Discrete Mathematics 41 (1982) 199–214
North-Holland Publishing Company

ENUMERATIONS OF ROOTED TREES WITH AN APPLICATION TO GROUP PRESENTATIONS

Phillip SCHULTZ

Department of Mathematics, University of Western Australia, Nedlands, W.A. 6009, Australia

Received 24 February 1981
Revised 17 August 1981

The combinatorial properties of the set of rooted trees can be viewed algebraically by considering this set as an algebra with one unary and one binary operation. This viewpoint yields solutions to several enumeration problems. In particular, using a correspondence between rooted trees and presentations of finite abelian p-groups devised by A.W. Hales, I enumerate all presentations of a given group.

1. Introduction

Rooted trees are ubiquitous in combinatorics, computer science, chemistry and physics, and the problem of finding an 'orderly algorithm' (Read [13]) to produce a catalogue of all trees satisfying various conditions is of continuing interest: see for example [3, 6, 7, 11, 12]. The parameters usually involved include number of edges, number of leaves, lengths of paths and number of vertices of each degree and at each level. Unfortunately the known enumerations of rooted trees of various types [5, 13] do not necessarily classify them according to these parameters, whereas those enumerations which do so classify them are of *ordered* trees [3]. An important current problem in combinatorics is to find an efficient uniform algorithm for finding all the ordered trees isomorphic to a given rooted tree.

In this paper I use algebraic methods to produce a number-theoretic representation of rooted trees which can to used to solve many of the problems mentioned above. It turns out that my enumeration is the same as Göbel's [5], and closely related to Read's [13]. However, my algebraic structure carries extra information about the parameters. Furthermore, the number-theoretic character of the representation means that it is easily implemented by computer.

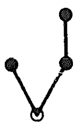
The main results are a recursive enumeration of all ordered trees isomorphic to a given rooted tree, methods of listing rooted trees satisfying various extra conditions, and as an application, the solution of a problem concerning presentations of finite abelian p-groups posed by Hales in 1971 [8]. Hales showed that each rooted tree can be used to define an abelian p-group. Call two such trees similar if the groups they define are isomorphic; then Hales' problem is to characterize the similarity classes of rooted trees. It turns out that the solution depends on a new parameter which I call the height distribution. On the one hand, it is related to the Ulm invariant of the abelian p-group, and on the other, it is related to the algebraic structure of the rooted tree.

I acknowledge the assistance of my son Jonathan who wrote all the computer programmes. Jonathan is a Year 9 student at Hollywood High School, Perth.

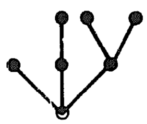
2. Algebra of rooted trees

Let T denote the set of rooted trees; for $\mathbf{u}, \mathbf{v} \in T$, let $\mathbf{u} \otimes \mathbf{v}$ be the rooted tree obtained by identifying the roots of $\mathbf{u}$ and $\mathbf{v}$, and let $p(\mathbf{u})$ be the rooted tree obtained by adjoining a new edge to the root of $\mathbf{u}$ and calling its vertex remote from the original root the new root.

For example, if

$\mathbf{u} =$  and $\mathbf{v} =$ 

then

$\mathbf{u} \otimes \mathbf{v} =$  and $p(\mathbf{u}) =$ 

Denote by $\mathbf{1}$ the rooted tree consisting of a single vertex, and by $\mathcal{T}$ the algebra $\langle T, \otimes, p, \mathbf{1} \rangle$. Clearly $\otimes$ is a commutative associative binary operation with identity $\mathbf{1}$, p is a unary operation, and $\mathcal{T}$ is generated by the singleton $\{\mathbf{1}\}$.

Now let $\mathcal{N} = \langle N, \times, p, 1 \rangle$ denote the algebra of positive integers under multiplication, with $p: N \to N$ defined by $p(n) =$ the nth prime in the natural order.

Theorem. *The mapping $\phi: 1 \mapsto \mathbf{1}$ extends uniquely to an algebra isomorphism of $\mathcal{N}$ onto $\mathcal{T}$.*

Proof. If $\phi(n)$ and $\phi(m)$ have been defined, let $\phi(n \times m) = \phi(n) \otimes \phi(m)$ and $\phi(p(n)) = p\phi(n)$. Since each $n \in N$ is a product of primes, unique up to order, and each prime is $p(n)$ for a unique n, this recursive definition extends ϕ to a well defined homomorphism of N into T.

