JOURNAL OF COMBINATORIAL THEORY, Series A 24, 318–338 (1978)

Formal Languages and Enumeration*

JAY R. GOLDMAN

School of Mathematics, University of Minnesota, Minneapolis, Minnesota 55455

Communicated by the Managing Editors

Received October 3, 1977

DEDICATED TO JOHN RIORDAN ON THE OCCASION OF HIS 75TH BIRTHDAY

We present a theory of generating functions in countably many non-commuting variables. This generalizes the theory of context free languages. Applications are given to compositions of a number, rooted planar tree, dissected polygons, and the theory of simple random walks.

1. INTRODUCTION

This paper develops a theory of generating functions in a countable number of non-commuting variables and their application to combinatorics and probability theory.

The genesis of this theory is the work of Polya [1], on picture writing, who regarded this approach as a mnemonic tool for deriving generating function identities. Independently of this Schützenberger [2, 3, 4] developed a theory of generating functions in a finite number of non-commuting variables to solve problems in context-free languages. Gross and Lentin [5] applied these methods to derive the generating function for triangular dissections of polygons and Kuich [6] applied them to count planted plane trees.

The basic ideas are the following:

(1) Each object to be counted is represented as a word in a free monoid (semigroup) on any number of generators. The words can be thought of as monomials in non-commuting variables. The set of words W corresponding to the set of objects to be counted is called a language.

(2) A formal substitution schema is found that generates the set W. This is a generalization to a countable set of variables of the notion of a context free grammar.

* Research partly supported by Air Force Contract AFOSR-73-2346 and by the NSF under Contract MCS74-06320.

0097-3165/78/0243–0318$02.00/0

(3) To each formal schema is associated a system of formal equations, one of whose solutions in the ring of formal power series over the monoid is the formal power series $S = \sum_{w \in W} w$.

(4) We choose those properties by which we wish to enumerate the objects e.g. planar trees by number of vertices or by edges or height or any combination of these. A homomorphism is then constructed which maps the semigroup into a commutative semigroup such that the induced map on the ring of formal power series maps S onto a generating function F in commuting variables which is the desired enumerator. F will satisfy the induced system of formal equations and this enables us to solve for F. A frequent tool at this stage is the Lagrange inversion formula in infinitely many variables.

This approach is essentially an invariant formalism for generating functions which in many cases allows us to deal with infinitely many variables as easily as with one variable.

Applications of this theory include enumeration of rooted planar trees, polygon dissections of any specification, compositions and partitions of a number, ballot problems and random walk paths. When we add a theory of probabilistic formal schema the combinatorial results can be used to derive functional equations of branching processes and the fluctuation theory of simple random walks. Cori and Richard [7] apply these ideas to the Tutte theory of planar graph enumeration.

The theory of formal schema unifies the classical counting and modern algorithmic approach to enumeration. Given an algorithm written in a specified form (formal schema) we can then derive a generating function from the algorithm.

In this section we give two examples which motivate the main ideas of formal language methods. This is followed by the general theory and applications to trees, polygon dissections, and random walks. In a second paper we shall give further applications including branching processes, functional composition symbols and partitions of a number.

I am grateful to M. Schützenberger, whose work inspired mine, for interesting conversations on the subject. I also thank A. Garsia and S.A. Joni for showing me tricks related to the Lagrange inversion formula. Garsia suggested that every language should be the homomorphic image of an unambiguous language.

An interesting question is whether or not the methods of this paper can be generalized to other species of generating functions along the lines of the Bender–Goldman theory of prefabs [8].

EXAMPLE. *Compositions of a number.* A *composition* of the positive integer n into k *parts* is a sequence $a_1, a_2, \ldots, a_k$ of k positive integers whose sum is n. We often denote the composition by $a_1 + a_2 + a_3 + \cdots + a_k$. Thus e.g.

the compositions of 3 are $1+1+1$, $1+2$, $2+1$, and 3. Compositions differ from partitions in that the order of the parts count. We shall derive generating functions and formulae for the number of compositions of n with k even parts (no restriction on the number of odd parts). In the course of the solution we shall actually set the stage for solving an infinite variety of composition problems.

The first step is to represent compositions as words in a free monoid. We take the set C of generators of the free monoid $M(C)$ to be t, (1), (2), (3),.... The composition $a_1 + a_2 + \cdots + a_k$ will be represented by the word $(a_1)(a_2) \cdots (a_k)\, t$ of $M(B)$. The generators can be thought of as non-commuting variables and the words (elements) of $M(C)$ are monomials in these variables. Any subset of $M(C)$ is called a *formal language* or just a *language*.

Our second step is to give a "formal schema" or algorithm to generate the language CM of those words which represent compositions of any number. The schema is a set of "substitution (production) rules,"

$$S \to (n)\, S, \qquad n = 1, 2, 3, \ldots,$$
$$S \to t,$$

together with an "axiom" or initial symbol S. A word $w \in M(C)$ is derivable from the schema if there exists a sequence $w_0 = S, w_1, w_2, \ldots, w_m = w$, of words of $M(C \cup \{S\})$, such that w_{i+1} is derived from w_i by replacing the leftmost occurrence of S in w_i by the right hand side of any substitution rule in the schema. (The above schema has only one S in any w_i; in general this is not true). We write the derivation

$$w_0 \to w_1 \to w_2 \to \cdots \to w_m .$$

For example the composition $(1)(5)(3)\, t$ of the number 9 is derived as follows

$$S \to (1)\, S \to (1)(5)\, S \to (1)(5)(3)\, S \to (1)(5)(3)\, t.$$

It should be clear that the set of all words derivable from the schema is exactly the language CM of compositions. The word t is taken to represent the empty composition.

