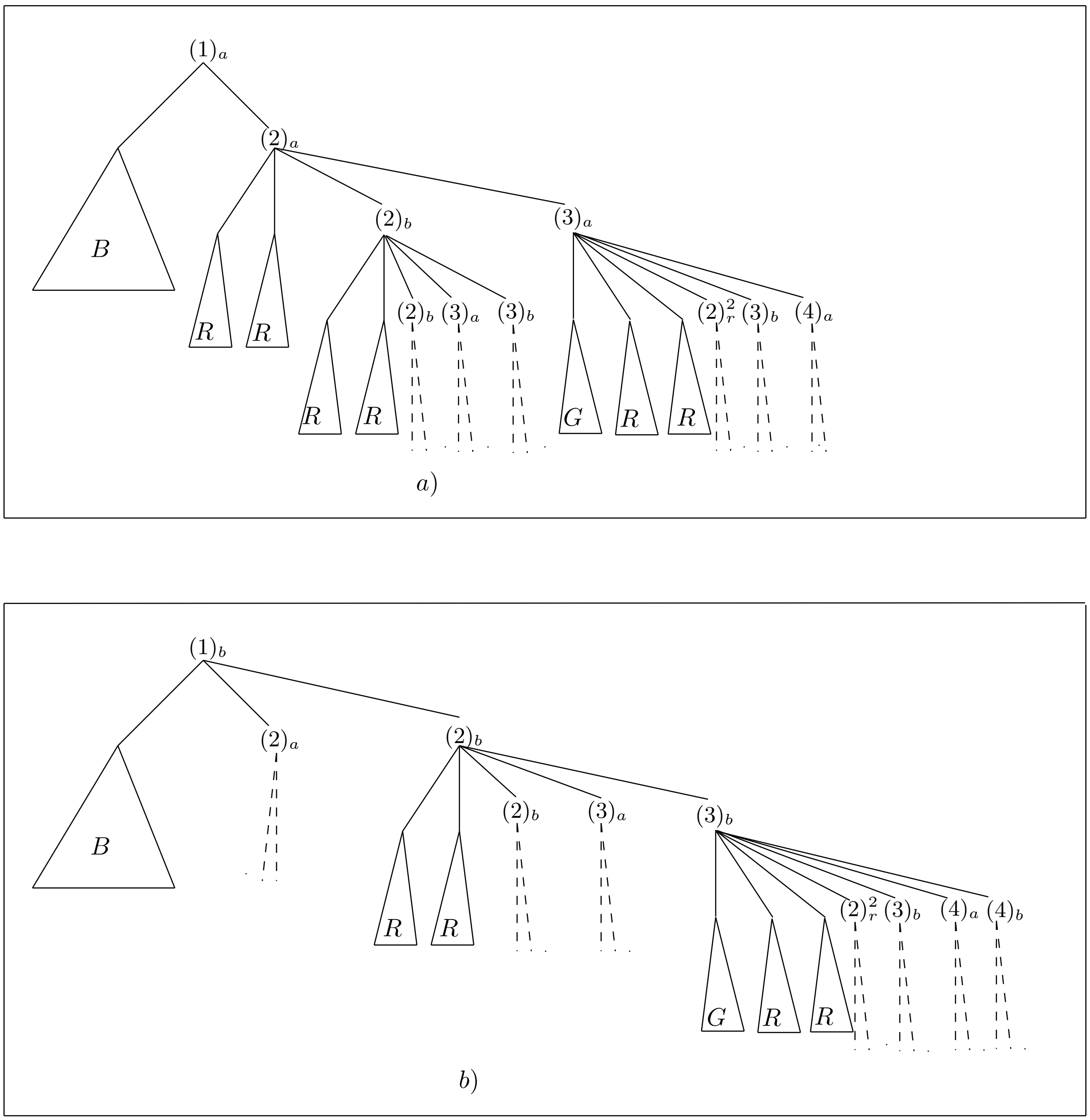

Figure 6.10: The first levels of Ω (a) and Ω'' (b).

1) v begins with $(1)_b$. The set of words in L_Ω having the form $w = (1)_a v$ is then equal to $(1)_a L_{\Omega''}$. Consequently
$$\sum_w m(w)w = (1)_a B.$$
For example, the word $(1)_a(1)_b(2)_a(1)_r(2)_r(3)_r(2)_g(2)_g(1)_g$ is a term of $(1)_a B$.

2) v begins with $(2)_a$. Six cases are possible:

a) $v \in L_Q$, L_Q being defined in Lemma 6.4.2. From Lemma 6.4.2 it holds that $L_Q = L_\Omega^\oplus$. Consequently
$$\sum_w m(w)w = (1)_a \sum_{v\in L_Q} m(v)v = (1)_a\ A^\oplus.$$
The word $(1)_a(2)_a(3)_a(3)_b(2)_r(3)_r(2)_g$ is a term of $(1)_a\ A^\oplus$.

b) $v = (2)_a(u_2)_{j_2}\dots(u_k)_{j_k}(1)_r w_2$, with $u_i > 1$ for $i \in \{1,\dots k\}$ and $j_k \in \{a,b\}$. The set of words in L_Ω having the form $(1)_a\ v$, is then equal to $(1)_a\ L_{\Omega^\oplus_{a,b}} L_{\Omega'}$. Consequently
$$\sum_w m(w)w = (1)_a \sum_{v\in L_{\Omega^\oplus_{a,b}} L_{\Omega'}} m(v)v = 2(1)_a\ A^\oplus_{a,b}R.$$
The word $(1)_a(2)_a(3)_a(3)_b(3)_b(4)_a(1)_r(2)_r(1)_g(1)_g$ is a term of $(1)_a\ A^\oplus_{a,b}R$.

