



UNIVERSITÀ DEGLI STUDI DI FIRENZE

Dottorato di Ricerca
in Ingegneria Informatica e dell'Automazione

XVIII Ciclo

Una struttura tira l'altra: generazione di oggetti combinatori

Candidato:
Maddalena Poneti

Tutore: Prof.ssa Elena Barucci

Cotutore: Prof. Jean-Guy Penaud

Coordinatore: Prof. Edoardo Mosca

Ringraziamenti

E' un vero piacere avere l'opportunità di ringraziare le persone che hanno contribuito in questi tre anni alla mia crescita scientifica e personale: persone tutte molto diverse tra loro, tanto da lasciare ognuna un segno particolare. Ringrazio Elena Barucci, donna tanto di classe quanto dal carattere forte, che mi ha insegnato che affrontare ogni situazione con pacatezza ma determinazione rende le persone degne di rispetto e le aiuta a fare (molto spesso) la cosa giusta.

Sono molto grata a Renzo Pinzani che, con il suo carattere al tempo stesso forte, volitivo ma anche paterno e pieno di ironia, mi ha aiutata in momenti non facili sia dal punto di vista scientifico che personale.

Ringrazio una donna esile ma con una grinta ed una forza di carattere invidiabili, oltre che con una disponibilità verso di me davvero notevole: Elisa Pergola.

Un grazie va ad Elisabetta Grazzini, che con la sua passione per l'organizzazione ha almeno tentato di insegnare a me, disordinata cronica, a riordinare le idee prima di tirare conclusioni altrimenti affrettate: consiglio particolarmente utile per le fasi di questa tesi.

Un'altra persona sicuramente importante è il Professor Penaud, che mi ha generosamente ospitato trattandomi come un membro della famiglia, e che con il suo comportamento ha ribadito l'importanza della dedizione al lavoro. Ringrazio gli amici oltre che collaboratori scientifici di Siena, inseparabili sempre e quindi anche in questa occasione: Simone Rinaldi e Andrea Frosini, sempre pronti a darmi una mano.

Inoltre come dimenticare le persone che hanno condiviso con me prima la stanza senza finestre nella vecchia sede del Dipartimento e poi quella luminosa della nuova sede? sto parlando di amici oltre che di compagni di ricerca: Antonio Bernini e Luca Ferrari, oltre alle due nuove "leve" Irene Fanti e Sara Ciaffarà.

A Gianni Aguzzi e Fabio Pippolini va un grazie come compagni "di caffè". Voglio inoltre ringraziare tutti i miei amici "esterni" e non al mondo della ricerca, che con la loro presenza mi hanno reso più sicura, in particolare Bar-

bara, Chiara, Silvia, Monica, Sabrina, Chiara, Elisa, Marco.

Grazie ad Antonello, che negli ultimi dieci anni ha condiviso con me molti momenti, facendomi crescere.

Un grazie che non ha bisogno di motivazioni va ai miei genitori: per la loro pazienza, la loro spinta morale ed emotiva, la loro dedizione alla famiglia ed al lavoro, per la loro fiducia in me.

Indice

Ringraziamenti	1
1 Premessa	8
1.1 Introduzione	8
1.2 Introduction	11
2 Generazione esaustiva di alcuni linguaggi regolari	17
2.1 Gli algoritmi di generazione	20
2.2 Analisi del costo dell'algoritmo	24
2.3 Conclusioni	25
3 Cammini culminanti	28
3.1 Allineamento di sequenze e cammini culminanti	29
3.2 Generazione esaustiva dei $\mathcal{CC}(n, h)$	32
3.3 Generazione casuale dei $\mathcal{CC}(n, h)$	35
4 Generazione a testa o croce	39
4.1 Parole di Fibonacci	40
4.2 Parole di Tribonacci	42
4.2.1 Complessità	45
4.3 Generalizzazione	48
4.4 Conclusioni	49
5 Dalle permutazioni di Fibonacci a quelle di Catalano	52
5.1 Permutazioni a motivo escluso	54
5.2 Da Fibonacci a 2^{n-1}	57
5.3 Da 2^{n-1} a Catalano	59
5.4 Da Fibonacci a Catalano in modo diretto	60
5.5 Conclusioni	61

6	Una struttura di reticolo distributiva.	65
6.1	Nozioni preliminari sull'insieme delle partizioni	67
6.2	Partizioni noncrossing e cammini di Dyck	69
6.3	Il reticolo di Bruhat per la partizione noncrossing	72
6.4	Collegamento con l'ordine di Bruhat forte sulle permutazioni .	77
6.5	Una possibile estensione	80
7	<i>m</i>-composizioni	87
7.1	Identità combinatorie	88
7.2	<i>m</i> -composizioni e linguaggi regolari	90
7.2.1	Parole su un alfabeto infinito	90
7.2.2	Parole su un alfabeto finito	92
7.3	Interpretazioni combinatorie	93
7.3.1	Partizioni lineari linearmente colorate	93
7.3.2	Famiglie suriettive	95
7.3.3	Bargraph etichettati	96
7.4	Identità del tipo Cassini	99
7.5	Comportamento asintotico e formule del tipo Binet	100
7.6	<i>m</i> -composizioni senza righe nulle	101
7.7	<i>m</i> -composizioni con righe palindromo	104
7.8	<i>m</i> -composizioni del tipo Carlitz	105
7.9	Matrici generate dalle <i>m</i> -composizioni	107
7.10	Generazione delle <i>m</i> -composizioni	109
7.11	Codice Gray per le <i>m</i> -composizioni	111
7.11.1	Definizione del codice Gray per le <i>m</i> -composizioni . . .	111
7.12	Generazione delle <i>m</i> -partizioni di interi	114

Elenco delle figure

3.1	Esempio di allineamento univocamente associato ad un cammino.	30
3.2	Esempio di cammino culminante in senso stretto con $n = 13$ ed $h = 5$. Viene evidenziata la legittimità di non considerare l'ultimo passo.	31
3.3	Generazione ECO dei cammini culminanti con $h = 4$	33
3.4	Generazione casuale dei cammini culminanti con $h = 4$ ed $n = 8$. Le linee continue rappresentano i possibili cammini corretti, mentre quelle tratteggiate sono i passi che portano a cammini non culminanti.	36
4.1	Rappresentazione grafica del polinomio $P(x)$ che definisce la probabilità nel caso di Fibonacci nell'intervallo $[0, 1]$, con i motivi e le loro probabilità.	41
4.2	Rappresentazione grafica del polinomio $P(x)$ che definisce la probabilità nel caso di Tribonacci.	43
4.3	Rappresentazione grafica della trasformazione dal polinomio $P(x)$ a $Q(t)$	44
5.1	Grafo delle permutazioni.	62
6.1	Diagramma di Hasse di $\Pi(4)$	68
6.2	La partizione flat di $\Pi(4)$	69
6.3	La partizione noncrossing $2 654 8731 9 \in NC(9)$	69
6.4	Cammino di Dyck associato a $2 654 8731 9$	70
6.5	Come si ottiene P' da P	76
6.6	Un Bell matching di una parola di Dyck di lunghezza 18	82
7.1	Rappresentazione della partizione con i 3 colori (cerchietto, rombo e quadrato rappresentano per esempio rosso, verde e blu rispettivamente).	94
7.2	(a) Un bargraph; (b) un bargraph etichettato di grado 4. . . .	96

7.3	Il bargraph di grado 4 associato ad M	97
7.4	Bargraph etichettati di grado 3: (a) con tutte le etichette in ogni colonna; (b) una 3-partizione; (c) uno stack etichettato di grado 3.	98
7.5	Bargraph associato alla 3-partizione di 11 $w = 1123\bar{1}22\bar{2}33\bar{1}$. .	115

Elenco delle tabelle

2.1	Codifica degli interi da 0 a 17 nel sistema di numerazione definito dai numeri di Pell.	20
2.2	Codifica di alcuni interi utilizzando la sequenza dei numeri di Fibonacci di ordine dispari come sistema di numerazione. . . .	23
3.1	Esempio di due sequenze di DNA con punteggi e sequenza binaria corrispondenti.	29
3.2	Risultati trovati per $h = 2, 3, 4$	34
3.3	Tabella che visualizza i coefficienti dei polinomi denominatori di $F(x)$	34
7.1	Tabella dei numeri $p_n^{(m)}$	104
7.2	Tabella dei numeri $z_n^{(m)}$	106

Capitolo 1

Premessa

1.1 Introduzione

Questa tesi di dottorato ha avuto origine inizialmente dall'argomento della mia tesi di laurea [MP]: la generazione di strutture descritte da ricorrenze lineari. I miei primi passi nel campo della ricerca mi hanno portato verso lo studio di argomenti da me mai trattati prima quali i collegamenti tra linguaggi regolari e sistemi di numerazione e le piccole scoperte personali quotidiane come il modo di esprimere sotto forma di matrice i vari oggetti descritti da ricorrenze lineari e quindi rilette come interi espressi in un dato sistema di numerazione e, ancora di più, la scoperta della relazione tra la struttura delle varie colonne della matrice e la struttura della ricorrenza lineare in questione. Tutte piccole scoperte le mie che hanno reso interessanti e stimolanti molti giorni di studio, ma che mi hanno anche portato a capire il significato di generazione, in particolare quello di generazione esaustiva. Non bastava infatti aver trovato il modo di generare tutti gli elementi di una certa dimensione, bisognava prendere in considerazione l'efficienza dell'algoritmo. A proposito di questo nel Capitolo 2 di questa tesi determiniamo appunto un algoritmo di generazione esaustiva per classi di oggetti combinatori che soddisfano particolari relazioni di ricorrenza della forma $x_n = ax_{n-1} + bx_{n-2}$. Per raggiungere questo nostro obiettivo, codifichiamo gli elementi di tali classi in termini di linguaggi regolari tramite i sistemi di numerazione. Infine proviamo che l'algoritmo gode della proprietà CAT, cioè utilizza un "tempo ammortizzato costante". Un punto di riferimento importante per la generazione esaustiva di oggetti combinatori di dimensione fissata n di una certa classe consiste in [P], dove viene introdotta una nuova metodologia di generazione al tempo stesso semplice ed efficace: il metodo ECO (Enumerating of Combinatorial Objects). Questo è un particolare metodo di combinatoria enumerativa e

costruttiva di cui la descrizione può essere trovata, per esempio, in [BDLPP]. Si comincia effettuando una partizione di una classe di oggetti secondo la loro dimensione (definita adeguatamente). L'obiettivo è poi quello di effettuare una specie di espansione locale su ogni oggetto di una data dimensione, producendo così tutti gli oggetti della dimensione successiva esattamente una volta. Di conseguenza un singolo oggetto produce un insieme di nuovi oggetti secondo un certo parametro. Di solito, se una costruzione siffatta è abbastanza regolare, può essere tradotta in codice utilizzando una *regola di successione* [W], che è un sistema puramente formale, espresso generalmente come segue:

$$\left\{ \begin{array}{l} (a) \\ (k) \rightsquigarrow (e_1(k)) \cdots (e_k(k)). \end{array} \right. \quad (1.1)$$

Le lettere denotano interi positivi, (a) si dice *assioma* e $(k) \rightsquigarrow (e_1(k)) \cdots (e_k(k))$ è la *produzione* di (k) . Una regola di successione può essere rappresentata tramite il suo *albero di generazione*, che è, dalla definizione, infinito, con una radice ed etichettato. E' un albero in cui la radice ha etichetta a (come l'assioma) e tale che ogni nodo etichettato k produce k figli, etichettati rispettivamente $e_1(k), \dots, e_k(k)$. Un'importante informazione enumerativa prodotta dalla regola di successione è la sequenza numerica delle cardinalità dei livelli dell'albero di generazione associato alla regola. Ci riferiamo a questa come a quella sequenza *determinata dalla* regola di successione.

La principale applicazione della generazione esaustiva è lo studio delle proprietà delle classi di oggetti o degli algoritmi che operano su di esse. In certi casi, anche se gli algoritmi di generazione esaustiva sono molto efficienti, poiché il numero di oggetti cresce in maniera esponenziale con le dimensioni diventa praticamente improponibile generare tutti gli oggetti di dimensioni elevate. In questi casi si può quindi utilizzare un campione "significativo" di oggetti di una certa dimensione e diventa quindi fondamentale poter generare tali oggetti in maniera casuale.

A differenza della generazione esaustiva, di solito generare in modo casuale ed uniforme un oggetto risulta essere più difficile, soprattutto per quanto riguarda l'uniformità della generazione e la complessità degli algoritmi trovati. Infatti per generare in modo casuale ed uniforme un oggetto bisogna tenere conto non solo della dimensione richiesta, ma anche della equiprobabilità di tutti gli oggetti della classe considerata con dimensione richiesta. Un esempio classico per spiegare il concetto è quello della lotteria: abbiamo un'urna contenente dieci palline numerate da 0 a 9; ad ogni estrazione la pallina estratta viene ributtata nell'urna, quindi ogni volta ciascun numero ha la stessa probabilità ($\frac{1}{10}$) di uscire degli altri. Il concetto di generazione casuale

ed uniforme è proprio questo: se S è la classe di oggetti di dimensione fissata n che possono essere estratti, ognuno ha probabilità $\frac{1}{|S|}$ di uscire. Lo scopo è quindi quello di generare algoritmi efficienti che estraggano un elemento di dimensione fissata senza avere a che fare con tutti gli altri [R]. Abbiamo affrontato il problema della generazione esaustiva e casuale di oggetti combinatori particolari nati per studiare il problema degli allineamenti di sequenze di DNA: i cosiddetti *cammini culminanti* (Capitolo 3). Li abbiamo generati tramite il metodo ECO e successivamente abbiamo cercato di approfondire o quanto meno illustrare le problematiche relative alla loro generazione casuale, risolta tramite un algoritmo piuttosto complesso da Ponty in [Po], e che noi vorremmo affrontare da un punto di vista più combinatorio. Da notare che consideriamo tali cammini con passi del tipo $(1, 1)$ e $(1, -1)$, anche se potremmo considerarli con passi più generici. Abbiamo affrontato il problema della generazione casuale di tali cammini tramite un metodo a rifiuto anticipato. I primi due capitoli che abbiamo qui sintetizzato introducono il capitolo successivo. Infatti sempre a proposito di linguaggi regolari descritti da ricorrenze lineari, nel Capitolo 4 estendiamo un algoritmo trovato da J. G. Penaud ed O. Roques in [PR] per il linguaggio di Fibonacci $L = \varepsilon + La + Lbb$, in cui, utilizzando un generatore uniforme di bit, generiamo in modo casuale ed uniforme il linguaggio di Tribonacci $L = \varepsilon + La + Lbb + Lccc$. La complessità di tale algoritmo risulta essere lineare. Dopo avere applicato tale algoritmo rivisto e adeguato alla luce delle proprietà del nuovo linguaggio, operiamo una generalizzazione a linguaggi regolari descritti da ricorrenze lineari con coefficienti interi a k termini.

E' risaputo che le permutazioni che evitano i motivi di lunghezza tre, che definiremo nel Capitolo 5, sono enumerate dai numeri di Catalano. Nel suddetto capitolo ci occupiamo di considerare la classe di permutazioni in cui i motivi proibiti sono 123, 132 e 213, questa rappresenta un caso particolare della classe enumerata da Catalano, ma allo stesso tempo, è una classe di oggetti enumerata da Fibonacci. Quindi nasce la necessità di vedere se, ed eventualmente come, passare da classi enumerate da Fibonacci a classi enumerate da Catalano e viceversa. Per far questo troviamo diverse classi di permutazioni di interi che si trovano tra i numeri di Fibonacci e quelli di Catalano. Come risultato introduciamo un qualche tipo di "continuità" tra le sequenze che enumerano queste classi.

Trattando di permutazioni a motivo escluso, ricordiamo che queste sono oggetti combinatori che, a seconda del numero e della lunghezza dei motivi che non possono contenere, sono insiemi di cardinalità diverse. Per enumerare gli oggetti di un determinato insieme un metodo classico è quello di mettere in biezione questo insieme con un altro di cui conosciamo già la cardinalità. A questo proposito, nel Capitolo 6 consideriamo le partizioni noncrossing, i

cammini di Dyck, e le permutazioni a motivo escluso 312. Diciamo che per trovare un “collegamento” tra loro utilizziamo le proprietà dei reticoli. Un *reticolo* è un poset (insieme parzialmente ordinato) in cui tutti i sottoinsiemi limitati non vuoti ammettono sia un estremo inferiore sia un estremo superiore. I reticoli possono anche essere caratterizzati come strutture algebriche che soddisfano determinate identità. Poiché entrambe le definizioni possono essere usate convenientemente, la teoria dei reticoli può essere applicata sia dalla teoria dell'ordine che dalla teoria dell'algebra universale. In [FP2] è definito un ordine naturale sui cammini di Dyck di lunghezza fissata che induce una struttura di reticolo distributivo. Il Capitolo 6 mostra come riusciamo a trasferire questo ordine alle partizioni noncrossing sfruttando una biiezione ben conosciuta [S], mostrando che le partizioni noncrossing possono essere dotate di una struttura di reticolo distributivo che ha molta rilevanza combinatoria. Infine, proviamo che i nostri reticoli sono isomorfi ai poset delle permutazioni che evitano il motivo 312 con l'ordine indotto dall'ordine di Bruhat forte del gruppo simmetrico.

Infine nel Capitolo 7 continuiamo a studiare delle proprietà algebriche di classi di oggetti combinatori, in particolare ci occupiamo della classe delle composizioni espresse sotto forma di matrice (in breve “ m -composizioni”), cioè matrici con elementi non negativi, con m righe, e le cui colonne sono diverse da vettori costituiti da tutti elementi nulli. Le m -composizioni ed in particolare quelle con $m = 2$ sono state introdotte e definite in [CFMRR]. Troviamo risultati enumerativi e varie interpretazioni combinatorie. In particolare, estendiamo al caso m -dimensionale molte delle proprietà delle composizioni ordinarie. Inoltre, sempre portando avanti l'argomento scatenante di questa tesi, troviamo un efficiente algoritmo di generazione per generare tutte e sole le m -composizioni (e poi anche m -partizioni di cui parleremo) di un intero n con m fissato.

1.2 Introduction

Cette thèse de doctorat se fonde sur mon travail de recherche initial, qui aboutit à l'écriture d'une thèse de *laurea* [MP]: la génération des structures décrites par des récurrences linéaires. En effet, mes premiers pas de chercheuse m'ont portée vers des sujets nouveau pour moi comme l'étude des relations entre les langages réguliers et les systèmes de numération. Ils m'ont ainsi conduite à des petites découvertes personnelles quotidiennes comme la façon d'exprimer sous forme matricielle les différents objets décrits par les récurrences linéaires et donc entièrement exprimés dans un système de numération donné, et, en particulier, la découverte de la relation entre la structure

des diverses colonnes de la matrice et la structure de la récurrence linéaire en question. Ces petits pas furent l'occasion d'intéressants, stimulants (et nombreux) jours d'étude, et m'ont amenée à comprendre le sens du mot *génération*, en particulier celui de génération exhaustive.

En effet il ne suffisait pas d'avoir trouvé la façon d'engendrer tous les éléments d'une certaine taille, il fallait prendre en considération l'efficacité de l'algorithme de génération.

Dans le Chapitre 2 de cette thèse, nous déterminons précisément un algorithme de génération exhaustive pour des classes d'objets combinatoires qui satisfont des relations particulières de récurrence de la forme $x_n = ax_{n-1} + bx_{n-2}$. Pour atteindre notre objectif, nous codons les éléments de telles classes en terme de langages réguliers par les systèmes de numération. Finalement nous prouvons que l'algorithme possède la propriété CAT, c'est-à-dire qu'il nécessite un " temps amorti constant". Un point de référence important pour la génération exhaustive d'objets combinatoires de taille fixée n d'une certaine classe se trouve dans [P], où est introduite une nouvelle méthodologie de génération à la fois simple et efficace : la méthode ECO (Enumerating of Combinatorial Objects). Il s'agit d'une méthode particulière de combinatoire énumérative et constructive: la description peut en être trouvée, par exemple, dans [BDLPP]. On commence à effectuer une partition d'une classe d'objets selon la taille (définie adéquatement). L'objectif est ensuite d'opérer une sorte d'expansion locale sur chaque objet de taille donnée, en produisant ainsi tous les objets de la taille suivante, et ce exactement une fois. Par conséquent un objet produit un ensemble de nouveaux objets selon un certain paramètre. En général, si une telle construction est assez régulière, elle peut être traduite codée par une *règle de succession* [W], qui est un système purement formel, exprimée généralement comme suit:

$$\left\{ \begin{array}{l} (a) \\ (k) \rightsquigarrow (e_1(k)) \cdots (e_k(k)). \end{array} \right. \quad (1.2)$$

Les lettres indiquent des entiers positifs, on dit que (a) est l'*axiome* et $(k) \rightsquigarrow (e_1(k)) \cdots (e_k(k))$ est la *production* de (k) . Une règle de succession peut être représentée par son *arbre de génération*, qui est, par définition, infini, avec une racine et étiqueté. C'est un arbre dans lequel la racine a étiquetée a (comme l'axiome) et tel que chaque noeud étiqueté k produit k fils, étiquetés respectivement $e_1(k), \dots, e_k(k)$. Une importante information énumérative produite par la règle de succession est la séquence numérique de la cardinalité des niveaux de l'arbre de génération associé à la règle. Nous nous référons à elle comme la séquence *déterminée par* la règle de succession. La principale application de la génération exhaustive est double, d'une part

l'étude des propriétés des classes d'objets et d'autre part celles des algorithmes qui agissent sur elles. En certains cas, même si les algorithmes de génération exhaustive sont très efficaces, le nombre d'objets croît de manière exponentielle avec la taille, ce qui rend pratiquement impossible d'engendrer tous les objets de grande taille. Dans ces cas on doit donc utiliser un représentant "significatif" de ces objets de grande taille et il devient donc fondamental de pouvoir engendrer des tels objets de façon aléatoire.

Par comparaison à la génération exhaustive, engendrer de façon aléatoire et uniforme un objet peut être plus difficile, surtout pour maintenir l'uniformité de la génération et la complexité des algorithmes trouvés. En effet pour engendrer de manière aléatoire et uniforme un objet il ne suffit pas de tenir compte de la taille demandée, mais aussi de l'équiprobabilité du tirage d'un objet parmi tous les objets de même taille. Un exemple classique pour expliquer ce concept est celui de la loterie: nous avons une urne contenant dix billes numérotées de 0 à 9; à chaque tirage la bille extraite est remise dans l'urne, donc à chaque tirage tout nombre a la même probabilité ($\frac{1}{10}$) d'être choisi. Le but est donc engendrer des algorithmes efficaces qui engendrent un élément de taille fixée, sans avoir à tenir compte des autres [R].

Nous avons abordé le problème de la génération, aussi bien exhaustive qu'aléatoire, d'objets combinatoires particuliers conçus pour étudier le problème des alignements de séquences de DNA : les *chemins culminants* (Chapitre 3). Nous avons engendrés ces chemins par la méthode ECO et puis nous avons cherché à approfondir ou au moins à illustrer les problématiques relatives à leur génération aléatoire. Cette génération est résolue par un algorithme plutôt complexe de Ponty dans [Po], et nous voudrions l'aborder d'un point de vue plus combinatoire. Il est à remarquer que nous considérons de tels chemins avec des pas du type $(1, 1)$ et $(1, -1)$, même si nous pourrions les considérer avec des pas plus génériques. Nous avons abordé le problème de la génération aléatoire de tels chemins par une méthode à rejet anticipé.

Les premiers deux chapitres que nous avons résumés ci-dessus introduisent le chapitre suivant. En effet, toujours à propos de langages réguliers décrits par des récurrences linéaires, dans le Chapitre 4 nous étendons un algorithme trouvé par J. G. Penaud et O. Roques en [PR] pour le langage de Fibonacci. Ainsi, en utilisant un générateur uniforme de bits, nous engendrons de manière aléatoire et uniforme le langage de Tribonacci, généralisant celui de Fibonacci. Il en résulte que la complexité d'un tel algorithme est linéaire. Après avoir appliqué cet algorithme revu et adapté à la lumière des propriétés du nouveau langage, nous en proposons une généralisation à des langages réguliers décrits par récurrences linéaires avec des coefficients entiers à k termes.

Nous savons que les permutations des entiers de 1 à n qui évitent un motif

de longueur trois, qui nous définirons dans le Chapitre 5, sont énumérées par les nombres de Catalan. Dans le chapitre susmentionné, nous présentons la classe des permutations dans lesquelles les motifs interdits sont 123, 132 et 213, ce qui représente un cas particulier de la classe énumérée par les nombres de Catalan, mais en même temps, c'est une classe d'objets énumérée par ceux de Fibonacci. Donc il apparaît la nécessité de savoir si, et éventuellement comment, on peut passer de classes énumérées par Fibonacci à des classes énumérées par Catalan et réciproquement. Pour faire ceci nous établissons différentes classes de permutations d'entiers qu'on peut trouver entre les nombres de Fibonacci et de Catalan. Comme résultat nous introduisons une relation de "continuité" entre les séquences qui énumèrent ces classes.

En traitant de permutations à motifs exclus, nous rappelons que celles-ci sont des objets combinatoires qui, selon le nombre et la longueur des motifs qu'ils ne doivent pas contenir, sont des ensemble de cardinalité différentes. Pour énumérer les objets d'un déterminé ensemble une méthode classique est de mettre cet ensemble bijection avec un autre ensemble dont nous connaissons déjà la cardinalité.

A cette fin, dans le Chapitre 6 nous considérons les partitions *non croisées*, les chemins de Dyck, et les permutations à motif exclu 312. Nous disons que pour trouver une "liaison" entre elles nous utilisons les propriétés des treillis. Un *treillis* est un ensemble partiellement ordonné dans lequel tous les sous-ensembles finis non vides admettent une borne inférieure et une borne supérieure. Les treillis peuvent également être caractérisés comme des structures algébriques satisfaisant des identités déterminées. Les deux définitions, celle issue de la théorie des ordres, ou celle issue de la théorie de l'algèbre universelle, peuvent être employées avantageusement, selon les circonstances. Dans [FP2], il est défini un ordre naturel sur les chemins de Dyck de longueur fixée qui induit une structure de treillis distributif. Le Chapitre 6 montre comment nous réussissons à transférer cet ordre aux partitions non croisées en exploitant des bijection bien connues (cf [S]), et en montrant que les partitions non croisées peuvent être dotées d'une structure de treillis distributif, structure qui a une grande importance combinatoire. Finalement, nous établissons que nos treillis sont isomorphes aux ensembles partiellement ordonnés des permutations évitant le motif 312, avec l'ordre induit par l'ordre de Bruhat fort du groupe symétrique.

Enfin dans le Chapitre 7 nous continuons à étudier des propriétés algébriques de classes d'objets combinatoires, en particulier de la classe de compositions exprimables sous forme de matrices (appelées en abrégé "*m*-compositions"), c'est-à-dire matrices dont les éléments sont non négatifs, avec *m* lignes, et

dont aucune colonne n'est identiquement nulle. Les m -compositions et en particulier celles avec $m = 2$ ont été introduites et définies dans [CFMRR]. Nous proposons des résultats énumératifs et quelques interprétations combinatoires. En particulier, nous étendons au cas m -dimensionnel beaucoup de propriétés des compositions ordinaires. Enfin, toujours en portant en avant l'argument déclencheur de cette thèse, nous présentons un algorithme efficace pour engendrer toutes les m -compositions d'un entier n avec m fixé.

Bibliografia

- [BDLPP] E. Barucci, A. Del Lungo, E. Pergola, R. Pinzani, *ECO: a methodology for the enumeration of combinatorial objects*, J. Difference Equ. Appl. 5 (1999), 435-490.
- [CFMRR] G. Castiglione, A. Frosini, E. Munarini, A. Restivo, S. Rinaldi, *Enumeration of L-convex polyominoes II. Bijection and area*, Proceedings of FPSAC'05, Taormina, 49.
- [FP2] L. Ferrari, R. Pinzani, *Lattices of lattice paths*, apparirà su J. Statist. Plann. Inference, consultabile su [arXiv:math.CO/0411427](https://arxiv.org/abs/math.CO/0411427).
- [MP] M. Poneti, *Generazione di strutture descritte da ricorrenze lineari*, Tesi di Laurea in Matematica, Università degli Studi di Firenze (2002).
- [P] E. Pergola, *ECO: a methodology for enumerating combinatorial objects*, Tesi di Dottorato, Università degli Studi di Firenze (1999).
- [Po] Y. Ponty, *Etudes combinatoire et génération aléatoire des structures secondaires d'ARN*, Mémoire du DEA Algorithmique (2003).
- [PR] J. G. Penaud, O. Roque, *Tirage à pile ou face de mots de Fibonacci*, Discrete Math. 256 (2002), 791-800.
- [R] O. Roque, *Tirages économes de structures combinatoires*, Tesi di dottorato, Université Bordeaux I (2001).
- [S] R. Simion, *Noncrossing partitions*, Discrete Math. 217 (2000), 367-409.
- [W] J. West, *Generating trees and the Catalan and Schröder numbers*, Discr. Math. 146 (1995), 247-262.

Capitolo 2

Generazione esaustiva di alcuni linguaggi regolari

Lo scopo della generazione esaustiva è lo sviluppo di algoritmi per elencare tutti gli oggetti di una certa classe. Gli algoritmi di generazione trovano applicazione in molte aree come il testing hardware e software, la chimica combinatoria, la teoria dei codici e la biologia computazionale. Comunque, tali algoritmi possono fornire molte informazioni circa le proprietà matematiche della classe di oggetti presa in considerazione. Al contrario, molte ricerche scientifiche e matematiche cominciano con un esame esaustivo di tutti i possibili casi di piccole istanze di un problema. Questi insiemi spesso hanno una struttura combinatoria; questo vuol dire che possiamo farne dei modelli con semplici strutture discrete, come cammini, grafi, parole di un linguaggio o permutazioni. C'è una metodologia in crescita e in maturazione per affrontare tali problemi in maniera sistematica. Nel contesto della generazione degli oggetti combinatori, lo scopo primario è che il costo della computazione sia proporzionale al numero degli oggetti generati. Un algoritmo per la generazione esaustiva sarà poi considerato *efficiente* se utilizza solo un costo di computazione costante per oggetto, in senso ammortizzato. Tali algoritmi si dicono godere della proprietà CAT (Constant Amortized Time, per esempio vedi <http://www.cs.uvic.ca/~fruskey/>). Molti articoli hanno studiato i linguaggi regolari in relazione ad automi a stati finiti [L, R, S], altri hanno trattato algoritmi con la proprietà CAT, ma nessuno ha considerato la possibilità di unire i due concetti. I risultati che invece otteniamo sono esattamente un connubio dei due concetti [BPP]. In questo capitolo definiremo un algoritmo CAT per la generazione esaustiva di una classe di linguaggi connessa con i sistemi di numerazione studiati da A. Fraenkel in relazione a molte differenti applicazioni (per esempio [F1]). In [F], l'autore prova il seguente teorema:

Teorema 2.1 *Sia $1 = x_0 < x_1 < \dots < x_n < \dots$ una sequenza di interi. Ogni intero non negativo m ha precisamente una rappresentazione nel sistema $\{x_n\}_{n \geq 0}$, della forma*

$$m = \sum_{i=0}^n d_i x_{n-i},$$

dove d_i sono interi non negativi che soddisfano la condizione:

$$d_n x_0 + \dots + d_{n-i} x_i < x_{i+1}, \quad i \geq 0.$$

Con $d_0 \dots d_n$ si intende la rappresentazione di m (di lunghezza n) nel sistema $\{x_n\}_{n \geq 0}$. Da notare che in tale rappresentazione aggiungiamo a sinistra gli zeri. Così un intero può avere più di una rappresentazione ma al massimo una di una data lunghezza.

$w(m, n)$ indica la rappresentazione di m di lunghezza n , sapendo che $m < x_{n+1}$ e che il sistema $\{x_n\}_{n \geq 0}$ è chiaramente fornito dal contesto. In un articolo successivo, Barcucci e Rinaldi [BR] applicano il Teorema 2.1 per dare un'interpretazione combinatoria ad alcune sequenze di interi in termini di linguaggi regolari. In particolare considerano sequenze $\{x_n\}_{n \geq 0}$ che soddisfano una ricorrenza della forma:

$$x_n = ax_{n-1} + bx_{n-2}, \quad (2.1)$$

con condizioni iniziali $x_0 = 1$, $x_1 = c$, dove $a, c \in \mathbb{N} \setminus \{0\}$, $b \in \mathbb{Z}$, e tali che $a > |b|$. Supponiamo che $x_{-1} = 0$, $x_0 = 1$, per avere $x_1 = c = a$ e $\Sigma = \{0, 1, \dots, a\}$. In questo caso per $b > 0$ la lettera a non compare come ultima lettera di una codifica perché la sequenza è $1, a, \dots$ quindi, per convenienza, assumiamo $c = a + 1$. Per garantire che $\{x_n\}_{n \geq 0}$ sia una sequenza di interi non negativi, consideriamo che $a^2 + 4b \geq 0$ come conseguenza della prima assunzione $a > |b|$; questa condizione implica anche che la sequenza definita è strettamente crescente eccetto che per il caso banale $x_n = x_{n-1}$. Alcuni studi sulle connessioni tra sistemi di numerazione, linguaggi regolari e ricorrenze lineari sono stati presentati anche in [L, S]. In questo capitolo utilizzeremo l'approccio presentato in [BR], ed avremo a che fare con sequenze che soddisfano (2.1). Pertanto richiamiamo brevemente alcune idee di base e risultati da [BR]. In poche parole, gli autori definiscono un linguaggio \mathcal{L} , sull'alfabeto $\Sigma = \{0, 1, 2, \dots, \alpha\}$, dove

$$\alpha = \max \left\{ a - 1, \left\lfloor \frac{a^2 + b - 1}{a} \right\rfloor \right\}.$$

L'insieme \mathcal{L} è definito come l'unione di insiemi mutualmente disgiunti \mathcal{L}_n , dove $\mathcal{L}_0 = \{\epsilon\}$ (ϵ è la parola vuota), e, per $n \geq 1$, $\mathcal{L}_n = \{w(m, n) : m < x_n\}$. \mathcal{L}_n è l'insieme di tutte le parole di lunghezza n in \mathcal{L} , e dalla costruzione abbiamo che la cardinalità di \mathcal{L}_n ($|\mathcal{L}_n|$) è x_n , per ogni $n \in \mathbb{N}$. Il linguaggio \mathcal{L} è regolare, e per caratterizzare le sue parole abbiamo bisogno di distinguere due casi:

1. $a \geq b \geq 0$. In questo caso l'insieme di simboli terminali è $\Sigma = \{0, 1, \dots, a\}$; il linguaggio \mathcal{L} può essere descritto come l'insieme di tutte le parole $w \in \Sigma^*$, $w = d_0 \cdots d_r$, con $d_i \in \Sigma$, e tali che se $d_i = a$, allora $d_{i+1} < b$, per ogni $i = 0, \dots, r$.
2. $a > -b \geq 0$, $c = a$. In questo caso l'insieme dei simboli terminali è $\Sigma = \{0, 1, \dots, a-1\}$, ed il linguaggio \mathcal{L} è l'insieme delle parole $w = u_0 \cdots u_r \in \Sigma^*$ tale che se $u_i = a-1$ allora $u_{i+1} \leq a-b-1$, e se $u_{i+1} = u_{i+2} = \dots = u_j = a-b-1$, $j > 0$, allora $u_{j+1} \leq a-b-1$.