We now form the formal sum $\sum_{w \in CM} w$ of all words in our language. This is, in fact, a formal power series in the infinitely many non-commuting variables t, (1), (2), (3),... .

Our third step is to set up a formal equation in the variable S such that $S = \sum_{w \in CM} w$ is a solution. The equation is

$$S = t + (1)\, S + (2)\, S + (3)\, S + \cdots = t + \sum_{n=1}^{\infty} (n)\, S \tag{1}$$

formed by putting the variable S on the left side and setting the right side equal to the formal sum of *all* words on the right hand side of *all* production rules in the schema.

To see that $\sum w$ is a solution note that $(1) \times \sum w = \sum (1)\, w$ is the sum of all compositions beginning with (1), $(2) \times \sum w = \sum (2)\, w$ is the sum of all compositions beginning with (2) and so on. Hence substituting $\sum w$ for every occurrence of S on the right hand side of the equation yields each composition exactly once which is exactly what we get by substituting $\sum w$ for S on the left hand side of the equation.

Equation (1) can be thought of as the defining equation for the generating function of all compositions. Since $\sum_{w \in CM} w$ is a solution of Eq. (1) we shall abuse terminology and write $S = \sum_{w \in CM} w$ when talking about the equation.

Let us now derive the generating function for composition of n

$$C(x) = \sum_{n=0}^{\infty} c_n x^n, \qquad c_0 = 1;$$

thus we ignore all information about a composition $a_1, a_2, \ldots, a_n$ except the value of its sum. $C(x)$ is derived by specializing $S = \sum w$ and Eq. (1). However, since we are dealing with formal power series, homomorphism replaces the operation of substituting special values for the variables $t, (1), (2), \ldots$ of the series S. Consider the homomorphism ϕ: $M(C) \to M(\{x\})$ (the monoid generated by x, i.e. all power series in x) defined on the generators by

$$(n) \to x^n,$$
$$t \to x^0 = 1.$$

Then for a composition $(a_1)(a_2) \cdots (a_k)\, t$

$$\phi[(a_1)(a_2) \cdots (a_k)\, t] = \Big(\prod_i \phi(a_i)\Big)\, \phi(t) = x^{a_1 + a_2 + \cdots + a_k}.$$

ϕ induces a homomorphism ϕ^* on the ring of formal power series over $M(C)$ into the ring of formal power series in one variable x and

$$\phi(S) = \phi\Big(\sum_{w \in CM} w\Big) = \sum_{n=0}^{\infty} c_n x^n = C(x).$$

Hence taking the image of Eq. (1) we see that $C(x)$ satisfies the induced equation

$$C(x) = 1 + \sum_{n=1}^{\infty} x^n C(x) = 1 + C(x) \frac{x}{1 - x}.$$

Solving for $C(x)$ we get

$$C(x) = \frac{1 - x}{1 - 2x}$$

and expanding

$$c_n = 2^{n-1}.$$

To derive the generating function $C(x, y) = \sum_{n,k} c_{n,k} x^n y^k$, where x and y commute and $c_{n,k}$ counts compositions of n with k even parts, we introduce the homomorphism Θ defined by

$$\begin{aligned} (n) &\to x^n y, \qquad && n \text{ even}, \\ (n) &\to x^n, && n \text{ odd}, \\ t &\to 1, \end{aligned}$$

of $M(C)$ onto the free commutative monoid generated by x and y.

$$\Theta[(a_1)(a_2) \cdots (a_m)\, t] = x^{a_1 + a_2 + \cdots + a_m}\, y^{\text{number of even parts}}$$

and Θ induces a homomorphism Θ^* of formal power series over the monoids. We then have

$$\begin{aligned} \Theta^*(S) &= \Theta^* \left(\sum w \right) \\ &= \sum c_{n,k} x^n y^k = C(x, y) \end{aligned}$$

and $C(x, y)$ satisfies the induced equation

$$C(x, y) = 1 + \sum_{n=0}^{\infty} x^{2n+1} C(x, y) + \sum_{n=1}^{\infty} x^{2n} y C(x, y)$$

from which $C(x, y)$ is immediately derived.

Thus given a specific composition counting problem, the fourth step of our procedure is to specialize the general series $\sum w$ and its defining Eq. (1), via a homomorphism, to look at only a special set of properties. This requires that the properties, e.g. values of compositions, number of even parts, number of parts of size 2, number of prime parts, etc., be representable in the multiplicative structure of the semigroup, that is by a homomorphism.

EXAMPLE. *Rooted Planar Trees.* By a *rooted planar tree* we mean a rooted unlabeled tree, in which the edges emanating from an arc (i.e. moving away from the root) are given an order i.e. we designate a final edge, second edge, etc. For example a family tree where the names of offspring are unimportant but the relative ages define the order of the edges. Topologically a rooted planar tree is a rooted unlabeled tree drawn in the plane with a cyclic ordering of the edges emanating from any vertex (this corresponds to the ordering in the combinatorial definition) which can be designated by rooting one of the edges incident to the root and taking the cyclic order as clockwise with the rooted edge first.

Two such trees are isomorphic if there is an *orientation* preserving homomorphism of the plane carrying one tree onto the other preserving vertices, edges, roots, and incidence relations.

Figure 1 shows two rooted planar trees. Figure 2 gives two isomorphic trees. Figure 3 shows two non-isomorphic trees. Figure 4 shows two alternate ways to represent the same rooted planar trees. In Fig. 4b the rooted vertex is at the head of the broken arrow and the rooted edge is the final edge emanating from the rooted vertex in a clockwise direction from the broken arrow. It will frequently be convenient to use this broken arrow representation.

We now represent these trees by words in a free monoid and generate the language consisting of all "tree" words by a formal schema.