c) $v = (2)_a(u_2)_{j_2}\dots(u_k)_r(1)_r w_2$, with $u_i > 1$ for $i \in \{1,\dots k\}$. The set of words in L_Ω having the form $(1)_a\ v$, is then equal to $(1)_a\ L_{\Omega^\oplus_r}$. Consequently
$$\sum_w m(w)w = (1)_a \sum_{v\in L_{\Omega^\oplus_r} L_{\Omega'}} m(v)v = (1)_a\ A^\oplus_r R.$$
The word $(1)_a(2)_a(3)_a(3)_b(3)_b(2)_r(1)_r(2)_r(1)_g(1)_g$ is a term of $(1)_a\ A^\oplus_r R$.

d) $v = (2)_a(u_2)_{j_2}\dots(u_k)_{j_k} g_1$, where $u_i > 1$, for $i \in \{2,\dots,k\}$, $j_k \in \{a,b\}$ and $g_1 \in L_G = (1)_g^+$. From the assertions of point *b*), we have
$$\sum_w m(w)w = (1)_a \sum_{v\in L_{\Omega^\oplus_{a,b}} L_G} m(v)v.$$
The value $m(v)$ also depends on the value u_k: if $u_k = j+1$ then, according to the rule Ω,
$$m(v) = (j-1)\ m\left((2)_a(u_2)_{j_2}\dots(j+1)_{j_k}\right)\ m(g_1).$$
Then we have:
$$\sum_w m(w)w = (1)_a \sum_{j\ge 2}(j-1) \sum_{v\in L_{\Omega^\oplus_{(j)_{a,b}}}} m(v)v \sum_{v\in L_G} m(v)v,$$
and consequently
$$\sum_w m(w)w = (1)_a \sum_{j\ge 2}(j-1)\ A^\oplus_{(j)_{a,b}} G.$$

By denoting $P_A = \sum_{j\geq 2} (j-1)\ A_{(j)_{a,b}}$ we have

$$\sum_w m(w)w = (1)_a\ P_A^{\oplus} G.$$

The word $(1)_a(2)_a(3)_a(3)_b(4)_b(1)_g(1)_g$ is a term of $(1)_a\ P_A^{\oplus} G$.

e) $v = (2)_a(u_2)_{j_2}\dots(u_k)_r g_1$, where $u_i > 1$, for $i \in \{2,\dots,k\}$, and $g_1 \in L_G = (1)_g^+$. From the assertions of point *c*), we have

$$\sum_w m(w)w = (1)_a \sum_{v\in L\Omega_r^{\oplus} L_G} m(v)v.$$

Also in this case the value $m(v)$ depends on the value u_k: if $u_k = j$ then, according to the rule Ω,

$$m(v) = (j-1)\ m\left((2)_a(u_2)_{j_2}\dots(j)_r\right)\ m(g_1).$$

Then we have:

$$\sum_w m(w)w = (1)_a \sum_{j\geq 2} (j-1) \sum_{v\in L\Omega^{\oplus}_{(j-1)_r}} m(v)v \sum_{v\in L_G} m(v)v,$$

and consequently

$$\sum_w m(w)w = (1)_a \sum_{j\geq 2} (j-1)\ A^{\oplus}_{(j-1)_r} G.$$

By denoting $Q_A = \sum_{j\geq 2}(j-1)\ A_{(j-1)_r}$ we have

$$\sum_w m(w)w = (1)_a\ Q_A^{\oplus} G.$$

The word $(1)_a(2)_a(3)_a(3)_b(4)_b(2)_r(1)_g$ is a term of $(1)_a\ Q_A^{\oplus} G$.

f) $v = (2)_a(u_2)_{j_2}\dots(u_k)_g g_1$, where $u_i > 1$, for $i \in \{2,\dots,k\}$, and $g_1 \in L_G = (1)_g^+$. The set of words in L_Ω having the form $(1)_a\ v$, is then equal to $(1)_a\ L\Omega_g^{\oplus} L_G$. Consequently

$$\sum_w m(w)w = (1)_a \sum_{v\in L\Omega_g^{\oplus} L_G} m(v)v.$$

The value $m(v)$ also depends on the value u_k: if $u_k = j$ then, according to the rule Ω,

$$m(v) = (j)\ m\left((2)_a(u_2)_{j_2}\dots(j)_g\right)\ m(g_1). \tag{6.19}$$

Then we have:

$$\sum_w m(w)w = (1)_a \sum_{j\geq 2} (j) \sum_{v\in L\Omega^{\oplus}_{(j-1)_g}} m(v)v \sum_{v\in L_G} m(v)v,$$

and consequently

$$\sum_w m(w)w = (1)_a \sum_{j\geq 2} j\ A^{\oplus}_{(j-1)_g} G.$$

By denoting $S_A = \sum_{j\geq 2} j\ A_{(j-1)_g}$ we have

$$\sum_w m(w)w = (1)_a\ S^{\oplus}_A G.$$

The word $(1)_a(2)_a(3)_a(3)_b(4)_b(2)_g(1)_g(1)_g$ is a term of $(1)_a\ S^{\oplus}_A G$.