Per comprendere appieno il cuore del problema, consideriamo un semplice esempio.

Esempio 2.1 Consideriamo i numeri di Pell, 1, 3, 7, 17, 41, 99, 239, \dots , (sequenza M1413 in [SP]) definiti dalla relazione di ricorrenza:

$$\begin{cases} x_n = 2x_{n-1} + x_{n-2} \\ x_0 = 1 \\ x_1 = 3 \end{cases} \quad (2.2)$$

Secondo il Teorema 2.1 ogni intero non negativo ha una sua rappresentazione in questo sistema, come mostrato in Tabella 2.1. Questa tabella viene utilizzata per gli esempi nel corso di tutto il capitolo. Da questa ricorrenza otteniamo un linguaggio \mathcal{L} sull'alfabeto $\Sigma = \{0, 1, 2\}$, poiché $\alpha = \max\{1, 2\}$. Ci riferiremo spesso ad \mathcal{L} come all'insieme delle *parole di Pell*. La relazione di ricorrenza definita in (2.1) rientra nel caso 1., ed il linguaggio è $\mathcal{L} = \{1 \cup 0 \cup 20\}^* \{2 \cup \epsilon\}$.

Lo scopo ultimo di [BR] è stato quello di mettere a punto una metodologia generale per generare sistematicamente molte classi di oggetti, basata sull'approccio dei sistemi di numerazione. L'importante risultato che presentiamo in questo capitolo è la definizione di un algoritmo per la generazione esaustiva di ogni sottoinsieme \mathcal{L}_n di un linguaggio \mathcal{L} , una volta fissati i coefficienti a, b , e dove i valori $\{x_n\}_{n \geq 0}$ soddisfano la (2.1). Poi studiamo il costo medio della procedura, provando che richiede tempo ammortizzato costante. In particolare, questo risultato fornisce la generazione esaustiva di ogni classe di oggetti \mathcal{O} , secondo un parametro p su \mathcal{O} , tale che

$$|\mathcal{O}_n| = |\{O \in \mathcal{O} : p(O) = n\}| = x_n.$$

17	7	3	1	i
0	0	0	0	0
0	0	0	1	1
0	0	0	2	2
0	0	1	0	3
0	0	1	1	4
0	0	1	2	5
0	0	2	0	6
0	1	0	0	7
0	1	0	1	8
0	1	0	2	9
0	1	1	0	10
0	1	1	1	11
0	1	1	2	12
0	1	2	0	13
0	2	0	0	14
0	2	0	1	15
0	2	0	2	16
1	0	0	0	17

Tabella 2.1: Codifica degli interi da 0 a 17 nel sistema di numerazione definito dai numeri di Pell.

2.1 Gli algoritmi di generazione

In questo paragrafo presentiamo un algoritmo di generazione, basato sui sistemi di numerazione, per produrre tutti gli elementi di \mathcal{L}_n , per n fissato, in ordine lessicografico. Utilizzeremo la seguente notazione: per ogni $m \geq 0$, $n \geq 1$, $w(m, n)$ indica la parola $w \in \mathcal{L}_n$ che codifica l'intero m . Per semplicità, quando non c'è ambiguità, tale parola sarà spesso denotata con $w(m)$. Allo stesso modo, per ogni $w \in \mathcal{L}$, $m(w) \in \mathbb{N}$ indicherà l'intero rappresentato dalla parola w . Per esempio, in riferimento all'Esempio 2.1, $w(53, 5) = 10112$, e $m(10020) = 47$. Le parole di \mathcal{L}_n sono ordinate naturalmente come segue:

$$w(0) < w(1) < w(2) < \dots < w(m) < w(m+1) < \dots < w(x_n - 1).$$

Si può facilmente verificare che questo ordine coincide con quello lessicografico. Anche in questo paragrafo consideriamo i due casi:

$$\begin{cases} x_n = ax_{n-1} + bx_{n-2}, b \geq 0 \\ x_0 = 1 \\ x_1 = a + 1, \end{cases} \quad (2.3)$$

$$\begin{cases} x_n = ax_{n-1} - bx_{n-2}, b \geq 0 \\ x_0 = 1 \\ x_1 = a. \end{cases} \quad (2.4)$$

La ragione per cui in (2.3) e (2.4) abbiamo scelto queste particolari condizioni iniziali è che permettono ad a di apparire in tutte le posizioni nelle codifiche di tutti gli interi (in particolare anche come ultimo termine). Il problema della generazione di parole di \mathcal{L}_n in ordine lessicografico è determinare il successore di una certa parola $w \in \mathcal{L}$ senza utilizzare la sua decodifica $m(w)$. Essenzialmente, cerchiamo un metodo efficiente per passare da $w(m)$ a $w(m+1)$, senza utilizzare pesanti operazioni di codifica e decodifica. L'idea di base di questi algoritmi è che, per la definizione di \mathcal{L} , ci sono sottostringhe che non possono comparire nella codifica degli interi m .

• **Ricorrenza (2.3):** sia \mathcal{L} il linguaggio associato alla ricorrenza (2.3).

Sia $w \in \mathcal{L}_n$, $w = w[1] \cdots w[n]$, $w[i] \in \Sigma$, $i = 1, \dots, n$; osserviamo che w non può contenere la sottostringa ab . Descriviamo un algoritmo per determinare il successore $s(w)$ di w , supponendo che $m(w) \neq x_n - 1$. La procedura comincia controllando la posizione più a destra di w , cioè, $w[n]$:

1. Se $w[n] \neq (b-1)$, e $w[n] \neq a$ allora facilmente,

$$s(w) = w[1] \cdots (w[n] + 1);$$

consideriamo l'Esempio 2.1, dove $a = 2$, $b = 1$, e sia $n = 5$. Se $w = 10101$, con $m(w) = 49$, allora $s(w) = 10102 (= 50)$.

2. se $w[n] = (b-1)$; ci sono due casi possibili:

- a. $w[n-1] = a$; se prendiamo $s(w) = w[1] \cdots (w[n] + 1)$, otteniamo la sottostringa "proibita" ab . Allora prendiamo $s(w)[n-1] = s(w)[n] = 0$, e incrementiamo di uno l'elemento in posizione $n-2$. Ancora, questo elemento potrebbe essere uguale a $b-1$, così in questo caso abbiamo bisogno di controllare la posizione precedente. Questa procedura va avanti fino a che troviamo una posizione $i_0 > 1$ tale che $w[i_0] \neq b-1$, o $w[i_0] = b-1$, ma $w[i_0-1] \neq a$. Se questa posizione non esiste, allora poniamo $i_0 = 1$. Una volta determinato i_0 ,

$$s(w) = w[1] \cdots w[i_0-1] (w[i_0] + 1) 0 \cdots 0.$$

Sempre in riferimento all'Esempio 2.1, sia $w = 01120$, con $m(w) = 30$; abbiamo $w[5] = 0 = b-1$, $w[4] = 2 = a$;

poiché $w[3] = 1 \neq b - 1$, allora $s(w) = 01200$
(= 31).

b. se $w[n - 1] \neq a$, semplicemenete

$$s(w) = w[1] \cdots w[n - 1] (w[n] + 1).$$

Sia $w = 11110$, con $m(w) = 68$; abbiamo $w[5] = 0 = b - 1$,
 $w[4] = 1 \neq a$, allora $s(w) = 11111$ (= 69).

3. se $w[n] = a$; ci sono ancora due possibilità:

a. $w[n - 1] = (b - 1)$ e $w[n - 2] = a$. Utilizzando le stesse motivazioni utilizzate per il caso 2.a, controlliamo la posizione più a destra $i_0 > 1$, tale che $w[i_0] \neq (b - 1)$ o $w[i_0 - 1] \neq a$. Se questa posizione non esiste, poniamo $i_0 = 1$. Una volta trovato,

$$s(w) = w[1] \cdots w[i_0 - 1] (w[i_0] + 1) 0 \cdots 0.$$

Sia $w = 10202$, con $m(w) = 57$; abbiamo $w[5] = 2 = a$,
 $w[4] = 0 = b - 1$, $w[3] = 2 = a$, $w[2] = 0 = b - 1$; allora
 $i_0 = 1$, e $s(w) = 11000$ (= 58).

b. $w[n - 1] \neq (b - 1)$ o $w[n - 2] \neq a$,

$$s(w) = w[1] \cdots (w[n - 1] + 1) 0.$$

Sia $w = 11112$, con $m(w) = 70$; abbiamo $w[5] = 2 = a$,
 $w[4] = 1 \neq b - 1$; allora $s(w) = 11120$ (= 71).

- **Ricorrenza (2.4):** Sia \mathcal{L} il linguaggio associato alla ricorrenza (2.3). Sia $w \in \mathcal{L}_n$; osserviamo che $w = w[1] \cdots w[n]$, $w[i] \in \Sigma$, $i = 1, \dots, n$, non può contenere sottostringhe della forma $(a - 1)(a - b - 1)^j(a - b)$, $j \geq 0$. Di seguito diamo un algoritmo, analogo al precedente, per determinare il successore $s(w)$ di w , ancora supponendo che $m(w) \neq x_n - 1$. Come al solito, leggiamo la parola w da destra verso sinistra a partire da $w[n]$.

1. Se $w[n] \neq (a - b - 1)$, e $w[n] \neq (a - 1)$

$$s(w) = w[1] \cdots w[n - 1] (w[n] + 1);$$

2. se $w[n] = (a - b - 1)$ dobbiamo controllare la posizione $n - 1$:
 se $w[n - 1] = (a - b - 1)$ troviamo la prima posizione $i_0 > 2$,
 se esiste, tale che $w[i_0] \neq (a - b - 1)$. Altrimenti poniamo
 $i_0 = 1$. Allora abbiamo i seguenti casi:

- a. se $w[i_0] = (a - 1)$,

$$s(w) = w[1] \cdots (w[i_0 - 1] + 1) 0 \cdots 0;$$

- b. altrimenti:

$$s(w) = w[1] \cdots w[i_0 - 1] w[i_0] w[i_0 + 1] \cdots (w[n] + 1);$$

3. se $w[n] = (a - 1)$, allora

$$s(w) = w[1] \cdots (w[n - 1] + 1) 0.$$

Esempio 2.2 Consideriamo la sequenza dei numeri di Fibonacci di indice dispari, $1, 3, 8, 21, 55, \dots$, (sequenza M2741 in [SP]) definita dalla relazione di ricorrenza:

$$\begin{cases} x_n = 3x_{n-1} - x_{n-2} \\ x_0 = 1 \\ x_1 = 3. \end{cases}$$

Codifichiamo gli interi non negativi utilizzando la sequenza $\{x_n\}_{n \geq 0}$ come sistema di numerazione (vedi anche la Tabella 2.2). Consideriamo

21	8	3	1	i
0	0	0	0	0
0	0	0	1	1
0	0	0	2	2
0	0	1	0	3
\vdots	\vdots	\vdots	\vdots	\vdots
0	2	0	2	18
0	2	1	0	19
\vdots	\vdots	\vdots	\vdots	\vdots
1	2	1	1	41
2	0	0	0	42

Tabella 2.2: Codifica di alcuni interi utilizzando la sequenza dei numeri di Fibonacci di ordine dispari come sistema di numerazione.

la parola $w = 0202$ ($= 18$), con $n = 4$; poiché $w[4] = 2 (= a - 1)$, siamo nella situazione 3, quindi $s(w) = 0210$ ($= 19$). Al contrario, per $w = 1211$ ($= 41$), abbiamo $w[4] = 1 = a - b - 1$, $w[3] = 1 = a - b - 1$; siamo nel caso 2.a.1, allora mettiamo $i_0 = 1$, ottenendo $s(w) = 2000$ ($= 42$).

2.2 Analisi del costo dell'algoritmo

Ora possiamo generare tutte le parole di \mathcal{L}_n , a cominciare da $w(0)$ fino a $w(x_n - 1)$. Ogni parola è ottenuta dalla precedente utilizzando uno dei due algoritmi. In questo paragrafo proveremo che il costo computazionale di ognuno dei due algoritmi è limitato da una costante. Di conseguenza, possiamo generare tutti gli elementi di \mathcal{L}_n in tempo ammortizzato costante. Per ogni $n \geq 1$, sia C_n il numero di tutte le sostituzioni di cifre necessarie per generare tutti gli elementi di \mathcal{L}_n , e sia P_n il numero di confronti necessari per generare tutti questi elementi. Inoltre, siano \overline{C}_n e \overline{P}_n il numero di sostituzioni e di confronti, rispettivamente, di cui abbiamo bisogno per generare tutte le parole di lunghezza n . Analizziamo separatamente i casi delle ricorrenze (2.3) e (2.4).

Ricorrenza (2.3): in riferimento alla tabella delle codifiche (per esempio la Tabella 2.1) consideriamo le $n - 1$ colonne più a sinistra. Il numero di sostituzioni necessarie per generare tutte le parole di lunghezza n è dato dal numero di sostituzioni necessarie per generare le parole di lunghezza $n - 1$ più quelle operate sull'ultima colonna della tabella, cioè, al massimo $x_n - 1$. Quindi abbiamo:

$$C_n = C_{n-1} + x_n - 1.$$

Da un semplice calcolo otteniamo $C_n = \sum_{i=0}^n x_i - n$. Poiché il numero di parole con lunghezza n è uguale a x_n , abbiamo

$$\overline{C}_n = \frac{1}{x_n} C_n = 1 + \sum_{i=0}^{n-1} \frac{x_i}{x_n} - \frac{n}{x_n}.$$

Consideriamo due possibili casi:

1. se $a \geq 2$ e $b \geq 0$, otteniamo

$$\frac{x_i}{x_{i+1}} = \frac{1}{a} - \frac{bx_{i-1}}{ax_{i+1}} \leq \frac{1}{a} \leq \frac{1}{2}.$$

Infine, abbiamo:

$$\frac{x_i}{x_n} \leq 2^{-(n-i)},$$

e poi

$$\overline{C}_n < 1 + \sum_{i=1}^n 2^{-i} < 2.$$

Il numero medio di sostituzioni per passare da un elemento al successivo è quindi al massimo due.

2. se invece $a = 1$, e $b = 1$, abbiamo

$$\frac{x_i}{x_{i+1}} = \frac{1}{\phi},$$

dove

$$\phi = \frac{-1 + \sqrt{5}}{2}.$$

Come nel primo caso abbiamo che \overline{C}_n è limitato da una costante.

Utilizzando le stesse argomentazioni per P_n abbiamo che il numero di confronti da fare è uguale al numero di sostituzioni realizzati più un confronto per ogni parola:

$$P_n = C_n + x_n - 1.$$

Quando sostituiamo una cifra nella parola, dobbiamo controllare al massimo la cifra precedente, di conseguenza

$$\overline{P}_n = \overline{C}_n + 1 - \frac{1}{x_n} < \overline{C}_n + 1.$$

Concludiamo che l'algoritmo per generare parole di \mathcal{L}_n nel caso della ricorrenza (2.3) utilizza un tempo ammortizzato costante.

Ricorrenza (2.4): Il calcolo è simile al caso precedente. Sia il numero medio di sostituzioni che quello di confronti sono limitati da costanti, quindi la generazione esaustiva gode della proprietà CAT.

2.3 Conclusioni

Abbiamo studiato semplici relazioni di ricorrenza, della forma (2.3) e (2.4). Comunque crediamo che sia interessante considerare altre classi di relazioni di ricorrenza ed i linguaggi ad esse associati utilizzando il metodo

dei sistemi di numerazione. Abbiamo cominciato a studiare il caso di relazioni di ricorrenza della forma

$$a_n = k_1 a_{n-1} + k_2 a_{n-2} + \cdots + k_r a_{n-r}, \quad n > r, \quad (2.5)$$

che soddisfano le condizioni

$$k_1 \geq k_2 \geq \cdots \geq k_r > 0, \quad (2.6)$$

ed i risultati sono simili a quelli in [BR] per le ricorrenze (2.1). Le cose si fanno più complicate rimuovendo o modificando le condizioni in (2.6) o considerando coefficienti non costanti.

Bibliografia

- [BPP] E. Barcucci, M. Poneti, R. Pinzani, *Exhaustive generation of some regular languages by using numeration systems*, pubblicato negli Atti del Convegno WORDS 2005;
- [BR] E. Barcucci, S. Rinaldi, *Some linear recurrences and their combinatorial interpretation by regular languages*, Theoret. Comput. Sci. 255 (2001), 679-686.
- [F] A. S. Fraenkel, *Systems of numeration*, Amer. Math. Monthly 92 (1985), 105-114.
- [F1] A. S. Fraenkel, *Arrays, numeration systems and Frankenstein games*, Theoret. Comput. Sci. 282/2 (2002), 271-284.
- [L] P. B. A. Lecomte, M. Rigo, *Numeration systems on a regular language*, Theory of Comput. Syst. 34 (2001,) 27-44.
- [R] M. Rigo, *Numeration systems on a regular language: arithmetic operations, recognizability and formal power series*, Theoret. Comput. Sci. 269 (2001), 469-498.
- [S] J. Shallit, *Numeration systems, Linear Recurrences, and Regular Sets*, Inform. and Comput. 113/2 (1994), 331-347.
- [SP] N. J. A. Sloane and S. Plouffe, *The encyclopedia of integer sequences*, Academic press (1996).

Capitolo 3

Cammini culminanti

Già nel Capitolo 2 abbiamo accennato al fatto che la combinatoria enumerativa è in stretto contatto con molte aree di ricerca; in questo capitolo affrontiamo brevemente un problema di biologia che viene trattato ed in parte risolto appunto con la combinatoria enumerativa. Stiamo parlando del problema degli allineamenti di DNA. Questo problema consiste nel rilevare le somiglianze sostanziali tra due sequenze proteiche o tra una sequenza ed un database. Infatti sapere se una proteina magari sconosciuta ha dei collegamenti o delle “somiglianze” con una conosciuta può portare a caratterizzarla, a scoprirne proprietà utili per lo sviluppo di vaccini o medicinali. Un classico e decisamente recente esempio può essere quello della SARS (Severe Acute Respiratory Syndrome). E' stata infatti scoperta la struttura tridimensionale della proteina che il coronavirus della SARS usa per infettare la cellula ospite. La struttura della proteina presenta caratteristiche molto particolari ed è una base di fondamentale importanza per arrivare alla creazione di specifici farmaci antivirali [ABC]. Un altro degli scopi primari della bioinformatica è quello di sviluppare dei database abbastanza completi da poter riconoscere il maggior numero di allineamenti possibili. Sono stati sviluppati molti algoritmi euristici per velocizzare il calcolo degli allineamenti locali: due di questi sono BLAST [LP] e Fasta [AGMLL]. Questi programmi sono basati sulla comparazione e riconoscimento di sottosequenze adiacenti: si suppone che le regioni di somiglianza siano ripartite in uno o molti piccoli frammenti, detti *semi*, che sono usati per rilevare le somiglianze potenziali. Più recente è lo studio di allineamenti di stringhe non contigue (dette *salti*) che porterebbero ad uno studio più selettivo del problema [CR, PW]. Per approfondimenti per quanto riguarda più specificatamente la parte di bioinformatica sono riferimenti importanti [Po] e [KNP].

Un allineamento è tanto più significativo quanto più alto è il suo *punteggio*. Il sistema più utilizzato è quello di assegnare un punteggio positivo ad

ogni confronto risultato positivo ed un punteggio negativo (penalità) ad ogni confronto risultato negativo.

Esempio 3.1 Consideriamo due sequenze di DNA ed associamo un punteggio ad ogni confronto, dove abbiamo assegnato $-p$ (o zero) ad ogni confronto negativo ed s (o uno) ad ogni confronto positivo.

sequenza 1	A	C	C	G	T	A	T	G	A	C	C	A	T	A
sequenza 2	A	A	C	G	A	A	T	G	G	C	A	T	T	A
punteggio	s	-p	s	s	-p	s	s	s	-p	s	-p	-p	s	s
codice binario	1	0	1	1	0	1	1	1	0	1	0	0	1	1

Tabella 3.1: Esempio di due sequenze di DNA con punteggi e sequenza binaria corrispondenti.

Come è facile immaginare, gli allineamenti interessanti sono quelli abbastanza lunghi da essere significativi e con piccole regioni con punteggio negativo. Per arrivare agli oggetti combinatori in relazione agli allineamenti diamo qualche utile definizione.

Consideriamo un allineamento senza salti che può essere in modo naturale riletto sotto forma di sequenza con punteggi positivi e negativi. Un tale allineamento si dice *omogeneo* se non contiene un segmento (sottosequenza) proprio con punteggio totale maggiore di quello dell'intero allineamento. Dato un allineamento, si definisce *segmento massimale con punteggio* il segmento omogeneo di lunghezza massima contenuto nell'allineamento. Quindi non considerando i possibili salti, gli allineamenti trovati quando si confrontano due sequenze sono di fatto segmenti massimali con punteggio. Infatti biologicamente i segmenti massimali con punteggio rivestono una posizione rilevante: se un allineamento non è omogeneo allora si può considerare come costituito da due allineamenti distinti; se un allineamento non è massimale, si può considerare come parte di un allineamento più grande.

3.1 Allineamento di sequenze e cammini culminanti

Consideriamo due sequenze di DNA, in particolare consideriamo un loro allineamento senza salti. Possiamo rappresentarlo come in tabella 3.1, cioè come sequenza binaria $A = (b_1, \dots, b_n)$, con $b_i \in \{0, 1\}$, dove 1 rappresenta un confronto positivo (*match*) e 0 uno negativo (*mismatch*). Supponiamo

di assegnare ad ogni match una costante positiva s e ad ogni mismatch una costante negativa $-p$. Quindi possiamo rappresentare A in modo alternativo, tramite le cosiddette *sequenze punteggio* indicate con $X_A = (x_1, \dots, x_n)$ con $x_i \in \{s, -p\}$ e $x_i = -p + b_i(s+p)$ dunque il punteggio dell'intero allineamento non è altro che $S(X) = \sum_{k=1}^n x_k$. Rileggiamo in questi termini la definizione di omogeneità che abbiamo dato precedentemente.

Definizione 3.1 Una sequenza binaria A (ed una sequenza punteggio associata X_A) si dice omogenea se $S(X_A)$ è strettamente maggiore di $S(X_A[i \dots j])$ per ogni segmento proprio $X_A[i \dots j] = (x_i, \dots, x_j)$ ($i > 1$ e $j < n$).

Per una sequenza binaria $A = (b_1, \dots, b_n)$ e la sequenza punteggio associata $X_A = (x_1, \dots, x_n)$, consideriamo l'evoluzione del punteggio dei prefissi $\sum_{i=1}^k x_i$ con $k = 1, \dots, n$. L'evoluzione può essere rappresentata da un cammino su \mathbb{Z}^2 partendo dal punto $(0, 0)$ con due passi possibili, $(1, s)$ e $(1, -p)$. La relazione biunivoca tra la sequenza binaria ed un cammino così costruito è

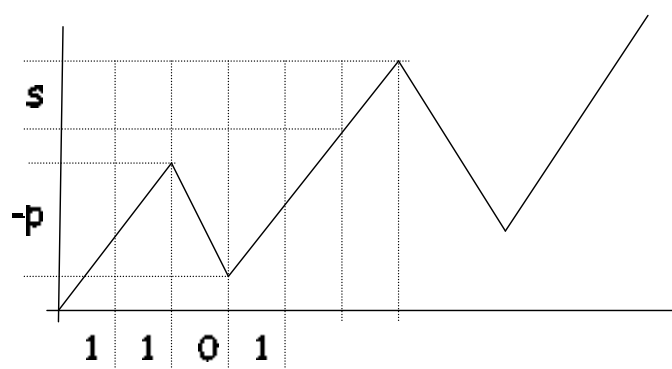


Figura 3.1: Esempio di allineamento univocamente associato ad un cammino.

\mathcal{C} su \mathbb{Z}^2 con punto iniziale in $(0, 0)$ e punto finale in (n, S) , che soddisfano le seguenti proprietà:

1. \mathcal{C} è positivo, cioè sta tutto sopra l'asse delle x ;
2. \mathcal{C} è *culminante*, cioè $\forall k < n, (k, y) \in \mathcal{C} \Rightarrow y < S$.

Un cammino che soddisfa le due condizioni si dice *cammino culminante (positivo)*. Siamo quindi arrivati a definire gli oggetti combinatori in “biezione” con gli allineamenti. Il nostro scopo è quindi generare esaustivamente proprio

questi oggetti combinatori. Vedremo che la loro generazione esaustiva, come quella casuale ed uniforme, dipende da più di un parametro. Osserviamo intanto che il valore S corrisponde all'altezza massima a cui può arrivare il cammino, quindi, d'ora in avanti, consideriamo il valore h che chiameremo *altezza del cammino* invece di S .

I cammini appena definiti allora possono essere detti *cammini culminanti in senso stretto* poiché cominciano nell'origine del piano e finiscono in un punto (n, h) senza aver mai raggiunto prima l'ordinata h . Possiamo invece indicare con *cammini culminanti in senso largo* (\mathcal{CC}) i cammini culminanti che cominciano nell'origine del piano e finiscono in un punto (n, h) che possono aver raggiunto ma non superato l'altezza h prima del punto di arrivo.

Osserviamo quindi che generare tutti i cammini culminanti in senso stretto di lunghezza $(n+1)$ ed altezza $(h+1)$ equivale a generare tutti quelli in senso largo (\mathcal{CC}) di lunghezza n e di altezza h .

Per semplicità studiamo i cammini \mathcal{CC} di lunghezza n ed altezza h (li indichiamo con $\mathcal{CC}(n, h)$ per avere sempre presenti i parametri in gioco). In generale abbiamo cammini $\mathcal{CC}(n, h)$ con passi del tipo (a, s) e $(b, -p)$ ma per comodità, visto che, per quanto riguarda la generazione esaustiva, le cose in sostanza non cambiano, consideriamo passi del tipo $(1, 1)$ ed $(1, -1)$ come possiamo osservare in Figura 3.2. Possiamo osservare come ogni cammino di

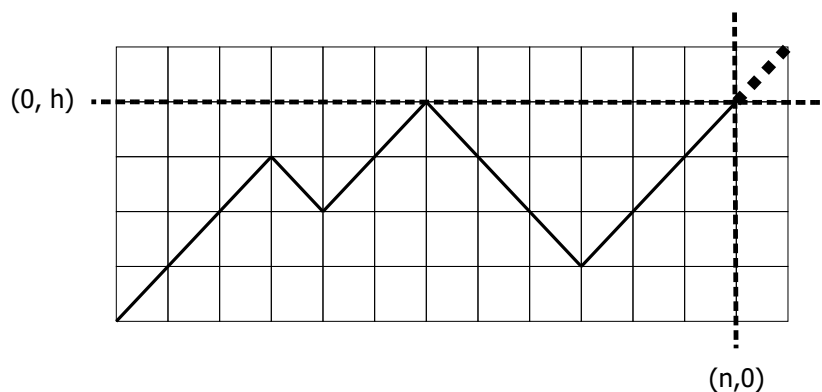


Figura 3.2: Esempio di cammino culminante in senso stretto con $n = 13$ ed $h = 5$. Viene evidenziata la legittimità di non considerare l'ultimo passo.

lunghezza n ed altezza h è come racchiuso in un rettangolo $n \times h$.

3.2 Generazione esaustiva dei $\mathcal{CC}(n, h)$

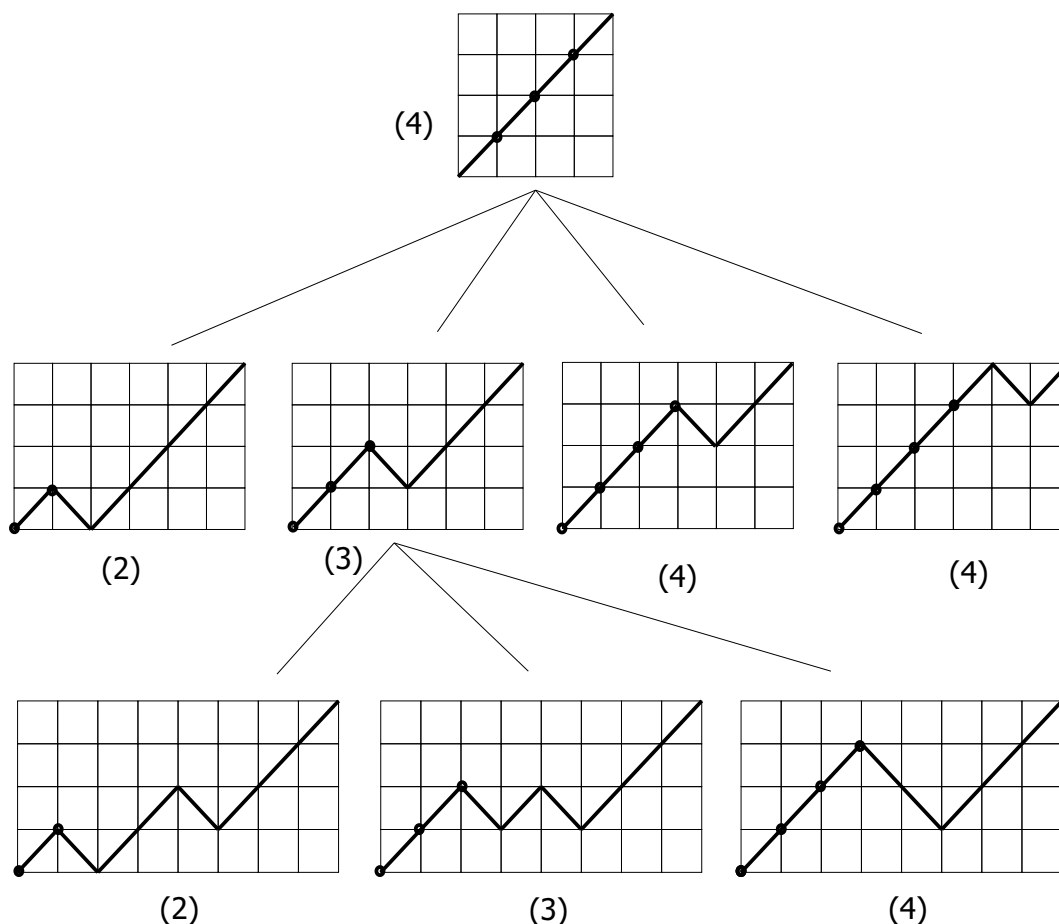
Come abbiamo già notato i \mathcal{CC} dipendono da due parametri tra loro distinti: la lunghezza n e l'altezza h . In realtà possiamo osservare che n deve essere maggiore o uguale ad h , in particolare $n = h + 2i$ con $i = 0, 1, \dots$. Vogliamo generare esaustivamente tutti i cammini $\mathcal{CC}(n, h)$, poichè questi cammini sono molto simili ai cammini di Dyck ¹ ed i cammini di Dyck possono essere agevolmente generati tramite il metodo ECO [BDLPP], applichiamo lo stesso metodo tenendo conto del fatto che h deve essere fissata a priori. Naturalmente, a differenza dei cammini di Dyck, che come radice dell'albero di generazione hanno il cammino vuoto, i $\mathcal{CC}(n, h)$ avranno come cammino di base quello di lunghezza $n = h$, per la dipendenza di n da h . Quindi al livello zero dell'albero di generazione avremo un cammino "diagonale" di lunghezza ed altezza h , quindi $h \times h$. Il passo seguente consiste nel determinare tutti i cammini del livello successivo, vale a dire quelli di lunghezza $n + 2$, cioè $h + 2$. Continuando così al livello i -esimo avremo tutti i $\mathcal{CC}(n, h)$ con $n = h + 2i$. In pratica passare da un livello al successivo significa aumentare di due la lunghezza dei cammini. Questo ancora di più suggerisce la stessa tecnica utilizzata per i cammini di Dyck: aggiungere un picco in "punti strategici" (i *siti attivi* definiti con il metodo ECO che stiamo utilizzando).

Dato un cammino di lunghezza n ed altezza h consideriamo la sua prima salita da sinistra, supponiamo che sia costituita da k passi ascendenti (con $k = 1, \dots, h$), per generare tutti i figli del suddetto cammino di lunghezza $n + 2$ attacchiamo un picco di altezza uno ad ogni nodo, che sarà il nostro sito attivo, della suddetta salita a partire da quello sull'asse. Un esempio con $h = 4$ fissata è rappresentato in figura 3.3, dove sono evidenziati i siti attivi con un cerchietto; e dove sono rappresentati i livelli dell'albero di generazione, a partire dalla radice (livello zero) in cui troviamo il cammino "diagonale" 4×4 . Una volta ottenuto l'albero di generazione possiamo scrivere la regola di successione che rappresenta la "crescita" dei cammini:

$$\begin{cases} (h) \\ (k) \rightsquigarrow (2)(3) \cdots (k)(k+1) & \text{con } k=2, 3, \dots, h-1 \\ (h) \rightsquigarrow (2)(3) \cdots (h-1)(h) \end{cases} \quad (3.1)$$

Come possiamo vedere nella regola di successione entrano in gioco due parametri: quello "classico" k che descrive l'evoluzione della generazione, e quello di partenza h . Da questo possiamo intuire che anche la funzione generatrice dovrà dipendere da h . Tramite il metodo del kernel [B *et al.*, Pr] dalla regola

¹Cammini positivi con passi del tipo $(1, 1)$ ed $(1, -1)$ che partono dal punto $(0, 0)$ ed arrivano al punto $(2n, 0)$.