The inverse ϕ^{-1} is also defined recursively: suppose ϕ^{-1} has been defined on rooted trees with less than k vertices, and $\mathbf{u}$ is a rooted tree with k vertices and root r. If r has degree >1, then $\mathbf{u} = \mathbf{u}_1 \otimes \mathbf{u}_2$, where $\mathbf{u}_1$ and $\mathbf{u}_2$ are rooted trees with less than k vertices, and we define $\phi^{-1}(\mathbf{u}) = \phi^{-1}(\mathbf{u}_1) \times \phi^{-1}(\mathbf{u}_2)$. If r has degree 1, then $\mathbf{u}=p(\mathbf{v})$, where $\mathbf{v}$ is a rooted tree with $k-1$ vertices, and we define $\phi^{-1}(\mathbf{u})=p(\phi^{-1}(\mathbf{v}))$. Clearly ϕ^{-1} is a homomorphism, and the inverse of ϕ.

Remarks. (1) The order $\mathbf{u}<\mathbf{v}$ iff $\phi^{-1}(\mathbf{u})\leq\phi^{-1}(\mathbf{v})$ is a well-ordering on $\mathcal{T}$ compatible with $\otimes$ and p.

(2) The 1–1 correspondence ϕ was first noted by Göbel [5], without all the algebraic paraphernalia. He used it to enumerate rooted trees. For the structural results to follow however, we need the abstract algebraic definition.

Our first application is to the enumeration of ordered trees corresponding to a given rooted tree, and for this we must establish a canonical form in the language of the algebra $\mathcal{T}$ for each rooted tree. The definition is recursive:

A *term* in $\mathcal{T}$ satisfies the following conditions:

(1) $\mathbf{1}$ is a term in $\mathcal{T}$.

(2) If $\mathbf{u}$ and $\mathbf{v}$ are terms in $\mathcal{T}$, then $\mathbf{u}\otimes\mathbf{v}$ is a term in $\mathcal{T}$.

(3) If $\mathbf{u}$ is a term in $\mathcal{T}$, then $p(\mathbf{u})$ is a term in $\mathcal{T}$.

(4) Nothing else is a term in $\mathcal{T}$.

A term in $\mathcal{T}$ is *reduced* if it satisfies (1), (3) and (4) with 'term' replaced by 'reduced term', and

(2′) If $\mathbf{u}$ and $\mathbf{v}$ are reduced terms in $\mathcal{T}$, neither of which is $\mathbf{1}$, then $\mathbf{u}\otimes\mathbf{v}$ is a reduced term in $\mathcal{T}$.

Finally, a reduced term in $\mathcal{T}$ is *completely reduced* if it satisfies (1), (3) and (4) with 'term' replaced by 'completely reduced term', and

(2″) If $\mathbf{u}$ and $\mathbf{v}$ are completely reduced terms in $\mathcal{T}$, neither of which is $\mathbf{1}$, such that $\mathbf{u}<\mathbf{v}$, then $\mathbf{u}\otimes\mathbf{v}$ is a completely reduced term in $\mathcal{T}$.

The effect of (2′) is to eliminate redundant occurrences in a term of the identity $\mathbf{1}$, and the effect of (2″) is to eliminate multiple occurrences of terms representing the same tree due to the commutativity of multiplication. Clearly each reduced term corresponds to an ordered rooted tree $\mathbf{u}$, in which the component subtrees are drawn in the plane in the order of the corresponding prime factors in an expression for $\phi^{-1}(\mathbf{u})$, while the completely reduced form corresponds to an expression for $\phi^{-1}(\mathbf{u})$ in which all such factors are in the natural order.

Occasionally we shall use 'completely reduced' to refer to terms in $\mathcal{N}$ as well as in $\mathcal{T}$.

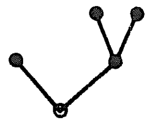
3. Coding of rooted trees

In any reduced term in $\mathcal{T}$, p occurs only in the context "$p($", $\mathbf{1}$ only in the context "$(\mathbf{1})$", and $\otimes$ only in the context "$)\otimes p$". Hence no information is lost if all p's, $\mathbf{1}$'s and $\otimes$'s are omitted, leaving a well-formed string of matched parentheses. Conversely every such string has an unambiguous interpretation as a reduced term, by insertion of appropriate p's, $\mathbf{1}$'s and $\otimes$'s.