The equation for B Let

$$L_{\Omega'''} = \{(1)_b\,,\ (1)_b\ (2)_a\ v\,,\ (1)_b\ (1)_b\ v \mid (1)_b\ (2)_a\ v\,,\ (1)_b\ (1)_b\ v \in L_{\Omega''}\}.$$

Then the words of $L_{\Omega'''}$ and those of L_Ω differ only for the first letter. Let $\xi(A)$ be the formal power series obtained from A by replacing, in each term, the first occurrence of $(1)_a$ with $(1)_b$. Then $\xi(A)$ is the formal power series of the words of $L_{\Omega'''}$. Let $w \in L_{\Omega''}$, it remains to study the case that $w = (1)_b v$ where v begins with $(2)_b$ (see Figure 6.10). Then six cases are possible:

a) $v \in L_R$, L_R being defined in Lemma 6.4.2. From Lemma 6.4.2 it holds that $L_R = L^{\oplus}_{\Omega''}$. Consequently

$$\sum_w m(w)w = (1)_b \sum_{v\in L_R} m(v)v = (1)_b\ B^{\oplus}.$$

The word $(1)_b(2)_b(2)_b(3)_b(4)_a(3)_r(2)_g$ is a term of $(1)_b\ B^{\oplus}$.

b) $v = (2)_b(u_2)_{j_2}\dots(u_k)_{j_k}(1)_r w_2$, with $u_i > 1$ for $i \in \{1,\dots k\}$ and $j_k \in \{a,b\}$. The set of words in L_Ω having the form $(1)_b\ v$, is then equal to $(1)_b\ L''^{\oplus}_{\Omega a,b} L_{\Omega'}$. Consequently

$$\sum_w m(w)w = (1)_b \sum_{v\in L_{\Omega''}{}^{\oplus}_{a,b} L_{\Omega'}} m(v)v = 2(1)_b\ B^{\oplus}_{a,b} R.$$

The word $(1)_b(2)_b(3)_b(4)_a(3)_b(1)_r(2)_r(1)_g$ is a term of $(1)_b\ B^{\oplus}_{a,b} R$.

c) $v = (2)_b(u_2)_{j_2}\dots(u_k)_r(1)_r w_2$, with $u_i > 1$ for $i \in \{1,\dots k\}$. The set of words in $L_{\Omega''}$ having the form $(1)_b\ v$, is then equal to $(1)_b\ L_{\Omega''}{}^{\oplus}_r\ L_{\Omega'}$. Consequently

$$\sum_w m(w)w = (1)_b \sum_{v\in L_{\Omega''}{}^{\oplus}_r L_{\Omega'}} m(v)v = (1)_b\ B^{\oplus}_r R.$$

The word $(1)_b(2)_b(3)_b(4)_a(3)_b(4)_a(3)_r(1)_r(2)_r(1)_g$ is a term of $(1)_b\ B^{\oplus}_r R$.

d) $v = (2)_b(u_2)_{j_2}\dots(u_k)_{j_k} g_1$, where $u_i > 1$, for $i \in \{2,\dots,k\}$, $j_k \in \{a,b\}$ and $g_1 \in L_G = (1)^+_g$. By using the same arguments as in the calculus for A we obtain

$$\sum_w m(w)w = (1)_b \sum_{j\geq 2} (j-1) \sum_{v\in L_{\Omega''}{}^{\oplus}_{(j)_{a,b}}} m(v)v \sum_{v\in L_G} m(v)v,$$

and consequently

$$\sum_{w} m(w)w = (1)_b \sum_{j\geq 2}(j-1)\ B^{\oplus}_{(j)_{a,b}} G.$$

By denoting $P_B = \sum_{j\geq 2} (j-1)\ B_{(j)_{a,b}}$ we have

$$\sum_{w} m(w)w = (1)_b\ P^{\oplus}_B G.$$

The word $(1)_b(2)_b(3)_b(4)_a(3)_b(1)_g(1)_g$ is a term of $(1)_b\ \ P^{\oplus}_B G$.

e) $v = (2)_b(u_2)_{j_2}\dots(u_k)_r g_1$, where $u_i > 1$, for $i \in \{2,\dots,k\}$, and $g_1 \in L_G = (1)^+_g$.

$$\sum_{w} m(w)w = (1)_b \sum_{j\geq 2}(j-1) \sum_{v\in L_{\Omega''^{\oplus}_{(j-1)_r}}} m(v)v \sum_{v\in L_G} m(v)v,$$

and consequently

$$\sum_{w} m(w)w = (1)_b \sum_{j\geq 2}(j-1)\ B^{\oplus}_{(j-1)_r} G.$$

By denoting $Q_B = \sum_{j\geq 2}(j-1)\ B_{(j-1)_r}$ we have

$$\sum_{w} m(w)w = (1)_b\ Q^{\oplus}_B G.$$

The word $(1)_b(2)_b(3)_b(4)_a(3)_b(4)_a(3)_r(1)_g$ is a term of $(1)_b\ \ Q^{\oplus}_B G$.

f) $v = (2)_b(u_2)_{j_2}\dots(u_k)_g g_1$, where $u_i > 1$, for $i \in \{2,\dots,k\}$, and $g_1 \in L_G = (1)^+_g$.

$$\sum_{w} m(w)w = (1)_b \sum_{j\geq 2}(j) \sum_{v\in L_{\Omega''^{\oplus}_{(j-1)_g}}} m(v)v \sum_{v\in L_G} m(v)v,$$

and consequently

$$\sum_{w} m(w)w = (1)_b \sum_{j\geq 2} j\ B^{\oplus}_{(j-1)_g} G.$$

By denoting $S_B = \sum_{j\geq 2} j\ B_{(j-1)_g}$ we have

$$\sum_{w} m(w)w = (1)_b\ S^{\oplus}_B G.$$

The word $(1)_b(2)_b(3)_b(4)_a(3)_b(4)_a(3)_r(2)_g(1)_g(1)_g$ is a term of $(1)_b\ \ S^{\oplus}_B G$.