 Figura 3.3: Generazione ECO dei cammini culminanti con $h = 4$.

otteniamo il sistema

$$\begin{cases} F_2(x) = x \sum_{l=2}^h F_l(x) \\ F_i(x) = x \sum_{l=i-1}^h F_l(x) \\ F_h(x) = x F_{h-1}(x) + 2x F_h(x) + x^h \end{cases} \quad \text{con } i = 3, \dots, h-1 \quad (3.2)$$

dove $F(x) = \sum_{l=2}^h F_l(x)$, da notare che partiamo da $h = 2$ perchè se l'altezza è uno, l'albero di generazione si riduce ad un semplice "filamento", cioè ogni volta abbiamo soltanto una possibilità di aggiunta: un picco sull'asse, ed otteniamo tutti i cammini culminanti con picchi tutti ad altezza uno. Di conseguenza è un caso banale. Dato il sistema 6.1, il problema è che possiamo calcolarne la soluzione solo assegnando a priori un'altezza specifica, cioè non riusciamo ad ottenere una soluzione generica dipendente da h . Quindi

h	2	3	4
F(x)	$\frac{x^2}{1-2x}$	$\frac{x^3}{1-3x+x^2}$	$\frac{x^4}{1-4x+3x^2}$
$[x]^n F(x)$	2^n	$(-\frac{3\sqrt{5}}{10} + \frac{1}{2})(\frac{3}{2} - \frac{\sqrt{5}}{2})^n + (\frac{3\sqrt{5}}{10} + \frac{1}{2})(\frac{3}{2} + \frac{\sqrt{5}}{2})^n$	$\frac{3^{n+1}}{2} - \frac{1}{2}$
Ricorrenza lineare	$f_n = 2^n$	$f_n = 3f_{n-1} - f_{n-2}$	$f_n = 4f_{n-1} - 3f_{n-2}$

 Tabella 3.2: Risultati trovati per $h = 2, 3, 4$.

h	x^0	x^1	x^2	x^3	x^4	x^5
1	1	-1				
2	1	-2				
3	1	-3	1			
4	1	-4	3			
5	1	-5	6	-1		
6	1	-6	10	-4		
7	1	-7	15	-10	1	
8	1	-8	21	-20	5	
9	1	-9	28	-35	15	-1

 Tabella 3.3: Tabella che visualizza i coefficienti dei polinomi denominatori di $F(x)$.

non siamo in grado di calcolare il generico coefficiente n -esimo della funzione generatrice, che è quello che conta tutti i cammini culminanti di altezza h e lunghezza n . Forniamo quindi dei risultati calcolati su casi specifici. In Tabella 3.2 vengono riassunti i risultati ottenuti per $h = 2, 3, 4$; abbiamo calcolato la funzione generatrice, il coefficiente n -esimo e la ricorrenza lineare corrispondente; tutto questo con l'aiuto del software applicativo Maple. Abbiamo cercato, a partire da questi risultati particolari, di trovare una relazione tra i coefficienti n -esimi delle funzioni generatrici con h variabile. Per far questo, poichè possiamo osservare che le funzioni generatrici considerate hanno la forma $F(x) = \frac{x^h}{P(x)}$ con $P(x)$ polinomio di grado h , organizziamo in Tabella 3.3 una specie di visione d'insieme dei coefficienti dei vari $P(x)$. Da questa abbiamo tratto delle ipotesi sulla relazione che cerchiamo. Indichiamo con $A = (a_{i,j})_{i,j=1,\dots,h}$ la matrice dei coefficienti di $P(x)$, possiamo notare che esiste la relazione

$$a_{i,j} = (-1)^j (|a_{i-1,j}| + |a_{i-2,j-1}|)$$

con $j = 1, 2, \dots, h + 1$. Quindi possiamo scrivere

$$F(x) = \frac{x^h}{\sum_{j=0}^h a_{h,j+1} x^j}.$$

Di conseguenza, osservando anche la Tabella 3.2, possiamo ipotizzare che il coefficiente n -esimo $[x]^n F(x)$ sia:

$$f_n = - \sum_{j=0}^h a_{h,j+2} f_{n-j}.$$

Da notare, come abbiamo accennato nell'introduzione di questa tesi, che potremmo considerare passi diversi da quelli di lunghezza uno per i cammini culminanti, grazie al fatto che potremmo dare pesi diversi ai confronti positivi o negativi effettuati. In ogni caso generalizzare l'algoritmo di generazione consta solo di qualche calcolo in più per quanto riguarda i siti attivi.

3.3 Generazione casuale dei $\mathcal{CC}(n, h)$

Abbiamo detto che un altro strumento utile alla risoluzione del problema degli allineamenti di sequenze proteiche sarebbe la generazione casuale dei cammini culminanti. Potremmo generare in modo casuale ed uniforme tali cammini sfruttando l'albero di generazione, come descritto in [P], ma il costo risulta essere troppo elevato ($O(n^3)$). Abbiamo quindi pensato di sfruttare un altro metodo di generazione casuale: quello *a rifiuto*, in particolare, il metodo *a rifiuto anticipato*. Il principio di base del metodo a rifiuto consiste nel generare oggetti di un sottoinsieme B di A costruendo prima quelli di A che dovranno essere più facili da ottenere. Per generare un oggetto di B si genera un oggetto di A , si controlla se è anche un oggetto di B , se lo è allora l'algoritmo termina con successo, se non lo è l'algoritmo ricomincia generando un altro oggetto di A . Il principio è semplice, ma il metodo è spesso difficile da applicare, per la difficoltà di generazione dell'insieme A . Infatti non solo bisogna costruire A in modo che la generazione di B sia uniforme, ma anche fare in modo che il rapporto tra le cardinalità dei suoi insiemi sia piccolo, meglio ancora se costante, per garantire l'efficienza media dell'algoritmo [R]. Mentre per i metodi a rifiuto si procede al controllo dell'appartenenza o meno di un oggetto all'insieme considerato alla fine della sua generazione, per quelli a rifiuto anticipato il controllo viene effettuato ad ogni passo. Ciò non toglie che le difficoltà siano le stesse. Per generare in modo casuale ed uniforme un cammino culminante di altezza h e lunghezza

n partiamo dal punto $(0,0)$, estraiamo in modo casuale un passo crescente o uno decrescente, e poi controlliamo se sono sempre rispettate le proprietà dei cammini culminanti: devono essere positivi (quindi con le estrazioni dei passi non possiamo scendere sotto l'asse) e non possono superare l'altezza h . Praticamente i cammini devono rimanere in un rettangolo $n \times h$. Osserviamo quindi che la generazione casuale di un cammino culminante in sé non è difficile, il problema è conoscere il rapporto tra le cardinalità dei due insiemi di cui parlavamo all'inizio. Infatti possiamo solo ipotizzare che la generazione casuale di questi cammini sia molto simile a quella dei prefissi dei cammini di Dyck $[R]$, ma non siamo in grado di calcolare, per h ed n generici, quanti sono in totale i cammini che possono essere generati tirando ogni volta un passo crescente o uno decrescente, e quante sono le probabilità di errore. A questo proposito bisogna aggiungere che dall'osservazione di molti casi particolari (Figura 3.4), si nota che non è proficuo assegnare la stessa probabilità di uscire ad entrambi i passi, in realtà noi abbiamo intenzione di continuare in futuro a studiare il problema attribuendo al passo crescente una probabilità di uscita pari a $\frac{n+h}{2n}$ ed a quello decrescente $\frac{n-h}{2n}$.

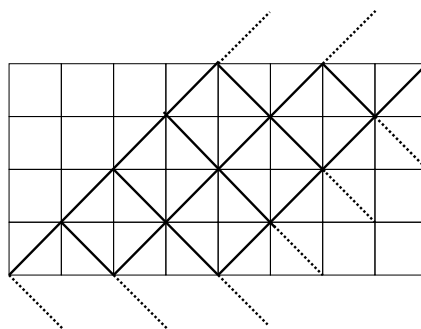


Figura 3.4: Generazione casuale dei cammini culminanti con $h = 4$ ed $n = 8$. Le linee continue rappresentano i possibili cammini corretti, mentre quelle tratteggiate sono i passi che portano a cammini non culminanti.

Bibliografia

- [ABC] F. Anselmi, A. Bernini, V. Causarone, S. Chiellini, A. Ciutti, F. Finetti, N. Menciassi, N. Niccolai, F. Prischi, O. Spiga, *Molecular modelling of S1 and S2 subunits of SARS coronavirus spike glycoprotein*, Biochemical and Biophysical Research Communications 310/1 (2003), 78-83.
- [AGMLL] S. Altschul, W. Gish, W. Miller, E. Myers, D. Lipman, *Basic local alignment search tool*, Journal of Molecular Biology 215 (1990), 403-410.
- [B *et al.*] C. Banderier, M. Bousquet-Mélou, A. Denise, P. Flajolet, D. Gardy, D. Gouyou-Beauchamps, *Generating functions for generating trees*, Discrete Math. 246 (2002), 29-55.
- [BDLPP] E. Barcucci, A. Del Lungo, E. Pergola, R. Pinzani, *ECO: a methodology for the enumeration of combinatorial objects*, J. Difference Equ. Appl. 5 (1999), 435-490.
- [CR] A. Califano, I. Rigoutsos, *A fast look-up algorithm for string homology*, Proceedings of the first International Conference on Intelligence Systems for Molecular Biology (1993), 56-64.
- [KNP] G.Kucherov, L. Noé, Y. Ponty, *Estimating seed sensitivity on homogeneous alignments*, Thème 2-Génie logiciel et calcul symbolique, Projets Adage & LRI, Rapport de recherche n.5047, INRIA (2003).
- [P] E. Pergola, *ECO: a methodology for enumerating combinatorial objects*, Tesi di Dottorato, Università degli Studi di Firenze (1999).
- [Po] Y. Ponty, *Etudes combinatoire et génération aléatoire des structures secondaires d'ARN*, Mémoire du DEA Algorithmique (2003).
- [Pr] E. Prodinger, *The kernel method: a collection of examples*, Sém. Lothar. Combin. 50:Article B50f (2003).

-
- [LP] D. Lipman, W. Pearson, *Improved tools for biological sequence comparison*, Proceedings Natl. Acsd. Sci. USA 85 (1988), 2444-2448.
- [PW] P. Pevzner, M. Waterman, *Multiple filtration and approximate pattern matching*, Algorithmica 13 (1995), 135-154, versione preliminare in Combinatorial Pattern Matching 1993.
- [R] O. Roques, *Tirages économes de structures combinatoires*, Tesi di dottorato, Université Bordeaux I (2001).

Capitolo 4

Generazione a testa o croce

I metodi per la generazione casuale permettono di studiare le numerose proprietà delle strutture combinatorie. Ad ognuna di queste strutture possiamo associare uno o più parametri di tipo numerico detti “*dimensioni*”. Estrarre in modo casuale e uniforme un oggetto combinatorio vuol dire generare un oggetto di dimensione fissata n in modo tale che tutti gli oggetti della stessa dimensione abbiano la stessa probabilità di essere estratti. In questo caso concentreremo la nostra attenzione sui numeri di Fibonacci (M1539 in [SP]) che enumerano molti oggetti combinatori e sono descritti da una semplice ricorrenza lineare dalla quale si costruisce il linguaggio regolare di Fibonacci. In [PR], gli autori hanno trovato un metodo per generare casualmente una parola del linguaggio di Fibonacci e hanno proposto una possibile estensione del loro algoritmo alla classe dei linguaggi razionali, detti “pattern star”, definiti da $L = (m_1 \cdots m_k)^*$, dove gli m_i sono pattern (motivi) di lunghezza qualsiasi. In questo caso ogni motivo di lunghezza l deve essere estratto con probabilità p^l dove p è l’unica radice reale positiva del polinomio $1 - \sum_{i=1}^k x^{\|m_i\|}$ (per la regola dei segni di Descartes [S, K]). Utilizzeremo questa idea prima per generare in modo casuale ed uniforme una parola del linguaggio di Tribonacci, e poi per fare la stessa cosa con una parola di un qualsiasi linguaggio regolare descritto da una ricorrenza lineare a tre termini con coefficienti interi che soddisfano determinate condizioni [BPeP].

4.1 Parole di Fibonacci

Cominciamo col richiamare qualche nozione di base [PR].

Sia

$$\begin{cases} f_n = f_{n-1} + f_{n-2} \\ f_0 = 0 \\ f_1 = 1 \end{cases} \quad (4.1)$$

la classica ricorrenza di Fibonacci, e $L = \epsilon + La + Lbb$ il linguaggio razionale di Fibonacci sull'alfabeto $\Sigma = \{a, b\}$, $L_n = \{w \in L \mid |w| = n\}$.

In [PR] gli autori generano in modo casuale una parola $w \in L_n$, con n fissato, considerando i possibili motivi mutualmente esclusivi che possono essere concatenati insieme: a e bb , il motivo a con probabilità p .

Utilizzando l'aritmetica floating point [DZ] il metodo classico di procedere consiste nell'estrarre un numero reale r in $[0, 1]$ in modo uniforme e continuo secondo la lunghezza, e scegliere a se $0 \leq r \leq p$ e bb se $p < r \leq 1$, dove p è la probabilità di a . Altrimenti, si può estrarre r considerando la sua rappresentazione binaria e successivamente paragonare le sue componenti binarie con quelle di p . Penaud e Roques definiscono un algoritmo che si ferma appena r differisce da p . Il primo bit più a sinistra r_i per cui r e p sono diversi è detto *bit decisivo*. Questo tipo di estrazione di r è detta "*tirage paresseux*" (cioè "estrazione pigra"), perchè l'estrazione dei bit si ferma appena il valore effettivo di r diventa più grande o più piccolo di quello di p . Sia w un codice binario, siano w_i l' i -esima cifra e $val(w)$ lo sviluppo di w come numero reale, cioè:

$$val(w) = \sum_{i \geq 1} w_i 2^{-i}.$$

In altre parole il bit decisivo è tale che:

1. $1 \leq j < i \Rightarrow r_j = p_j$;
2. $(r_i = 1 \wedge p_i = 0) \vee (r_i = 0 \wedge p_i = 1)$.

Da notare che, in questo caso, p è trattato come se fosse conosciuto e finito, ma in realtà è uguale a $\frac{1}{\phi}$, con $\phi = \frac{1+\sqrt{5}}{2}$ (*radice aurea*), quindi ha un numero infinito di decimali. Per questa ragione gli autori suppongono di non conoscere il valore di p e valutano il segno del polinomio su $val(r)$, definendo così la probabilità. Determiniamo il polinomio come segue: assumendo che a abbia probabilità p e che bb abbia probabilità q , per rispettare l'uniformità secondo la lunghezza dobbiamo avere $q = p^2$ e $p + q = 1$. Quindi otteniamo l'identità $1 - p - p^2 = 0$. Consideriamo il polinomio $P(x) = 1 - x - x^2$, avente

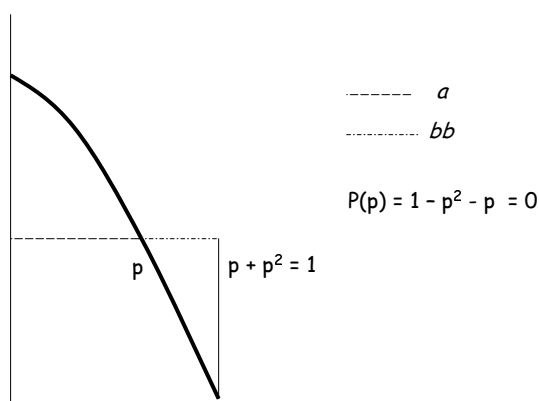


Figura 4.1: Rappresentazione grafica del polinomio $P(x)$ che definisce la probabilità nel caso di Fibonacci nell'intervallo $[0, 1]$, con i motivi e le loro probabilità.

p come radice, tale polinomio è ovviamente decrescente in $[0, 1]$, come si vede in figura 4.1.

Supponiamo che il valore $p \in [0, 1]$ sia sconosciuto e che $P(x)$ sia decrescente. La Proprietà 3 in [PR] è fondamentale per capire come valutare $P(x)$, questo porta alla generazione casuale ed uniforme di una parola di Fibonacci di lunghezza fissata.

Proposizione 4.1 *Dato p , se r è estratto uniformemente bit a bit, allora:*

1. $r_i = 1 \Rightarrow \{p_i = 0 \Leftrightarrow (\text{val}(r_1 r_2 \dots r_{i-1} 1) > \text{val}(p))\}$;
2. $r_i = 0 \Rightarrow \{p_i = 1 \Leftrightarrow (\text{val}(r_1 r_2 \dots r_{i-1} 1) \leq \text{val}(p))\}$.

Dopo ogni estrazione di un bit r_i , per controllare se questo è il bit decisivo, è sufficiente calcolare $P(\text{val}(r_1 r_2 \dots r_{i-1} 1))$. Un numero r è generato bit a bit e, dopo ogni estrazione, vengono calcolati il valore di r ($= \text{val}(r)$) e $P(\text{val}(r))$ quindi:

- se $r_i = 0$ e $P(\text{val}(r_1 r_2 \dots r_{i-1} 1)) \geq 0$ allora r_i è il bit decisivo, e così $\text{val}(r_1 r_2 \dots r_{i-1} 0) \leq \text{val}(p)$; in questo caso si aggiunge il motivo a avente probabilità p ;
- se $r_i = 1$ e $P(\text{val}(r_1 r_2 \dots r_{i-1} 1)) < 0$ allora r_i è il bit decisivo, e così $\text{val}(r_1 r_2 \dots r_{i-1} 1) > \text{val}(p)$; in questo caso si aggiunge il motivo bb avente probabilità $(1 - p) = p^2$.

Naturalmente, l'estrazione dei bit continua fino a che non viene raggiunta una delle due situazioni precedenti e poi fino a che non viene generata in modo completo una parola di lunghezza fissata n . Questo è un metodo a rifiuto,

poiché, se avessimo $w \in L_{n-1}$ e, con l'applicazione dell'algoritmo, venisse aggiunto il motivo bb , si otterrebbe la parola $w' \in L_{n+1}$, che dovrebbe essere rifiutata perchè sarebbe di lunghezza $n + 1$. In tale situazione l'algoritmo ricomincia dal primo passo (Algoritmo 1 in [PR]).

4.2 Parole di Tribonacci

Introduciamo ora un metodo simile a quello visto per generare in modo uniforme e casuale parole di una particolare famiglia di numeri che è una generalizzazione di Fibonacci: i numeri di Tribonacci. Tali numeri sono generati dalla seguente ricorrenza lineare:

$$\begin{cases} f_n = f_{n-1} + f_{n-2} + f_{n-3} \\ f_{-1} = 1 \\ f_0 = 1 \\ f_1 = 2 \end{cases} \quad (4.2)$$

Il linguaggio L di Tribonacci è razionale sull'alfabeto $\Sigma = \{a, b, c\}$, $L = \epsilon + La + Lbb + Lccc$ e $L = \{\epsilon, a, aa, bb, aaa, abb, bba, ccc, \dots\}$ è l'insieme infinito delle parole, ordinate secondo la lunghezza crescente. Sia L_n l'insieme delle parole di Tribonacci di lunghezza n . Possiamo ottenere una parola di lunghezza n in tre modi possibili: $L_{n-1}a$, $L_{n-2}bb$, $L_{n-3}ccc$. Naturalmente, se il motivo estratto è bb e la parola corrente a cui dobbiamo aggiungerlo ha lunghezza $(n - 1)$, la nuova parola avrà lunghezza $(n + 1)$, quindi tale parola sarà rifiutata e la procedura ricomincia dall'inizio, come mostrato nell'Algoritmo 1. Per estrarre ogni parola di L_n in modo equiprobabile, la parola viene generata motivo dopo motivo tramite la scelta di quello da aggiungere alla parola corrente: il motivo a con probabilità p , il motivo bb con probabilità q , il motivo ccc con probabilità v . Per uniformità secondo la lunghezza, abbiamo: $q = p^2$ e $v = p^3$, inoltre $p + q + v = 1$. In questo modo p è la radice sconosciuta del polinomio $P(x) = 1 - x - x^2 - x^3$ in $[0, 1]$. Ad ogni passo della costruzione della parola abbiamo una generazione casuale di r per scegliere uno dei tre motivi tramite una generalizzazione del metodo trovato per i numeri di Fibonacci. Fondamentalmente il numero reale r deve essere estratto in forma binaria e successivamente viene valutato in $val(r1)$ il polinomio $P(x)$ per la scelta del motivo da aggiungere alla parola corrente di Tribonacci. Sfortunatamente in questo caso non è troppo semplice valutare il segno del polinomio perchè, quando la valutazione è negativa, il motivo da aggiungere non è determinato in modo univoco: il numero reale r è in $(p, 1] = (p, p + p^2] \cup (p + p^2, 1 - p - p^2 + p^3]$, questi due ultimi intervalli corrispondono rispettivamente alla scelta dei motivi bb

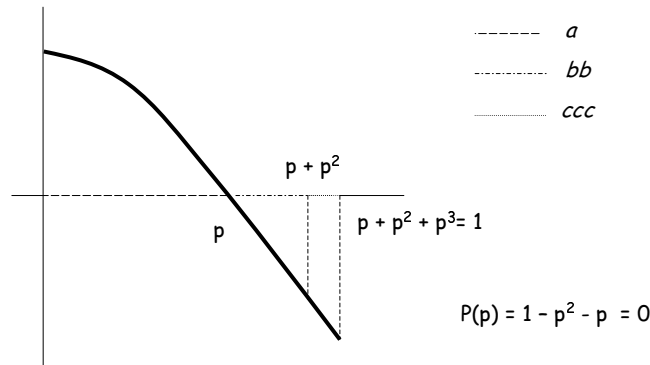


Figura 4.2: Rappresentazione grafica del polinomio $P(x)$ che definisce la probabilità nel caso di Tribonacci.

e *ccc* (vedi figura 4.2), sempre considerando le probabilità dei singoli motivi. Per ottenere un algoritmo a terminazione assicurata possiamo introdurre un

Algoritmo 1 *EstrazioneUniforme*(n)

Require: un intero n .

Ensure: una parola casuale uniforme w di lunghezza n .

```

 $w = \epsilon$ ;
while  $|w| < n$  do
     $m = \text{RandomPattern}$ ;
     $w = wm$ ;
    if  $|w| > n$  then
         $w = \epsilon$ ;
    end if
end while
if  $|w| > n$  then
    “algoritmo che finisce”.
end if
    
```

intero T che rappresenti il numero massimo di iterazioni permesse. In questo modo consideriamo una variabile k che conta il numero di estrazioni effettuate e l'algoritmo termina appena si verifica la situazione $k > T$. Dobbiamo confrontare il corrente codice binario costruito tramite l'algoritmo, con la sconosciuta ma ben determinata probabilità p . Per fare questo dobbiamo valutare il segno del polinomio $P(x) = 1 - x - x^2 - x^3$ calcolato in $(\text{val}(r1))$, secondo il nuovo bit estratto per ottenere r . Quindi:

- se $r_i = 0$ ed il segno di $P(\text{val}(r1))$ è positivo, allora è stato trovato

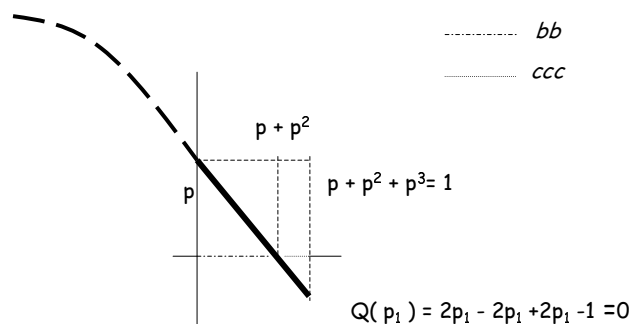


Figura 4.3: Rappresentazione grafica della trasformazione dal polinomio $P(x)$ a $Q(t)$.

il bit decisivo e r risulta essere più piccolo di p , quindi a è il motivo estratto (cioè il motivo avente probabilità p);

- se $r_i = 1$ ed il segno di $P(\text{val}(r1))$ è negativo, allora è stato trovato il bit decisivo e r risulta essere più grande di p ma non sappiamo ancora se r è più piccolo o più grande di $(p + p^2)$, che è la probabilità del motivo bb . Di conseguenza dobbiamo fare una seconda valutazione di $P(x)$.

Operiamo a questo proposito un cambio di variabile per considerare come nuova radice $(p + p^2)$, come in figura 4.3. Il vecchio polinomio è $P(x) = 1 - x - x^2 - x^3$, con $P(p) = 1 - p - p^2 - p^3 = 0$.

Consideriamo una nuova variabile p_1 :

$$p_1 = \frac{p^2}{1-p}.$$

Da notare che $1 - p = p^2 + p^3$, quindi:

$$p_1 = \frac{1}{1+p} \longrightarrow p = \frac{1-p_1}{p_1}.$$

Con questa sostituzione otteniamo un nuovo polinomio $Q(t) = 2t^3 - 2t^2 + 2t - 1$ tale che $Q(p_1) = 2p_1^3 - 2p_1^2 + 2p_1 - 1 = 0$.

Secondo la regola di Descartes il grafico del polinomio $Q(t)$ è analogo a quello del precedente in $[0, 1]$.

Adesso dobbiamo valutare $Q(\text{val}(r1))$ ¹:

¹Da ricordare che $r_i = 1$.

- se $Q(\text{val}(r1))$ è negativo, allora è stato trovato il bit decisivo e r è più grande di $(p + p^2)$, quindi il motivo estratto è *ccc* con probabilità p^3 ;
- altrimenti il bit non è decisivo e viene estratto un altro bit, z_1 , che sarà l' r_{i-1} -esimo bit del numero reale r e viene valutato ancora il polinomio $Q(t)$, in $\text{val}(r_1 \cdots r_{i-1}11)$ con la stessa procedura utilizzata per il polinomio iniziale: il bit è decisivo e si estraggono rispettivamente *ccc* e *bb* a seconda che la valutazione di $Q(t)$ in $\text{val}(r1)$ sia negativa con $z_1 = 1$ e positiva con $z_1 = 0$, altrimenti r deve essere aggiornato a rz_1 e deve essere generato un altro bit.

La procedura rimane sulla valutazione di $Q(t)$ finchè non viene trovato il bit decisivo, poi ricomincia a generare in modo casuale e uniforme un bit z per r e per la “ P -valutazione”.

Ovviamente la procedura prosegue fino alla completa generazione di una parola di Tribonacci di lunghezza fissata n ed è descritta nell’ Algoritmo 2, dove r_i e p_i sono i singoli bit rispettivamente di r e p e l’ i -esimo bit è quello decisivo.

4.2.1 Complessità

La complessità media dell’Algoritmo 1 è il prodotto del numero medio di chiamate alla funzione *MotivoCasuale* per la generazione di una parola e del numero medio di estrazioni richieste per ottenere una parola di Tribonacci di lunghezza n . Consideriamo un calcolo analogo a quello fatto in [PR] per le parole di Fibonacci: sia $A_n = \sum_{f \in L_n} |f|_a + |f|_{bb} + |f|_{ccc}$ il numero totale di motivi *a*, *bb*, *ccc* presenti in una parola di lunghezza n . Poichè $L = L_a + L_{bb} + L_{ccc}$, consideriamo $l_n = \text{cardinalità di } L_n$. Possiamo scrivere:

$$\begin{aligned}
 A_n &= \sum_{f \in L_{n-1}a} |f|_a + |f|_{bb} + |f|_{ccc} + \sum_{f \in L_{n-2}bb} |f|_a + |f|_{bb} + |f|_{ccc} + \\
 &\quad + \sum_{f \in L_{n-3}ccc} |f|_a + |f|_{bb} + |f|_{ccc} = \\
 &= \sum_{f \in L_{n-1}} (|f|_a + 1) + |f|_{bb} + |f|_{ccc} + \sum_{f \in L_{n-2}} |f|_a + (|f|_{bb} + 1) + |f|_{ccc} + \\
 &\quad + \sum_{f \in L_{n-3}} |f|_a + |f|_{bb} + (|f|_{ccc} + 1) = \\
 &= A_{n-1} + A_{n-2} + A_{n-3} + L_{n-1} + L_{n-2} + L_{n-3}.
 \end{aligned}$$

con condizioni iniziali $A_0 = 0$, $A_1 = 1$, $A_2 = 3$, $A_3 = 8$, $A_4 = 17$, $A_5 = 43$, e, dal momento che $L_n = L_{n-1} + L_{n-2} + L_{n-3}$, per $n \geq 6$, otteniamo l’espressione:

$$A_n = 2A_{n-1} + A_{n-2} - 3A_{n-4} - 2A_{n-5} - A_{n-6},$$

Algoritmo 2 *MotivoCasuale* ($P(x), Q(t)$)

Require: : $P(x), Q(t)$.**Ensure:** : il motivo *a* con probabilità p , il motivo *bb* con probabilità p^2 , o il motivo *ccc* con probabilità p^3 .

```

r =  $\epsilon$ ;
BitDecisivo = FALSE;
while BitDecisivo = FALSE do
  z = un bit casuale uniforme
  if [z = 1 &&  $P(\text{val}(r1)) < 0$ ] then
    z è decisivo per il suo confronto con  $p$ , ma:
    if  $Q(\text{val}(r1)) < 0$  then
      z è definitivamente decisivo
      BitDecisivo = TRUE;
      viene estratto ccc
    else
      r = rz
      z1 = un bit casuale uniforme
      if [z1 = 1 &&  $Q(\text{val}(r1)) < 0$ ] then
        z1 è definitivamente decisivo
        BitDecisivo = TRUE;
        viene estratto ccc
      else
        if [z1 = 0 &&  $Q(\text{val}(r1)) \geq 0$ ] then
          z1 è definitivamente decisivo
          BitDecisivo = TRUE;
          viene estratto bb
        end if
      end if
      r = rz1
    end if
  else
    if [z = 0 &&  $P(\text{val}(r1)) \geq 0$ ] then
      z è definitivamente decisivo
      BitDecisivo = TRUE;
      viene estratto a
    end if
  end if
  r = rz
end while

```

quindi

$$A_n = [x_n] \left(\frac{4x^5 - 2x^4 + x^3 + x^2 + x}{2x^5 + 3x^4 - x^2 - 2x + 2} \right).$$

A differenza del caso di Fibonacci, ora non possiamo trovare una buona forma per il coefficiente n -esimo della funzione generatrice. Indicando con $E(X)$ il numero di chiamate alla funzione `MotivoCasuale`, è tuttavia semplice osservare che tale numero è compreso nell'intervallo $[\lceil \frac{n}{3} \rceil, n]$ ed il numero medio quindi sarà compreso nell'intervallo $[\frac{1}{3}, 1]$.

Allo stesso tempo consideriamo una variabile Y che conta il numero di estrazioni necessarie per ottenere una parola corretta; il numero medio quindi è $E(Y) = L_n(\frac{1}{\gamma})^n$, dove L_n è il coefficiente n -esimo di $\frac{1}{(1-x-x^2-x^3)}$, che è la funzione generatrice dei numeri di Tribonacci e $\frac{1}{\gamma}$ è la radice più grande del polinomio $(1-x-x^2-x^3)$. Possiamo applicare il seguente teorema studiato da Flajolet e Sedgewick in [FS].

Teorema 4.1 (*Comportamento asintotico per ricorrenze lineari*) *Supponiamo che una funzione generatrice $\frac{f(x)}{g(x)}$, con $f(x)$ e $g(x)$ relativamente prime tra loro e $g(0) \neq 0$, abbia un unico polo $\frac{1}{\gamma}$ di modulo minimo (cioè, $g(\frac{1}{\alpha}) = 0$ e $\alpha \neq \gamma$ implica che $|\frac{1}{\alpha}| > |\frac{1}{\gamma}|$, o $|\alpha| < |\gamma|$). Allora, se la molteplicità di $\frac{1}{\gamma}$ è ν , abbiamo*

$$[x^n] \frac{f(x)}{g(x)} \sim C \gamma^n n^{\nu-1} \quad \text{con} \quad C = \nu \frac{(-\gamma)^\nu f(\frac{1}{\gamma})}{g^{(\nu)}(\frac{1}{\gamma})}.$$

Nel nostro caso $\frac{1}{(1-x-x^2-x^3)} = \frac{f(x)}{g(x)}$, $\nu = 1$, $\frac{1}{\gamma} = 0.54$, quindi $L_n = C(0.54)^{-n}$ e

$$E(Y) = L_n(\frac{1}{\gamma})^n = C(0.54)^{-2n}$$

di conseguenza, se definiamo \mathcal{C}_n la complessità media dell'algoritmo, otteniamo

$$\mathcal{C}_n = E(X)E(Y) \leq nC(0.54)^{-2n}$$

Quindi dobbiamo considerare la lunghezza delle possibili parole da generare; per esempio, per generare una parola di Tribonacci di lunghezza n , se si è generata una parola di lunghezza $(n-2)$ ci sono due possibilità per arrivare alla parola finale in modo corretto senza avere un rifiuto: aggiungere due volte il motivo a oppure una volta il motivo bb . In questo modo dobbiamo considerare un altro tipo di probabilità e così non possiamo valutare lo stesso polinomio.

Il numero di rifiuti è $O(1)$ ed il numero medio di motivi da scegliere per generare una parola di lunghezza fissata n decresce al crescere della lunghezza dei motivi.

4.3 Generalizzazione

Si noti che è possibile generalizzare il concetto espresso per la sequenza di Tribonacci a ricorrenze lineari del tipo

$$f_n = af_{n-1} + bf_{n-2} + cf_{n-3}, \quad (4.3)$$

con $a, b, c \in \mathbb{N}$, $a \geq b \geq c \geq 0$ e condizioni iniziali f_0, f_1, f_2 . Definiamo p la probabilità che un motivo di lunghezza 1 venga estratto; otteniamo:

$$1 - ap - bp^2 - cp^3 = 0 \Rightarrow P(x) = 1 - ax - bx^2 - cx^3$$

In [BR] Barcucci e Rinaldi hanno studiato una connessione tra i linguaggi regolari, le ricorrenze lineari ed i sistemi di numerazione. Grazie a queste nozioni possiamo considerare i possibili motivi da aggiungere ad una parola $w \in L$ per ottenere un'altra parola w' in modo tale che anche w' stia in L , ma, ovviamente, di lunghezza più grande di quella della parola di partenza w .

Esempio 4.1 Consideriamo la ricorrenza lineare per i numeri di Tribonacci come in (4.2) con sequenza $1, 1, 2, 4, 7, 13, 24, \dots$ (A081172 in [SP]) e linguaggio $\mathcal{L} = \{0 \cup 10 \cup 110\}^*$. Quindi, data una parola $w \in \mathcal{L}$, possiamo ottenere un'altra parola di lunghezza maggiore grazie alla possibilità di aggiungere un motivo di lunghezza uno (cioè 0), due (10), o tre (110). Per esempio, data $w = 01100010 \in \mathcal{L}$, abbiamo tre possibilità di aggiungere un motivo: 0, 10, o 110, ottenendo rispettivamente le nuove parole 011000100, 0110001010, 01100010110, che appartengono ancora al linguaggio.

Utilizzando queste osservazioni il concetto può essere generalizzato a ricorrenze lineari del tipo (4.3).