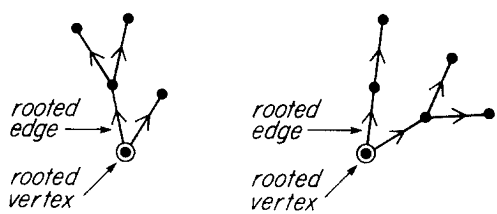

FIGURE 1

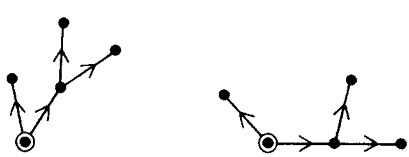

FIGURE 2

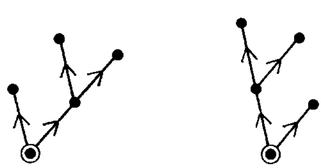

FIGURE 3

The free semigroup $M(T)$ is generated by the set $T = \{(\ ,\ ), t\}$ and the schema is

$$S \to (\overset{n \text{ times}}{S \cdots S}),\ n = 1, 2, \ldots$$
$$S \to t.$$

As for compositions, we say a word $w \in M(T)$ is derivable from the schema if there exists a sequence $w_0 = S, w_1, w_2, \ldots, w_m = w$, of words in $M(T \cup \{S\})$, such that w_{i+1} is derived from w_i by replacing the *left most* occurrence of S in w_1 by the expression on the right hand side of any substitution rule in the schema. We call the sequence a *derivation* of w. It is easy to prove that this schema is non-ambiguous in the sense that any word w is derivable by at most one derivation.

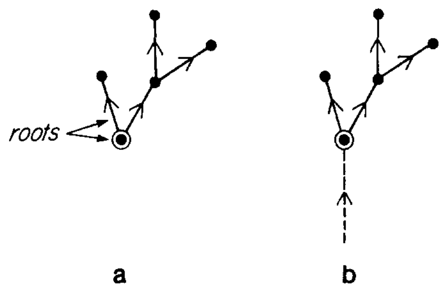

FIGURE 4

To connect this with trees consider the following example of a derivation of the word $((tt)\, t(t))$:

$$S \to (SSS) \to ((SS)\, SS) \to ((tS)\, SS) \to ((tt)\, SS)$$
$$\to ((tt)\, tS) \to ((tt)\, t(S)) \to ((tt)\, t(t))$$

We associate with this word *and* its derivation a *derivation tree* or *parsing tree* (Fig. 5a) and ignoring the labels on the derivation tree we get a rooted planar tree (Fig. 5b) in which the broken arrow is not part of the tree but only designates the rooting as discussed earlier. To go from the tree back to the word we just associate with each vertex a pair of parentheses and as many S's between them as there are arrows emanating from the vertex.

By associating with each word the rooted planar tree determined by the derivation tree, we get a one-one correspondence between the words generated by our schema and the set of all rooted planar trees. Let RT denote this language which codes rooted planar tree.

To solve a variety of tree enumeration problems we form the formal sum

$\sum_{w \in RT} w$ of all tree words. This is a formal power series in the non-commuting variables $\{(,), t\}$. Proceeding as with compositions we see that $S = \sum_{w \in RT} w$ is a solution of the formal equation

$$S = t + (S) + (SS) + (SSS) + \cdots \tag{2}$$
$$= t + \sum_{n=1}^{\infty} (\overbrace{S \cdots S}^{n \text{ times}})$$

formed by setting the left side equal to S and the right side equal to the formal sum of all expressions on the right hand side of all production rules in the schema.

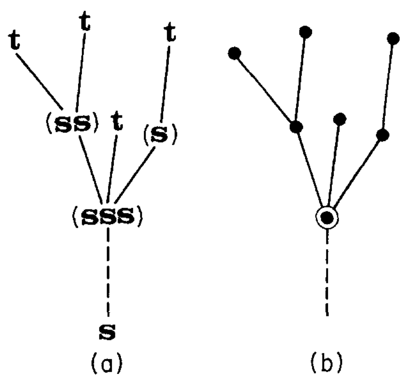

FIGURE 5

To see this substitute $\sum w$ into the equation and note that every tree word then appears exactly once on the left side. Now on the right hand side $t + \sum_{n=1}^{\infty} (\overbrace{\sum w \sum w \cdots \sum w}^{n \text{ times}})$ the term $(\sum w) = \sum(w)$ is the sum of all words which use $S \to (S)$ as the first step of their derivation (trees with one edge emanating from the root) and each such word appears once, $(\sum w \sum w) = \sum_w \sum_{w'} (ww')$ is the sum of all words which use $S \to (SS)$ as the first step of their derivation (trees with two edges emanating from the root) and each such word appears once, and so on with the terms $(\sum w \sum w \sum w)$, $(\sum w \sum w \sum w \sum w)$,.... Hence each tree word appears exactly once on the right side of the equation.

As for compositions, Eq. (2) can be thought of as the defining equation to derive various generating functions of rooted planar trees. For example, we first derive the generating function $T(x) = \sum_{n=1}^{\infty} t_n x^n$, where t_n is the number of rooted planar trees with n vertices. Consider the homomorphism ϕ: $M(T) \to M(\{x\})$ (the monoid generated by x) defined on the generators of $M(T)$ by

$$(\to x, \qquad t \to x.$$

The ϕ maps a word w onto x^n, where n is the number of vertices in the tree represented by w. ϕ induces a homomorphism ϕ^* of the ring of formal power series over $M(T)$ into the ring of formal power series in one variable x and $\phi^* \sum_{w \in RT} w = T(x)$. Taking the image of Eq. (2) we see that $T(x)$ satisfies the induced equation