To conclude the proof we must verify that the remaining formal power series satisfy the equations of the systems (6.17) and (6.18). The equations of system (6.17) can be easily deduced from the equations for A and B. For instance the equation for $A_{a,b}$ is obtained from the equation for A, by observing that the terms ending with R and G do not contribute to $A_{a,b}$.

Now we compute the equations for P_A, Q_A, and S_A. We omit the calculus of P_B, Q_B, and S_B since the procedure is exactly the same. Let us recall that $P_A = \sum_{j\geq 2} (j-1)\ A_{(j)_{a,b}}$. From the equation for $A_{a,b}$ we have

$$A_{(i)_{a,b}} = (1)_a\ B_{(i)_{a,b}} + (1)_a\ A^{\oplus}_{(i-1)_{a,b}} \quad \text{for } i > 1.$$

Then by substituting $A_{(j)_{a,b}}$ we obtain

$$\begin{aligned} P_A &= (1)_a \sum_{j\geq 2} (j-1)\, B_{(j)_{a,b}} + (1)_a \sum_{j\geq 2} (j-1) A^{\oplus}_{(j-1)_{a,b}} \\ &= (1)_a\, P_B + (1)_a\, A^{\oplus}_{(1)_{a,b}} + (1)_a \sum_{j\geq 3} (j-1) A^{\oplus}_{(j-1)_{a,b}} \\ &= (1)_a\, P_B + (1)_a\, A^{\oplus}_{a,b} + (1)_a\, P^{\oplus}_A. \end{aligned}$$

Now, we compute $Q_A = \sum_{j\geq 2}(j-1)\, A_{(j-1)_r}$. From the equation for A_r we have

$$\begin{aligned} A_{(i)_r} &= (1)_a\, B_{(i)_r} + (1)_a\, A^{\oplus}_{(i-1)_r} + 2(1)_a\, A^{\oplus}_{a,b}\, C_(i) + (1)_a\, A^{\oplus}_r\, C_(i) \quad \text{if } i>1, \\ A_{(1)_r} &= (1)_a\, B_{(1)_r} + +2(1)_a\, A^{\oplus}_{a,b}\, C_(1) + (1)_a\, A^{\oplus}_r\, C_(1). \end{aligned} \tag{6.20}$$

By substituting $A_{(j-1)_r}$ in Q_A we have

$$\begin{aligned} Q_A = \quad & A_{(1)_r} + (1)_a\, \sum_{j\geq 3}(j-1)\;\, B_{(j-1)_r} + (1)_a\, \sum_{j\geq 3}(j-1) A^{\oplus}_{(j-2)_r} + \\ & +2(1)_a\, A^{\oplus}_{a,b} \sum_{j\geq 3}(j-1)\, C_{(j-1)} + (1)_a\, A^{\oplus}_r \sum_{j\geq 3}(j-1)\, C_{(j-1)}. \end{aligned}$$

Then, by substituting $A_{(1)_r}$ we have

$$\begin{aligned} Q_A &= (1)_a\, Q_B + 2(1)_a\, A^{\oplus}_{a,b}\, P + (1)_a\, A^{\oplus}_r\, P + (1)_a\, \sum_{j\geq 3}(j-1) A^{\oplus}_{(j-2)_r} \\ &= (1)_a\, Q_B + +2(1)_a\, A^{\oplus}_{a,b}\, P + (1)_a\, A^{\oplus}_r\, P + (1)_a Q^{\oplus}_A + (1)_a\, A^{\oplus}_r \end{aligned}$$

Finally we determine S_A. From the equation for A_g we have

$$\begin{aligned} A_{(i)_g} = \quad & (1)_a\, B_{(i)_g} + (1)_a\, A^{\oplus}_{(i-1)_g} + 2(1)_a\, A^{\oplus}_{a,b}\, R_{(i)_g} + (1)_a\, A^{\oplus}_r\, R_{(i)_g} \quad \text{if } i>1, \\ A_{(1)_g} = \quad & (1)_a\, B_{(1)_g} + 2(1)_a\, A^{\oplus}_{a,b}\, R_{(1)_g} + (1)_a\, A^{\oplus}_r\, R_{(1)_g} + (1)_a\, P^{\oplus}_A\, G + \\ & (1)_a\, Q^{\oplus}_A\, G + (1)_a\, S^{\oplus}_A\, G. \end{aligned} \tag{6.21}$$

We recall that $S_A = \sum_{j\geq 2} j\, A_{(j-1)_g}$. Then, by substituting $A_{(j-1)_g}$ we obtain

$$\begin{aligned} S_A = \quad & 2A_{(1)_g} + (1)_a \sum_{j\geq 3} j\, B_{(j-1)_g} + (1)_a\, \sum_{j\geq 3} j A^{\oplus}_{(j-2)_g} + \\ & 2(1)_a\, A^{\oplus}_{a,b}\, \sum_{j\geq 3} j\, R_{(j-1)_g} + (1)_a\, A^{\oplus}_r\, \sum_{j\geq 3} j\, R_{(j-1)_g}. \end{aligned}$$