Esempio 4.2 Consideriamo $a = 2$, $b = 1$, $c = 0$; questa è una ricorrenza che descrive i numeri di Pell (A000129 in [SP])²:

$$\begin{cases} f_n = 2f_{n-1} + f_{n-2} \\ f_0 = 1 \\ f_1 = 2 \end{cases}$$

In questo caso il linguaggio regolare è $\mathcal{L} = \{0 \cup 1 \cup 20\}^*$.

Data una parola $w \in \mathcal{L}$ abbiamo due possibilità di aggiungere un motivo di lunghezza uno (cioè 0, 1) ed una possibilità di aggiungerne uno di lunghezza due (20).

²Naturalmente questo è un caso particolare, perchè ci sono solo due termini nella ricorrenza, e la generazione di una parola di Pell è più simile alla generazione di una parola di Fibonacci piuttosto che a quella di una parola di Tribonacci.

Possiamo notare che, per ricorrenze a due termini, ci sono a possibilità di aggiungere un motivo di lunghezza uno (cioè $0, 1, \dots, a-1$) e b possibilità di aggiungere un motivo di lunghezza due ($a0, a1, \dots, a(b-1)$), vedremo che è possibile applicare la stessa idea a ricorrenze con molti termini [BR]. In questa situazione, con una ricorrenza a tre termini, ap è la probabilità di avere a motivi di lunghezza uno e quindi c'è il primo cambio di variabile:

$$p_1 = ap \rightarrow p = \frac{1}{a}p_1$$

e

$$1 - p_1 - \frac{b}{a^2}p_1^2 - \frac{c}{a^3}p_1^3 = 0 \Rightarrow P_1(t) = 1 - t - \frac{b}{a^2}t^2 - \frac{c}{a^3}t^3.$$

Applichiamo l'Algoritmo 1 al polinomio $P_1(t)$, e, in questo caso, abbiamo rispettivamente probabilità $p_1, \frac{b}{a^2}p_1^2, \frac{c}{a^3}p_1^3$ di avere un motivo di lunghezza uno, due, tre. Poiché la ricorrenza ha coefficienti interi non nulli a, b, c , abbiamo una sostituzione di variabile un po' più complicata:

$$\begin{aligned} p_2 &= \frac{b}{a^2}p_1^2 \frac{1}{1-p_1} = \frac{b}{a^2}p_1^2 \frac{1}{\frac{b}{a^2}p_1^2 + \frac{c}{a^3}p_1^3} = \frac{1}{1 + \frac{c}{ab}p_1} \\ \Rightarrow p_1 &= \frac{ab}{c} \frac{1-p_2}{p_2}. \end{aligned}$$

Operiamo una sostituzione ed abbiamo la nuova equazione polinomiale con radice p_2 :

$$\begin{aligned} -b^3 + 2b^3p_2 - (abc + b^3)p_2^2 + (abc + c^3)p_2^3 &= 0 \\ \Rightarrow Q(z) = -b^3 + 2b^3z - (abc + b^3)z^2 + (abc + c^3)z^3. \end{aligned}$$

Questo algoritmo trova la lunghezza di un motivo che deve essere aggiunto, non sappiamo come viene estratto il motivo specifico, ma possiamo considerare E_i definito come un insieme di tutti i motivi leciti di lunghezza i . In particolare:

$$\begin{aligned} E_1 &= \{0, 1, \dots, a-1\} && \Rightarrow \#E_1 = a \\ E_2 &= \{a0, a1, \dots, a(b-1)\} && \Rightarrow \#E_2 = b \\ E_3 &= \{ab0, ab1, \dots, ab(c-1)\} && \Rightarrow \#E_3 = c \end{aligned}$$

Grazie a questo metodo, basta estrarre in modo casuale un intero tra 0 e $(a-1)$, $(b-1)$, o $(c-1)$ secondo l'insieme scelto.

4.4 Conclusioni

Il problema della generalizzazione dell'algoritmo in [PR] ai linguaggi razionali del tipo "pattern star" (linguaggi che appaiono spesso nella generazione di cammini nel piano) è risolto dall'algoritmo presentato in questo

capitolo.

Stiamo inoltre studiando un'applicazione di questo algoritmo: la generazione di involuzioni di lunghezza n usando un procedimento simile per generare il numero di punti fissi in modo che la generazione risulti uniforme.

Bibliografia

- [B] E. Barucci, A. Bernini, M. Poneti, *From Fibonacci to Catalan numbers*, sottomesso.
- [BPeP] E. Barucci, J.-G. Penaud, M. Poneti, *Toss up for heads or tails of Tribonacci words and a generalization*, sottomesso.
- [BR] E. Barucci, S. Rinaldi, *Some linear recurrences and their combinatorial interpretation by regular languages*, Theoret. Comput. Sci. 255 (2000), 680-686.
- [DZ] A. Denise, P. Zimmermann, *Uniform random generation of decomposable structures using floating-point arithmetic*, Theoret. Comput. Sci. 218 (1999), 233-248.
- [FS] P. Flajolet, R. Sedgewick, *An introduction to the analysis of algorithms*, Addison Wesley (1996).
- [PR] J. G. Penaud, O. Roque, *Tirage à pile ou face de mots de Fibonacci*, Discrete Math. 256 (2002), 791-800.
- [SP] N. J. A. Sloane, S. Plouffe, *The encyclopedia of integer sequences*, Academic press (1996).
- [K] D. E. Knuth, *Seminumerical Algorithms*, The art of computer programming.
- [S] D. J. Struik (Ed.), *A source Book in Mathematics 1200-1800*, Princeton University Press, New York (1986).

Capitolo 5

Dalle permutazioni di Fibonacci a quelle di Catalano

I numeri di Fibonacci e Catalano sono sequenze molto ben conosciute, sono presenti in molti problemi combinatori perchè enumerano una gran quantità di oggetti combinatori. Per esempio, i numeri di Fibonacci sono coinvolti nel problema della pavimentazione di una striscia, nella crescita della popolazione dei conigli, nel calcolo degli antenati delle api, mentre i numeri di Catalano compaiono nell'enumerazione di molti tipi di cammini, alberi, permutazioni, poliomini.

I numeri di Fibonacci sono descritti da una famosa ricorrenza:

$$\begin{cases} f_n = f_{n-1} + f_{n-2} \\ f_0 = 1 \\ f_1 = 1 \end{cases}$$

la cui funzione generatrice è:

$$f(x) = \frac{1}{1 - x - x^2}.$$

La sequenza comincia con 1, 2, 3, 5, 8, 13, 21, 34, . . . , (sequenza M0692 in [SP]) e l' n -esimo numero della sequenza può essere approssimato da $\frac{1}{\sqrt{5}}\phi^{n+1}$ dove

$$\phi = \frac{1 + \sqrt{5}}{2}$$

è la *radice aurea* (di cui abbiamo già parlato anche nel capitolo 4), il numero “magico” studiato fino dagli antichi Greci per la sua proprietà di apparire ovunque il rapporto tra due misure è “armonioso”, non solo in architettura, ma anche in biologia e nel corpo umano.

I numeri di Catalano forse non sono così magici, ma sono stati ampiamente studiati; appaiono in numerose relazioni, anche connessi ad altre sequenze o da soli e sono descritti dalla ricorrenza

$$\begin{cases} C_n = \sum_{i=0}^{n-1} C_{n-1-i}C_i \\ C_0 = 1 \\ C_1 = 1 \end{cases}$$

L'espressione

$$C_n = \frac{1}{n+1} \binom{2n}{n},$$

che deriva dalla funzione generatrice

$$C(x) = \frac{1 + \sqrt{1 - 4x}}{2x},$$

è considerata una formula chiusa per l' n -esimo numero di Catalano. La sequenza comincia con 1, 2, 5, 14, 42, 132, \dots , (sequenza M1459 in [SP]).

La domanda che ci siamo posti è: *Esiste una connessione tra i numeri di Fibonacci e quelli di Catalano?*

Per esempio, le sequenze:

- 1, 2, 4, 7, 13, 24, \dots , numeri di Tribonacci (sequenza M1074 in [SP]);
- 1, 2, 4, 8, 16, 32, \dots , 2^{n-1} (sequenza M1129 in [SP]);
- 1, 2, 5, 12, 29, 70, \dots , numeri di Pell (sequenza M1413 in [SP]);
- 1, 2, 5, 13, 34, 89, \dots , numeri di Fibonacci di indice pari (sequenza M1439 in [SP]),

si trovano tra i numeri di Fibonacci e quelli di Catalano (noi chiamiamo la sequenza M1439 numeri di Fibonacci di indice *pari* mentre altri autori la chiamano numeri di Fibonacci di indice *dispari*, ma tutto dipende dalle condizioni iniziali che si danno per la sequenza di Fibonacci). Il nostro obiettivo è dare una interpretazione combinatoria unificante per tutte queste sequenze, e altre ancora [BBP]. A questo scopo questo utilizzeremo permutazioni cosiddette *a motivo escluso*, cioè permutazioni in cui non compaiono alcune particolari sequenze e successivamente estenderemo i risultati ottenuti a cammini ed alberi.

5.1 Permutazioni a motivo escluso

Indichiamo con S_n l'insieme delle permutazioni su $[n] = 1, 2, \dots, n$. Siano $\Pi = \pi_1\pi_2 \cdots \pi_n \in S_n$ e $\Gamma = \gamma_1\gamma_2 \cdots \gamma_k \in S_k$. Diciamo che Π non contiene una sottosequenza del tipo Γ (o il *motivo* Γ) se non esiste nessuna sequenza $j_1 < j_2 < \cdots < j_k$ tale che $\Pi_{j_i} < \Pi_{j_h}$ se e solo se $\gamma_i < \gamma_h$. Definiamo con $S_n(\Gamma)$ l'insieme delle permutazioni su $[n]$ non contenenti una sottosequenza del tipo Γ .

Per esempio $7465312 \in S_7(123)$ mentre $7154326 \notin S_7(123)$ perché la sottosequenza 146 è del tipo 123.

Se $\Gamma_1, \Gamma_2, \dots, \Gamma_j$ sono permutazioni denotiamo con

$$S_n(\Gamma_1, \Gamma_2, \dots, \Gamma_j) = S_n(\Gamma_1) \cap S_n(\Gamma_2) \cap \cdots \cap S_n(\Gamma_j)$$

l'insieme delle permutazioni su $[n]$ che non contengono nessuna delle sequenze $\Gamma_1, \Gamma_2, \dots, \Gamma_j$. Per esempio $6745231 \in S_7(123, 132, 213)$ mentre $6475231 \notin S_7(123, 132, 213)$ perché la sequenza 475 è del tipo 132.

Permutazioni che evitano delle sottosequenze proibite sono state ampiamente studiate da molti autori [BK, BDPP1, BDPP2, Chen, Chung, EM, Gi, Gu, Kratt, Krem, SS, S, W1, W2, W3].

Un metodo naturale ma molto efficiente per enumerare classi di permutazioni è stato proposto da Chung et al. [Chung], Rogers [R], e, più tardi, da West [W1]. Tale metodo consiste nel generare permutazioni di S_n a partire da permutazioni di S_{n-1} inserendo n in tutte le posizioni tali che non compaia mai una sottosequenza proibita (indichiamo queste posizioni con un ' \diamond '). Queste posizioni sono conosciute con il nome di *siti attivi*, mentre un *sito* è una qualsiasi posizione tra due elementi consecutivi in una permutazione, o davanti al primo elemento o dopo l'ultimo.

Se una permutazione di $S_{n-1}(\Gamma_1, \dots, \Gamma_j)$ contiene k siti attivi, genera k permutazioni di $S_n(\Gamma_1, \dots, \Gamma_j)$.

Per mostrare come possiamo enumerare classi di permutazioni con questo metodo, consideriamo la classe $S_n(123)$. Sia $\Pi = \pi_1\pi_2 \cdots \pi_n$ una permutazione di $S_n(123)$ tale che $\pi_1 > \pi_2 > \cdots > \pi_{k-1} < \pi_k$. Allora i primi k siti sono attivi, perché l'inserzione di $n+1$ in una di queste posizioni non crea una sottosequenza del tipo 123. Al contrario, i siti a destra di π_k non sono attivi, perché l'inserzione di $n+1$ produce la sottosequenza $\pi_{k-1}\pi_k(n+1)$ che è del tipo 123. Quindi dalla permutazione

$$\diamond\pi_1 \diamond \pi_2 \diamond \cdots \diamond \pi_{k-1} \diamond \pi_k \pi_{k+1} \cdots \pi_n$$

otteniamo le seguenti:

$$\diamond(n+1) \diamond \pi_1 \diamond \pi_2 \diamond \cdots \diamond \pi_{k-1} \diamond \pi_k \pi_{k+1} \cdots \pi_n$$

$$\begin{aligned}
 & \diamond \pi_1 \diamond (n+1) \pi_2 \cdots \pi_n \\
 & \diamond \pi_1 \diamond \pi_2 \diamond (n+1) \pi_3 \cdots \pi_n \\
 & \vdots \\
 & \diamond \pi_1 \diamond \pi_2 \diamond \cdots \diamond \pi_{k-1} \diamond (n+1) \pi_k \cdots \pi_n
 \end{aligned}$$

che hanno rispettivamente $(k+1), 2, 3, \dots, k$ siti attivi.

Sottolineiamo che da una permutazione con k siti attivi otteniamo sempre k permutazioni aventi $(k+1), 2, 3, \dots, k$ siti attivi, indipendentemente dalla lunghezza della permutazione. Possiamo “condensare” questa proprietà in una *regola di successione*:

$$\left\{ \begin{array}{l} (2) \\ (k) \rightsquigarrow (2)(3) \cdots (k)(k+1) \end{array} \right. \quad (5.1)$$

dove (2) è il numero di siti attivi della permutazione minimale che ha lunghezza 1.

La costruzione ricorsiva della permutazione di $S_n(123)$ può anche essere rappresentata da un *albero di generazione*, dove ogni nodo è una permutazione, le permutazioni ottenute da Π appaiono come figli di Π e la radice è la permutazione minimale di lunghezza 1. Quindi, sull’ n -esimo livello abbiamo tutte le permutazioni di lunghezza n (se assumiamo che il livello della radice sia 1). La regola di successione 6.1 mette in relazione i gradi di ogni nodo dell’albero con quelli dei suoi figli. Da una regola di successione possiamo ottenere una equazione funzionale per la funzione generatrice $f(t) = \sum_{n \geq 0} a_n t^n$ dove a_n è il numero di oggetti al livello n . In questo caso, visto che la 6.1 è anche la regola di riscrittura dei numeri di Catalanò, otteniamo

$$|S_n(123)| = C_n = \frac{1}{n+1} \binom{2n}{n}.$$

Quindi, essendo $S_n(123)$ il nostro punto di arrivo, per avere un punto di partenza, consideriamo $S_n(123, 132, 213)$. Nelle permutazioni di questa classe solo i primi due siti possono essere attivi, perché l’inserzione di $n+1$ in un’altra posizione produrrebbe una sottosequenza $\pi_1 \pi_2 (n+1)$ che è del tipo 123 oppure 213. Se $\pi_1 < \pi_2$ allora solo il primo sito è attivo perché l’inserzione di $n+1$ nel secondo sito produrrebbe la sottosequenza $\pi_1 \pi_2 (n+1)$ che è del tipo 132.

Sia $\Pi = \pi_1 \pi_2 \cdots \pi_n$ una permutazione di $S_n(123, 132, 213)$; se $\pi_1 < \pi_2$, da $\diamond \pi_1 \pi_2 \cdots \pi_n$ otteniamo $\diamond(n+1) \diamond \pi_1 \pi_2 \cdots \pi_n$ che ha due siti attivi; se $\pi_1 > \pi_2$, da $\diamond \pi_1 \diamond \pi_2 \cdots \pi_n$ otteniamo $\diamond(n+1) \diamond \pi_1 \pi_2 \cdots \pi_n$ e $\diamond \pi_1(n+1) \pi_2 \cdots \pi_n$ che hanno due siti attivi e un sito attivo rispettivamente.

Così la regola di successione è

$$\left\{ \begin{array}{l} (2) \\ (1) \rightsquigarrow (2) \\ (2) \rightsquigarrow (1)(2) \end{array} \right. \quad (5.2)$$

Le regole di successione 6.1 e 5.2 sono *finite* perché vi compare solo un numero limitato di etichette. Questa regola di successione corrisponde ai numeri di Fibonacci, quindi

$$|S_n(123, 132, 213)| = f_n.$$

Consideriamo due sottosequenze proibite, cioè la classe $S_n(123, 213)$. Ogni permutazione di questa classe ha esattamente due siti attivi (i primi due siti attivi), di conseguenza la regola di successione corrispondente è

$$\left\{ \begin{array}{l} (2) \\ (2) \rightsquigarrow (2)(2) \end{array} \right. \quad (5.3)$$

e

$$|S_n(123, 213)| = 2^{n-1}.$$

Se consideriamo le sottosequenze 123 e 132, troviamo ancora

$$|S_n(123, 132)| = 2^{n-1}$$

ma con una regola di successione diversa, cioè:

$$\left\{ \begin{array}{l} (2) \\ (k) \rightsquigarrow (1)^{k-1}(k+1) \end{array} \right. \quad (5.4)$$

In effetti se $\Pi = \diamond \pi_1 \diamond \pi_2 \diamond \cdots \diamond \pi_k \pi_{k+1} \cdots \pi_n \in S_n(123, 132)$, quando inseriamo $n+1$ nel primo sito, tutti i siti attivi sono ancora attivi e lo è anche il nuovo sito a sinistra di $n+1$. Quindi $\diamond(n+1) \diamond \pi_1 \diamond \pi_2 \diamond \cdots \diamond \pi_k \pi_{k+1} \cdots \pi_n$ ha $k+1$ siti attivi. Quando inseriamo $n+1$ in qualsiasi altro sito, solo il primo sito è attivo perché l'inserzione di $n+2$ in un'altra posizione produrrebbe la sottosequenza $\pi_1(n+2)(n+1)$ che è del tipo 132, o la sottosequenza $\pi_1(n+1)(n+2)$ che è del tipo 123.

Ricordiamo che le “permutazioni di Fibonacci” evitano i tre motivi 123, 132 e 213, mentre le permutazioni enumerate da 2^{n-1} evitano due di questi motivi (123 e 132 oppure 123 e 213) e le “permutazioni di Catalano” evitano solo un motivo. Inoltre la sequenza 2^{n-1} si trova tra Fibonacci e Catalano, cioè $f_n \leq 2^{n-1} \leq C_n$.

5.2 Da Fibonacci a 2^{n-1}

Noi analizzeremo sequenze che si trovano tra Fibonacci e 2^{n-1} . Per esempio le sequenze corrispondenti alle ricorrenze che generalizzano la ricorrenza di Fibonacci, con adeguate condizioni iniziali, stanno tra Fibonacci e 2^{n-1} . Queste sequenze si dicono numeri di “Tribonacci”, “Tetranacci”, \dots , “k-nacci”; queste sequenze e le ricorrenze che le definiscono sono illustrate nella tabella seguente.

<i>Fibonacci</i>	1, 2, 3, 5, 8, 13, \dots	$f_n = f_{n-1} + f_{n-2}$
<i>Tribonacci</i>	1, 2, 4, 7, 13, 24, \dots	$t_n = t_{n-1} + t_{n-2} + t_{n-3}$
<i>Tetranacci</i>	1, 2, 4, 8, 15, 29, \dots	$q_n = q_{n-1} + q_{n-2} + q_{n-3} + q_{n-4}$
\vdots	\vdots	\vdots
<i>k - nacci</i>	1, 2, 4, 8, 16, 32, \dots	$v_n = \sum_{i=1}^k v_{n-i}$
\vdots	\vdots	\vdots
2^{n-1}	1, 2, 4, 8, 16, 32, \dots	$d_n = 2d_{n-1} = \sum_{i=1}^n v_{n-i}$

Per andare da $S_n(123, 132, 213)$ a $S_n(123, 213)$, il motivo 132 deve scomparire; per fare questo con “continuità”, “generalizziamo” questo motivo aumentando la sua lunghezza, cioè consideriamo i motivi 1432, 15432, \dots , $1(k+1)k \cdots 2, \dots$

Come primo passo esaminiamo $S_n(123, 213, 1432)$. Utilizzando la tecnica classica dei siti attivi, otteniamo la seguente regola di successione:

$$\left\{ \begin{array}{l} (2_1) \\ (1) \rightsquigarrow (2_1) \\ (2_1) \rightsquigarrow (2_1)(2_2) \\ (2_2) \rightsquigarrow (2_1)(1) \end{array} \right.$$

Abbiamo due differenti etichette (2_1) e (2_2) per i due differenti sviluppi delle permutazioni con due siti attivi. Questa regola di successione corrisponde ai numeri di Tribonacci.

Quando consideriamo, in generale, $S_n(123, 213, 1(k+1) \cdots 2)$, otteniamo la regola di successione:

$$\left\{ \begin{array}{l} (2_1) \\ (1) \rightsquigarrow (2_1) \\ (2_1) \rightsquigarrow (2_1)(2_2) \\ \vdots \\ (2_j) \rightsquigarrow (2_1)(2_{j+1}) \\ \vdots \\ (2_{k-1}) \rightsquigarrow (2_1)(2_k) \\ (2_k) \rightsquigarrow (2_1)(1) \end{array} \right.$$

che corrisponde alla sequenza generata dalla ricorrenza $v_n = v_{n-1} + \dots + v_{n-k}$ con le adeguate condizioni iniziali. Quando $k \rightarrow \infty$, questa successione diventa

$$\left\{ \begin{array}{l} (2) \\ (2) \rightsquigarrow (2)(2) \end{array} \right.$$

Inoltre, quando $k \rightarrow \infty$, il motivo $1(k+1) \dots 2$ non è più rilevante ed otteniamo $S_n(123, 213)$.

Poichè anche le permutazioni di $S_n(123, 132)$ sono enumerate da 2^{n-1} , possiamo applicare la tecnica precedente per andare da Fibonacci a 2^{n-1} . In questo caso, il motivo 213 deve scomparire e dobbiamo generalizzare questo motivo considerando i motivi $3214, 43215, \dots, k(k-1) \dots 1(k+1) \dots$.

Al primo passo troviamo che anche la classe $S_n(123, 132, 3214)$ è enumerata dai numeri di Tribonacci, quindi la regola di successione è:

$$\left\{ \begin{array}{l} (2) \\ (1) \rightsquigarrow (2) \\ (2) \rightsquigarrow (1)(3) \\ (3) \rightsquigarrow (1)(1)(3) \end{array} \right.$$

Al passo k troviamo che la classe $S_n(123, 132, k(k-1) \dots 1(k+1))$ è caratterizzata dalla regola di successione:

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

ed ancora una volta, quando $k \rightarrow \infty$, il motivo $k(k-1) \dots 1(k+1)$ non è più rilevante.

Come risultato abbiamo trovato due famiglie di classi di permutazioni enumerate da sequenze tra Fibonacci e 2^{n-1} . La loro evoluzione è descritta dalle

regole di successione che generalizzano la regola di successione di Fibonacci e tende alle due regole di successione per 2^{n-1} .

5.3 Da 2^{n-1} a Catalano

Utilizzeremo la tecnica introdotta nel paragrafo precedente per trovare classi di permutazioni enumerate da sequenze che si trovano tra 2^{n-1} ed i numeri di Catalano.

Per andare da 2^{n-1} a Catalano per prima cosa consideriamo la classe $S_n(123, 213)$. Il motivo 213 deve scomparire, quindi lo generalizziamo considerando ancora i motivi 3214, 43215, \dots , $k(k-1) \cdots 1(k+1) \cdots$. Al primo passo abbiamo $S_n(123, 3214)$ e la regola di successione:

$$\left\{ \begin{array}{l} (2) \\ (2) \rightsquigarrow (2)(3) \\ (3) \rightsquigarrow (2)(3)(3) \end{array} \right.$$

Quello che si ottiene è la ben conosciuta regola di successione dei numeri di Fibonacci di indice pari (1, 2, 5, 13, 34, \dots). Al generico passo k abbiamo $S_n(123, k(k-1) \cdots 1(k+1))$ e la regola di successione:

$$\left\{ \begin{array}{l} (2) \\ (2) \rightsquigarrow (2)(3) \\ (h) \rightsquigarrow (2)(3) \cdots (h-1)(h)(h+1) \quad h < k \\ (k) \rightsquigarrow (2)(3) \cdots (k-1)(k)(k) \end{array} \right.$$

che, quando $k \rightarrow \infty$, diventa la regola di successione di Catalano (6.1). Inoltre, il secondo motivo non è più rilevante ed otteniamo $S_n(123)$.

In modo analogo possiamo partire da $S_n(123, 132)$ e generalizzare il motivo 132 come 2143, 32154, \dots . Al primo passo otteniamo $S_n(123, 2143)$ e la regola di successione:

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

che è la regola di successione infinita dei numeri di Fibonacci di indice pari. Al passo k troviamo la regola di successione:

$$\left\{ \begin{array}{l} (2) \\ (h) \rightsquigarrow (2) \cdots (h)(h+1) \quad h < k \\ (h) \rightsquigarrow (2) \cdots (k-2)(k-1)_{h-k+2}(h+1) \quad h \geq k \end{array} \right.$$

ed il limite di questa regola di successione è ancora la regola di successione di Catalano.

5.4 Da Fibonacci a Catalano in modo diretto

Per andare da Fibonacci a Catalano consideriamo ancora la classe $S_n(123, 132, 213)$ per Fibonacci e $S_n(123)$ per Catalano e generalizziamo i due motivi 132 e 213 allo stesso tempo. Al primo passo troviamo la regola di successione:

$$\left\{ \begin{array}{l} (2) \\ (2) \rightsquigarrow (2)(3) \\ (3) \rightsquigarrow (2)(2)(3) \end{array} \right.$$

per la classe $S_n(123, 2143, 3214)$. Questa classe è enumerata dai numeri di Pell che sono descritti dalla ricorrenza:

$$\left\{ \begin{array}{l} p_n = 2p_{n-1} + p_{n-2} \\ p_0 = 0 \\ p_1 = 1 \end{array} \right.$$

Al passo k abbiamo la classe $S_n(123, (k-1) \cdots 1(k+1)k, k(k-1) \cdots 1(k+1))$ e la regola di successione corrispondente è

$$\left\{ \begin{array}{l} (2) \\ (h) \rightsquigarrow (2)(3) \cdots (h-1)(h)(h+1) \quad h < k \\ (k) \rightsquigarrow (2)(3) \cdots (k-1)(k-1)(k) \end{array} \right.$$

il cui limite quando $k \rightarrow \infty$ è ancora una volta la regola di successione di Catalano.

Infine possiamo anche andare da Pell ai numeri di Fibonacci di indice pari. Applichiamo la solita tecnica cominciando con la classe $S_n(123, 2143, 3214)$ enumerata dai numeri di Pell e generalizziamo il motivo 2143 o 3214. Otteniamo le classi $S_n(123, 3214, 21(k+1) \cdots 3)$ e $S_n(123, 2143, (k-1) \cdots 1(k+1)(k))$ e le corrispondenti regole di successione:

$$\left\{ \begin{array}{l} (2) \\ (2) \rightsquigarrow (2)(3_1) \\ (3_1) \rightsquigarrow (2)(3_1)(3_2) \\ \vdots \\ (3_{k-3}) \rightsquigarrow (2)(3_1)(3_{k-2}) \\ (3_{k-2}) \rightsquigarrow (2)(2)(3_1) \end{array} \right.$$

e

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

che, quando $k \rightarrow \infty$, diventano le regole di successione dei numeri di Fibonacci di indice pari. Ricordiamo che le due classi sono enumerate dal numero di sequenze generate dalla ricorrenza

$$y_n = 2y_{n-1} + y_{n-2} + \cdots + y_{n-k+1}$$

e, quando $k \rightarrow \infty$, troviamo una ricorrenza per numeri di Fibonacci di indice pari:

$$f_{2n} = 2f_{2n-2} + f_{2n-4} + \cdots + f_2 + f_0.$$

Possiamo riunire i risultati ottenuti in Figura 7.5 dove una linea continua rappresenta un passo diretto per andare da una famiglia all'altra, ed una linea tratteggiata unisce famiglie di permutazioni per cui possiamo andare dall'una all'altra tramite passaggi intermedi, cioè operando la generalizzazione di uno o due motivi.

5.5 Conclusioni

Partendo da due classi di permutazioni che evitano sottosequenze proibite, enumerate dai numeri di Fibonacci e Catalano, ed utilizzando una tecnica che consiste nel generalizzare le sottosequenze proibite abbiamo trovato molte famiglie di classi di permutazioni che sono enumerate da sequenze poste tra i numeri di Fibonacci e quelli di Catalano. In questo modo abbiamo introdotto un certo tipo di continuità tra le differenti classi di permutazioni e tra le sequenze di numeri che le enumerano. I risultati che abbiamo trovato possono facilmente essere estesi ai cammini di Dyck ed agli alberi planari attraverso il metodo ECO [BDPP3, BDPP4]. Le regole di successione finite che abbiamo introdotto, infatti, descrivono anche particolari classi di cammini, con vincoli sull'altezza o sul livello delle valli, oppure alberi con vincoli sul grado o livello dei nodi.

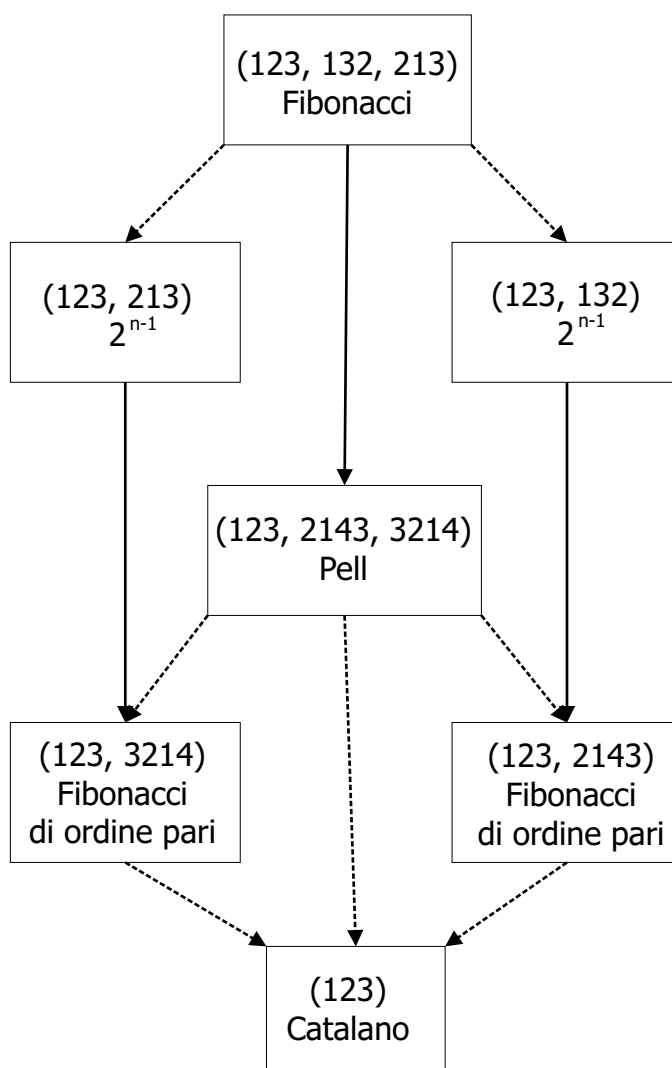


Figura 5.1: Grafo delle permutazioni.

Bibliografia

- [BBP] E. Barucci, A. Bernini, M. Poneti, *From Fibonacci to Catalan permutations*, sottomesso.
- [BK] J. Bandlow, K. Killpatrick, *An area-to-inv bijection Dyck paths and 312-avoiding permutations*, *Elect. J. Combin.* 8 (2001).
- [BDPP1] E. Barucci, A. Del Lungo, E. Pergola, R. Pinzani, *From Motzkin to Catalan permutations*, *Discrete Math.* 217 (2000), 33-49.
- [BDPP2] E. Barucci, A. Del Lungo, E. Pergola, R. Pinzani, *Some permutations with forbidden subsequences and their inversion number*, *Discrete Math.* 234 (2001), 1-15.
- [BDPP3] E. Barucci, A. Del Lungo, E. Pergola, R. Pinzani, *A methodology for plane trees enumeration*, *Discrete Math.* 180 (1998), 45-64.
- [BDPP4] E. Barucci, A. Del Lungo, E. Pergola, R. Pinzani, *ECO: A methodology for enumeration of combinatorial objects*, *J. Diff. Equ. Appl.* 5 (1999), 435-490.
- [Chen] W.Y.C. Chen, Y.P. Deng, L.L.M. Yang, *Motzkin paths and reduced decompositions for permutations with forbidden patterns*, *Elect. J. of Combin.* 9 (2003).
- [Chung] F.R.K. Chung, R.L. Graham, V.E. Hoggatt Jr., M. Kleiman, *The number of Baxter permutations*, *J. Combin. Theory Ser. A* 24 (1978), 382-394.
- [EM] E. S. Egge, T. Mansour, *Permutations which avoid 1243 and 2143, continued fractions, and Chebyshev polynomials*, <http://www.arXiv:math.CO/0208046> v1. (2002).
- [Gi] S. Gire, *Arbres, permutations à motifs exclus et cartes planaires: quelques problèmes algorithmiques et combinatoires*, Tesi di dottorato, Université Bordeaux I (1993).

-
- [Gu] O. Guibert, *Combinatoire des permutations à motifs exclus en liaison avec mots, cartes planaires et tableaux de Young*, Tesi di dottorato, Université Bordeaux I (1995).
- [Kratt] C. Krattenthaler, *Permutations with restricted patterns and Dyck paths*, Adv. Applied Math. 27 (2001), 510-530.
- [Krem] D. Kremer, *Permutations with forbidden subsequences and a generalized Schröder number*, Discrete Math. 218 (2000), 121-130.
- [R] D. G. Rogers, *Ascending sequences in permutations*, Discrete Math. 22 (1978), 35-40.
- [SS] R. Simion, W. Schmidt, *Restricted permutations*, Europ. J. Combin. 6 (1985), 383-406.
- [SP] N. J. A. Sloane, S. Plouffe, *The Encyclopedia of Integer Sequences*, Academic press (1996).
- [S] Z. E. Stankova, *Forbidden subsequences*, Discrete Math. 132 (1994), 291-316.
- [W1] J. West, *Permutations with forbidden subsequences and stack-sortable permutations*, Ph.D. Thesis, M.I.T. Cambridge, MA (1990).
- [W2] J. West, *Generating trees, the Catalan and Schröder numbers*, Discrete Math. 146 (1995), 247-262.
- [W3] J. West, *Generating trees, forbidden subsequences*, Discrete Math. 157 (1996), 363-374.