$$T(x) = x + \sum_{n=1}^{\infty} xT(x)^n$$
$$= x \frac{1}{1 - T}$$

Thus T satisfies the quadratic equation

$$T^2 - T + x = 0$$

whose solutions are

$$T = \frac{1 \pm (1 - 4x)^{1/2}}{2}$$
$$= \frac{1 \pm \sum_{n=0}^{\infty} \binom{\frac{1}{2}}{n}(-4x)^n}{2}$$

where we recall that for $n \geqslant 1$, $\binom{\frac{1}{2}}{n} = (-1)^{n-1}\binom{2n-2}{n-1}/n2^{2n-1}$ and $\binom{\frac{1}{2}}{0} = 1$. Since $\binom{\frac{1}{2}}{0} = 1$ and we need a zero constant term we choose the solution corresponding to the negative sign. Hence

$$T(x) = \frac{1}{2} \sum_{n=1}^{\infty} \binom{\frac{1}{2}}{n}(-4x)^n$$

and

$$t_n = \frac{1}{n}\binom{2n-2}{n-1}.$$

We can of course specialize our schema to count a subclass of trees. Suppose e.g. we wish to count binary trees i.e. the outdegree of every vertex equals 2. These are generated by the schema

$$S \to (SS),$$
$$S \to t.$$

Using the homomorphism $(\to x$, $t \to x$ we get that $T^b(x) = \sum_{n=1}^{\infty} t_n{}^b x^n$, the generating function for rooted planar binary trees satisfies

$$T^b(x) = x + xT^b(x)^2.$$

Solving this quadratic equation, similar to the previous one, we get

$$t_{2n}^b = 0,\ t_{2n-1}^b = \frac{1}{n}\binom{2n-2}{n-1},\ n = 1, 2, \ldots,$$

i.e. the number of rooted planar binary trees with $2n - 1$ vertices equals the number of all rooted planar trees with n vertices.

2. Formal Languages

In this section we develop a theory of formal languages and generating functions in a countable number of non-commuting variables to the extent needed for combinatorial applications. The basic reference for the finite case is the paper of Chomsky-Schützenberger [4].

A *free monoid* (*semi-group*) $M(G)$ is a triple $M(G) = (G, W, \circ)$ where:

(a) the set G of *generators* is an arbitrary set,

(b) W is the set of *words* on G, i.e. all finite sequences of elements of G,

(c) $\circ$ is the binary operation of concatenation on the words, i.e. $w_1 \circ w_2 = w_1 w_2$.

If we say that two words in $M(G)$ are equivalent when the elements of one are a rearrangement of the elements of the other, then we have an equivalence relation and $A(G)$, the quotient monoid with respect to this relation, is called the *free Abelian monoid* on G. Intuitively we can think of the words of $M(G)$ as monomials in non-commuting variables (elements of G) and the words of $A(G)$ as monomials in same variables considered as commuting variables.

A *language* L is a subset of a free monoid.

A *formal grammar* is a 3-tuple $F = (A, B, P)$ where:

(a) A, the non-terminal or variable symbols, is a countable set whose elements are generally denoted by upper case letters,

(b) B, the set of terminal symbols, is a countable set whose elements are generally denoted by lower case letters,

(c) P is a countable set of *substitution rules or productions*, i.e. expressions of the form $T \to w$, where T is an element of A and w is a word of $M(A \cup B)$.

A *formal schema* is a 4-tuple $F_S = (A, B, S, P)$ where (A, B, P) is a formal grammar and S is a fixed element of A called the *axiom*.

A word $w \in M(B)$ is *derivable* form the schema F_S if there exists a sequence $w_0 = S, w_1, w_2, \ldots, w_m = w$, of words in $M(A \cup B)$, such that w_{i+1} is derived from w_i by replacing the *left most* occurrence of a variable (element of A) in w_i by the right hand side of any substitution rule in the schema in which the variable appears on the left. When describing a schema we generally write down the set of substitution rules with the sets of terminal and non-terminal symbols being understood. The language $L(F_S)$ of the schema F_S is the set of all words derivable from F_S. When A, B and P are finite sets the language is called a context free language.

It is possible for a word w derivable from a schema F_S to be derived in more than one way i.e. there can be more than one derivation sequence whose last term is w. E.g. consider the schema $S \to T, S \to t, T \to S$. The word $w = t$ is derivable by the sequences S, T, t and S, t. In fact t is the only word derivable from this language and there are infinitely many ways to derive it. We let $N(w, F_S)$ be the number of ways to derive w from the schema F_S with with $N(w, F_S) = \infty$ if there are infinitely many such ways.

HYPOTHESIS. We restrict ourselves to those schema F_S, such that $N(w, F_S)$ is finite for all $w \in L(F_S)$.

$L(F_S)$ is *unambiguous* if $N(w, F_S) = 1$ for all $w \in L(F_S)$.

By a *formal power series* (FPS) over a free monoid $M(G)$ (or $A(G)$) we understand a map $r: M \to Z$, written symbolically as $r = \sum_{w \in M} r(w)\, w$, where $r(w)$ is called the coefficient of w. We could just as easily choose the elements of any field to be our coefficients but there is no need for such generality here.

The set of all power series over a monoid M can be made into a ring $R(M)$ by introducing the following operations:

(a) addition — $(r_1 + r_2)(w) = r_1(w) + r_2(w)$,

(b) multiplication — $r_1 \cdot r_2(w) = \sum_{w_1, w_2; w_1 \circ w_2 = w} r_1(w_1)\, r_2(w_2)$. This ring also has an operation of scalar multiplication by elements of Z viz.

(c) $(\alpha \cdot r)(w) = \alpha(r(w))$.

If $F_S = (A, B, S, P)$ is a formal schema and $L(F_S)$ is the corresponding language we can associate with L its characteristic formal power series over $M(B)$ viz. $f_S = \sum N(w, F)\, w$.