Then, by substituting $A_{(1)_g}$ we have

$$\begin{aligned} S_A &= (1)_a\ S_B + (1)_a\ \sum_{j\geq 3} j A^{\oplus}_{(j-2)_g} + 2(1)_a\ A^{\oplus}_{a,b}\ Q + (1)_a\ A^{\oplus}_r\ Q + \\ &\quad +2(1)_a\ P^{\oplus}_A\ G\ + 2(1)_a\ Q^{\oplus}_A\ G + 2(1)_a\ S^{\oplus}_A\ G. \\ &= (1)_a\ S_B + 2(1)_a\ A^{\oplus}_{a,b}\ Q + (1)_a\ A^{\oplus}_r\ Q + 2(1)_a\ P^{\oplus}_A\ G + \\ &\quad +2(1)_a\ Q^{\oplus}_A\ G + 2(1)_a\ S^{\oplus}_A\ G + (1)_a\ S^{\oplus}_A + (1)_a\ A^{\oplus}_g. \end{aligned}$$

□

Finally, by taking the commutative image of the equations of the system obtained in Theorem 6.4.1, we have a system of functional equations and we can compute the generating function $A(x)$ of A. The system that we obtain is the following where $R(x)$, $C(x)$, $P(x)$, $Q(x)$, and $G(x)$ are already known from Subsection 6.3.

$$\begin{aligned} A(x) = \ & x\ +\ x\ B(x)\ +\ x\ A(x)\ +\ 2x\ A_{a,b}(x)\ R(x)\ +\ x\ A_r(x)\ R(x) + \\ & x\ P_A(x)\ G(x)\ +\ x\ Q_A(x)\ G(x) + \\ & x\ S_A(x)\ G(x), \end{aligned} \tag{6.22}$$

where

$$\begin{aligned} B(x) = \ & A(x)\ +\ x\ B(x)\ + 2x\ \ B_{a,b}(x)\ R(x) +\ x\ B_r(x)\ R(x) + \\ & x\ P_B(x)G(x) + x\ Q_B(x)G(x) + \\ & x\ S_B(x)G(x), \end{aligned} \tag{6.23}$$

$$\begin{aligned} A_{a,b}(x) = \ & x + x\ B_{a,b}(x) + x\ A_{a,b}(x) \\ B_{a,b}(x) = \ & A_{a,b}(x) + x\ B_{a,b}(x) \\ A_r(x) = \ & x\ B_r(x) + x\ A_r(x) + 2x\ A_{a,b}(x)\ C(x) + x\ A_r(x)\ C(x) \\ B_r(x) = \ & A_r(x) + x\ B_r(x) + 2x\ B_{a,b}(x)\ C(x) + x\ B_r(x)\ C(x), \\ A_g(x) = \ & x\ B_g(x) + x\ A_g(x) + 2x\ A_{a,b}(x)\ R_g(x) + x\ A_r(x)\ R_g(x) + \\ & x\ P_A(x)\ G(x) + x\ Q_A(x)\ G(x) + x\ S_A(x)\ G(x) \\ B_g(x) = \ & A_g(x) + x\ B_g(x) + 2x\ B_{a,b}(x)\ R_g(x) + x\ B_r(x)\ R_g(x) + \\ & x\ P_B(x)\ G(x) + x\ Q_B(x)\ G(x) + x\ S_B(x)\ G(x) \\ R_g(x) = \ & xR_g(x) + xR_g(x) + xC(x)R_g(x) + xP(x)G(x) + xQ(x)G(x), \end{aligned} \tag{6.24}$$

$$
\begin{aligned}
P_A(x) = \quad & x\ P_B(x) + x\ P_A(x) + x\ A_{a,b}(x) \\
P_B(x) = \quad & P_A(x) + x\ P_B(x) + x\ B_{a,b}(x) \\
Q_A(x) = \quad & x\ Q_B(x) + 2x\ A_{a,b}(x)\ P(x) + x\ A_r(x)\ P(x) + x\ Q_A(x) + x\ A_r(x) \\
Q_B(x) = \quad & Q_A(x) + 2x\ B_{a,b}(x)\ P(x) + x\ B_r(x)\ P(x) + x\ Q_B(x) + x\ B_r(x) \\
S_A(x) = \quad & x\ S_B(x) + 2x\ A_{a,b}(x)\ Q(x) + x\ A_r(x)\ Q(x) + 2x\ P_A(x)\ G(x)\ + 2x\ Q_A(x)\ G(x) + \\
& 2x\ S_A(x)\ G(x)\ + x\ S_A(x) + x\ A_g(x) \\
S_B(x) = \quad & S_A(x) + 2x\ B_{a,b}(x)\ Q(x) + x\ B_r(x)\ Q(x) + 2x\ P_B(x)\ G(x) + 2x\ Q_B(x)\ G(x) + \\
& 2x\ S_B(x)\ G(x) + x\ S_B(x) + x\ B_g(x).
\end{aligned}
\tag{6.25}
$$

In this system, the series $C(x)$, $R(x)$, $P(x)$, $Q(x)$, and $G(x)$ are known from the previous calculations and the equations for $A_{a,b}(x)$, $B_{a,b}(x)$, $A_r(x)$, and $B_r(x)$ are linear. Once these series have been obtained, all other equations are linear. Solving the system we get

$$
A(x) = \sum_{n\geq 0} f_n x^n = \frac{1 - 6x + 11x^2 - 4x^3 - 4x^2\sqrt{1-4x}}{(1-4x)^2}, \tag{6.26}
$$

then we determine the generating function for convex polyominoes,

$$
C(x) = x^2 f_\Omega(x) = xA(x).
$$

Bibliography

[1] C. Banderier, M. Bousquet-Mélou, A. Denise, P. Flajolet, D. Gardy, and D. Gouyou-Beauchamps. Generating functions for generating trees. *Discrete Mathematics*, 246:29–55, 2002.