Capitolo 6

Una struttura di reticolo distributiva.

Ogni articolo che ha a che fare con i numeri di Catalano contiene una frase del tipo: “In [S2] Stanley dà 66 interpretazioni combinatorie differenti dei numeri di Catalano”. Invece, l’esercizio 6.19 di [S2] è forse la migliore fonte di informazioni sulla famiglia di Catalano, almeno da un punto di vista puramente enumerativo. Un passo successivo dovrebbe essere quello di definire alcune interessanti strutture d’ordine sugli oggetti della famiglia di Catalano e di cercare di confrontarle. Quello che cerchiamo di fare è affrontare la prima istanza del problema.

Cominciamo col considerare le partizioni “noncrossing”. Queste possono essere dotate di un ordine di raffinamento, per ottenere i reticoli, studiati in prima istanza da Kreweras [Kre], che sono stati utilizzati in molti e differenti contesti. Tali reticoli possiedono interessanti proprietà ma non risultano essere distributivi. Esiste la possibilità di definire una interessante struttura reticolare distributiva su partizioni noncrossing? Pensiamo che la risposta possa essere affermativa se troviamo un ordine sulle partizioni noncrossing che sia isomorfo almeno a due reticoli distributivi con una interpretazione combinatoria.

Per prima cosa consideriamo i cammini di Dyck e su di essi definiamo la seguente relazione d’ordine: dati due cammini di Dyck P, Q della stessa lunghezza, diciamo che $P \leq Q$ se P sta interamente sotto Q (al limite può coincidere con Q in qualche punto). E’ possibile mostrare [FP2] che l’insieme dei cammini di Dyck di lunghezza fissata dotati di questa relazione d’ordine è un reticolo distributivo. Questi *reticoli di Dyck* non sono molto conosciuti; sono stati studiati in [FP2] (seguendo delle idee generali di Narayana [N]), ed in [CJ] gli autori mostrano la loro importanza nello studio

di alcuni problemi in relazione con le algebre di Temperley-Lieb. La nostra idea è quella di trasferire tale relazione d'ordine sulle partizioni noncrossing secondo una famosa biiezione [S]. Abbiamo chiamato *reticoli di Bruhat delle partizioni noncrossing* i reticoli distributivi delle partizioni noncrossing così definiti; il Paragrafo 6.3 è dedicato allo studio di alcune proprietà di questi reticoli. Inoltre, i reticoli di Bruhat delle partizioni noncrossing risultano essere isomorfi ad un'altra interessante classe di reticoli. Non è difficile trovare esplicitamente una biiezione tra le partizioni noncrossing e le permutazioni che evitano il motivo 312. Più precisamente, mostriamo che tale biiezione porta un isomorfismo d'ordine tra il reticolo di Bruhat delle partizioni noncrossing di n e la classe $S_n(312)$ delle permutazioni di n che evitano il motivo 312 dotate dell'ordine di Bruhat. Come sottoprodotto abbiamo che $S_n(312)$ è un sottoreticolo distributivo del gruppo simmetrico di ordine n con l'ordine di Bruhat. Questi risultati si trovano nella sezione 6.4, dove è stato anche trovato un criterio per determinare massimo e minimo di due permutazioni in $S_n(312)$. Il solo articolo che tratta questo tipo di problemi è [P], dove l'autore determina i poset di Bruhat (a partire dai gruppi di Weyl) che sono reticoli. In ogni modo, il linguaggio ed i propositi di [P] sono totalmente diversi dai nostri. Sarebbe interessante confrontare i nostri risultati con quelli di Proctor. Comunque, ci sembra che il risultato che abbiamo ottenuto [BBFP1, BBFP2] sia il primo concernente la struttura d'ordine indotta dall'ordine di Bruhat forte sulla classe delle permutazioni che evitano un motivo. Precisiamo che risultati simili sono stati trovati per l'ordine debole: citiamo [BW], dove gli autori mostrano che $S_n(312)$ con l'ordine debole è un sottoreticolo di S_n , isomorfo al reticolo di Tamari, e [Dr], dove sono caratterizzati quei modelli τ tali che $S_n(\tau)$ è un reticolo.

La parte finale di questa introduzione è dedicata alla spiegazione di alcune notazioni che utilizziamo e alla presentazione dei fondamenti delle teorie generali a cui intendiamo riferirci.

La relazione di copertura di ogni poset che considereremo è indicata con il simbolo \prec , in modo tale che $x \prec y$ significa che “ x è coperto da y ”. L'insieme (ed il reticolo) delle partizioni di $[n] = \{1, 2, \dots, n\}$ sarà denotato da $\Pi(n)$. Se $\pi \in \Pi(n)$, utilizzeremo sempre la notazione $\pi = B_1|B_2|\dots|B_k$, dove i B_i sono i blocchi di π , gli elementi all'interno di ogni blocco sono in ordine decrescente e $\max B_i < \max B_j$, per $i < j$.

Avremo spesso a che fare con i cammini di Dyck e, a seconda del contesto, li descriveremo in modi diversi. Comunque un cammino di Dyck sarà alternativamente descritto come un particolare percorso nel reticolo nel piano discreto $\mathbf{N} \times \mathbf{N}$ (denotato con lettere maiuscole come P, Q, R, \dots) o come una funzione $f : \mathbf{N} \rightarrow \mathbf{N}$ che soddisfa certe proprietà (denotato

tramite lettere minuscole come f, g, h, \dots) o anche come una particolare parola su un alfabeto a due lettere $\{U, D\}$ (denotato con lettere greche come $\omega(U, D), \psi(U, D), \dots$).

Nella sezione 6.1 richiamiamo brevemente delle proprietà dei reticoli di partizioni utilizzando il linguaggio del metodo ECO. Referenze aggiuntive per il metodo ECO sono [B *et al.*, FP1, PPR, GPP]. Nella sezione 6.4 utilizziamo il concetto di permutazione a motivo escluso (generalizzato), di cui abbiamo già parlato, ma a proposito di altre proprietà, con altri tipi di argomentazioni e soprattutto con altri scopi, nel Capitolo 5. Possiamo trovare una quantità enorme di articoli che si occupa dell'esclusione di motivi, vedi per esempio [SS, F, Kra]. In [BS] gli autori hanno introdotto i *motivi generalizzati* per studiare le statistiche Mahoniane sulle permutazioni. Un motivo generalizzato è una permutazione $\sigma \in S_k$ dotata di trattini fra coppie dei suoi elementi (per esempio 1-32 e 23-1 sono motivi generalizzati di 3) ed una permutazione π contiene un motivo generalizzato quando elementi adiacenti nel motivo generalizzato corrispondono ad elementi adiacenti in π . Classi di permutazioni che evitano un motivo generalizzato sono state ampiamente studiate negli ultimi anni (vedi [BS, BFP, C, CM], per citarne qualcuno).

6.1 Nozioni preliminari sull'insieme delle partizioni

Cominciamo col richiamare alcune proprietà dell'insieme delle partizioni riguardanti l'ordine classico di raffinamento. Questa sezione quindi deve essere intesa come un'indagine selezionata di alcune classiche proprietà dei reticoli delle partizioni. Per non ripetere parola per parola i contenuti di alcuni testi classici, abbiamo scelto di utilizzare un linguaggio alternativo. Quello che abbiamo ottenuto è una presentazione di alcuni risultati classici sull'insieme delle partizioni inseriti nella struttura del metodo ECO e delle regole di successione.

Date $\pi, \gamma \in \Pi(n)$, definiamo $\pi \leq \gamma$ quando ogni blocco di π è contenuto in un blocco di γ . Le numerose proprietà di questo ordine classico si possono trovare in molte fonti, tra cui [S1, A]. Qui diciamo solo che $\Pi(n)$ dotato di questo *ordine di raffinamento* è un reticolo che non è né distributivo né modulare. Tuttavia, possiede una funzione rango: il rango di $\pi = B_1|B_2| \dots |B_k$ è $(n - k)$. I numeri di Whitney sui reticoli delle partizioni sono i ben conosciuti numeri di Stirling di seconda specie. Il diagramma di Hasse di $\Pi(4)$ è mostrato in Figura 6.1.

Una costruzione ricorsiva dell'insieme delle partizioni lavora come segue:

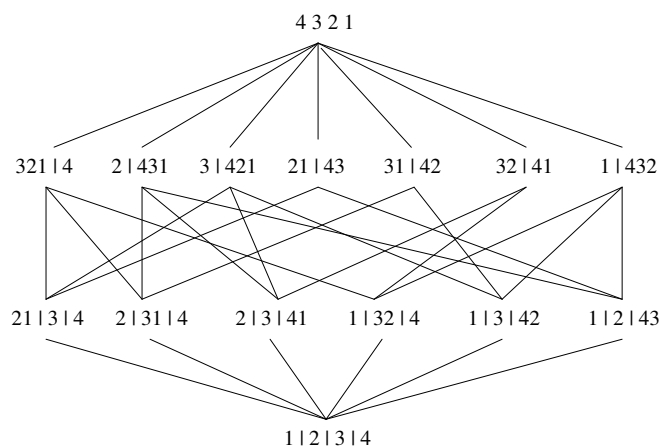


Figura 6.1: Diagramma di Hasse di $\Pi(4)$.

dato $\pi = B_1|B_2|\dots|B_k \in \Pi(n)$, si aggiunge $(n+1)$ ad ogni blocco di π o si inserisce il nuovo blocco $B_{k+1} = \{n+1\}$. In questo modo otteniamo $k+1$ nuove partizioni di $[n+1]$, dette $\pi_i = B_1|\dots|B_{i-1}|B_{i+1}|\dots|B_k|(n+1)B_i$, $i = 1, \dots, k$ e $\pi_{k+1} = B_1|\dots|B_k|(n+1)$. Osserviamo che questa costruzione classica può essere riletta alla luce del metodo ECO e codificata tramite la seguente regola di successione:

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

Questa regola di successione dovrebbe essere letta come segue. Si etichetta ogni partizione con il numero di relativi blocchi, più 1. Quindi una partizione etichettata (k) produce k nuove partizioni; di queste, $k-1$ sono ancora etichettate (k) , mentre l'altra è etichettata $(k+1)$.

Diciamo che $\pi, \gamma \in \Pi(n)$ sono *ECO-equivalenti* se sono prodotte dalla stessa partizione di $\Pi(n-1)$ (cioè, se hanno lo stesso padre nell'albero di generazione della suddetta regola di successione). Risulta quindi evidente che ogni classe di equivalenza con l'ordine indotto è un "flat" [DP], cioè un poset con minimo in cui tutti gli elementi rimanenti sono massimali. Tuttavia, grazie alla costruzione ECO precedente, otteniamo una "partizione flat" di ogni $\Pi(n)$ (Figura 6.1).

La seguente proposizione, la cui dimostrazione è diretta, contiene alcune proprietà di questa partizione flat.

Proposizione 6.1 *Il reticolo $\Pi(n)$ è partizionato in \mathcal{B}_{n-1} flat¹; più preci-*

¹ \mathcal{B}_n è l' n -th numero di Bell.

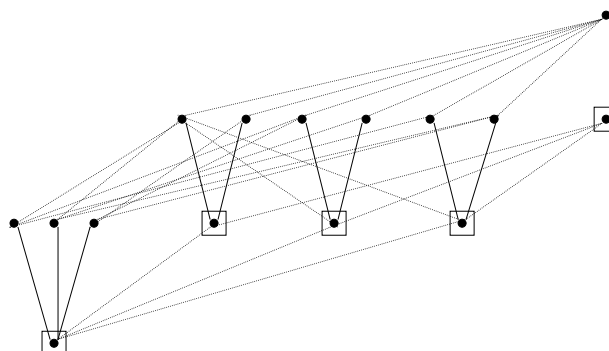


Figura 6.2: La partizione flat di $\Pi(4)$.

samente, per $1 \leq k \leq n - 1$, ci sono $\mathcal{S}_{n-1,k}$ flat² di cardinalità $n + 1 - k$, i cui minimi hanno rango $k - 1$. L'ordine indotto sull'insieme di questi minimi è isomorfo a $\Pi(n - 1)$. Se calcoliamo i numeri di Whitney di $\Pi(n)$ utilizzando questa partizione flat otteniamo la nota ricorsione per gli $\mathcal{S}_{n,k}$, $\mathcal{S}_{n+1,k} = \mathcal{S}_{n,k-1} + k\mathcal{S}_{n,k}$.

6.2 Partizioni noncrossing e cammini di Dyck

Una partizione di $1, 2, \dots, n$ è *noncrossing* quando, dati quattro elementi, $1 \leq a < b < c < d \leq n$, tali che a, c sono nello stesso blocco e b, d sono nello stesso blocco, allora i due blocchi coincidono. L'insieme di tutte le partizioni noncrossing di un n -insieme saranno denotate con $NC(n)$ (Figura 6.3). In [S] si possono trovare abbondanti applicazioni di queste nozioni.

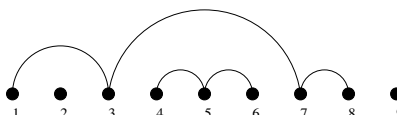


Figura 6.3: La partizione noncrossing $2|654|8731|9 \in NC(9)$.

L'ordine di raffinamento può essere ristretto alle partizioni noncrossing:

² $\mathcal{S}_{n,k}$ è l' n, k entrata del triangolo dei numeri di Stirling di seconda specie.

quello che otteniamo è ancora un reticolo, che viene solitamente detto *reticolo della partizione noncrossing*. Fra le caratteristiche principali di questi reticoli ricordiamo che non sono distributivi e che le operazioni sui reticoli sono diverse da quelle tra i reticoli delle partizioni (il massimo tra due partizioni noncrossing non è necessariamente noncrossing).

Le partizioni noncrossing sono enumerate dai numeri di Catalano, quindi, è possibile trovare una biiezione con i cammini di Dyck. Quella che descriviamo può essere trovata anche in [De, S]. Fissiamo un cammino di Dyck ed etichettiamo i suoi passi crescenti enumerandoli da sinistra verso destra (in modo che il k -esimo passo crescente sia etichettato k). Successivamente assegniamo ad ogni passo decrescente la stessa etichetta del passo crescente corrispondente. Ora consideriamo la partizione i cui blocchi sono costituiti dalle etichette di ogni sequenza di passi discendenti. Si vede facilmente che questa partizione è noncrossing. In Figura 6.4 è illustrato un esempio di questa biiezione; le etichette in grassetto sui passi discendenti sono gli elementi della corrispondente partizione noncrossing, considerando che i passi crescenti sono semplicemente etichettati in ordine crescente.

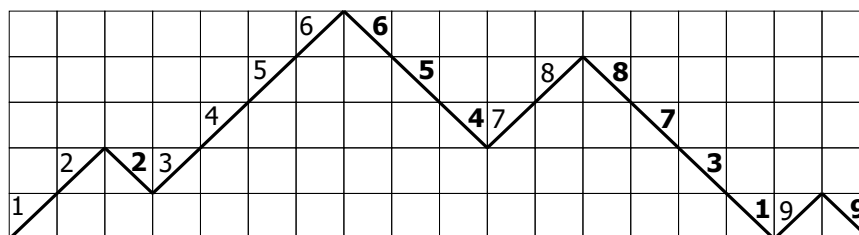


Figura 6.4: Cammino di Dyck associato a $2|654|8731|9$.

Indichiamo con \mathcal{D}_n l'insieme dei cammini di Dyck di lunghezza $2n$. Ricordiamo che su \mathcal{D}_n è possibile definire una relazione d'ordine tale che $f \leq g$ se $f(i) \leq g(i)$, per ogni $i \in \mathbf{N}$. Ciò vuol dire che $f \leq g$ quando f “sta sotto” g . L'insieme \mathcal{D}_n , dotato di tale ordine, risulta essere un reticolo distributivo che è stato studiato in dettaglio in [FP2] con il nome di *reticolo di Dyck (di ordine n)*. Precisiamo che i reticoli di Dyck sono stati considerati in [CJ], dove gli autori parlano di *inclusione geometrica* di cammini. La nostra idea è quella di trasferire la relazione d'ordine dei reticoli di Dyck lungo la biiezione precedentemente descritta. In questo modo definiamo un nuovo ordine sulle partizioni noncrossing. I reticoli distributivi così ottenuti saranno chiamati *reticoli di Bruhat per le partizioni noncrossing*. La ragione di questo nome, che per ora può risultare abbastanza oscura, verrà spiegata nella Sezione 6.4.

Nostro obiettivo è quello di fornire una descrizione soddisfacente di questi reticoli.

Nella restante parte di questa sezione proponiamo una nuova costruzione per le partizioni noncrossing tramite il trasferimento di una nota costruzione ECO dei cammini di Dyck descritta in [BDLPP] lungo la precedente biiezione. La costruzione opera come segue: un cammino di Dyck P di lunghezza $2n$ genera un insieme di cammini di Dyck di lunghezza $2n+2$ inserendo un picco in ogni punto dell'ultima sequenza di passi discendenti. Come sottoprodotto troveremo una statistica sulle partizioni noncrossing la cui distribuzione è data dai numeri "ballot".

Fissiamo una partizione $\pi = B_1 | \dots | B_{k-1} | B_k \in NC(n)$, con $B_k = a_1 \dots a_r$. Cominciando da π costruiamo $r+1$ nuove partizioni:

$$\begin{cases} \pi_1 = B_1 | \dots | B_k | (n+1) \\ \pi_i = B_1 | \dots | B_{k-1} | a_1 \dots a_{r-i+1} | (n+1) a_{r-i+2} \dots a_r & \text{per } i = 2, \dots, r \\ \pi_{r+1} = B_1 | \dots | B_{k-1} | (n+1) a_1 \dots a_r \end{cases} \quad (6.1)$$

Questa costruzione della partizione noncrossing è "isomorfa" alla sopra citata costruzione ECO per i cammini di Dyck. La proposizione 6.2 trasla sulle partizioni noncrossing alcuni risultati enumerativi ottenuti grazie a questa costruzione.

Proposizione 6.2 *Ogni partizione di $NC(n)$ è ottenuta esattamente una volta per mezzo della suddetta costruzione dalle partizioni in $NC(n-1)$. Più nel dettaglio, se una partizione noncrossing è etichettata con la cardinalità del blocco che contiene il suo minimo, più 1 (cosicché $\pi = B_1 | \dots | B_s$ è etichettato $(|B_s| + 1)$), allora la costruzione può essere descritta dalla seguente regola di successione:*

$$\Omega : \begin{cases} (1) \\ (k) \end{cases} \rightsquigarrow (2)(3)(4) \dots (k)(k+1) \quad .$$

Di conseguenza, abbiamo che il numero di partizioni noncrossing di $[n]$ tali che il blocco contenente n abbia cardinalità $k-1$ è uguale al numero di cammini di Dyck di lunghezza $2n$ in cui l'ultima discesa ha $k-1$ passi discendenti, che è il numero ballot $\frac{k+1}{n} \binom{2n-k-2}{n-k-1}$ (vedi [BDLPP] per la combinatoria enumerativa della regola Ω).

Osservazione. Precisiamo che la precedente costruzione delle partizioni noncrossing è simile ad una data nella seconda sezione di [A1], dove l'autore illustra le basi di un metodo di enumerazione tramite le tavole ballot. Sarebbe interessante mettere in relazione questo approccio con la metodologia ECO.

6.3 Il reticolo di Bruhat per la partizione noncrossing

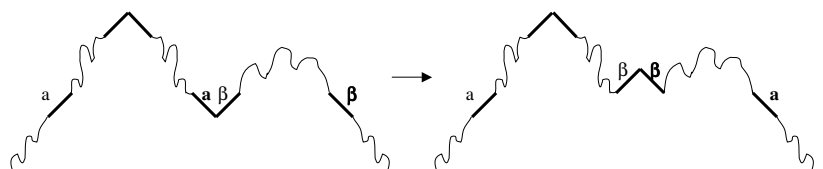
D'ora in avanti assumiamo che le partizioni noncrossing siano dotate dell'ordine di Bruhat definito sopra.

Date due partizioni noncrossing π, γ cerchiamo delle condizioni per riconoscere se $\pi \prec \gamma$. Se $\pi \prec \gamma$ diciamo anche che γ "ricopre" π .

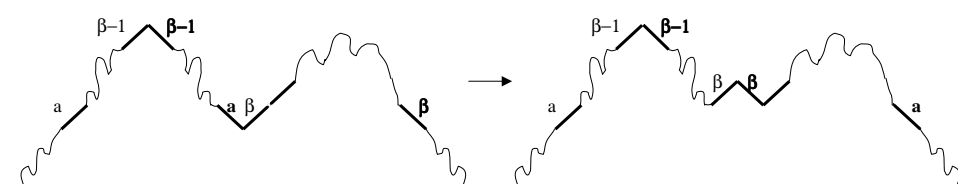
Teorema 6.1 (*Caratterizzazione della copertura*) *Date due partizioni noncrossing $\pi, \gamma \in NC(n)$, $\pi \prec \gamma$ se e solo se γ è ottenuta da π muovendo il minimo di un blocco B di π nel blocco \tilde{B} contenente l'elemento $\beta = \max B + 1$, prendiamo*

1. β dentro \tilde{B} , se $\beta = \max \tilde{B}$, o
2. un nuovo blocco $\bar{B} = \{\beta\}$, se $\beta \neq \max \tilde{B}$.

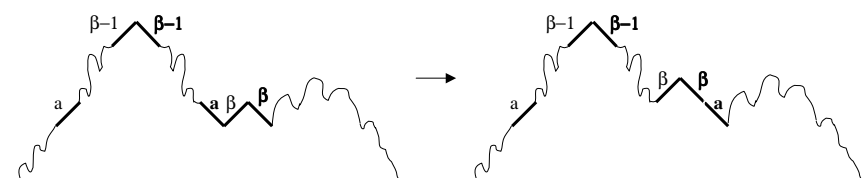
Dim. Supponiamo che P_π, P_γ siano i cammini di Dyck associati a π, γ , rispettivamente. Il fatto che $P_\pi \prec P_\gamma$ in \mathcal{D}_n vuol dire che P_γ è ottenuto da P_π sostituendo una valle con un picco. Nel contesto delle partizioni noncrossing questo equivale a muovere il minimo a di un blocco, poiché il passo discendente di una valle è l'ultimo passo di una discesa. L'elemento a viene spostato nel blocco che contiene l'elemento corrispondente al passo discendente abbinato con il passo crescente della valle. Segue direttamente dalla biiezione che tale passo discendente ha etichetta uguale a $\beta = \max B + 1$, dove B è il blocco contenente a in π . La seguente figura illustra questi fatti.



Ed ora cosa succede con l'elemento β ? I casi possibili sono due. Se il passo ascendente della valle in P_π è seguito da un altro passo ascendente, allora β non è il massimo del suo blocco in π , ed è facile verificare che in γ diventa un blocco fatto da un solo elemento, dal momento che in P_γ il passo corrispondente è preceduto e seguito da un passo ascendente (vedi figura seguente).



Se il passo ascendente della valle è seguito da uno discendente, allora β è il massimo del suo blocco in π , e rimane nello stesso blocco anche in γ , come illustrato nella figura seguente.



□

Esempio 6.1 Data la partizione $2|54|631 \in NC(6)$, ci sono esattamente due partizioni che la ricoprono, che sono $3|54|621$ (2 si sposta e 3 non è il massimo del suo blocco) e $2|5|6431$ (4 si sposta e 6 è il massimo del suo blocco).

E' interessante osservare che le due "istruzioni" 1. e 2. nel teorema 6.2 hanno una stretta analogia con la definizione di un *filler* data in [DS]. Ricordiamo che un punto $i \in \{2, 3, \dots, n\}$ è detto filler di $\pi \in NC(n)$ se è soddisfatta una delle seguenti condizioni:

1. $i - 1$ e i sono nello stesso blocco e i è l'elemento più grande del suo blocco;

2. i forma un blocco fatto solo da un elemento e $i - 1$ non è l'elemento più grande del suo blocco.

Inoltre, un filler è prodotto ogni volta che una valle preceduta da un passo ascendente è cambiata in un picco nel cammino di Dyck associato. Così un filler in una partizione noncrossing corrisponde ad un passo discendente preceduto da una lunga salita nel cammino di Dyck associato (dove per lunga salita si intende una sequenza di due o più passi ascendenti consecutivi). Comunque, il numero di partizioni noncrossing di un n -insieme avente k filler coincide con il numero $T_{n,k}$ di cammini di Dyck di lunghezza $2n$ aventi k lunghe salite, vale a dire [S1]:

$$T_{n,k} = \frac{1}{n+1} \binom{n+1}{k} \sum_{j=0}^{n-2k} \binom{k+j-1}{k-1} \binom{n+1-k}{n-2k-j}.$$

Un ulteriore risultato che abbiamo ottenuto è la definizione di un criterio per confrontare due partizioni noncrossing assegnate. Definiamo *vettore max* di una partizione noncrossing $\pi \in NC(n)$ il vettore $\max(\pi) = (\mu_1, \dots, \mu_n)$ tale che μ_i è il massimo dei primi i elementi di π . Quindi, per esempio, se $\pi = 2|31|54$, allora $\max(\pi) = (2, 3, 3, 5, 5)$. E' facile verificare che il vettore \max determina in modo univoco la partizione noncrossing associata. Questa osservazione sarà molto importante nel seguito.

Teorema 6.2 (*Caratterizzazione dell'ordine di Bruhat di NC*) Siano π, γ due partizioni in $NC(n)$. Allora $\pi \leq \gamma$ se e solo se $\max(\pi) \leq \max(\gamma)$ nell'ordine coordinata per coordinata, cioè se $\mu_i \leq \nu_i$ per ogni $i = 1, \dots, n$.

Dim. Siano $\omega_1 = \omega_1(U, D)$ e $\omega_2 = \omega_2(U, D)$ due cammini di Dyck corrispondenti a π e γ , rispettivamente. Allora è chiaro che $\omega_1 \leq \omega_2$ se e solo se ogni prefisso di ω_1 contiene almeno tanti D quanti il prefisso corrispondente di ω_2 . Questo può essere tradotto sulle partizioni utilizzando il vettore \max . In effetti, se $\max(\pi) = (\mu_1, \dots, \mu_n)$ e $\max(\gamma) = (\nu_1, \dots, \nu_n)$, consideriamo i due vettori $(\bar{\mu}_1, \dots, \bar{\mu}_n)$ e $(\bar{\nu}_1, \dots, \bar{\nu}_n)$, dove $\bar{\mu}_i = \mu_i + i$ e $\bar{\nu}_i = \nu_i + i$. Allora, non è difficile osservare che $\bar{\mu}_i$ e $\bar{\nu}_i$ codificano la posizione dell' i -esimo D nel corrispondente cammino di Dyck. Dalle ipotesi, abbiamo che l' i -esimo D di ω_1 occorre prima dell' i -esimo D di ω_2 , e quindi $\bar{\mu}_i \leq \bar{\nu}_i$. Poichè questo vale per ogni $i \leq n$, la tesi è dimostrata.

□

Esempio 6.2 Siano $\pi = 2|43|51|6$, $\gamma = 43|52|61 \in NC(6)$. Troviamo facilmente che $\max(\pi) = (2, 4, 4, 5, 5, 6)$ e $\max(\gamma) = (4, 4, 5, 5, 6, 6)$. E' immediato vedere che $\max(\pi) \leq \max(\gamma)$, da cui $\pi \leq \gamma$.

Osservazione. Se $\pi \prec \gamma$, allora $\max(\pi)$ e $\max(\gamma)$ differiscono esattamente di una posizione.

A questo punto è bene osservare che i reticoli delle partizioni noncrossing possono essere descritti in modo alternativo utilizzando le *funzioni "parking" crescenti*. Ricordiamo che una funzione parking crescente è una sequenza $a_1 \leq a_2 \leq \dots \leq a_n$ tale che $a_i \leq i$, per ogni $i \leq n$. Il poset delle funzioni parking crescenti di $[n]$ con l'ordine coordinata per coordinata è chiaramente un reticolo distributivo. Non è difficile fare vedere che un reticolo è isomorfo al duale del reticolo di Bruhat delle partizioni noncrossing su n elementi. Inoltre, la corrispondenza che rappresenta $(a_1, a_2, \dots, a_{n-1}, a_n)$ nella sequenza $(n+1-a_n, n+1-a_{n-1}, \dots, n+1-a_2, n+1-a_1)$ è una biezione che inverte l'ordine tra le funzioni parking crescenti di $[n]$ e l'insieme dei vettori max delle partizioni noncrossing di $[n]$.

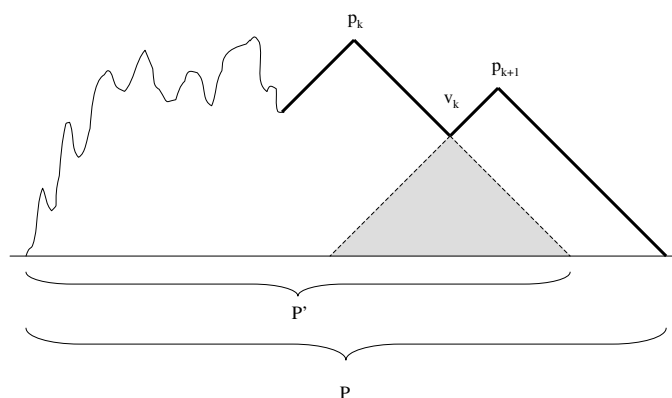
E' noto [FP2] che i reticoli di Dyck possiedono una funzione rango (semplicemente perchè sono reticoli distributivi) che è data essenzialmente dall'*area* limitata da un cammino di Dyck e l'asse delle x . Per area di un cammino di Dyck si intende la somma di tutte le altezze dei punti di partenza dei passi del cammino. Più nel dettaglio, se $A(P)$ è l'area di un cammino di Dyck P di lunghezza n , allora il rango di P nel suo reticolo di Dyck è dato da $r(P) = \frac{A(P)-n}{2}$. Il prossimo obiettivo è tradurre il parametro "area sotto i cammini di Dyck" in un parametro sulle partizioni noncrossing, per definire un rango sui reticoli di Bruhat delle partizioni noncrossing.

Il primo risultato è una formula per l'area dei cammini di Dyck in funzione dei loro picchi e delle loro valli. Poichè non abbiamo trovato tale formula in letteratura, ne proponiamo anche una dimostrazione.

Lemma 6.1 *Sia P un cammino di Dyck. Definiamo con p_i e v_j le altezze dell' i -esimo picco e della j -esima valle di P , rispettivamente. Per convenzione consideriamo anche l'ultimo punto del cammino come ultima valle di altezza zero :*

$$A(P) = \sum_{i=1}^n (p_i^2 - v_i^2). \quad (6.2)$$

Dim. Procediamo per induzione sul numero dei picchi. Se un cammino di Dyck ha solo un picco, allora è il massimo del suo reticolo di Dyck e la formula segue immediatamente. Ora supponiamo che P abbia $k+1$ picchi. Consideriamo il cammino P' ottenuto da P togliendo l'ultimo picco, cioè coincidente con P fino al k -esimo picco e poi finisce con una sequenza di passi discendenti (vedi Figura 6.5).


 Figura 6.5: Come si ottiene P' da P .

E' facile vedere che

$$A(P) = A(P') + p_{k+1}^2 - v_k^2,$$

da cui, per ipotesi induttiva:

$$A(P) = \sum_i (p_i^2 - v_i^2)$$

□

Ora siamo pronti per trovare una formula per esprimere il rango di una partizione nel reticolo di Bruhat delle partizioni noncrossing.

Teorema 6.3 $NC(n)$ è un reticolo distributivo, di conseguenza ha rango. Più precisamente, se $\pi = B_1 | \dots | B_k \in NC(n)$, allora il suo rango è dato da:

$$r_n(\pi) = \frac{A(\pi) - n}{2}, \quad (6.3)$$

dove

$$A(\pi) = \sum_{i=1}^k \left(|B_i| \left(2b_i - 2 \sum_{j=1}^{i-1} |B_j| - |B_i| \right) \right) \quad (6.4)$$

(qui $b_i = \max B_i$).

Dim. Sia P un cammino di Dyck e $\pi = B_1 | \dots | B_k$ la partizione noncrossing ad esso associata. Chiaramente il numero di picchi di P coincide con il numero di blocchi di π . Se (x, p_i) è l' i -esimo picco di P , allora p_i è la differenza tra il numero u di passi ascendenti ed il numero d di passi discendenti da (x, p_i) . Nel linguaggio delle partizioni noncrossing, u è il massimo b_i dell' i -esimo blocco di π , e d è semplicemente la somma delle cardinalità dei primi $i - 1$ blocchi di π . Quindi abbiamo:

$$p_i = b_i - \sum_{j=1}^{i-1} |B_j|.$$

In modo analogo, se v_i è l'altezza della i -esima valle di P , troviamo:

$$v_i = b_i - \sum_{j=1}^i |B_j|.$$

Sostituendo queste quantità nella formula 6.1, si ottiene infine:

$$\begin{aligned} A(\pi) &= \sum_{i=1}^k \left(\left(b_i - \sum_{j=1}^{i-1} |B_j| \right)^2 - \left(b_i - \sum_{j=1}^i |B_j| \right)^2 \right) \\ &= \sum_{i=1}^k \left(|B_i| \left(2b_i - 2 \sum_{j=1}^{i-1} |B_j| - |B_i| \right) \right). \end{aligned}$$

□

6.4 Collegamento con l'ordine di Bruhat forte sulle permutazioni

L'ultima formula data per il rango di una partizione noncrossing nel suo reticolo di Bruhat non è tanto semplice da capire come la funzione rango per i cammini di Dyck. Per trovare un modo migliore per esprimere questo parametro, utilizziamo il concetto di permutazione a motivo escluso (generalizzato). Ciò che otteniamo è ancora un'altra descrizione di reticoli di Bruhat per le partizioni noncrossing che fornisce alcune importanti informazioni sull'ordine di Bruhat (forte) dei gruppi simmetrici.

Proposizione 6.3 *Rimuovendo le barre nelle partizioni noncrossing si definisce una biiezione tra $NC(n)$ e l'insieme $S_n(312)$ delle permutazioni di n che evitano il motivo 312 , per ogni $n \in \mathbf{N}$.*

Dim. Per prima cosa osserviamo che, per ogni $n \in \mathbf{N}$, $S_n(312) = S_n(31-2)$, perchè sappiamo che questi due insiemi finiti sono entrambi enumerati dai numeri di Catalano ed ovviamente $S_n(312) \subseteq S_n(31-2)$. Ora, se un motivo 31-2 appare in una partizione noncrossing, allora, indicando con $b < c < a$ i tre elementi corrispondenti ad un tale motivo, a e b devono appartenere allo stesso blocco, ed il massimo d del blocco che contiene c deve essere più grande di a (dal momento che il massimo di un blocco in una partizione noncrossing è più grande di ogni altro elemento che lo precede). Quindi, i quattro elementi a, b, c, d dovrebbero costituire un incrocio, contro l'ipotesi.