Now in line with our introductory examples we introduce the equation of a grammar. If $F = (A, B, P)$ is a grammar and $w_{S,1}, w_{S,2}, \ldots, \in A$, are all words such that $S \to w_{S,i}$ is a substitution rule then the set of formal equations

$$S = \sum_i w_{S,i}, \text{ for all } S \in A$$

is the *set of equations* of the grammar F. For any $S \in A$ we also call these equations the equations of the schema F_S.

Before stating the main theorem we need one additional notion. A *homomorphism* from a monoid M to a monoid N is a map $\phi: M \to N$ such that $\phi(a \circ b) = \phi(a) \circ \phi(b)$ for all $a, b \in M$. Such a homomorphism induces a homomorphism ϕ^* of the formal power series rings, $\phi^*: R(M) \to R(N)$, by $\phi^* \sum r(w)\, w) = \sum r(w)\, \phi(w)$. If M and N are free monoids or free Abelian monoids and a set E of power series satisfies a set of equations over $R(M)$, then the image set $\phi^*(E)$ satisfies the induced set of equations over $R(N)$ since these are just identities in the power series ring. Furthermore if $F_S = (A, B, S, P)$ is a grammar and $\phi: M(B) \to M(B')$ is a homomorphism then ϕ induces a grammar $\phi(F) = (A, B', P')$ in a natural way where the image of the substitution rule arises by taking the image of each element of B in the rule e.g. $\phi[S \to abS] = S \to \phi(a)\, \phi(b)\, S$. The language $L(\phi(F_S))$ is called the homomorphic image of $L(F_S)$ under ϕ.

THEOREM. *If $F = (A, B, P)$ is a formal grammar, $\{F_S : S \in A\}$ is the set of all schema which can be formed from A, and $\{f_S\}$ is the associated set of formal power series, then $\{f_S\}$ satisfies the set of equations for F, i.e., if we substitute f_S for each occurrence of S in the set of equations we get a true set of identities in the corresponding ring of formal power series.*

A proof of this theorem can be given by the iterative technique used by Chomsky and Schützenberger [4] in the finite case. Here we present a different proof for the case where the grammar has only one variable, to stress the intuitive idea behind the proof. Our proof is easily generalized to countably many variables.

To prove this theorem we need the following lemma.

LEMMA. *Given a grammar $F = (A, B, P)$, there exists a grammar $F' = (A, B', P')$ and a homomorphism $\phi: M(B') \to M(B)$ such that* (a) $\phi(F') = F$, (b) *every language $L(F_S')$ associated with F' is unambiguous, and* (c) *every language $L(F_S)$ is the homomorphic image of $L(F_S')$.*

Proof. We construct F' as follows: For each substitution rule $S \to w$ in P we have a substitution rule $S \to ({}_S w)$ in P'. We must take the set $C = \{), ({}_S : S \in A\}$ to be distinct from any of the symbols in B. Now set $B' = B \cup C$.

The map ϕ that takes a word in $M(B')$ and erases all occurrences of parentheses from C yields a word in $M(B)$ and $\phi: M(B') \to M(B)$ is clearly a homomorphism. Since $\phi[S \to ({}_S w)] = S \to w$ we have $\phi(F') = F$. Similarly $\phi(L(F_S')) = L(F_S)$ for all $S \in A$.

To prove unambiguity we fix $S \in A$, choose a word $w \in L(F_S')$ and show that w uniquely determines its derivation sequence. Reading the letters of w from right to left we locate the first left parenthesis from the set C, say $({}_T$. We then find the matching right parenthesis so that $w = v_1({}_T v_2)\, v_3$ where

v_2 and v_3 contain no left parentheses from C. Since our derivation always proceeds by substitution for the left most occurrence of a variable the next to the last word in the derivation sequence *must* be $w_2 = v_1 T v_3$. Applying the same procedure to w_2 we get the unique second to the last word in the deviation sequence and iterating this process we arrive at the unique derivation sequence for w.

Proof of the theorem. By the preceeding lemma we need only consider those grammars $F = (A, B, P)$ whose associated languages F_S are unambiguous. Since we are restricting ourselves to grammars with one variable, we let our substitution rules be $P = \{S \to w_{S,i} : i = 1, 2,...\}$ and we have one equation

$$S = \sum_i w_{S,i}.$$

Let $\bar{w}_{S,i}$ be the power series resulting from the substituion of f_S, the characteristic series of $L(F_S)$, for each occurrence of S in $w_{S,i}$. Then we must prove that

$$\sum_{w \in L(F_S)} w = \sum_i \bar{w}_{S,i} \tag{3}$$

is a true identity in the ring $R(B)$. On the left hand side of the equation we have the set of words of $L(F_S)$ each appearing once. We must show the right hand side consists of the same set.

Let $w \in L(F_S)$ and let $S \to w_{S,j}$ be the first substitution rule used in the unique derivation of w. Then $w_{S,j} = v_1 S \cdots S v_2 \cdots S v_n$ where the v_i are empty or words of $M(B)$. Now $w = v_1 w_1 \cdots w_{i_1} v_2 \cdots w_{i_n} v_n$ where each $w_i \in L(F_S)$ since it is obtained from a series of substitutions beginning with S. Hence it must appear as a term in $\bar{w}_{S,j} = v_1 f_S \cdots f_S v_2 \cdots f_S v_n$ since each w_i is a term in f_S. Hence every $w \in L(F_S)$ appears on the right hand side of Eq. (3).