[2] C. Banderier and P. Flajolet. Basic analytic combinatorics of directed lattice paths. *Theoretical Computer Science*, 281:37–80, 2002.

[3] E. Barcucci, A. Del Lungo, J.M. Fédou, and R. Pinzani. Steep polyominoes, q-Motzkin numbers and q-Bessel functions. *Discrete Mathematics*, 189:21–42, 1998.

[4] E. Barcucci, A. Del Lungo, A. Frosini, and S. Rinaldi. A technology for reverse-engineering a combinatorial problem from a rational generating function. *Advances in Applied Mathematics*, 26:129–153, 2001.

[5] E. Barcucci, A. Del Lungo, and E. Pergola. Random generation of trees and other combinatorial objects. *Theoretical Computer Science*, 218:219–232, 1999.

[6] E. Barcucci, A. Del Lungo, E. Pergola, and R. Pinzani. ECO: a methodology for the enumeration of combinatorial objects. *Journal of Difference Equations and Applications*, 5:435–490, 1999.

[7] E. Barcucci, A. Del Lungo, E. Pergola, and R. Pinzani. Some combinatorial interpretations of q-analogs of Schröder numbers. *Annals of Combinatorics*, 3:173–192, 1999.

[8] E. Barcucci, E. Pergola, R. Pinzani, and S. Rinaldi. ECO method and hill-free paths. In *Séminaire Lotharingien de Combinatoire*, volume 46, 2001. Article B46b.

[9] E. Barcucci and S. Rinaldi. Some linear recurrences and their combinatorial interpretation by means of regular languages. *Theoretical Computer Science*, 255:679–686, 2001.

[10] D. Beauquier and M. Nivat. On translating one polyomino to tile the plane. *Discrete Comput. Geom.*, 6:575–592, 1991.

[11] F. Bergeron, G. Labelle, and P. Leroux. *Theory of species and tree like structures.* Cambridge University Press, 1997.

[12] J. Bonin, L. Shapiro, and R. Simion. Some q-analogues of the Schröder numbers arising from combinatorial statistics on lattice paths. *Journal of Statistical Planning and Inference*, 34:35–55, 1993.

[13] M. Bousquet-Mélou. A method for the enumeration of various classes of column-convex polygons. *Discrete Mathematics*, 154:1–25, 1996.

[14] M. Bousquet-Mélou. Rapport scientifique d'habilitation à diriger des recherches. Rapport Technique, LaBRI, Université de Bordeaux I, Bordeaux, 1996.

[15] M. Bousquet-Mélou. Multi-statistic enumeration of two-stack sortable permutations. *Electronic Journal of Combinatorics*, 5:R21, 1998.

[16] M. Bousquet-Mélou and J. M. Fédou. The generating function of convex polyominoes: The resolution of a q-differential system. *Discrete Mathematics*, 137(1-3):53–75, 1995.

[17] M. Bousquet-Mèlou and A. J. Guttmann. Enumeration of three dimensional convex polygons. *Annals of Combinatorics*, 1:27–53, 1997.

[18] M. Bousquet-Mélou and M. Petkovšek. Linear recurrences with constant coefficients: the multivariate case. *Discrete Mathematics*, 225:51–75, 2000.

[19] R. Brak and A. J. Guttmann. Algebraic approximants: a new method of series analysis. *J. Phys. A: Math. Gen.*, 23(24):1331–1337, 1990.

[20] S. Brlek, E. Duchi, E. Pergola, and R. Pinzani. Bijective construction of equivalent ECO-systems. In Birkhauser, editor, *Proceedings of Internation Colloquium on Mathematics and Computer Science: trees, algorithms and probability(MathInfo'2002)*, Versailles, France, 2002.

[21] S. Brlek, E. Duchi, E. Pergola, and S. Rinaldi. On the equivalence problem of succession rules. In Birkhauser, editor, *Proceedings of FPSAC'02*, Melbourne, Australia, 2002.

[22] S. Chaiken, D. J. Kleitman, M. Saks, and J. Shearer. Covering regions by rectangles. *SIAM J. Discr. and Alg. Meth.*, 2:394–410, 1981.

[23] S.J. Chang and K. Y. Lin. Rigorous results for the number of convex polygons on the square and honeycomb lattices. *J. Phys. A: Math. Gen.*, 21:2635–2642, 1988.

[24] N. Chomsky and M. P. Schützenberger. The algebraic theory of context-free languages. In *Computer Programming and Formal Systems*, pages 118–161, North Holland. ed. P. Braffort and D. Hirschberg.

[25] L. Chottin. Etude syntaxique de certains langages solutions d'équations avec opérateurs. *Theoretical Computer Science*, 5:51–84, 1977.

[26] F. R. K. Chung, R. L. Graham, V. E. Hoggatt, and M. Kleimann. The number of Baxter permutations. *J. Combin. Theory Ser. A*, 24:382–394, 1978.

[27] J. H. Conway and J. C. Lagarias. Tiling with polyominoes and combinatorial group theory. *Journal of Combinatorial Theory A*, 53:183–208, 1990.

[28] R. Cori. Un code pour les graphes planaires et ses applications. *Ásterisque*, 27, 1975.

[29] R. Cori and J. Richard. Énumération des graphes planaires à l'aide des séries formelles en variables non commutatives. *Discrete Mathematics*, 2:115–162, 1972.

[30] S. Corteel. Séries génératrices exponentielles pour les eco-systèmes signés. In *Proceedings of the 12-th International Conference on Formal Power Series and Algebraic Combinatorics, Moscow*, 2000.