□

Osservazione. Nella parte restante di questa sezione utilizzeremo molto la biiezione sopra descritta. In particolare passeremo liberamente da una partizione noncrossing alla permutazione a motivo escluso 312 ad essa associata senza dirlo esplicitamente. Comunque utilizzeremo sempre le lettere greche ($\pi, \gamma, \sigma, \dots$) per indicare sia una partizione noncrossing sia la permutazione che evita 312 ad essa associata. Infine, osserviamo che ogni massimo di un blocco di una partizione noncrossing corrisponde al massimo da sinistra verso destra della corrispondente permutazione, tale massimo è un elemento più grande di ogni altro elemento alla sua sinistra.

Da notare che la composizione della biiezione tra cammini di Dyck e partizioni noncrossing con quella tra le partizioni noncrossing e le permutazioni che evitano il motivo 312 è esattamente la biiezione considerata in [BK] ed in [F]. Compare anche in [Kra], come biiezione tra cammini di Dyck e permutazioni che evitano 132. Una semplice visualizzazione di questa biiezione è fornita in [R, EP]. Utilizzando questa descrizione, alcune delle proprietà dell'ordine di Bruhat sulle partizioni noncrossing, come il Teorema 6.2, possono essere adeguatamente riformulate.

Fra le caratteristiche della sopra citata biiezione, una molto interessante è illustrata in [BK], dove gli autori mostrano che l'area di un cammino di Dyck corrisponde al numero di inversioni delle permutazione associata. Poiché la funzione rango dell'ordine di Bruhat sulle permutazioni è precisamente il numero di inversioni, siamo portati a congetturare una relazione chiusa tra i nostri reticoli delle partizioni noncrossing ed i sottoposet indotti dall'ordine di Bruhat sulle permutazioni che escludono 312. Il prossimo teorema dimostra la veridicità di questa congettura.

Teorema 6.4 *Sia $(S_n(312); \leq)$ il poset ottenuto dal trasferimento della struttura del reticolo delle permutazioni noncrossing di Bruhat $NC(n)$ lungo la biiezione precedente. Questo è precisamente il sottoposet indotto su $S_n(312)$*

dall'ordine di Bruhat forte del gruppo simmetrico S_n . Quindi $S_n(312)$ è un sottoreticolo distributivo di S_n dotato dell'ordine di Bruhat forte.

Dim. Dobbiamo mostrare che il diagramma di Hasse del reticolo delle permutazioni noncrossing di Bruhat è isomorfo a quello di $S_n(312)$ con l'ordine di Bruhat forte indotto. Per fare questo, è sufficiente provare che gli insiemi di elementi che coprono una partizione noncrossing e la permutazione che evita 312 ad esso associata coincidono, considerando la biezione che ad una partizione noncrossing $B_1|B_2|\dots|B_k$ (dove i B_i sono blocchi di interi) associa la permutazione $B_1B_2\dots B_k$.

Siano π, γ due partizioni, supponiamo che $\pi \prec \gamma$ nel reticolo delle permutazioni noncrossing di Bruhat. Questo vuol dire che γ è ottenuta da π utilizzando una delle due regole del Teorema 6.1. In ogni caso, la permutazione γ è ottenuta dalla permutazione π sostituendo il minimo a di un blocco B con $\beta = \max B + 1$. Sulle permutazioni questo vuol dire che il numero di inversioni di γ è più grande di quello di π (perché $a < \beta$). Per concludere che $\pi \prec \gamma$ in $S_n(312)$ rimane solo da mostrare che la suddetta trasposizione non genera altre inversioni o, equivalentemente, che tutti gli elementi tra a e β in π sono comunque più piccole di a o più grandi di β . In effetti, $\beta - 1$ è il massimo di B , così appare prima a in π . Quindi, se c'è un elemento di x tale che $a < x < \beta$ e x è tra a e β in π , allora dovremmo avere un motivo 312, che invece è escluso. Di conseguenza abbiamo mostrato che, se $\pi \prec \gamma$ in $NC(n)$, allora anche $\pi \prec \gamma$ in $S_n(312)$.

Per concludere la dimostrazione mostreremo che, se $\pi \prec \gamma$ in $S_n(312)$, allora necessariamente γ è ottenuto da π come nel Teorema 6.1. Dalle ipotesi segue che γ differisce da π per una trasposizione di una coppia di elementi a e β . Supponiamo che $a < \beta$ e quindi a compare prima di β in π . Se a non era un minimo nella partizione noncrossing associata con π , allora dovrebbe esserci un'elemento $x < a$ che appare dopo a , e quindi in γ gli elementi β, x, a mostrerebbero un motivo 312. Di conseguenza a deve essere il minimo del suo blocco B nella partizione noncrossing π . Definiamo $b = \max B$. Deve essere $\beta = b + 1$. Infatti, se non fosse così, allora $\beta - 1$ non dovrebbe apparire tra a e β in π (perché altrimenti γ conterrebbe troppe inversioni). Chiaramente $\beta - 1$ non può comparire prima di un b , perché ogni elemento prima di b deve essere più piccola di b . Quindi $\beta - 1$ sta necessariamente a destra di β in π . Ma in questo caso la permutazione γ conterrebbe un motivo 312 negli elementi $\beta, a, \beta - 1$, questa è una contraddizione. Quindi $\beta = b + 1$, ed il teorema è infine dimostrato.

□

A questo punto è opportuno menzionare il seguente, notevole, corollario.

Corollario 6.1 *Per ogni $n \in \mathbf{N}$, il reticolo di Dyck \mathcal{D}_n è isomorfo al reticolo $S_n(312)$ con l'ordine di Bruhat forte.*

Nostro prossimo obiettivo è quello di trovare una descrizione sintetica delle operazioni di massimo e minimo nei reticoli di Bruhat delle permutazioni che evitano 312.

Sia $\pi = \pi_1 \cdots \pi_n$, $\gamma = \gamma_1 \cdots \gamma_n \in S_n(312)$. Definiamo la permutazione $\pi \vee \gamma = \sigma_1 \cdots \sigma_n$ ponendo σ_i uguale al più grande elemento fra quelli minori o uguali a $\max\{\pi_1, \dots, \pi_i, \gamma_1, \dots, \gamma_i\}$ non ancora apparsi nelle posizioni precedenti. Analogamente, la permutazione $\pi \wedge \gamma = \tau_1 \cdots \tau_n$ è definita ponendo τ_i uguale al più grande elemento fra quelli minori o uguali a $\min\{\max\{\pi_1, \dots, \pi_i\}, \max\{\gamma_1, \dots, \gamma_i\}\}$ non ancora apparsi nelle posizioni precedenti. Per esempio, data $\pi = 32657481$, $\gamma = 24378651$ abbiamo $\pi \vee \gamma = 34678521$ e $\pi \wedge \gamma = 23467581$. Nella proposizione seguente mostriamo che le operazioni sopra definite coincidono esattamente con quelle di massimo e di minimo in $S_n(312)$.

Proposizione 6.4 *Per ogni $\pi, \gamma \in S_n(312)$, le permutazioni $\pi \vee \gamma$ e $\pi \wedge \gamma$ sono rispettivamente il massimo ed il minimo di π e γ nel reticolo di Bruhat $S_n(312)$.*

Dim. Siano $\max(\pi)$ e $\max(\gamma)$ i vettori max delle partizioni noncrossing associate con π e γ , rispettivamente. Il massimo dei due cammini di Dyck associati a π e γ corrispondono al cammino di Dyck massimo determinato coordinata per coordinata di $\max(\pi)$ e $\max(\gamma)$, chiamato $\max(\pi) \vee \max(\gamma)$, che è quindi il vettore max del massimo di π e γ in $S_n(312)$. C'è un'unica permutazione che evita 312 associata a $\max(\pi) \vee \max(\gamma)$, che può essere ottenuta come segue: l' i -esimo elemento della permutazione è il più grande elemento tra quelli minori o uguali alla i -esima componente del vettore max non ancora apparso nella permutazione. Questo corrisponde esattamente alla nostra definizione di $\pi \vee \gamma$. Per quanto riguarda il minimo il ragionamento è lo stesso, quindi la dimostrazione è completa.

□

6.5 Una possibile estensione

Alla fine del nostro lavoro, vorremmo sottolineare la possibilità di estendere alle classi di partizioni senza restrizioni i risultati ottenuti ed illustrati in questo capitolo. Più precisamente, potrebbe essere interessante trovare

una relazione d'ordine sull'insieme delle partizioni che coincida con l'ordine di Bruhat quando ci si restringe alle partizioni noncrossing, così come determinare un'opportuna classe di cammini associati ad un tale ordine ed analizzare le connessioni con le permutazioni a motivo escluso. A proposito di questo, osserviamo che, se fossimo in grado di rappresentare le partizioni senza restrizioni in modo analogo a quelle noncrossing (cioè, gli elementi in ogni blocco sono in ordine decrescente, ed i blocchi sono elencati in modo tale che i loro massimi siano in ordine crescente), allora esisterebbe un'ovvia biiezione tra l'insieme delle partizioni e le permutazioni che evitano 3-12 (si adattano gli argomenti della Proposizione 6.3). Sfortunatamente, l'ordine della struttura indotta dall'ordine di Bruhat forte su $S_n(3-12)$ in generale non è un reticolo.

Un primo passo in questa direzione sarà l'oggetto dell'ultimo risultato che presentiamo. Consideriamo l'insieme Ω delle parole di Dyck, cioè l'insieme di tutte le parole ω su un alfabeto con due simboli $\{U, D\}$ che soddisfa la condizione nota di Dyck, ovvero ω contiene lo stesso numero di D e di U , ed ogni prefisso di ω contiene almeno tante U quante D . Chiamiamo *matching* di una parola di Dyck $\omega = \omega(U, D)$ un accoppiamento tra le U e le D di ω . Rappresentiamo i matching con degli archi come in figura:



Definiamo *Bell matching* un matching di $\omega \in \Omega$ che soddisfa le seguenti due condizioni:

1. per ogni insieme di D consecutive, la D più a sinistra è accoppiata con la prima U alla sua sinistra;
2. ogni altro D è accoppiato con una U alla sua sinistra, in modo tale che non ci siano incroci degli archi originati dall'insieme delle D consecutive.

La seguente proposizione mostra la connessione tra i Bell matching e l'insieme delle partizioni.

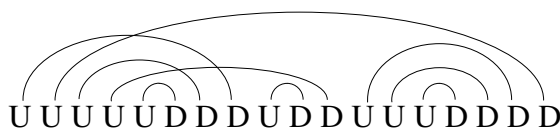


Figura 6.6: Un Bell matching di una parola di Dyck di lunghezza 18

Proposizione 6.5 *Esiste una biezione tra i Bell matching delle parole di Dyck di lunghezza $2n$ e l'insieme delle partizioni di $[n]$.*

Dim. Data una parola di Dyck ω , si contano le U di ω da sinistra a destra, poi si etichetta ogni D con il numero della U che si accoppia con lei. Gli insiemi delle etichette di ogni sottoparola di D consecutive costituisce il blocco della partizione. E' facile vedere che questa costruzione può essere invertita per arrivare alla biezione desiderata.

□

Osserviamo che le partizioni noncrossing corrispondono ai Bell matching senza incroci. Chiaramente, per ogni parola di Dyck di lunghezza $2n$, c'è esattamente un Bell matching senza incroci.

La classe dei Bell matching delle parole di Dyck di lunghezza fissata può essere partizionata come segue. Diciamo che due Bell matching sono *equivalenti* se sono matching della stessa parola di Dyck. Questa relazione di equivalenza può essere trasportata sull'insieme delle partizioni, in modo tale che ogni classe di equivalenza contiene esattamente una partizione noncrossing. Il risultato finale che abbiamo ottenuto è l'enumerazione dell'insieme delle partizioni tramite il conteggio degli elementi di ogni classe di equivalenza. Di conseguenza, otteniamo un'espressione dei numeri di Bell in termini di parametri naturali dei cammini di Dyck (altezza dei picchi e delle valli) che a quanto ci risulta non compare in letteratura.

Teorema 6.5 *Sia $\pi = B_1 | \dots | B_k \in NC(n)$. Allora la classe di equivalenza $[\pi]$ ha cardinalità:*

$$|[\pi]| = \prod_{i=1}^k \binom{b_i - \sum_{j=1}^{i-1} |B_j| - 1}{|B_i| - 1}, \quad (6.5)$$

dove $b_i = \max B_i$, come di consueto. In modo equivalente, la biezione tra le partizioni noncrossing ed i cammini di Dyck trasla la formula (6.5) nel modo

seguinte:

$$|[P]| = \prod_{i=1}^k \binom{p_i - 1}{v_i}, \quad (6.6)$$

dove P è il cammino di Dyck associato con π e p_k e v_k sono le altezze del k -esimo picco e della k -esima valle di P , rispettivamente.

Sommando le cardinalità di tutte le classi di equivalenza, otteniamo la seguente espressione per i numeri di Bell:

$$\mathcal{B}_n = \sum_{P \in \mathcal{D}_n} \prod_{i=1}^k \binom{p_i - 1}{v_i}. \quad (6.7)$$

Dim. Data la partizione noncrossing π , possiamo ottenere la formula (6.5) considerando la parola di Dyck ω associata a π e poi contare il numero di Bell matching definibile su tale parola. Per fare questo, consideriamo il primo insieme di D consecutive in ω . La D iniziale deve essere accoppiata con la U ad essa adiacente, così c'è solo una possibile scelta; tutte le rimanenti D di questo primo gruppo possono essere accoppiate con qualche precedente U . Poiché la cardinalità del fattore iniziale di U è b_1 (cioè il massimo del blocco B_1), e la cardinalità del primo insieme di D consecutive è $|B_1|$, abbiamo $\binom{b_1-1}{|B_1|-1}$ possibili scelte. Ora consideriamo il secondo fattore di D consecutive in ω . In questo caso, dobbiamo accoppiare tutte quelle D con ognuna delle precedenti U non già accoppiate, eccetto la D più a sinistra, che deve essere accoppiata con la U ad essa adiacente. Poiché dobbiamo scegliere $|B_2| - 1$ U fuori dall'insieme di $b_2 - |B_1| - 1$ U , le scelte possibili sono $\binom{b_2-|B_1|-1}{|B_2|-1}$. Iterando il procedimento otteniamo la formula (6.5), come desiderato.

Per ottenere la formula (6.6) dobbiamo trasportare i parametri sulle partizioni noncrossing in quelli sui cammini di Dyck, come fatto nel Teorema 6.3.

□

Bibliografia

- [A] M. Aigner, *Combinatorial Theory*, Springer-Verlag, Berlin-New York (1979).
- [A1] M. Aigner, *Catalan and other numbers: a recurrent theme*, in: H. Crapo, D. Senato (Eds.), *Algebraic Combinatorics and Computer Science*, Springer, Milano (2001), 347-390.
- [BBFP1] E. Barucci, A. Bernini, L. Ferrari, M. Poneti, *A distributive lattice structure on noncrossing partition*, presentato come poster a FPSAC 2005.
- [BBFP2] E. Barucci, A. Bernini, L. Ferrari, M. Poneti, *A distributive lattice structure connecting Dyck paths, noncrossing partitions and 312-avoiding permutations*, Order Journal, pubblicato online, apparirà su rivista.
- [BS] E. Babson, E. Steingrímsson, *Generalized permutation patterns and a classification of the Mahonian statistics*, Sémin. Lothar. Combin. 44 (2000), Art. B44b, 18 pp.
- [B et al.] C. Banderier, M. Bousquet-Mélou, A. Denise, P. Flajolet, D. Gardy, D. Gouyou-Beauchamps *Generating functions for generating trees*, Discrete Math. 246 (2002), 29-55.
- [BK] J. Bandlow, K. Killpatrick, *An area-to-inv bijection between Dyck paths and 312-avoiding permutations*, Electron. J. Combin. 8 (2001), #R40 (16 pp.).
- [BDLPP] E. Barucci, A. Del Lungo, E. Pergola, R. Pinzani *ECO: a methodology for the enumeration of combinatorial objects*, J. Difference Equ. Appl. 5 (1999), 435-490.
- [BFP] A. Bernini, L. Ferrari, R. Pinzani, *Enumerating permutations avoiding three Babson-Steingrímsson patterns*, Ann. Comb. 9 (2005), 137-162.

-
- [BW] A. Bjorner, M. Wachs, *Shellable nonpure complexes and posets. II*, Trans. Amer. Math. Soc. 349 (1997), 3945-3975.
- [CJ] S. Cautis, D. M. Jackson, *The matrix of chromatic joins and the Temperley-Lieb algebra*, J. Combin. Theory Ser. B 89 (2003), 109-155.
- [C] A. Claesson, *Generalized pattern avoidance*, European J. Combin. 22 (2001), 961-971.
- [CM] A. Claesson, T. Mansour, *Enumerating permutations avoiding a pair of Babson-Steingrímsson patterns*, Ars Combin. apparirà.
- [DP] B. A. Davey, H. A. Priestley, *Introduction to Lattices and Order*, Cambridge University Press, Cambridge (1990).
- [DS] A. Denise, R. Simion, *Two combinatorial statistics on Dyck paths*, Discrete Math. 137 (1995). 155-176.
- [De] E. Deutsch, *Dyck path enumeration*, Discrete Math. 204 (1999). 167-202.
- [Dr] B. Drake, *The weak order on pattern-avoiding permutations*, proceedings di FPSAC 2005.
- [EP] S. Elizalde, I. Pak, *Bijections for refined restricted permutations*, J. Combin. Theory Ser. A 105 (2004), 207-219.
- [F] M. Fulmek, *Enumeration of permutations containing a prescribed number of occurrences of a pattern of length three*, Adv. in Appl. Math. 30 (2003), 607-632.
- [FP1] L. Ferrari, R. Pinzani, *A linear operator approach to succession rules*, Linear Algebra Appl. 348 (2002), 231-246.
- [FP2] L. Ferrari, R. Pinzani, *Lattices of lattice paths*, J. Statist. Plann. Inference apparirà, consultabile su [arXiv:math.CO/0411427](https://arxiv.org/abs/math/0411427).
- [GPP] O. Guibert, E. Pergola, R. Pinzani, *Vexillary involutions are enumerated by Motzkin numbers*, Ann. Comb. 5 (2001), 153-174.
- [Kra] C. Krattenthaler, *Permutations with restricted patterns and Dyck paths*, Adv. in Appl. Math. 27 (2001), 510-530.
- [Kre] G. Kreweras, *Sur les partitions non croisées d'un cycle*, Discrete Math. 1 (1972), 333-350.

-
- [N] T. V. Narayana, *Lattice Path Combinatorics with Statistical Applications*, University of Toronto Press, Toronto (1979).
- [PPR] E. Pergola, R. Pinzani, S. Rinaldi, *ECO-approximation of algebraic functions*, in: D. Krob, A. A. Mikhalev, A. V. Mikhalev (Eds.), FPSAC 2000, Springer, Berlin.
- [P] R. A. Proctor, *Bruhat lattices, plane partitions generating functions, and minuscule representations*, European J. Combin., 5 (1984), 331-350.
- [R] A. Reifegerste, *On the diagram of 132-avoiding permutations*, European J. Combin., 24 (2003), 759-776.
- [S] R. Simion, *Noncrossing partitions*, Discrete Math. 217 (2000), 367-409.
- [SS] R. Simion, F. W. Schmidt, *Restricted Permutations*, European J. Combin. 6 (1985), 383-406.
- [Sl] N. J. A. Sloane, *The On-Line Encyclopedia of Integer Sequences*, at <http://www.research.att.com/~njas/sequences/index.html>.
- [S1] R. P. Stanley, *Enumerative Combinatorics 1*, Cambridge University Press, New York (1997).
- [S2] R. P. Stanley, *Enumerative Combinatorics 2*, Cambridge University Press, Cambridge (1999).
- [W] J. West, *Generating trees and the Catalan and Schröder numbers*, Discrete Math. 146 (1995), 247-262.

Capitolo 7

m -composizioni

Una *composizione* di un numero naturale n è una k -upla $\gamma = (x_1, \dots, x_k)$ di interi positivi tali che $x_1 + \dots + x_k = n$. Gli elementi x_i , k e n sono, rispettivamente, le parti, la lunghezza e l'ordine di γ . Sappiamo che ci sono $\binom{n-1}{k-1}$ composizioni di lunghezza k di n e 2^{n-1} composizioni di n , quando $n \geq 1$. Le composizioni sono oggetti molto ben conosciuti [A, C, Co] e molte delle loro proprietà sono state studiate in alcuni articoli recenti, come in [BS, CH, HM04, HM05, KP, KR, MUV].

In [CFMRR] è stata estesa la definizione di composizioni ordinarie definendo la classe delle 2-composizioni per ottenere una biezione tra questa classe e la classe dei poliomini L -convessi, dove all'ordine di una composizione viene associato il perimetro di un poliomino. Tale estensione al caso bidimensionale può essere immediatamente generalizzata al caso m -dimensionale. Il nostro scopo è ora quello di studiare la classe delle composizioni m -dimensionali ed generarle [MPR].

Sia m un intero positivo. Definiamo *composizione in forma di matrice con m righe*, o più semplicemente *m -composizione*, una matrice $m \times k$ con elementi interi non negativi

$$M = \begin{bmatrix} x_{11} & \dots & x_{1k} \\ \vdots & & \vdots \\ x_{m1} & \dots & x_{mk} \end{bmatrix}$$

le cui colonne sono diverse da vettori nulli. Diciamo che il numero k di colonne è la *lunghezza* della matrice. Inoltre diciamo che M è una *m -composizione* di un intero non negativo n se la somma di tutti i suoi elementi è esattamente n . Inoltre, $\sigma(M)$ indica la somma di tutti gli elementi della matrice M .

Le composizioni in forma di matrice così definite sono molto simili alle *composizioni in forma di vettori* [A, p.57] definite da P. A. MacMahon e studiate per esempio in [A75, A76a, A76b].

7.1 Identità combinatorie

Cominciamo il nostro studio sulle m -composizioni ottenendo molte identità su questi oggetti utilizzando elementari concetti combinatori.

Per prima cosa, comunque, richiamiamo alcune definizioni e proprietà di base dei multiset. Un *multiset* su un insieme X è una funzione $\mu : X \rightarrow \mathbb{N}$. La *molteplicità* di un elemento $x \in X$ è $\mu(x)$. L' *ordine* di μ è la somma $\text{ord}(\mu)$ delle molteplicità degli elementi X , cioè $\text{ord}(\mu) = \sum_{x \in X} \mu(x)$. Il numero di tutti i multiset di ordine k su un insieme di dimensione n è il *coefficiente del multiset*

$$\binom{n}{k} = \frac{n^{\bar{k}}}{k!} = \frac{n(n+1)\dots(n+k-1)}{k!}.$$

Siano $\mathcal{C}_{n,k}^{(m)}$ l'insieme di tutte le m -composizioni di n di lunghezza k e $c_{n,k}^{(m)} = |\mathcal{C}_{n,k}^{(m)}|$. Allo stesso modo siano $\mathcal{C}_n^{(m)}$ l'insieme di tutte le m -composizioni di n e $c_n^{(m)} = |\mathcal{C}_n^{(m)}|$.

Data $M \in \mathcal{C}_{n+m,k+1}^{(m)}$, la sua prima colonna è equivalente al multiset di $[m]$ di ordine non nullo i ed il resto della matrice è una m -composizione di $n+m-i$ di lunghezza k . Deriva quindi la ricorrenza:

$$c_{n+m,k+1}^{(m)} = \sum_{i=1}^{n+m-k} \binom{m}{i} c_{n+m-i,k}^{(m)}. \quad (7.1)$$

Gli stessi ragionamenti portano all'identità

$$c_{n+m}^{(m)} = \sum_{i=1}^{n+m} \binom{m}{i} c_{n+m-i}^{(m)}. \quad (7.2)$$

Ora utilizzeremo dei concetti basati sul Principio di Inclusione-Esclusione. Sia A_i l'insieme di tutte le m -composizioni M di $n+m$ con elementi positivi in posizione $i1$. Allora, poichè la prima colonna di M è diversa da un vettore di zeri, segue che $\mathcal{C}_n^{(m)} = A_1 \cup \dots \cup A_m$ e per il Principio di Inclusione-Esclusione

$$c_n^{(m)} = |A_1 \cup \dots \cup A_m| = \sum_{\substack{S \subseteq [m] \\ S \neq \emptyset}} (-1)^{|S|-1} \left| \bigcap_{i \in S} A_i \right|.$$

L'insieme $\bigcap_{i \in S} A_i$ è formato da tutte le m -composizioni $M = [x_{i,j}]$ di $n+m$ con elementi positivi nella prima colonna nella posizione indicata da S . Sostituendo $x_{i,1} - 1$ ad ogni elemento $x_{i,1}$ (con $i \in S$), abbiamo due casi: la prima colonna di M è un vettore di zeri o non lo è. Nel primo caso

rimuovendo la prima colonna abbiamo una m -composizione di $n + m - |S|$ mentre nel secondo abbiamo m -composizioni di $n + m - |S|$. Inoltre

$$\left| \bigcap_{i \in S} A_i \right| = 2c_{n+m-|S|}^{(m)}.$$

Poichè questo risultato dipende solo dalla dimensione di S segue che

$$c_{n+m}^{(m)} = 2 \sum_{i=1}^m \binom{m}{i} (-1)^{i-1} c_{n+m-i}^{(m)}. \quad (7.3)$$

Per esempio, per $m = 2, 3, 4$ abbiamo le ricorrenze

$$\begin{aligned} c_{n+2}^{(2)} &= 4c_{n+1}^{(2)} - 2c_n^{(2)} \\ c_{n+3}^{(3)} &= 6c_{n+2}^{(3)} - 6c_{n+1}^{(3)} + 2c_n^{(3)} \\ c_{n+4}^{(4)} &= 8c_{n+3}^{(4)} - 12c_{n+2}^{(4)} + 8c_{n+1}^{(4)} - 2c_n^{(4)}. \end{aligned}$$

Esattamente con lo stesso ragionamento otteniamo la relazione di ricorrenza

$$c_{n+m,k+1}^{(m)} = \sum_{i=1}^m \binom{m}{i} (-1)^{i-1} c_{n+m-i,k}^{(m)} + \sum_{i=1}^m \binom{m}{i} (-1)^{i-1} c_{n+m-i,k+1}^{(m)}. \quad (7.4)$$

Consideriamo ora gli insiemi A_i di tutte le matrici $M \in \mathcal{M}_{m,k}(\mathbb{N})$ con la i -esima colonna uguale al vettore di zeri tale che $\sigma(M) = n$. Allora $\mathcal{C}_{n,k}^{(m)} = A'_1 \cap \dots \cap A'_k$ e per il Principio di Inclusione-Esclusione, poichè questo risultato dipende solo dalla dimensione di S segue che

$$c_{n,k}^{(m)} = \sum_{i=0}^k \binom{k}{i} \left(\binom{m(k-i)}{n} \right) (-1)^i. \quad (7.5)$$

Allora

$$c_n^{(m)} = \sum_{k=0}^n c_{n,k}^{(m)} = \sum_{k=0}^n \sum_{i=0}^k \binom{k}{i} \left(\binom{m(k-i)}{n} \right) (-1)^i. \quad (7.6)$$

Questo concetto può facilmente essere generalizzato come segue. Consideriamo l'insieme $\mathcal{C}_k^{(m)}(r_1, \dots, r_m)$ di tutte le m -composizioni di lunghezza k dove la i -esima riga ha somma uguale a r_i , per ogni $i = 1, \dots, m$, e quindi $c_k^{(m)}(r_1, \dots, r_m)$ la sua cardinalità. Poniamo inoltre A_i come l'insieme di tutte le matrici $M \in \mathcal{M}_{m,k}(\mathbb{N})$ con la i -esima colonna uguale

al vettore di zeri e la somma delle righe sia uguale a (r_1, \dots, r_m) . Allora $\mathcal{C}_k^{(m)}(r_1, \dots, r_m) = A'_1 \cap \dots \cap A'_k$ e per il Principio di Inclusione-Esclusione

$$c_k^{(m)}(r_1, \dots, r_m) = |A'_1 \cap \dots \cap A'_k| = \sum_{S \subseteq [k]} (-1)^{|S|} \left| \bigcap_{i \in S} A_i \right|.$$

L'insieme $\bigcap_{i \in S} A_i$ è formato da tutte le matrici $M \in \mathcal{M}_{m,k}(\mathbb{N})$ con il vettore di zeri in ogni colonna con indice dato dagli elementi di S . La i -esima riga di tale matrice M corrisponde al multiset di ordine r_i su un insieme di dimensione $k - |S|$. Quindi segue che

$$\left| \bigcap_{i \in S} A_i \right| = \binom{k - |S|}{r_1} \cdots \binom{k - |S|}{r_m}.$$

Poiché il risultato dipende ancora solo dalla dimensione di S segue che

$$c_k^{(m)}(r_1, \dots, r_m) = \sum_{i=0}^k \binom{k}{i} \binom{k-i}{r_1} \cdots \binom{k-i}{r_m} (-1)^i. \quad (7.7)$$

7.2 m -composizioni e linguaggi regolari

7.2.1 Parole su un alfabeto infinito

Le m -composizioni possono essere viste come parole su un alfabeto infinito in modo molto semplice. Considerando ogni m -composizione come la concatenazione delle sue colonne segue immediatamente che l'insieme $\mathcal{C}^{(m)}$ di tutte le m -composizioni è equivalente al linguaggio libero \mathcal{A}^* sull'alfabeto infinito $\mathcal{A}^{(m)} = \{a_\mu : \mu \in \mathcal{M}_{\neq 0}^{(m)}\}$, dove $\mathcal{M}_{\neq 0}^{(m)}$ è l'insieme di tutti i multiset $\mu : [m] \rightarrow \mathbb{N}$ con ordine positivo ed il simbolo a_μ corrisponde alla colonna $[\mu(1) \ \dots \ \mu(m)]^T$. Sostituendo ogni simbolo a_μ con una indeterminata x_μ , segue immediatamente che la funzione generatrice di $\mathcal{C}^{(m)}$ è

$$c(X) = \frac{1}{1 - \sum_{\mu \in \mathcal{M}_{\neq 0}^{(m)}} x_\mu} \quad (7.8)$$

dove $X = \{x_\mu : \mu \in \mathcal{M}_{\neq 0}^{(m)}\}$. In particolare, per $x_\mu = x^{\text{ord}(\mu)}$ otteniamo la funzione generatrice

$$c^{(m)}(x) = \sum_{n \geq 0} c_n^{(m)} x^n = \frac{1}{1 - h(x)} \quad (7.9)$$

dove

$$h(x) = \sum_{k \geq 1} \binom{m}{k} x^k = \frac{1}{(1-x)^m} - 1.$$

Inoltre abbiamo

$$c^{(m)}(x) = \frac{(1-x)^m}{2(1-x)^m - 1}. \quad (7.10)$$

Segue la ricorrenza (7.3) che abbiamo già ottenuto nel precedente paragrafo.

In modo simile, per $x_\mu = x^{\text{ord}(\mu)}y$ otteniamo la funzione generatrice

$$c(x, y)^{(m)} = \sum_{n, k \geq 0} c_{n, k}^{(m)} x^n y^k = \frac{1}{1 - h(x)y}. \quad (7.11)$$

Dalla funzione generatrice (7.10) possiamo generalizzare la seguente identità soddisfatta dai numeri c_n delle 2-composizioni: $c_n = 3c_{n-1} + c_{n-2} + \dots + c_0$, $n \geq 2$, ottenuti in [CFMRR] manipolando la semplice ricorrenza $c_n = 4c_{n-1} - 2c_{n-2}$.

Da (7.10) abbiamo

$$c^{(m)}(x) = \frac{1}{2 - \frac{1}{(1-x)^m}} = \frac{(1-x)}{2 - 2x - \frac{1}{(1-x)^{m-1}}}$$

e quindi

$$c^{(m)}(x) \left(2 - 2x - \frac{1}{(1-x)^{m-1}} \right) = 1 - x$$

che porta all'equazione:

$$c_m(x) = 1 - x + 2xc^{(m)}(x) + \left(\frac{1}{(1-x)^{m-1}} - 1 \right) c^{(m)}(x).$$

Ora prendendo il rapporto incrementale di entrambi i lati abbiamo

$$Rc^{(m)}(x) = -1 + 2c^{(m)}(x) + R \left(\frac{1}{(1-x)^{m-1}} - 1 \right) c^{(m)}(x)$$

da cui arriviamo ad avere

$$c_{n+1}^{(m)} = -\delta_{n,0} + 2c_n^{(m)} + \sum_{k=1}^{n+1} \binom{m-1}{k} c_{n-k+1}^{(m)} = -\delta_{n,0} + 2c_n^{(m)} + \sum_{k=0}^n \binom{m-1}{k+1} c_{n-k}^{(m)}$$

cioè

$$c_{n+1}^{(m)} = -\delta_{n,0} + 2c_n^{(m)} + \sum_{k=0}^n \binom{m+k-1}{k+1} c_{n-k}^{(m)}. \quad (7.12)$$

7.2.2 Parole su un alfabeto finito

Estendendo la codifica utilizzata in [BS] per le composizioni ordinarie, proveremo adesso in modo più generale che le m -composizioni sono equivalenti a particolari parole su un alfabeto finito $\mathcal{A}_m = \{a_1, \dots, a_m, b_1, \dots, b_m\}$. Definiremo una associazione $\ell : \mathcal{C}^{(m)} \rightarrow \mathcal{A}_m^*$ come segue

$$\begin{array}{cccc} 1 & & 0 & & 1 & & 0 \\ 0 & \xrightarrow{\ell} a_1, & \dots & \vdots \xrightarrow{\ell} a_m, & + & \vdots \xrightarrow{\ell} b_1, & \dots & + & \vdots \xrightarrow{\ell} b_m \\ \vdots & & & 0 & + & \vdots & & + & 0 \\ 0 & & & 1 & & 0 & & & 1 \end{array}$$

e operando come segue. Prima di tutto scriviamo una m -composizione M come somma formale (cioè giustapposizione) delle sue colonne. Scriviamo poi ogni colonna come una giustapposizione di colonne semplici dove una colonna semplice è una colonna in cui tutti gli elementi eccetto uno sono zero. Ordiniamo le colonne semplici secondo la posizione dell'elemento diverso da zero. A questo punto scriviamo ogni colonna semplice come una giustapposizione di colonne elementari, dove una colonna elementare è una colonna in cui tutti gli elementi sono zero tranne uno uguale a 1. Inoltre, se l'elemento diverso da zero di una colonna semplice è k allora sarà scritta come la giustapposizione di k colonne elementari. Infine sostituiamo ogni colonna elementare con il corrispondente simbolo secondo la codifica data.