Now let w be any word appearing on the right hand side of Eq. (3) and suppose it appears in the series $\bar{w}_{S,j}$. Then as above w must be of the form $v_1 w_1 \cdots w_{i_1} v_2 \cdots w_{i_n} v_n$ where each v_i is empty or $\in M(B)$ and each $w_i \in L(F_S)$. Thus by starting with the substitution rule $S \to v_1 S \cdots S v_2 \cdots S v_n$ and then listing the derivation sequences for $w_1, w_2, ..., w_n$, leaving out the first term of each viz. S, we have produced a derivation sequence for w and therefore $w \in L(F_S)$. Hence the right hand side of equation (3) consist of exactly the words of $L(F_S)$ and the theorem is proved.

3. Rooted Planar Trees

In example (2) we were able to count rooted planar trees by the total number of vertices or the number of end vertices and non-end vertices, using different variables for each property. Suppose however we wish to count each tree by the number of vertices of each given outdegree (i.e. by the number of branches hanging from each vertex, going away from the root). We shall derive the generating function

$$A(\bar{x}) = \sum a(\lambda_0, \lambda_1, \ldots)\, x_0^{\lambda_0} x_1^{\lambda_1} \cdots$$

where $A(\bar{x}) = A(x_0, x_1, \ldots)$ is a generating function in countably many commuting variables $x_0, x_1, \ldots$, the summation is over all sequences $\lambda_0, \lambda_1, \ldots$ of non-negative integers and $a(\lambda_0, \lambda_1, \ldots)$ equals the number of rooted planar trees with λ_i vertices of outdegree i (λ_0 counts end vertices). Note that many of the a's will be zero. In fact $a(\lambda_0, \lambda_1, \ldots) = 0$ unless $1 + \sum_{i=1}^{\infty} (i-1)\lambda_i = \lambda_0$ a condition easily proved by induction on the process of growing a tree one edge at a time. Although the information as to the number of vertices of each degree is in the coding of trees to words we used in example (2), it is not in the "multiplicative structure" of the word, i.e., we cannot extract the information by a homomorphism. To remedy this consider the following formal schema:

$$S \to ({}_n\overbrace{S \cdots S}^{n \text{ times}}), \; n = 1, 2, \ldots, \quad S \to u.$$

The derivation trees of this schema will be just as in example (2), the set of rooted planar trees. We note that this again yields an unambiguous language. Now, however, we have put some redundant information in the coding, namely each left parenthesis in a word, which corresponds to a vertex in the tree, has a subscript which denotes the outdegree of that vertex. E.g. the derivation sequence

$$\begin{aligned} S &\to ({}_2SS) \to ({}_2({}_1S)\,S) \to ({}_2({}_1u)\,S) \to ({}_2({}_1u)({}_2SS)) \\ &\to ({}_2({}_1u)({}_2uS)) \to ({}_2({}_1u)({}_2uu)) \end{aligned}$$

corresponds to the tree in Fig. 6. We have of course extended the finite

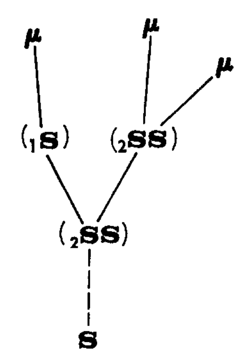

FIGURE 6

alphabet $\{(,), u\}$ of example (2) to the countable alphabet $\{), u, (_n : n = 1, 2, \ldots\}$ but we now have the desired information in the generators of the monoid.

The equation of this schema is

$$S = u + \sum_{n=1}^{\infty} ({}_nS \cdots S). \tag{4}$$

To derive the generating function $A(\bar{x})$ consider the homomorphism

$$({}_n \to x_n \,, u \to x_0 \,,) \to 1.$$

Under this map a word w representing a tree with λ_i vertices of outdegree i gets mapped onto $x_0^{\lambda_0} x_0^{\lambda_1} \cdots$ and $S = \sum_{w \text{ a tree word}} w$ maps onto $A(\bar{x})$. This induces a map of the equation of the schema and we have

$$A(\bar{x}) = x_0 + \sum_{n=1}^{\infty} x_n A(\bar{x})^n = \sum_{n=0}^{\infty} x_i A^i. \tag{5}$$

By the introduction of a new variable we shall now solve this functional equation for $A(\bar{x})$ by means of the Lagrange inversion formula.

Now A is a formal power series in $x_0 \,, x_1 \,, \ldots$ such that for each N, there are only a finite number of monomials of degree N occurring in A.

We set $A_N =$ the homogeneous part of A of degree N (note $A_1 \equiv x_0$) and also set

$$A(t) = \sum_{n=1}^{\infty} A_n t^n,$$

a fps in t with coefficients in $R[x_1 \,, x_2 \,, \ldots,]$. Taking the homogenous parts of deg $N + 1$ (for $N \geqslant 1$) on both sides of (5) gives

$$A_{N+1} = x_1 A_N + x_2 \sum_{\substack{v_1+v_2=N \\ v_i \geqslant 1}} A_{v_1} A_{v_2} + \cdots + x_N \sum_{\substack{v_1+\cdots+v_N=N \\ v_i \geqslant 1}} A_{V_1} \cdots A_{V_N}$$

or equivalently (for all $N \geqslant 1$)

$$\begin{aligned} A(t)\Big|_{t^{N+1}} &= x_1 A(t)\Big|_{t^N} + x_2 A(t)^2\Big|_{t^N} + \cdots + x_N A(t)^N\Big|_{t^N} \\ &= \sum_{v=1}^{N} x_v A(t)^v \Big|_{t^N} \\ &= \sum_{v=1}^{\infty} x_v A(t)^v \Big|_{t^N} = t \sum_{v=1}^{\infty} x_v A(t)^v \Big|_{t^{N+1}} . \end{aligned}$$