[31] A. Del Lungo, E. Duchi, A. Frosini, and S. Rinaldi. Enumeration of convex polyominoes using the ECO method. In *Proceeding of DMCS*, Lyon, 2003. To appear.

[32] M. Delest, J. P. Dubernard, and I. Dutour. Parallelogram polyominoes and corners. *J. Symbolic Computation*, 20(5-6):503–515, 1995.

[33] M. Delest and J. M. Fédou. Attribute grammars are useful for combinatorics. *Theoretical Computer Science*, 98:65–76, 1992.

[34] M. Delest and X. Viennot. Algebraic languages and polyominoes enumeration. *Theoretical Computer Science*, 34:169–206, 1984.

[35] A. Denise and R. Simion. Two combinatorial statistics on Dyck paths. *Discrete Mathematics*, 137, 1995.

[36] E. Deutsch. Operations on production matrices. Personal communication.

[37] E. Duchi, J. M. Fédou, C. Garcia, and E. Pergola. On the inversion of succession rules. Rapport numero I3S/RR-2002-38-FR, 2002.

[38] E. Duchi, J-M. Fédou, and S. Rinaldi. From object grammars to ECO-systems. Accepted for publication in Theoretical Computer Science.

[39] E. Duchi, A. Frosini, R. Pinzani, and S. Rinaldi. A note on rational succession rule. *Journal of Integer Sequences*, 6(Issue 1), 2003. Article 03.1.7.

[40] P. Duchon. On the enumeration and generation of generalized Dyck words. In *In the Proceedings of FPSAC*, 1998.

[41] S. Dulucq, S. Gire, and O. Guibert. Enumération de permutations à motifs exclus. In *Actes du 30ième séminaire Lotharingien de Combinatoire, Hesselberg*, 1993.

[42] S. Dulucq, S. Gire, and O. Guibert. A combinatorial proof of J.West's conjecture. *Discrete Mathematics*, 187:71–96, 1998.

[43] S. Dulucq, S. Gire, and J. West. Permutations with forbidden subsequences and nonseparable planar maps. *Discrete Mathematics*, 153:85–103, 1996.

[44] I. Dutour. *Grammaires d'objets : énumération, bijections et génération aléatoire*. PhD thesis, Thèse de l'Université de Bordeaux I, 1996.

[45] I. Dutour and J.M. Fédou. Object grammars and random generation. *Discrete Mathematics and Theoretical Computer Science*, 2:47–61, 1998.

[46] I. Dutour and J.M. Fédou. Object grammars and bijections. *Theoretical Computer Science*, 290(3):1915–1929, 2003.

[47] S. Eilenberg and J. B. Wright. Automata in general algebra. *Information and Control*, 11:452–470, 1967.

[48] E.Pergola and R. A. Sulanke. Schröder triangles, paths, and parallelogram polyominoes. *Electronic Journal of Integer Sequences*, 1, 1998. Art. 98.1.7.

[49] G. Fayolle and R. Iasnogorodski. Solutions of functional equations arising in the analysis of two-server queueing models. *Performance of computer systems*, pages 289–303, 1979. North-Holland.

[50] J.M. Fédou and C. Garcia. Algebraic succession rules. In *Proceedings of the 14-th International Conference on Formal Power Series and Algebraic Combinatorics, Melbourne*, 2002.

[51] L. Ferrari. *On the foundations of the ECO method*. PhD thesis, University of Florence, 2003.

[52] L. Ferrari, E. Pergola, R. Pinzani, and S. Rinaldi. Jumping succession rules and their generating functions. Discrete Mathematics, to appear.

[53] L. Ferrari, E. Pergola, R. Pinzani, and S. Rinaldi. An algebraic characterization of the set of succession rules. *Theoretical Computer Science*, 281:351–367, 2002.

[54] P. Flajolet. Annals of discrete mathematics. *Performance of computer systems*, 8:217–222, 1980.

[55] P. Flajolet, J. Francon, and J. Vuillemin. Sequences of operations analysis for dynamic data structures. *J. of Algorithms*, 1:111–141, 1980.

[56] P. Flajolet and A. Odlyzko. Singularity analysis of generating functions. *SIAM J. Discrete Math.*, 3:216–240, 1990.

[57] P. Flajolet, B. Salvy, and P. Zimmermann. Lambda-upsilon-omega: An assistant algorithms analyzer. In T. Mora, editor, *Proceedings of Applied Algebra, Algebraic Algorithms and Error-Corretting Codes*, pages 201–212, Rome, Italy, 1988. Springer Verlag.

[58] P. Flajolet, B. Salvy, and P. Zimmermann. Lambda-upsilon-omega: The 1989 cookbook. Rapporte technique n. 1073, Institut National de Recherche en Informatique et en Automatique, 1989.

[59] P. Flajolet, B. Salvy, and P. Zimmermann. Automatic average-case analysis of algorithms. *Theoretical Computer Science*, 1:37–109, 1991.

[60] P. Flajolet and R. Sedgewick. The average case analysis of algorithms: complex asympotics and generating functions. Rapport de recherche n. 2026, Institut National de Recherche en Informatique et en Automatique, 1993.

[61] P. Flajolet and R. Sedgewick. *Analytic Combinatorics - Symbolic Combinatorics*. First Edition. 2002.

[62] P. Flajolet, P. Zimmermann, and B. Van Cutsem. A calculus for the random generation of labeled combinatorial structures. *Theoretical Computer Science*, 132:1–35, 1994.

[63] S. W. Golomb. Checker boards and polyominoes. *Amer. Math. Monthly*, 61(10):675–682, 1954.