Un esempio spiegherà meglio la corrispondenza. Consideriamo la 3-composizione di 11

$$M = \begin{bmatrix} 2 & 0 & 1 & 2 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 2 \end{bmatrix}.$$

Seguendo la procedura descritta abbiamo:

$$\begin{array}{cccccccccccc} 2 & 0 & 1 & 2 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ M = 0 + 1 + 0 + 1 = 0 & 0 & 0 + 1 + 0 & 0 + 0 & 0 & 1 & 0 & 0 & & & & & & & \\ 1 & 0 & 1 & 2 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \end{array}$$

e quindi $\ell(M) = a_1 a_1 a_3 b_2 b_1 a_3 b_1 a_1 a_2 a_3 a_3$.

Sia $\mathcal{L}_m = \ell(\mathcal{C}^{(m)})$ il linguaggio sull'alfabeto \mathcal{A}_m corrispondente alle m -composizioni. Le parole di \mathcal{L}_n soddisfano le seguenti condizioni:

1. ogni parola comincia con un simbolo $\in \{a_1, \dots, a_m\}$;
2. ogni simbolo a_i o b_i può essere seguito da un qualsiasi b_j mentre può essere seguito da una simbolo a_j solo quando $i \leq j$.

Questo implica che le parole in \mathcal{L}_n hanno una fattorizzazione unica della forma xy dove:

1. x è una parola non vuota della forma $a_1^{i_1} \dots a_m^{i_m}$, con $i_1, \dots, i_m \geq 0$;
2. y è una parola (volendo anche vuota) $y = y_1 \dots y_k$, con $y_r = b_j a_j^{q_j} \dots a_m^{q_m}$, with $q_j, \dots, q_m \geq 0$.

Secondo tale caratterizzazione \mathcal{L}_m è un linguaggio regolare definito dall'espressione regolare non ambigua:

$$\varepsilon + (a_1^+ a_2^* \dots a_m^* + a_2^+ a_3^* \dots a_m^* + \dots + a_m^+) (b_1 a_1^* a_2^* \dots a_m^* + b_2 a_2^* \dots a_m^* + \dots + b_m a_m^*)^*$$

dove, ε denota la parola vuota.

Questa codifica ci permette di ottenere un sistema ECO per le m -composizioni. Inoltre, poiché a_i e b_i possono essere seguite dalle simboli $a_i, \dots, a_m, b_1, \dots, b_m$, etichettando a_i e b_i con $(2m - i + 1)$, segue il sistema ECO

$$\left\{ \begin{array}{l} (*) \\ (*) \rightsquigarrow (m+1) \cdots (2m) \\ (k) \rightsquigarrow (m+1)^2 \cdots (k)^2 (k+1) \cdots (2m) \quad (m+1 \leq k \leq 2m). \end{array} \right.$$

7.3 Interpretazioni combinatorie

7.3.1 Partizioni lineari linearmente colorate

Le m -composizioni possono essere interpretate in termini di specie lineari [J] come segue. Sia $C = \{c_1, \dots, c_m\}$ un insieme di colori totalmente ordinato nel modo naturale $c_1 < \dots < c_m$. Diciamo che l'insieme linearmente ordinato $[n] = \{1, 2, \dots, n\}$ è m -colorato quando ogni elemento è colorato con un colore in C rispettando la seguente condizione: se due elementi x e y , con $x \leq y$, appartengono allo stesso blocco ed hanno colori c_i e c_j , rispettivamente, allora $i \leq j$. Inoltre una m -colorazione di $[n]$ è un ordine che preserva l'associazione $\gamma : [n] \rightarrow C$. Definiamo una partizione lineare m -colorata di $[n]$ come una partizione lineare in cui ogni blocco è m -colorato.

Le m -composizioni di n di lunghezza k sono equivalenti alla partizione lineare m -colorata di $[n]$ con k blocchi. Viceversa, ad ogni m -composizione M di n , di lunghezza k , possiamo associare una partizione lineare m -colorata π di $[n]$ come segue. Alla i -esima colonna (h_1, \dots, h_m) di M si associa l' i -esimo blocco di π di dimensione $h_1 + \dots + h_m$ avente il primo elemento h_1 colorato con c_1, c_2, \dots , fino all'ultimo elemento h_m colorato con c_m . Per esempio, la

3-composizione

$$M = \begin{bmatrix} 2 & 0 & 1 & 2 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 2 \end{bmatrix}$$

corrisp

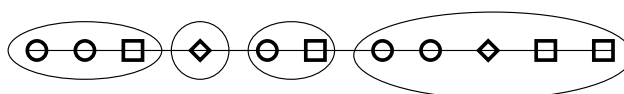


Figura 7.1: Rappresentazione della partizione con i 3 colori (cerchietto, rombo e quadrato rappresentano per esempio rosso, verde e blu rispettivamente).

dove i tre simboli rappresentano tre colori (per esempio la stella, il rombo ed il cerchio sono rispettivamente il rosso, il blu ed il verde).

Siano \mathbf{Comp}_m le specie lineari delle m -composizioni, cioè delle partizioni lineari m -colorate. Dare una struttura di queste specie su di un insieme linearmente ordinato L equivale ad assegnare una partizione lineare π su L e poi una m -colorazione, cioè un ordine che preservi la associazione su C , su ogni blocco di π . Allora, se \mathbf{G} indica le specie lineari uniformi e $\mathbf{Map}_{\neq \emptyset}^{(m)}$ indica le specie lineari dell'ordine che preserva le associazioni da un ordine lineare non vuoto all'insieme dei colori C , abbiamo che

$$\mathbf{Comp}_m = \mathbf{G} \circ \mathbf{Map}_{\neq \emptyset}^{(m)}.$$

Un ordine che preserva la associazione $f : [k] \rightarrow [m]$ equivale ad un multiset di ordine k sull'insieme $[m]$. Quindi segue che

$$\mathbf{Card}(\mathbf{Map}_{\neq \emptyset}^{(m)}; x) = \sum_{k \geq 1} \binom{m}{k} x^k = \frac{1}{(1-x)^m} - 1$$

e di conseguenza $\mathbf{Card}(\mathbf{Comp}_m; x) = \mathbf{Card}(\mathbf{G}; x) \circ \mathbf{Card}(\mathbf{Map}_{\neq \emptyset}^{(m)}; x) = c^{(m)}(x)$.

Utilizzando questa interpretazione possiamo ottenere alcune utili identità. Sia $\pi \in \mathbf{Comp}_m[L]$, dove $L = \{x_1, \dots, x_{i+1}, \dots, x_{i+j+1}\}$ ha dimensione $i+j+1$. L'elemento x_{i+1} appartiene ad un blocco della forma $\{x_{i-h+1}, \dots, x_i, x_{i+1}, x_{i+2}, \dots, x_{i+k+2}\}$ dove $h, k \in \mathbb{N}$. Rimuovendo tale blocco, π si divide in

due partizioni lineari m -colorate di un ordine lineare di dimensione $i - h$ ed un ordine lineare di dimensione $j - k$, rispettivamente. Quindi segue che

$$c_{i+j+1}^{(m)} = \sum_{h,k \geq 0} \binom{m}{h+k+1} c_{i-h}^{(m)} c_{j-k}^{(m)}. \quad (7.13)$$

Ricordiamo che $\binom{m}{i+j+1}$ fornisce il numero di tutte le associazioni d'ordine $f : [i+j+1] \rightarrow [m]$. Supponiamo che $f(i+1) = k$, con $k \in [m]$. Poiché f è l'ordine che preserva, segue che $f(x) \in [k]$ per ogni $x \in [i]$ e $f(x) \in \{k, \dots, m\}$ per ogni $x \in \{i+2, \dots, i+j+1\}$. Quindi

$$\binom{m}{i+j+1} = \sum_{k=1}^m \binom{k}{i} \binom{m-k+1}{j} = \sum_{k=0}^{m-1} \binom{i+k}{i} \binom{m-k}{j}. \quad (7.14)$$

7.3.2 Famiglie suriettive

Le m -composizioni ammettono anche la seguente interpretazione combinatoria. Siano P_1, \dots, P_m e Q insiemi ordinati linearmente. Consideriamo una famiglia $\{f_i : P_i \rightarrow Q\}_{i=1}^m$ di ordine che preserva le associazioni con la seguente proprietà: per ogni elemento $q \in Q$ esiste almeno un indice i ed un elemento $p \in P_i$ tale che $q = f_i(p)$. Le singole associazioni non sono necessariamente suriettive ma l'intera famiglia lo è, nel senso che ogni elemento del codominio ammette almeno una retroimmagine lungo una delle associazioni della famiglia. Quindi noi diciamo che una famiglia con questa proprietà è una *famiglia suriettiva*.

Ora possiamo chiederci quante famiglie suriettive ci sono, quando $|P_1| = r_1, \dots, |P_m| = r_m$ e $|Q| = k$. La risposta è: $c_k^{(m)}(r_1, \dots, r_m)$. Al contrario, data una famiglia suriettiva $\{f_i : P_i \rightarrow Q\}_{i=1}^m$ possiamo costruire una m -composizione M di lunghezza k come segue. La i -esima riga di M è generata dalla associazione $f_i : P_i \rightarrow Q$ prendendo come elementi i numeri delle retroimmagini degli elementi di Q lungo f_i , cioè definendo questo come $[|f_i^\bullet(1)| \dots |f_i^\bullet(k)|]$, dove $f_i^\bullet(y)$ indica l'insieme di tutte le retroimmagini di y lungo f_i . Chiaramente la somma di questa riga è $|P_i| = r_i$. Comunque ogni colonna di M è diversa dal vettore di zeri per la proprietà che caratterizza le famiglie suriettive. Così, abbiamo che M è una m -composizione di lunghezza k con vettore somma delle righe uguale a (r_1, \dots, r_m) .

7.3.3 Bargraph etichettati

Un *bargraph* è un poliomino convesso per colonne, tale che la base si trova tutta sull'asse orizzontale. Un bargraph è univocamente definito dall'altezza delle sue colonne (Figura 7.2, (a)). L'enumerazione dei bargraph secondo perimetro e area è stata trattata in [OP, PB], in relazione allo studio dei modelli di percolazione, e, più recentemente, da un punto di vista analitico, in [BMR].

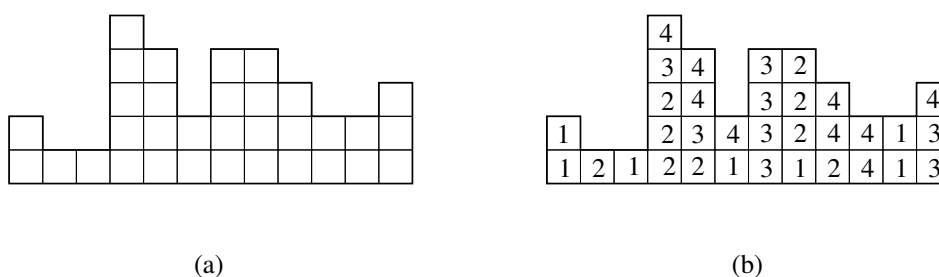


Figura 7.2: (a) Un bargraph; (b) un bargraph etichettato di grado 4.

Qui consideriamo *bargraphs etichettati*, cioè bargraph le cui celle sono tutte etichettate con interi positivi, e tali che, per ogni colonna, l'etichetta di una cella è minore o uguale di quella della cella immediatamente sopra (se c'è), (Figura 7.2, (b)). Il grado è l'etichetta massima del bargraph. Per ogni $m \geq 1$ dato, tutte le m -composizioni di un intero n possono essere rappresentate come un bargraph etichettato di grado $j \leq m$ avente n celle. Questa rappresentazione è strettamente in relazione con la rappresentazione sotto forma di parola delle m -composizioni, spiegata nel paragrafo 7.2.

Mostriamo questa semplice corrispondenza utilizzando un esempio. Consideriamo la seguente 4-composizione di 18,

$$M = \begin{bmatrix} 2 & 0 & 1 & 2 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 3 & 2 \\ 2 & 0 & 1 & 2 \end{bmatrix}.$$

Dalle considerazioni precedenti M è rappresentata attraverso la parola

$$\ell(M) = a_1^2 a_3 a_4^2 b_2 b_1 a_3^3 a_4 b_1 a_1 a_2 a_3^2 a_4^2,$$

dove la notazione della potenza è utilizzata per le ripetizioni. La parola è decomposta, in modo naturale, in sottoparole da ogni occorrenza del simbolo b_i ,

$$a_1^2 a_3 a_4^2 \mid b_2 \mid b_1 a_3^3 a_4 \mid b_1 a_1 a_2 a_3^2 a_4^2.$$

La parola è poi trasformata nella parola

$$a_1^2 a_3 a_4^2 \mid a_2 \mid a_1 a_3^3 a_4 \mid a_1^2 a_2 a_3^2 a_4^2,$$

semplicemente cambiando ogni occorrenza di b_i con un'occorrenza di a_i .

Costruiamo poi un bargraph dove la i -esima colonna corrisponde alla i -esima componente in ogni decomposizione. In pratica, la i -esima componente ha tante celle quante il numero di elementi presenti nella i -esima componente, e queste celle sono etichettate leggendo da sinistra verso destra gli indici degli elementi nella componente. Da questo otteniamo un bargraph al massimo di grado m , come in Figura 7.3.

			4
			4
4		4	3
4		3	3
3		3	2
1		3	1
1	2	1	1

Figura 7.3: Il bargraph di grado 4 associato ad M .

Naturalmente, le composizioni ordinarie (cioè le 1-composizioni) sono rappresentate come bargraph di grado 1, cioè i bargraph classici.

Alcune sottoclassi di m -composizioni

La semplice corrispondenza tra m -composizioni e bargraph etichettati può essere applicata per determinare biiezioni tra sottoclassi particolari. Quindi consideriamo:

1. l'insieme di bargraph aventi tutte le m etichette in ogni colonna; corrisponde all'insieme delle m -composizioni non contenenti elementi nulli. Per esempio, il bargraph in Figura 7.4 (a) corrisponde alla composizione:

$$M = \begin{bmatrix} 1 & 1 & 1 & 3 & 2 \\ 1 & 1 & 1 & 2 & 1 \\ 1 & 3 & 1 & 1 & 1 \end{bmatrix}.$$

La funzione generatrice di tali oggetti è quindi facilmente determinata:

$$1 + \frac{\frac{x^m}{(1-x)^m}}{1 - \frac{x^m}{(1-x)^m}} = \frac{1}{1 - \left(\frac{x}{1-x}\right)^m}.$$

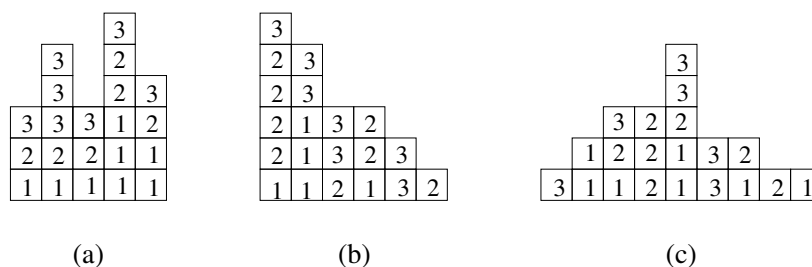


Figura 7.4: Bargraph etichettati di grado 3: (a) con tutte le etichette in ogni colonna; (b) una 3-partizione; (c) uno stack etichettato di grado 3.

2. l'insieme dei *diagrammi di Ferrers etichettati*, cioè quei bargraph etichettati per cui ogni colonna ha altezza maggiore o uguale di quella della colonna alla sua destra, (Figura 7.4, (b)). Un diagramma di Ferrers etichettato di grado m corrisponde ad una m -composizione tale che la somma degli elementi di ogni colonna è maggiore o uguale della somma degli elementi della colonna alla propria destra. Chiamiamo questi oggetti m -partizioni. Questa definizione è motivata dal fatto che le partizioni ordinarie corrispondono ai diagrammi di Ferrers, che possono essere visti come diagrammi di Ferrers generalizzati di grado 1. Per esempio, il bargraph di Figura 7.4 (b) corrisponde alla 3-partizione di 20:

$$\begin{bmatrix} 1 & 3 & 0 & 1 & 0 & 0 \\ 4 & 0 & 1 & 2 & 0 & 1 \\ 1 & 2 & 2 & 0 & 2 & 0 \end{bmatrix}$$

E' interessante determinare il numero di m -partizioni di un intero n , e studiare alcune delle loro proprietà combinatorie;

3. l'insieme degli *stack etichettati* hanno la forma dei poliomini stack, (Figura 7.4, (c)). Uno stack etichettato di grado m corrisponde alla m -composizione tale che la sequenza c_1, \dots, c_k è unimodale, essendo c_i la somma degli elementi della i -esima colonna.

Per esempio, il bargraph di Figura 7.4 (c) corrisponde alla 3-composizione:

$$\begin{bmatrix} 0 & 2 & 1 & 0 & 2 & 0 & 0 \\ 0 & 0 & 1 & 3 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 2 & 2 & 0 \end{bmatrix}.$$

7.4 Identità del tipo Cassini

In [CFMRR] è stato provato che i numeri $c_n^{(2)}$ di tutte le 2-composizioni di n soddisfano la seguente identità del tipo Cassini: $c_n^{(2)} c_{n+2}^{(2)} - (c_{n+1}^{(2)})^2 = -2^{n-1}$, per ogni $n \geq 1$. Qui proveremo che tale identità può essere generalizzata ai numeri $c_n^{(m)}$. Più precisamente proveremo che

$$\begin{vmatrix} c_n^{(m)} & c_{n+1}^{(m)} & \cdots & c_{n+m-1}^{(m)} \\ c_{n+1}^{(m)} & c_{n+2}^{(m)} & \cdots & c_{n+m}^{(m)} \\ \vdots & \vdots & & \vdots \\ c_{n+m-1}^{(m)} & c_{n+m}^{(m)} & \cdots & c_{n+2m-2}^{(m)} \end{vmatrix} = (-1)^{\lfloor m/2 \rfloor} 2^{n-1} \quad (7.15)$$

per ogni $m \geq 1$ e $n \geq 1$.

Sia $C_n^{(m)} = [c_{n+i+j}^{(m)}]_{i,j=0}^{m-1}$. La sua i -esima riga è $\mathbf{r}_i = [c_{n+i+j}^{(m)}]_{j=0}^{m-1}$. In particolare, dalla ricorrenza (7.3), l'ultima riga è

$$\mathbf{r}_m = 2 \sum_{k=1}^m \binom{m}{k} (-1)^{k-1} \mathbf{r}_{m-k} = 2 \sum_{k=1}^{m-1} \binom{m}{k} (-1)^{k-1} \mathbf{r}_{m-k} + (-1)^{m-1} 2 \mathbf{r}_0$$

dove $\mathbf{r}_0 = [c_{n-1+j}^{(m)}]_{j=0}^{m-1}$. Allora sottraendo all'ultima riga la combinazione lineare seguente delle colonne precedenti

$$2 \sum_{k=1}^{m-1} \binom{m}{k} (-1)^{k-1} \mathbf{r}_{m-k}$$

l'ultima riga di $\det C_n^{(m)}$ diventa $(-1)^{m-1} 2 \mathbf{r}_0$. Estraeando $(-1)^{m-1} 2$ dall'ultima linea e poi scalando ciclicamente tutte le righe verso il basso otteniamo che

$$\det C_n^{(m)} = 2 \det C_{n-1}^{(m)}.$$

Allora, per ogni $n \geq 1$, segue che

$$\det C_n^{(m)} = 2^{n-1} \det C_1^{(m)}.$$

Ora dobbiamo considerare il determinante della matrice $C_1^{(m)} = [c_{i+j+1}^{(m)}]_{i,j=0}^{m-1}$. Dall'identità (7.13) abbiamo l'identità $C_1^{(m)} = L_m M_m L_m^T$ dove $L_m = [c_{i-j}^{(m)}]_{i,j=0}^{m-1}$,

$M_m = \left[\binom{m}{i+j+1} \right]_{i,j=0}^{m-1}$ e T indica la trasposizione. Poiché L_m è una matrice triangolare con elementi diagonali unitari, segue che $\det C_1^{(m)} = \det M_m$. Ora l'identità (7.14) implica l'identità $M_m = B_m \tilde{B}_m$ dove $B_m = \left[\binom{i+j}{i} \right]_{i,j=0}^{m-1}$ e $\tilde{B}_m = \left[\binom{m-i}{j} \right]_{i,j=0}^{m-1}$. Ma $\tilde{B}_m = J_m B_m$ dove $J_m = [\delta_{i+j,m-1}]_{i,j=0}^{m-1}$. Allora $M_m = B_m J_m B_m$ e $\det M_m = \det J_m (\det B_m)^2$. Poiché, come ben sappiamo, $\det J_m = (-1)^{\lfloor m/2 \rfloor}$ e $\det B_m = 1$, segue che $\det M_m = (-1)^{\lfloor m/2 \rfloor}$ e di conseguenza $\det C_1^{(m)} = (-1)^{\lfloor m/2 \rfloor}$. Infine abbiamo $\det C_n^{(m)} = (-1)^{\lfloor m/2 \rfloor} 2^{n-1}$, per ogni $n \geq 1$.

7.5 Comportamento asintotico e formule del tipo Binet

Le radici del polinomio al denominatore della funzione (7.10) sono

$$x_k = 1 - \frac{1}{\sqrt[m]{2}} \omega_m^k \quad k = 0, 1, \dots, m-1$$

dove $\omega_m = e^{2\pi i/m}$ è una radice primitiva dell'unità.

Ora scriviamo la funzione generatrice (7.10) nel modo seguente

$$c^{(m)}(x) = \frac{1}{2} \frac{2(1-x)^m - 1 + 1}{2(1-x)^m - 1} = \frac{1}{2} \left[1 + \frac{1}{2(1-x)^m - 1} \right].$$

Allora otteniamo la seguente espansione in frazioni parziali

$$\frac{1}{2(1-x)^m - 1} = \frac{A_0}{x - x_0} + \dots + \frac{A_{m-1}}{x - x_{m-1}}$$

dove i coefficient A_k sono definiti da

$$A_k = \lim_{x \rightarrow x_k} \frac{x - x_k}{2(1-x)^m - 1}.$$

Applicando la regola di De l'Hopital segue che

$$A_k = \lim_{x \rightarrow x_k} \frac{1}{-2m(1-x)^{m-1}} = \frac{1}{-2m(1-x_k)^{m-1}} = -\frac{\omega_m^k}{m \sqrt[m]{2}}.$$

Inoltre la funzione 7.10 si decompone come segue

$$c^{(m)}(x) = \frac{1}{2} \left[1 - \frac{1}{x} \sum_{k=0}^{m-1} \frac{A_k}{1 - x/x_k} \right] = \frac{1}{2} \left[1 + \frac{1}{m \sqrt[m]{2} x} \sum_{k=0}^{m-1} \frac{\omega_m^k}{1 - x/x_k} \right].$$

Di conseguenza abbiamo la seguente formula del tipo Binet

$$c_n^{(m)} = \frac{1}{2} \left[\delta_{n,0} + \frac{1}{m \sqrt[m]{2}} \sum_{k=0}^{m-1} \frac{\omega_m^k}{(x_k)^{n+1}} \right]$$

Da questa espressione è facile ottenere una formula asintotica per i coefficienti $c_n^{(m)}$ quando n tende all'infinito. Al contrario, la singolarità dominante, cioè la radice con modulo minimo, è $x_0 = 1 - 1/\sqrt[m]{2}$. Allora segue che

$$c_n^{(m)} \sim -\frac{A_0}{2x_0^{n+1}} = \frac{1}{2m(\sqrt[m]{2}-1)} \left(\frac{\sqrt[m]{2}}{\sqrt[m]{2}-1} \right)^n \quad \text{se } n \rightarrow \infty.$$

In particolare abbiamo

$$c_{n+1}^{(m)} \sim \frac{\sqrt[m]{2}}{\sqrt[m]{2}-1} c_n^{(m)} \quad \text{se } n \rightarrow \infty.$$

7.6 m -composizioni senza righe nulle

In questo paragrafo studieremo le m -composizioni in cui ogni riga è diversa da un vettore di zeri. Cominciamo determinando un'espressione per il numero $f_n^{(m)}$ di tutte queste m -composizioni di n . Sia A_i l'insieme di tutte le m -composizioni $M \in \mathcal{C}_n^{(m)}$ dove l' i -esima riga è zero. Allora

$$f_n^{(m)} = |A'_1 \cap \dots \cap A'_m| = \sum_{S \subseteq [m]} (-1)^{|S|} \left| \bigcap_{i \in S} A_i \right|.$$

Poiché $\bigcap_{i \in S} A_i$ è chiaramente in corrispondenza biunivoca con l'insieme di tutte le $(m - |S|)$ -composizioni di n , segue che

$$f_n^{(m)} = \sum_{k=0}^m \binom{m}{k} (-1)^k c_n^{(m-k)} = \sum_{k=0}^m \binom{m}{k} (-1)^{n-k} c_n^{(k)}. \quad (7.16)$$

D'altra parte, l'insieme $\mathcal{C}_n^{(m)}$ può essere partizionato secondo il numero di righe fatte di zeri e questo porta alle identità seguenti

$$c_n^{(m)} = \sum_{k=0}^m \binom{m}{k} f_n^{(k)}. \quad (7.17)$$

Ovviamente questa formula può anche essere ottenuta formalmente invertendo (7.16).

Dalla forma dell'identità (7.16) segue che la funzione generatrice per i numeri $f_n^{(m)}$ è

$$f^{(m)}(x) = \sum_{k=0}^m \binom{m}{k} (-1)^{n-k} c^{(k)}(x) = \sum_{k=0}^m \binom{m}{k} (-1)^{n-k} \frac{(1-x)^k}{2(1-x)^k - 1}.$$

Quindi questa funzione ha la forma

$$f^{(m)}(x) = \frac{x^m p_m(x)}{(1-2x)(1-4x+2x^2) \cdots (2(1-x)^m - 1)} \quad (7.18)$$

dove $p_m(x)$ è il polinomio con grado (minore o) uguale a $\binom{m}{2}$. Questo implica che, per $n \geq 1$, i numeri $f_n^{(m)}$ soddisfano una ricorrenza lineare omogenea con coefficienti costanti di ordine $\binom{m+1}{2}$, che possono essere dedotti dal denominatore della funzione (7.18). Per esempio per $m = 2$ abbiamo la funzione

$$f^{(2)}(x) = \frac{3x^2 - 2x^3}{(1-2x)(1-4x+2x^2)} = \frac{3x^2 - 2x^3}{1-6x+10x^2-4x^3}$$

e di conseguenza la ricorrenza

$$f_{n+3}^{(2)} = 6f_{n+2}^{(2)} - 10f_{n+1}^{(2)} + 4f_n^{(2)}.$$

In modo simile per $m = 3$ abbiamo la funzione

$$f^{(3)}(x) = \frac{13x^3 - 24x^4 + 16x^5 - 4x^6}{1-12x+52x^2-102x^3+96x^4-44x^5+8x^6}$$

e la ricorrenza

$$f_{n+6}^{(3)} = 12f_{n+5}^{(3)} - 52f_{n+4}^{(3)} + 102f_{n+3}^{(3)} - 96f_{n+2}^{(3)} + 44f_{n+1}^{(3)} - 8f_n^{(3)}.$$

Cerchiamo adesso una formula esplicita per i numeri $f_n^{(m)}$. Poiché gli $f_n^{(m)}$ contano tutte le m -composizioni in cui ogni somma per riga è non nulla, segue immediatamente che

$$f_n^{(m)} = \sum_{k \geq 0} \sum_{\substack{(r_1, \dots, r_m) \in \mathbb{N}_0^m \\ r_1 + \dots + r_m = n}} c_k^{(m)}(r_1, \dots, r_m) = \sum_{k \geq 0} \sum_{\substack{\rho \in \mathbb{N}_0^m \\ |\rho| = n}} c_k^{(m)}(\rho)$$

dove $\mathbb{N}_0 = \mathbb{N} \setminus \{0\}$, $\rho = (r_1, \dots, r_m)$ e $|\rho| = r_1 + \dots + r_m$. Allora, utilizzando (7.7), abbiamo la formula

$$f_n^{(m)} = \sum_{\substack{\rho \in \mathbb{N}_0^m \\ |\rho| = n}} \sum_{k \geq 0} \sum_{i=0}^k \binom{k}{i} \binom{k-i}{r_1} \cdots \binom{k-i}{r_m} (-1)^i. \quad (7.19)$$

Chiaramente $f_n^{(m)} = 0$ ogni volta che $n < m$. Consideriamo ora il caso $m = n$. In questo caso abbiamo solo il vettore $\rho = (1, \dots, 1)$ e l'identità (7.19) diventa

$$f_n^{(n)} = \sum_{k \geq 0} \left[\sum_{i=0}^k \binom{k}{i} (k-i)^n (-1)^i \right].$$

La somma tra le parentesi è ben conosciuta e fornisce il numero di funzioni suriettive da un insieme di dimensione n ad uno di dimensione k . Tale numero può essere espresso in termini di numeri di Stirling di seconda specie, più precisamente è uguale a $\left\{ \begin{smallmatrix} n \\ k \end{smallmatrix} \right\} k!$. Allora

$$f_n^{(n)} = \sum_{k=0}^n \left\{ \begin{smallmatrix} n \\ k \end{smallmatrix} \right\} k!.$$

Anche questa somma è ben conosciuta e fornisce il numero t_n di tutte le sistemazioni preferenziali sull'insieme di dimensione n ed è la sequenza A000670 in [S]. Così, in conclusione, abbiamo l'importante risultato che $f_n^{(n)} = t_n$.

Questo risultato può essere generalizzato. Al contrario nella formula per $f_{n+1}^{(n)}$ abbiamo solo gli n vettori $\rho = (1, \dots, 1, 2, 1, \dots, 1)$. Inoltre (7.19) diventa

$$f_{n+1}^{(n)} = \frac{n}{2} \sum_{k \geq 0} \sum_{i=0}^k \binom{k}{i} (k-i)^n (k-i+1) (-1)^i = \frac{n}{2} \left(\sum_{k \geq 0} \left\{ \begin{smallmatrix} n+1 \\ k \end{smallmatrix} \right\} k! - \sum_{k \geq 0} \left\{ \begin{smallmatrix} n \\ k \end{smallmatrix} \right\} k! \right)$$

cioè

$$f_{n+1}^{(n)} = \frac{n}{2} (t_{n+1} + t_n).$$

In modo simile, quando consideriamo $f_{n+2}^{(n)}$, abbiamo solo gli n vettori $\rho = (1, \dots, 1, 3, 1, \dots, 1)$ e gli $\binom{n}{2}$ vettori $\rho = (1, \dots, 1, 2, 1, \dots, 1, 2, 1, \dots, 1)$. In più (7.19), dopo la semplificazione, diventa

$$f_{n+2}^{(n)} = \frac{n}{24} [(3n+1)t_{n+2} + 6(n+1)t_{n+1} + (3n+5)t_n].$$

Tutti questi risultati suggeriscono che esistono polinomi $p_i^{(k)}(x)$ tali che

$$f_{n+k}^{(n)} = \sum_{i=0}^k p_i^{(k)}(n) t_{n+i}.$$

m / n	0	1	2	3	4	5	6	7	8	9	10	11
0	1	0	0	0	0	0	0	0	0	0	0	0
1	1	1	2	2	4	4	8	8	16	16	32	32
2	1	2	5	8	18	28	62	96	212	328	724	1120
3	1	3	9	19	48	96	236	468	1146	2270	5556	11004
4	1	4	14	36	101	240	648	1520	4082	9560	25660	60088
5	1	5	20	60	185	501	1470	3910	11390	30230	88002	233530
6	1	6	27	92	309	930	2939	8640	27048	79280	247968	726672

 Tabella 7.1: Tabella dei numeri $p_n^{(m)}$.

7.7 m -composizioni con righe palindrome

Nella letteratura sono state studiate spesso le composizioni palindrome, cioè composizioni i cui elementi sono gli stessi nell'ordine dato o in quello inverso. Noi generalizziamo questa definizione alle m -composizioni considerando tutte le m -composizioni con righe palindrome.

Chiaramente ogni m -composizione con righe palindrome ha la forma $[M|M_s]$ quando la sua lunghezza è pari ed ha forma $[M|\mathbf{v}|M_s]$ quando la sua lunghezza è dispari, dove M è una m -composizione arbitraria, M_s è la m -composizione speculare ottenuta da M invertendo ogni riga e \mathbf{v} è un vettore colonna arbitrario. Inoltre la funzione generatrice per le m -composizioni con righe palindrome è data da

$$p^{(m)}(x) = \sum_{n \geq 0} p_n^{(m)} x^n = c^{(m)}(x^2) + \left(\frac{1}{(1-x)^m} - 1 \right) c^{(m)}(x^2)$$

cioè

$$p^{(m)}(x) = \frac{1}{(1-x)^m} c^{(m)}(x^2) = \frac{(1+x)^m}{2(1-x^2)^m - 1}.$$

Da questa identità segue immediatamente che

$$p_n^{(m)} = \sum_{k=0}^{\lfloor n/2 \rfloor} \binom{m}{n-2k} c_k^{(m)}.$$

Sia ora $q_n^{(m)}$ il numero di tutte le m -composizioni di n con righe palindrome. Con argomentazioni del tutto simili a queste utilizzate nel caso delle m -composizioni ordinarie abbiamo che

$$p_n^{(m)} = \sum_{k=0}^m \binom{m}{k} q_n^{(k)}$$

$$q_n^{(m)} = \sum_{k=0}^m \binom{m}{k} (-1)^{m-k} p_n^{(k)}.$$

Da notare che quando $n = m$ c'è già una n -composizione con righe palindrome, data dal vettore colonna con tutti gli elementi uguali a 1. Inoltre $q_n^{(n)} = 1$.

7.8 m -composizioni del tipo Carlitz

Una m -composizione è detta del tipo Carlitz se non contiene due colonne adiacenti uguali. Quando $m = 1$ otteniamo le classiche composizioni di Carlitz [C].