Thus we have

$$\begin{aligned} A(t) &= A_1 t + t \sum_{v=1}^{\infty} x_v A(t)^v \\ &= x_0 t + t \sum_{v=1}^{\infty} x_v A(t)^v \\ &= t\left(x_0 + \sum_{v=1}^{\infty} x_v A(t)^v\right) = t \sum_{v=0}^{\infty} x_v A(t)^v \end{aligned}$$

or equivalently

$$\frac{A(t)}{\sum_{v=0}^{\infty} x_v A(t)^v} = t. \tag{6}$$

We set

$$a(t) = \frac{t}{\sum_{v=0}^{\infty} x_v t^v} \text{ (an invertible } fps \text{ under functional composition)}$$

and (6) $\Rightarrow$

$$a(A(t)) = t.$$

The Lagrange inversion theorem states: given $f(t)$, $F(t)$ such that

$$f(F(t)) = t, \qquad \text{then for all } N > 0,$$

$$F(t)\Big|_{t^N} = \frac{1}{N}\left(\frac{t}{f(t)}\right)^N\Big|_{t^{N-1}}.$$

Thus in our case, we have

$$\begin{aligned} A_{N+1} = A(t)\Big|_{t^{N+1}} &= \frac{1}{N+1}\left(\sum_{v=0}^{\infty} x_v t^v\right)^{N+1}\Big|_{t^N} \\ &= \frac{1}{N+1} \sum_{\substack{v_1+v_2+\cdots+v_{N+1}=N \\ v_i \geqslant 0}} x_{v_1} x_{v_2} \cdots x_{v_{N+1}}. \end{aligned} \tag{7}$$

Now $v_1 + v_2 + \cdots + v_{N+1} = N$ and $v_i \geqslant 0$ implies $v_i \leqslant N$. Thus if we let λ_i = number of v's which are equal to i we have that the coefficient of $x_0^{\lambda_0} x_1^{\lambda_1} \cdots x_N^{\lambda_N}$ under the summation sign in (7) is given by the multinomial coefficient $\binom{N+1}{\lambda_0, \lambda_1, \ldots, \lambda_N}$. Hence

$$A_{N+1} = \frac{1}{N+1} \sum \binom{N+1}{\lambda_0, \ldots, \lambda_N} x_0^{\lambda_0} x_1^{\lambda_1} \cdots x_N^{\lambda_N}$$

and

$$A(\lambda_0, \lambda_1, \ldots) = \frac{1}{N+1}\binom{N+1}{\lambda_0, \ldots, \lambda_N}$$
$$= \frac{N!}{\lambda_1! \cdots \lambda_N!}$$
$$= \frac{(\lambda_0 + \lambda_1 + \cdots + \lambda_N - 1)!}{\lambda_0! \, \lambda_1! \cdots \lambda_N!}$$

which was our goal.

If we wish to count only those rooted planar trees with specific restrictions on the types of outdegrees we can either specialize our Eq. (4) or the formal schema. The latter specialization often lead to a simpler equation as, e.g., in the case of binary trees and will prove most useful in later sections.

4. Polygon Dissections

We consider regular n-sided polygons in the plane whose vertices are labeled $1, 2, \ldots, n$ in a counterclock-wise direction. A dissection of such a polygon is a collection of *non-overlapping* diagonals which decompose the polygon into smaller polygonal regions. A *dissected polygon* is a polygon together with a dissection. Examples are given in Fig. 7.

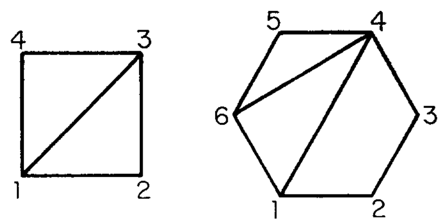

Figure 7

In this section we describe a method for counting all dissected polygons as well as subclasses given by restrictions on the subpolygons. Polya [1] counted the number of dissected polygons, allowing only triangular subpolygons, by using a picture writing identity.

The essential idea is to associate with each dissected polygon a rooted planar tree and thus reduce the enumeration to a tree counting problem. This correspondence is illustrated in Fig. 8. We place a vertex of the tree in each subpolygon as well as in the outside region near the middle of each edge of the polygon. The edges of the tree are the lines connecting the vertices across each line of the dissected polygon. The edge crossing the $\overline{12}$ side of the polygon is a broken edge and the vertex in the subpolygon containing the $\overline{12}$ side is the root.

It is easy to prove that our procedure sets up a one-one correspondence between dissected polygons and rooted planar trees in which no vertex has outdegree one. The tree with one vertex corresponds to the trivial dissected polygon with one side. The number of non-terminal nodes in the tree corresponds to the number of subpolygons and a node of outdegree i corresponds to a subpolygon with $i + 1$ sides.

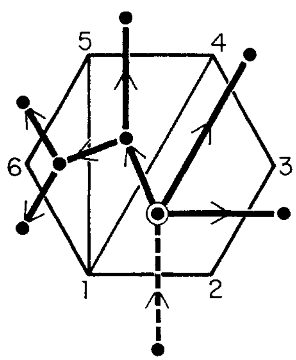

FIGURE 8

Thus, e.g., we can derive Polya's formula [1] for P_n, the number of dissected polygons consisting of n-triangular subpolygons. Since, in this case, an $n + 2$ sided dissected polygon has exactly n triangles, we are in fact counting the number of ways of dissecting all $n + 2$ sided polygons into triangles. Under our correspondence these dissected polygons correspond to the binary rooted planar trees of example (2) given by the schema

$$S \to (SS), \qquad S \to t$$

We choose a slightly different homomorphism than in example (2) viz.,

$$(\to x, \qquad t \to 1.$$

Under this homomorphism a term x^n in the generating function corresponds to a tree with n non-terminal nodes or a dissected polygon with n-triangles. The quadratic equation which arises is easily solved and yields

$$D_n = \text{the number of dissected } n\text{-sided polygons (into } n - 2 \text{ triangles)}$$
$$= -\tfrac{1}{2}\binom{\frac{1}{2}}{n-1}.$$

To count dissected polygons with each subpolygon having k sides we would use the schema

$$S \rightarrow (\overbrace{S \cdots S}^{k-1 \text{ times}}), S \rightarrow t$$

with the same homomorphism $(\rightarrow x, t \rightarrow 1$. This gives the functional equation

$$T(x) = xT(x)^{k-1} + 1$$

the so-called trinomial equation which can be solved by Lagrange inversion [9].