[64] I.P. Goulden and D.M. Jackson. *Combinatorial Enumeration*. John Wiley & Sons, New York, 1983.

[65] R. L. Graham, D. E. Knuth, and O. Patashnik. *Concrete Mathematics*. Addison-Wesley Pub. Co, Reading, Massachusetts, 1989.

[66] O. Guibert. *Combinatoire des permutations à motifs exclus en liason avec mots, cartes planaries et tableaux de Young*. PhD thesis, Thèse de l'Université de Bordeaux I, 1996.

[67] O. Guibert, E. Pergola, and R. Pinzani. Vexillary involutions are enumerated by motzkin numbers. *Annals of Combinatorics*, 5:153–174, 2001.

[68] A. J. Guttmann and I. G. Enting. The number of convex polygons on the square and honeycomb lattices. *J. Phys.*, A 21:L467–L474, 1988.

[69] A. Joyal. Une théorie combinatoire des séries formelles. *Advanced in Math.*, 42(1):1–82, 1981.

[70] D. A. Klarner and R. L. Rivest. A procedure for improving the upper bound for the number of n-ominoes. *Canad. J. Math.*, 25(3):585–602, 1973.

[71] D. E. Knuth. *The Art of Computer Programming, Vol. 1: Fundamental Algorithms*. Addison-Wesley, 1968. Exercises 4 and 11, Section 2.2.1.

[72] D. E. Knuth. Semantics of context-free languages. *Math. Systems Theory*, 2:127–145, 1968.

[73] Labelle and Yeh. Generalized Dyck paths. *Discrete Mathematics*, 82(1):1–6, 1990.

[74] P. A. MacMahon. *Combinatorial Analysis*. Cambridge University Press, 1916. Reprint by Chelsea, New York, 1960.

[75] D. Merlini, R. Sprugnoli, and M.C. Verri. An algebra for proper generating trees. In *Proceedings of CMCS, Versailles*, pages 127–139, 2000.

[76] D. Merlini and M.C. Verri. Generating trees and proper riordan arrays. *Discrete Mathematics*, 218:167–183, 2000.

[77] J. G. Penaud, E. Pergola, R. Pinzani, and O. Roques. Chemins de Schröder et hiérarchies aléatoires. *Theoretical Computer Science*, 255:345–361, 2001.

[78] E. Pergola. *ECO: a Methodology for Enumerating Combinatorial Objects*. PhD thesis, University of Florence, 1999.

[79] E. Pergola and R. Pinzani. A combinatorial interpretation of the area of Schröder paths. *Electronic Journal of Combinatorics*, 6, 1999.

[80] E. Pergola, R. Pinzani, and S. Rinaldi. A set of well-defined operations on succession rules. In *Proceedings of CMCS, Versailles*, pages 141–152, 2000.

[81] T. Prellberg and R. Brak. Critical exponents from non-linear functional equations for partially directed cluster models. *J. Statist. Phys.*, (78):701–730, 1995.

[82] D. H. Redelmeier. Counting polyominoes: yet another attack. *Discrete Mathematics*, 36:191–203, 1981.

[83] S. Rinaldi. *Succession rules : the whole story.* PhD thesis, University of Florence, 2001.

[84] E. Roblet. *Une interprétation combinatoire des approximants de Padé.* PhD thesis, Université de Bordeaux I, Bordeaux, 1994.

[85] M. Robson. Personal communication.

[86] G. Rozenberg and A. Salomaa. *The mathematical theory of L systems.* Academic Press, INC., London, 1980.

[87] A. Salomaa. *Jewels of formal languages.* Computer Science Press, Potomac, Maryland, 1981.

[88] A. Salomaa and M. Soittola. *Automata-theoretic aspects of formal power series.* Springer-Verlag, New York, 1978.

[89] E. Schröder. Vier combinatorische problem. *Z. für Math. Physik*, 15:361–370, 1870.

[90] M. P. Schützenberger. On context-free languages and push-down automata. *Information and Control*, 6:246–264, 1963.

[91] M. P. Schützenberger. Algebraic automata and context-free sets. *Information and Control*, 11:3–29, 1967.

[92] N. J. A. Sloane. *The On-Line Encyclopedia of Integer Sequences.* at http://www.research.att.com/ njas/sequences/index.html.

[93] N. J. A. Sloane and S. Plouffe. *The encyclopedia of integer sequences.* Academic Press, New York, 1996.

[94] R. P. Stanley. *Enumerative Combinatorics, Vol I.* Wadworth & Brooks/Cole, Monterey, Cal., 1986.

[95] R. A. Sulanke. Bijective Recurrences concerning Schröder paths. *Electronic Journal of Combinatorics*, 5(1):R47, 1998.

[96] R. A. Sulanke. Moments of generalized motzkin paths. *J. of Integer Sequences*, 3(00.1.1):191–203, 2000.

[97] R. A. Sulanke. Bijective Recurrences for Motzkin Paths. *Adv. in Applied Math.*, 27:627–640, 2001.

[98] X.G. Viennot. A combinatorial theory for general orthogonal polynomials with extensions and applications. *Lecture Notes in Mathematics*, 1171:139–157, 1985.

[99] J. West. Generating trees and the Catalan and Schröder numbers. *Discrete Mathematics*, 146:247–262, 1995.

[100] H. S. Wilf. *Generatingfunctionology.* Academic Press, San Diego, 1992.

[101] D. Zeilberger. Self avoiding walks, the language of science, and Fibonacci numbers. *Journal of Statistical Planning and Inference*, 54:135–138, 1996.