Come nel paragrafo 7.2, anche una m -composizione del tipo Carlitz può essere vista come una parola sull'alfabeto infinito $\mathcal{A}^{(m)} = \{a_\mu : \mu \in \mathcal{M}_{\neq 0}^{(m)}\}$. Sia \mathcal{Z} l'insieme di tutte le parole corrispondenti alle m -composizioni del tipo Carlitz e sia \mathcal{Z}_μ il sottoinsieme di \mathcal{Z} formato esattamente dalle parole che finiscono con a_μ , per ogni $\mu \in \mathcal{M}_{\neq 0}^{(m)}$. Allora segue immediatamente che

$$\mathcal{Z} = 1 + \sum_{\mu \in \mathcal{M}_{\neq 0}^{(m)}} \mathcal{Z}_\mu \quad \text{e} \quad \mathcal{Z}_\mu = (\mathcal{Z} - \mathcal{Z}_\mu) a_\mu \quad \forall \mu \in \mathcal{M}_{\neq 0}^{(m)}.$$

Adesso per ottenere le funzioni generatrici associate ai linguaggi \mathcal{Z} and \mathcal{Z}_μ è sufficiente sostituire ogni simbolo a_μ con una indeterminata x_μ . Quindi otteniamo il sistema lineare

$$z(X) = 1 + \sum_{\mu \in \mathcal{M}_{\neq 0}^{(m)}} z_\mu(X) \quad \text{e} \quad z_\mu(X) = (z(X) - z_\mu(X)) x_\mu \quad \forall \mu \in \mathcal{M}_{\neq 0}^{(m)}$$

da cui

$$z_\mu(X) = \frac{x_\mu}{1 + x_\mu} z(X)$$

e quindi

$$z(X) = \frac{1}{1 - \sum_{\mu \in \mathcal{M}_{\neq 0}^{(m)}} \frac{x_\mu}{1 + x_\mu}}.$$

Mettendo $x_\mu = x^{\text{ord}(\mu)}$, otteniamo la funzione generatrice per i coefficienti $z_n^{(m)}$ che contano tutte le m -composizioni di n del tipo Carlitz. In particolare abbiamo

$$z^{(m)}(x) = \sum_{n \geq 0} z_n^{(m)} x^n = \frac{1}{1 - \sum_{k \geq 1} \binom{m}{k} \frac{x^k}{1 + x^k}}. \quad (7.20)$$

n / m	0	1	2	3	4	5	6
0	1	1	1	1	1	1	1
1	0	1	2	3	4	5	6
2	0	1	5	12	22	35	51
3	0	3	18	58	136	265	458
4	0	4	53	255	793	1925	3984
5	0	7	162	1137	4660	14056	34788
6	0	14	505	5095	27434	102720	303902
7	0	23	1548	22749	161308	750255	2654064
8	0	39	4756	101625	948641	5480235	23179743
9	0	71	14650	454116	5579224	40031030	202445610
10	0	124	45065	2028939	32811986	292408771	1768099107
11	0	214	138622	9065145	192971168	2135917405	15442052496

 Tabella 7.2: Tabella dei numeri $z_n^{(m)}$.

Per $m = 1$ otteniamo di nuovo la funzione generatrice per le composizioni ordinarie di Carlitz.

La sequenza $z_n^{(1)}$ appare in [S] come la sequenza #A003242, mentre per $m \geq 2$ le sequenze corrispondenti sono assenti.

Dalla funzione (7.20) è possibile ottenere una formula esplicita per i coefficienti $z_n^{(m)}$. Inoltre

$$\begin{aligned}
 z^{(m)}(x) &= \sum_{k \geq 0} \left(\sum_{n \geq 1} \binom{m}{n} \frac{x^n}{1+x^n} \right)^k \\
 &= \sum_{k \geq 0} \sum_{a_1 \geq 1} \binom{m}{a_1} \frac{x^{a_1}}{1+x^{a_1}} \cdots \sum_{a_k \geq 1} \binom{m}{a_k} \frac{x^{a_k}}{1+x^{a_k}} \\
 &= \sum_{k \geq 0} \sum_{a_1, \dots, a_k \geq 1} \binom{m}{a_1} \cdots \binom{m}{a_k} \frac{x^{a_1}}{1+x^{a_1}} \cdots \frac{x^{a_k}}{1+x^{a_k}} \\
 &= \sum_{k \geq 0} \sum_{a_1, \dots, a_k \geq 1} \binom{m}{a_1} \cdots \binom{m}{a_k} \sum_{b_1 \geq 1} (-1)^{b_1-1} x^{a_1 b_1} \cdots \sum_{b_k \geq 1} (-1)^{b_k-1} x^{a_k b_k} \\
 &= \sum_{k \geq 0} \sum_{\substack{a_1, \dots, a_k \geq 1 \\ b_1, \dots, b_k \geq 1}} \binom{m}{a_1} \cdots \binom{m}{a_k} (-1)^{b_1 + \dots + b_k - k} x^{a_1 b_1 + \dots + a_k b_k}.
 \end{aligned}$$

Quindi

$$z^{(m)}(x) = \sum_{n \geq 0} \left(\sum_{k \geq 0} \sum_{\substack{\alpha, \beta \in \mathbb{N}_0^k \\ \alpha \cdot \beta = n}} \binom{m}{\alpha} (-1)^{|\beta| - k} \right) x^n$$

dove se $\alpha = (a_1, \dots, a_k)$ e $\beta = (b_1, \dots, b_k)$ allora $\alpha \cdot \beta = a_1 b_1 + \dots + a_k b_k$, $|\beta| = b_1 + \dots + b_k$ e

$$\binom{m}{\alpha} = \binom{m}{a_1} \dots \binom{m}{a_k}.$$

Così, infine, abbiamo la seguente espressione

$$z_n^{(m)} = \sum_{k \geq 0} \sum_{\substack{\alpha, \beta \in \mathbb{N}_0^k \\ \alpha \cdot \beta = n}} \binom{m}{\alpha} (-1)^{|\beta| - k}.$$

Sia ora $g_n^{(m)}$ il numero di tutte le m -composizioni di n del tipo Carlitz senza righe nulle. Con argomenti del tutto simili a quelli utilizzati nel caso delle m -composizioni ordinarie abbiamo che

$$z_n^{(m)} = \sum_{k=0}^m \binom{m}{k} g_n^{(k)}$$

$$g_n^{(m)} = \sum_{k=0}^m \binom{m}{k} (-1)^{m-k} z_n^{(k)}.$$

7.9 Matrici generate dalle m -composizioni

Consideriamo la matrice infinita $C = [c_n^{(m)}]_{m, n \geq 0}$ generata dalle m -composizioni. Questa ammette una decomposizione LU su \mathbb{N} , cioè esiste una matrice triangolare inferiore L ed una matrice triangolare superiore U , entrambe con elementi non negativi, tali che $C = LU$. Inoltre tale decomposizione è equivalente all'identità (7.17) che implica che L è la matrice binomiale ordinaria

$B = [(\binom{m}{n})]_{m,n \geq 0}$ ed U è la matrice

$$F = [f_n^{(m)}]_{m,n \geq 0} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 4 & 8 & 16 & 32 & 64 & 128 & 256 \\ 0 & 0 & 3 & 16 & 66 & 248 & 892 & 3136 & 10888 & 37536 \\ 0 & 0 & 0 & 13 & 132 & 924 & 5546 & 30720 & 162396 & 834004 \\ 0 & 0 & 0 & 0 & 75 & 1232 & 13064 & 114032 & 893490 & 6550112 \\ 0 & 0 & 0 & 0 & 0 & 541 & 13060 & 195020 & 2327960 & 24418640 \\ 0 & 0 & 0 & 0 & 0 & 0 & 4683 & 155928 & 3116220 & 48697048 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 47293 & 2075948 & 53476920 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 545835 & 30532384 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 7087261 \\ \dots & & & & & & & & & \end{bmatrix}$$

generata dalle m -composizioni senza righe nulle.

Tale decomposizione si mantiene anche quando consideriamo le matrici finite $C_k = [c_n^{(m)}]_{m,n \geq 0}^k$, $B_k = [(\binom{m}{n})]_{m,n \geq 0}^k$ e $F_k = [f_n^{(m)}]_{m,n \geq 0}^k$, cioè $C_k = B_k F_k$. Inoltre possiamo facilmente calcolare il determinante di C_k . Poiché sia B_k che F_k sono triangolari, per il teorema di Binet $\det C_k = \det B_k \det F_k$, cioè

$$\det C_k = t_0 t_1 \cdots t_k.$$

Questi risultati si mantengono anche per le m -composizioni del tipo Carlitz. Inoltre abbiamo $Z = BG$ dove $Z = [z_n^{(m)}]_{m,n \geq 0}$ e

$$G = [g_n^{(m)}]_{m,n \geq 0} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 3 & 4 & 7 & 14 & 23 & 39 & 71 \\ 0 & 0 & 3 & 12 & 45 & 148 & 477 & 1502 & 4678 & 14508 \\ 0 & 0 & 0 & 13 & 108 & 672 & 3622 & 18174 & 87474 & 410379 \\ 0 & 0 & 0 & 0 & 75 & 1056 & 10028 & 79508 & 570521 & 3850376 \\ 0 & 0 & 0 & 0 & 0 & 541 & 11520 & 155840 & 1705915 & 16529925 \\ 0 & 0 & 0 & 0 & 0 & 0 & 4683 & 140256 & 2566554 & 37084794 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 47293 & 1894032 & 45082170 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 545835 & 28159872 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 7087261 \\ \dots & & & & & & & & & \end{bmatrix}$$

e $Z_k = B_k G_k$ dove $Z_k = [z_n^{(m)}]_{m,n=0}^k$ e $G_k = [g_n^{(m)}]_{m,n=0}^k$. Si può ipotizzare che

$$\det Z_k = t_0 t_1 \cdots t_k.$$

Dobbiamo dimostrare che $g_n^{(m)} = t_n$.

Infine un risultato simile si mantiene anche per le m -composizioni con righe palindrome, cioè $P = BQ$ dove $P = [p_n^{(m)}]_{m,n \geq 0}$,

$$Q = [q_n^{(m)}]_{m,n \geq 0} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 2 & 4 & 4 & 8 & 8 & 16 & 16 \\ 0 & 0 & 1 & 4 & 10 & 20 & 46 & 80 & 180 & 296 \\ 0 & 0 & 0 & 1 & 6 & 24 & 74 & 204 & 558 & 1334 \\ 0 & 0 & 0 & 0 & 1 & 8 & 44 & 192 & 706 & 2384 \\ 0 & 0 & 0 & 0 & 0 & 1 & 10 & 70 & 400 & 1930 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 12 & 102 & 724 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 14 & 140 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 16 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \dots & & & & & & & & & \end{bmatrix}$$

e $P_k = B_k Q_k$, dove $P_k = [p_n^{(m)}]_{m,n=0}^k$ e $Q_k = [q_n^{(m)}]_{m,n=0}^k$. Comunque, poiché $q_n^{(n)} = 1$, abbiamo

$$\det P_k = 1.$$

7.10 Generazione delle m -composizioni

Nell'ultima parte di questo capitolo affrontiamo il problema della generazione delle m -composizioni, fornendo un algoritmo che gode della proprietà CAT. Tenendo in considerazione le proprietà delle m -composizioni e soprattutto la loro caratterizzazione, come illustrata nel paragrafo 7.3, per comodità introduciamo due alfabeti $\Sigma^1 = \{1, 2, \dots, m\}$, $\Sigma^2 = \{\bar{1}, \bar{2}, \dots, \bar{m}\}$, $\Sigma = \Sigma^1 \cup \Sigma^2$. Possiamo rileggere le parole definite in 7.3 come parole sull'alfabeto Σ , ottenute come sequenze di indici dei simboli a_i e b_j come segue: si associa ad a_i la cifra $i \in \Sigma^1$, ed a b_j la cifra $j \in \Sigma^2$. Definiamo questa rappresentazione una *rappresentazione a indici* di M . Per esempio, la 3-composizione di 6

$$M = \begin{pmatrix} 1 & 0 \\ 2 & 2 \\ 0 & 1 \end{pmatrix}$$

equivale alla parola $a_1 a_2 a_2 b_2 a_2 a_3$ ed ora anche a $122\bar{2}23$.

Indichiamo dunque con \mathcal{L}_n l'insieme delle parole $w = w_1 w_2 \dots w_n$ rappresentanti le m -composizioni di n .

\mathcal{L}_n è definito da:

$$\mathcal{L}_1 = 1, 2, \dots, m$$

$\mathcal{L}_n = \{wi | w = w_1w_2\dots w_{n-1} \in \mathcal{L}_{n-1}, i \in \Sigma^1, i \geq w_{n-1}\} \cup \{wi | w = w_1w_2\dots w_{n-1} \in \mathcal{L}_{n-1}, i \in \Sigma^2\}$ ovvero, gli elementi di \mathcal{L}_n sono ottenuti concatenando agli elementi di \mathcal{L}_{n-1}

1. un simbolo in Σ^1 che sia maggiore o uguale all'ultimo simbolo di w ;
2. un simbolo in Σ^2 .

Quindi i primi insiemi per $m = 2$ sono

$$\mathcal{L}_1 = \{1, 2\}$$

$$\mathcal{L}_2 = \{11, 12, 22, 1\bar{1}, 1\bar{2}, 2\bar{1}, 2\bar{2}\}$$

$$\mathcal{L}_3 = \{111, 112, 122, 222, 1\bar{1}1, 1\bar{1}2, 1\bar{2}2, 2\bar{1}1, 2\bar{1}2, 2\bar{2}2, 11\bar{1}, 11\bar{2}, 12\bar{1}, 12\bar{2}, 22\bar{1}, 22\bar{2}, 1\bar{1}\bar{1}, 1\bar{1}\bar{2}, 1\bar{2}\bar{1}, 1\bar{2}\bar{2}, 2\bar{1}\bar{1}, 2\bar{1}\bar{2}, 2\bar{2}\bar{1}, 2\bar{2}\bar{2}\}.$$

Proposizione 7.1 \mathcal{L}_n contiene tutte le m -composizioni di n senza ripetizioni.

Dim. La dimostrazione si fa per induzione su n .

\mathcal{L}_1 contiene per costruzione tutte le m -composizioni di 1 senza ripetizioni. Supponiamo che \mathcal{L}_n contenga tutte le m -composizioni di n senza ripetizioni. Le operazioni possibili per ottenere da una m -composizione μ di n , rappresentata dalla parola $w = w_1w_2\dots w_n$, una m -composizione di $n + 1$ sono:

- (i) incrementare di 1 un elemento;
- (ii) aggiungere una colonna con un solo elemento uguale a 1 in riga k ($1 \leq k \leq m$) a sinistra di una colonna in μ ;
- (iii) aggiungere una colonna con un solo elemento uguale a 1 in riga k ($1 \leq k \leq m$) a destra dell'ultima colonna in μ .

Si può facilmente notare che le operazioni (i) e (ii) consistono nell'aggiungere un carattere, diciamo ω , fra due caratteri di w o a sinistra di w_1 , ottenendo quindi una parola del tipo

$$\alpha^1 = w_1w_2\dots w_i\omega w_{i+1}\dots w_{n-1}w_n$$

oppure del tipo

$$\alpha^2 = \omega w_1w_2\dots w_{n-1}w_n.$$

Le parole

$$\beta^1 = w_1w_2\dots w_i\omega w_{i+1}\dots w_{n-1} \text{ e } \beta^2 = \omega w_1w_2\dots w_{n-1}$$

sono m -composizioni di n e quindi appartengono, per ipotesi induttiva, a \mathcal{L}_n e poichè se $w_n \in \Sigma^1$ allora $w_n \geq w_{n-1}$, si ha che $\alpha^1 = \beta^1 w_n$ e $\alpha^3 = \beta^2 w_n$ appartengono a \mathcal{L}_{n+1} . Le parole che si ottengono con l'operazione iii) sono del tipo $\alpha^2 = w_1 w_2 \dots w_{n-1} w_n \bar{w}$ con $\bar{w} \in \Sigma^2$, $1 \leq \bar{w} \leq m$, e poichè $\beta^3 = w_1 w_2 \dots w_n$ appartiene per ipotesi induttiva a \mathcal{L}_n allora α^3 appartiene a \mathcal{L}_{n+1} . Inoltre, poichè le parole che si ottengono da una singola parola in \mathcal{L}_n applicando i passi 1. e 2. sono tutte diverse, se \mathcal{L}_n contiene le m -composizioni di n senza ripetizione, allora anche le parole in \mathcal{L}_{n+1} sono uniche.

□

7.11 Codice Gray per le m -composizioni

E' interessante vedere come possiamo generare tutte le m -composizioni di n partendo dalla prima. Per fare questo rileggiamo le m -composizioni come codici Gray. Con il termine *codice* si indica un insieme X tale che ogni parola in X^+ può essere scritta unicamente come prodotto di parole in X , cioè ha una fattorizzazione unica in parole in X [BP]. In tale definizione rientrano anche i codici Gray, dove per codice Gray si intende la codifica di numeri in modo che numeri consecutivi differiscano tra loro esattamente di una cifra. Vogliamo quindi generare le m -composizioni in modo che due successive differiscano solo per una cifra.

Consideriamo una m -composizione di n come parola $w \in \mathcal{L}_n$. Indichiamo come successori di w_n (cioè l'ultima cifra di w) le cifre che possiamo concatenare a w per ottenere una m -composizione di $n+1$ secondo l'algoritmo di generazione. Si osservi che se $w_n = i$ (con $i \in \Sigma^1 \cup \Sigma^2$) i successori sono:

$$\langle i, i+1, \dots, m, \bar{1}, \bar{2}, \dots, \bar{m} \rangle .$$

Tutte le liste dei successori contengono m e \bar{m} ; possiamo quindi fare in modo che tutte le liste di successori inizino per m e finiscano per \bar{m} o viceversa. Indichiamo con $s(i, m)$ e $s(i, \bar{m})$ queste due liste che definiamo così:

$$\begin{cases} s(i, m) &= \langle m, m-1, \dots, i+1, i, \bar{1}, \bar{2}, \dots, \bar{m} \rangle \\ s(i, \bar{m}) &= \langle \bar{m}, \bar{m}-1, \dots, \bar{2}, \bar{1}, i, i+1, \dots, m \rangle \end{cases}$$

Pertanto la lista dei successori di una parola termina sempre per m o \bar{m} (alternativamente).

7.11.1 Definizione del codice Gray per le m -composizioni

La definizione che qui diamo del codice Gray per le m -composizioni è un'applicazione dei concetti espressi in [BGPP]. La definizione ricorsiva

del codice Gray si basa sulla generazione di liste di lunghezza crescente. Utilizziamo la seguente notazione.

- \mathcal{L}_k = lista delle codifiche di tutti gli oggetti di dimensione k ;
- l_i^k = i -esimo elemento di \mathcal{L}_k ;
- $|\mathcal{L}_k|$ = cardinalità di \mathcal{L}_k ;
- se x è una sequenza di caratteri, \overrightarrow{x} indica il carattere più a destra di x ;
- data una lista L :
 - $\text{first}(L)$ = primo elemento della lista L ;
 - $\text{last}(L)$ = ultimo elemento della lista L ;
 - $x \circ L$ = lista ottenuta concatenando ad x tutti gli elementi di L .

Costruzione della lista L_n

La lista di tutti gli elementi di lunghezza n è ottenuta concatenando $|\mathcal{L}_{n-1}|$ liste L_n^i definite da

$$\begin{cases} L_n^1 = l_1^{n-1} \circ s(\overrightarrow{l_1^{n-1}}, \overrightarrow{\text{last}(L_n^{i-1})}) \\ L_n^i = l_i^{n-1} \circ s(\overrightarrow{l_i^{n-1}}, \overrightarrow{\text{last}(L_n^{i-1})}) \quad i > 1 \end{cases}$$

Quindi, posto $|\mathcal{L}_{n-1}| = M$, si ha

$$\begin{cases} \mathcal{L}_1 = \langle 1, 2, \dots, m \rangle \\ \mathcal{L}_n = \Theta_{i=1}^M \mathcal{L}_n^i \quad n > 1 \end{cases}$$

Esempio 7.1 Vediamo come vengono generate le 2-composizioni di 3. In questo caso una parola termina sempre con un elemento $i \in \{1, 2\} \cup \{\overline{1}, \overline{2}\}$, troviamo tutti gli insiemi $s(i, j)$ con $i, j \in \{1, 2\} \cup \{\overline{1}, \overline{2}\}$:

$$\begin{aligned} s(1, 2) &= s(\overline{1}, 2) = \langle 2, 1, \overline{1}, \overline{2} \rangle \\ s(1, \overline{2}) &= s(\overline{1}, \overline{2}) = \langle \overline{2}, \overline{1}, 1, 2 \rangle \\ s(2, 2) &= s(\overline{2}, 2) = \langle 2, \overline{1}, \overline{2} \rangle \\ s(2, \overline{2}) &= s(\overline{2}, \overline{2}) = \langle \overline{2}, \overline{1}, 2 \rangle \end{aligned}$$

L'insieme degli oggetti di dimensione 1 è

$$\mathcal{L}_1 = \langle 1, 2 \rangle,$$

costruiamo quindi le singole liste L_2^i che, insieme, costituiranno \mathcal{L}_2 :

$$\begin{aligned} L_2^1 &= 1 \circ s(1, 2) = \langle 12, 11, 1\bar{1}, 1\bar{2} \rangle \\ L_2^2 &= 2 \circ s(2, \bar{2}) = \langle 2\bar{2}, 2\bar{1}, 22 \rangle \end{aligned}$$

$$\Rightarrow \mathcal{L}_2 = \langle 12, 11, 1\bar{1}, 1\bar{2}, 2\bar{2}, 2\bar{1}, 22 \rangle$$

quindi:

$$\begin{aligned} L_3^1 &= 12 \circ s(2, 2) = \langle 122, 12\bar{1}, 12\bar{2} \rangle \\ L_3^2 &= 11 \circ s(1, \bar{2}) = \langle 11\bar{2}, 11\bar{1}, 111, 112 \rangle \\ L_3^3 &= 1\bar{1} \circ s(\bar{1}, 2) = \langle 1\bar{1}\bar{2}, 1\bar{1}1, 1\bar{1}\bar{1}, 1\bar{1}\bar{2} \rangle \\ L_3^4 &= 1\bar{2} \circ s(\bar{2}, \bar{2}) = \langle 1\bar{2}2, 1\bar{2}\bar{1}, 1\bar{2}\bar{2} \rangle \\ L_3^5 &= 22 \circ s(2, 2) = \langle 122, 12\bar{1}, 12\bar{2} \rangle \\ L_3^6 &= 21 \circ s(2, 2) = \langle 122, 12\bar{1}, 12\bar{2} \rangle \\ L_3^7 &= 22 \circ s(2, 2) = \langle 122, 12\bar{1}, 12\bar{2} \rangle \end{aligned}$$

$$\Rightarrow \mathcal{L}_3 = \langle 122, 12\bar{1}, 12\bar{2}, 11\bar{2}, 11\bar{1}, 111, 112, 1\bar{1}\bar{2}, 1\bar{1}1, 1\bar{1}\bar{1}, 1\bar{1}\bar{2}, 1\bar{2}2, 1\bar{2}\bar{1}, 1\bar{2}\bar{2}, 222, 22\bar{1}, 22\bar{2} \rangle$$

Teorema 7.1 *Due parole consecutive della lista \mathcal{L}_n differiscono solo per un carattere.*

Dim. La dimostrazione si fa per induzione su n .

Per $n = 1$ si ha che \mathcal{L}_1 è costituito, per definizione, da parole fatte da un solo carattere quindi il teorema vale. Supponiamo che due elementi consecutivi di \mathcal{L}_{n-1} differiscano solo per un carattere e dimostriamo che questo vale anche per \mathcal{L}_n . Gli elementi delle liste L_n^i differiscono, per costruzione, di un solo carattere, quindi è sufficiente dimostrare che anche $last(L_n^i)$ e $first(L_n^{i+1})$, con $1 \leq i \leq |\mathcal{L}_{n-1}| - 1$, differiscono per un solo carattere. Dalla definizione delle liste L_n^i si ha, assumendo che l'ultimo elemento della lista $s(\overrightarrow{l_i^{n-1}}, \overrightarrow{last(L_n^{i-1})})$ sia J :

$$last(L_n^i) = l_i^{n-1} J.$$

D'altra parte:

$$L_n^{i+1} = l_{i+1}^{n-1} \circ s(\overrightarrow{l_{i+1}^{n-1}}, \overrightarrow{last(L_n^i)}) = l_{i+1}^{n-1} \circ s(\overrightarrow{l_{i+1}^{n-1}}, J).$$

Per definizione il primo elemento della lista $s(i, k)$, è sempre k , con

$k \in \{m, \bar{m}\}$, quindi

$$first(L_n^{i+1}) = l_{i+1}^{n-1} J$$

e poichè, per ipotesi induttiva, l_i^{n-1} e l_{i+1}^{n-1} differiscono solo di un carattere, allora anche $last(L_n^i)$ e $first(L_n^{i+1})$ differiscono di un solo carattere.

□

7.12 Generazione delle m -partizioni di interi

Nel paragrafo 7.3 abbiamo considerato i bargraph etichettati in riferimento alle m -composizioni. Per esempio, se consideriamo la seguente 3-composizione di 9:

$$M = \begin{bmatrix} 1 & 0 & 1 \\ 2 & 1 & 1 \\ 0 & 3 & 0 \end{bmatrix}$$

può essere rappresentata, in termini di bargraph etichettati, in questo modo:

	3	
2	3	
2	3	2
1	2	1

ed in termini di parole: $a_1a_2a_2b_2a_3a_3a_3b_1a_2$ oppure $122\bar{2}333\bar{1}2$. Possiamo notare come una m -composizione qualunque può corrispondere ad un bargraph con colonne di altezza variabile. Consideriamo ora le partizioni, che sono un sottoinsieme delle composizioni. Infatti, sia $\lambda = (\lambda_1, \dots, \lambda_k)$ una sequenza di interi, si definisca *peso* di λ la quantità $\lambda_1 + \dots + \lambda_k$, se tale sequenza di peso n ha tutte parti non negative, si dice che è una composizione di n in k parti non negative, se, inoltre, λ è una sequenza non crescente, si dice che è una *partizione* di n in k parti.

Questa differenza tra composizioni e partizioni di interi riletta sui bargraph si traduce nel fatto che se una m -composizione è una partizione, allora il bargraph che si ottiene è un bargraph a scale con colonne di altezza non crescente. Quindi vuol dire che, se la somma degli interi presenti nella i -esima colonna della matrice è s_i , $\forall j > i$ si ha $s_j \geq s_i$. Per cui se, invece delle composizioni consideriamo le partizioni, possiamo “estendere” il concetto di m -composizioni e studiare le m -partizioni.

Esempio 7.2 Sia

$$M = \begin{bmatrix} 2 & 1 & 0 & 1 \\ 1 & 2 & 1 & 0 \\ 1 & 0 & 2 & 0 \end{bmatrix}$$

una 3-partizione di 11, la parola ad essa associata è $a_1a_1a_2a_3b_1a_2a_2b_2a_3a_3b_1$, o, in modo equivalente, $1123\bar{1}22\bar{2}33\bar{1}$. Il bargraph seguente la rappresenta

secondo l'associazione m -composizione-bargraph che abbiamo definito nel paragrafo 7.3.2 e, come possiamo facilmente notare, $(s_1 = 4) > (s_2 = 3) = (s_3 = 3) > (s_4 = 1)$ (Figura 7.5).

3			
2	2	3	
1	2	3	
1	1	2	1

Figura 7.5: Bargraph associato alla 3-partizione di 11 $w = 1123\bar{1}22\bar{2}33\bar{1}$.

Abbiamo fornito, nelle pagine precedenti, un metodo per generare tutte e sole le m -composizioni di n a partire da quelle di $(n - 1)$, con m fissato. Aggiungendo un piccolo controllo sulla procedura di generazione, possiamo generare anche le m -partizioni di n . E' intuitivo che la differenza di altezza tra le ultime due colonne del bargraph corrisponde al numero di simboli di Σ^1 compresi tra gli ultimi due simboli di Σ^2 nella parola. In riferimento all'esempio 7.2, avevamo quattro colonne con altezza h_i (con $i \in 1, 2, 3, 4$) in cui la parola corrispondente è $w = 1123\bar{1}22\bar{2}33\bar{1}$ dove $w_8 = \bar{2}$ e $w_1 = \bar{1}$ sono gli ultimi due simboli appartenenti a Σ_2 in w ed indicano gli ultimi due cambi di colonna. Il numero di simboli che stanno tra questi ultimi due cambi di colonna è $h_3 - h_4 = 2$.

Sia k il numero di colonne della matrice o, in modo equivalente, il numero di colonne del bargraph. Sia s_i la somma degli interi della i -esima colonna della matrice, e sia h_i l'altezza della i -esima colonna del bargraph. Ovviamente si ha $s_i = h_i$. Sia d_1 il numero di simboli appartenenti all'alfabeto Σ^1 presenti tra gli ultimi due elementi di Σ^2 di $w \in \mathcal{L}_n$, e d_2 il numero di simboli dell'alfabeto Σ^1 che si trovano dopo l'ultimo simbolo di Σ^2 in w . Sempre tenendo conto della relazione tra matrice e bargraph, si ha:

$$d_1 = h_{k-1} - h_k \quad \text{e} \quad d_2 = h_k - 1.$$

Considerando un bargraph con n celle, è ovvio osservare che, per generare i bargraph con $(n + 1)$ celle sempre con altezza delle colonne non crescente, possiamo aggiungere una cella sopra o accanto all'ultima colonna (cioè la colonna più a destra). Naturalmente dovremo tenere conto del fatto che tale colonna non può superare in altezza quella precedente e che l'etichetta della nuova cella dovrà essere maggiore o uguale di quella della cella in testa all'ultima colonna e minore o uguale a m . Inoltre potrà sempre essere aggiunta una colonna di altezza uno con una qualsiasi etichetta a destra dell'ultima colonna del bargraph.

Possiamo affermare che l'algoritmo di generazione del linguaggio descritto dalle m -partizioni è in sostanza lo stesso del linguaggio definito dalle m -composizioni, con un controllo in più. Diamo quindi un algoritmo per la generazione delle m -partizioni di $(n + 1)$ a partire da quelle di n . Siano

$$w = w_1 \dots w_{k-j} \dots w_k \dots w_n \quad \text{con} \quad w_{k-j}, w_k \in \Sigma^2 \text{ e } w_{k+1}, \dots, w_n \in \Sigma^1$$

e

$$d_1 = j + 1 = |w_{k-d_1} \dots w_{k-1}| \quad , \quad d_2 = |w_{k+1} \dots w_n|.$$

Per facilitare i controlli consideriamo una “nuova parola” $w' = d_1 d_2 w_1 \dots w_n$ di lunghezza $(n + 2)$. Infatti, in questo modo, ad ogni passo dovremo solo controllare e poi aggiornare, le prime due componenti di w' , invece di contare i simboli. Dunque:

- se $d_1 > d_2$ allora
 1. $w = w_1 \dots w_n i$ con $i \in \Sigma^1$ e $i \geq w_n$ e vengono aggiornati i due contatori: $d_1 = d_1 - 1$ e $d_2 = d_2 + 1$;
 2. $w = w_1 \dots w_n i$ con $i \in \Sigma^2$ e vengono aggiornati i contatori: $d_1 = d_2$ e $d_2 = 0$;
- se invece $d_1 = d_2$ allora possiamo aggiungere solo come nel caso 2.

Vista la somiglianza tra le m -composizioni e le m -partizioni, ci sembra ragionevole ipotizzare l'esistenza di un codice Gray anche per le m -partizioni e quindi studiare un possibile confronto anche da questo punto di vista.

Bibliografia

- [A] G. E. Andrews, *The Theory of Partitions*, Encyclopedia of Mathematics and its Applications, Addison-Wesley Publishing Company, Reading, Massachusetts (1976).
- [A75] G. E. Andrews, *The Theory of Compositions, II: Simon Newcom's problem*, Utilitas Math. 7 (1975), 33-54.
- [A76a] G. E. Andrews, *The Theory of Compositions, I: The ordered factorizations of n and a conjecture of C. Long*, Canadian Math. Bull. 18 (1976), 479-484.
- [A76b] G. E. Andrews, *The Theory of Compositions, III: The MacMahon formula and the Stanton-Cowan numbers*, Utilitas Math. 9 (1976), 283-290.
- [BS] A. Björner, R. Stanley, *An analogue of Young's lattice for compositions*, accettato (2005).
- [BGPP] A. Bernini, E. Grazzini, E. Pergola, R. Pinzani, *Gray code for Catalan structures*, sottomesso.
- [BMR] M. Bousquet-Mélou, A. Rechnitzer, *The site-perimeter of bargraphs*, Advances in Applied Mathematics 31 (2003), 86-112.
- [BP] J. Berstel, D. Perrin, *Theory of codes*, Mathematics, Academic Press (1985).
- [C] L. Carlitz, *Restricted compositions*, The Fibonacci Quarterly 14 (1976), 254-264.
- [CH] P. Chinn, S. Heubach, *Composizioni of n with no occurrence of k* , Proceedings of the Thirty-Fourth Southeastern International Conference on Combinatorics, Graph Theory and Computing. Congr. Numer. 164 (2003), 33-51.

- [CFMRR] G. Castiglione, A. Frosini, E. Munarini, A. Restivo, S. Rinaldi, *Enumeration of L-convex polyominoes II. Bijection and area*, Proceedings di FPSAC'05, Taormina, #49.
- [Co] L. Comtet, *“Advanced Combinatorics”*, Reidel, Dordrecht-Holland, Boston (1974).
- [HM04] S. Heubach, T. Mansour, *Composizioni of n with parts in a set*, Proceedings of the Thirty-Fifth Southeastern International Conference on Combinatorics, Graph Theory and Computing. Congr. Numer. 168 (2004), 127-143.
- [HM05] S. Heubach, T. Mansour, *Counting Rises, Levels, and Drops in Compositions*, Integers A11 (2005).
- [J] A. Joyal, *Une théorie combinatoire des séries formelles*, Adv. in Math. 42 (1981), 1-82.
- [KP] A. Knopfmacher, H. Prodinger, *On Carlitz composizioni*, European J. Combin. 19 (1998), 579-589.
- [KR] A. Knopfmacher, N. Robbins, *Composizioni with parts constrained by the leading summand*, apparirà su in Ars Combinatorica.
- [MPR] E. Munarini, M. Poneti, S. Rinaldi, *Matrix compositions*, sottomesso a FPSAC 2006.
- [MUV] D. Merlini, F. Uncini, C. Verri, *A unified approach to the study of general and palindromic compositions*, Integers 4 (2004), A23.
- [OP] A. L. Owczarek, T. Prellberg, *Exact Solution of the Discrete (1+1)-dimensional SOS Model with Field and Surface Interactions*, J. Stat. Phys. 70 (1993), 1175-1194.
- [PB] T. Prellberg, R. Brak, *Critical Exponents from Non-Linear Functional Equations for Partially Directed Cluster Models*, J. Stat. Phys. 78 (1995), 701-730.
- [S] N.J.A. Sloane, *“On-Line Encyclopedia of Integer Sequences”*, pubblicato online <http://www.research.att.com/~njas/sequences/>.