To count dissected polygons with a specified number of subpolygons of each type we use the general formula derived in Section 3.

5. Random Walks

The theory of discrete random walks is treated both extensively and beautifully in Feller, Vol. 1 [10], in particular, the results in this section are derived in Chapter 3 on fluctuations in coin tossings and random walks.

We review some of the basic concepts, for others the reader should consult Feller.

Definition. *A lattice path of length r* is a sequence of points $(0, 0)$, $(1, n_1)$, $(2, n_2)$,..., (r, n_r) satisfying

(a) n_i are non-negative integers,

(b) $n_{i+1} = n_i \pm 1$, $i = 0, 1,..., r - 1$.

If $n_r = 0$ we say the path is *closed.*

All paths considered in this section are closed paths and we thus drop the adjective.

We can also represent each path by a polygonal curve whose edges are the line segments connecting (i, n_i) to $(i + 1, n_{i+1})$; see Fig. 9.

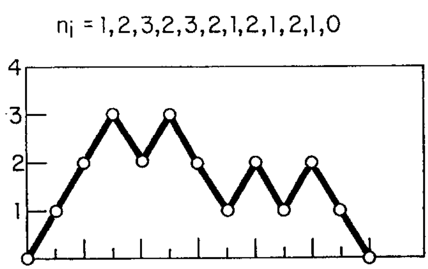

Figure 9

Our problem is to count all paths of length r. To do this we use a very pretty correspondence, due to Harris [11], between paths and a class of rooted planar trees. A formal description and proof of this very simple correspondence would be rather long and tedious so we illustrate the process (going from tree to path) by an example which should make the general idea completely clear.

The tree in Fig. 10 corresponds to the lattice path of Fig. 9. Starting at the root we traverse the contour of the tree as shown in the figure and we begin the lattice path at (0, 0). Each time we traverse an edge of the tree going away from the root we add a new segment to the lattice path going

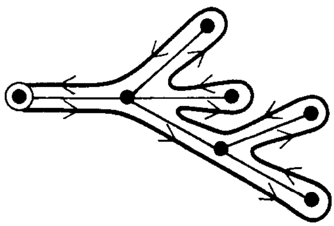

FIGURE 10

up, i.e. the ordinate of its endpoint is increased by one, and each time we traverse an edge going in toward the root we add a segment going down (decrease the ordinate by one). It is easy to see that this sets up a one-one correspondence between all rooted planar trees whose roots have degree one and all lattice paths.

The class of rooted planar trees, with the degree of the root equal to one, is generated by the schema

$$S \to (T),$$
$$T \to (_n \overbrace{T \cdots T}^{n \text{ times}}), \; n = 1, \ldots,$$
$$T \to t.$$

This schema then leads to the pair of equations

$$S = (T),$$
$$T = t + (_1T) + (_2TT) + \cdots.$$

We want to compute the generating function $P(x) = \sum_{n=1}^{\infty} p_n x^n$ where p_n counts closed lattice paths of length n. Noting that each edge of the tree gives two segments in the path we choose the homomorphism

$$(\to x^2, \qquad (_n \to x^{2n}, \qquad t \to 1.$$

This maps S onto $P(x)$, T onto another generating function $Q(x)$ and we get the induced set of equations

$$P = xQ$$

$$Q = 1 + \sum_{n=1}^{\infty} x^{2n}Q^n = 1 + \frac{x^2Q}{1 - x^2Q}.$$

Thus we have a quadratic equation for Q and we can find $Q(x)$, $P(x)$ and p_n as in previous sections.

The enumeration of non-closed paths can also be handled by Harris' correspondence but this requires a fancier schema and will be done in another paper.

References

1. G. Polya, On picture writing, *Amer. Math. Monthly* (Dec. 1956), 689–697.
2. M. P. Schützenberger, "Some Remarks on Chomsky's Context-Free Languages," Quarterly Progress Report No. 58, Research Laboratory of Electronics, M.I.T., October 1961.
3. M. P. Schützenberger, On a theorem of R. Jungen, *Proc. Amer. Math. Soc.* **13** (1962), 885–890.
4. N. Chomsky and M. P. Schützenberger, The algebraic theory of context-free. languages, *in* "Computer Programming and Formal Systems" (P. Braffort and D. Hirschberg, Eds.), North-Holland, Amsterdam, 1963.
5. M. Gross and Lentin, "Introduction to Formal Grammars," Springer, New York, 1971.
6. W. Kuich, Languages and the enumeration of planted plane trees, *Indagationes Math.* **32** (1970), 268–280.
7. R. Cori and J. Richard, Énumération des graphes planaires à l'aide des séries formelles en variables non commutatives, *Discrete Math.* **2** (1972), 115–162.
8. E. A. Bender and J. R. Goldman, Enumerative uses of generating functions, *Indiana Univ. Math. J.* **20** (1971), 753–765.
9. G. N. Raney, Functional composition patterns and power series reversion, *Trans. Amer. Math. Soc.* **94** (1960), 441–451.
10. W. Feller, "An Introduction to Probability Theory and Its Applications," Vol. 1, 3rd ed., Wiley, New York, 1968.
11. T. E. Harris, "First Passage and Recurrence Distributions," *Trans. Amer. Math. Soc.* **73** (1952), 471–486.