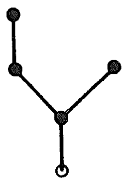
Table 1

2	()	3	(())	4	()()
5	((()))	6	()(())	7	(()())
8	()()()	9	(())(())	10	()((()))
11	(((())))	12	()()(())	13	(()(()))
14	()(()())	15	(())((()))	16	()()()()
17	((()()))	18	()(())(())	19	(()()())
20	()()((()))	21	(())(()())	22	()(((())))
23	((())(()))	24	()()()(())	25	((()))((()))
26	()(()(()))	27	(())(())(())	28	()()(()())
29	(()((())))	30	()(())((()))	31	((((()))))
32	()()()()()	33	(())(((())))	34	()((()()))
35	((()))(()())	36	()()(())(())	37	(()()(()))
38	()(()()())	39	(())(()(()))	40	()()()((()))
41	((()(())))	42	()(())(()())	43	(()(()()))
44	()()(((())))	45	(())(())((()))	46	()((())(()))
47	((())((())))	48	()()()()(())	49	(()())(()())
50	()((()))((()))	51	(())((()()))	52	()()(()(()))
53	(()()()())	54	()(())(())(())	55	((()))(((())))
56	()()()(()())	57	(())(()()())	58	()(()((())))
59	(((()())))	60	()()(())((()))	61	(()(())(()))
62	()((((()))))	63	(())(())(()())	64	()()()()()()
65	((()))(()(()))	66	()(())(((())))	67	((()()()))
68	()()((()()))	69	(())((())(()))	70	()((()))(()())
71	(()()((())))	72	()()()(())(())	73	((())(()()))
74	()(()()(()))	75	(())((()))((()))	76	()()(()()())
77	(()())(((())))	78	()(())(()(()))	79	(()(((()))))
80	()()()()((()))	81	(())(())(())(())	82	()((()(())))
83	(((())(())))	84	()()(())(()())	85	((()))((()()))
86	()(()(()()))	87	(())(()((())))	88	()()()(((())))
89	(()()()(()))	90	()(())(())((()))	91	(()())(()(()))
92	()()((())(()))	93	(())((((()))))	94	()((())((())))
95	((()))(()()())	96	()()()()()(())	97	(((()))((())))
98	()(()())(()())	99	(())(())(((())))	100	()()((()))((()))
101	(()(()(())))	102	()(())((()()))	103	((())(())(()))
104	()()()(()(()))	105	(())((()))(()())	106	()(()()()())
107	(()()(()()))	108	()()(())(())(())	109	((()((()))))
110	()((()))(((())))	111	(())(()()(()))	112	()()()()(()())
113	(()(())((())))	114	()(())(()()())	115	((()))((())(()))
116	()()(()((())))	117	(())(())(()(()))	118	()(((()())))
119	(()())((()()))	120	()()()(())((()))	121	(((())))(((())))
122	()(()(())(()))	123	(())((()(())))	124	()()((((()))))
125	((()))((()))((()))	126	()(())(())(()())	127	(((((())))))
128	()()()()()()()	129	(())(()(()()))	130	()((()))(()(()))
131	(()()()()())	132	()()(())(((())))	133	(()())(()()())
134	()((()()()))	135	(())(())(())((()))	136	()()()((()()))
137	((())(((()))))	138	()(())((())(()))	139	(()((()())))
140	()()((()))(()())	141	(())((())((())))	142	()(()()((())))
143	(((())))(()(()))	144	()()()()(())(())	145	((()))(()((())))
146	()((())(()()))	147	(())(()())(()())	148	()()(()()(()))
149	(((()))(()()))	150	()(())((()))((()))	151	(()()(())(()))
152	()()()(()()())	153	(())(())((()()))	154	()(()())(((())))
155	((()))((((()))))	156	()()(())(()(()))	157	((()()(())))
158	()(()(((()))))	159	(())(()()()())	160	()()()()()((()))
161	(()())((())(()))	162	()(())(())(())(())	163	(()(()()()))
164	()()((()(())))	165	(())((()))(((())))	166	()(((())(())))
167	((())(()(())))	168	()()()(())(()())	169	(()(()))(()(()))
170	()((()))((()()))	171	(())(())(()()())	172	()()(()(()()))
173	(()()()((())))	174	()(())(()((())))	175	((()))((()))(()())
176	()()()()(((())))	177	(())(((()())))	178	()(()()()(()))
179	(((()(()))))	180	()()(())(())((()))	181	(()(())(()()))
182	()(()())(()(()))	183	(())(()(())(()))	184	()()()((())(()))
185	((()))(()()(()))	186	()(())((((()))))	187	(((())))((()()))
188	()()((())((())))	189	(())(())(())(()())	190	()((()))(()()())
191	((()(()())))	192	()()()()()()(())	193	(()()(((()))))
194	()(((()))((())))	195	(())((()))(()(()))	196	()()(()())(()())
197	((())(())((())))	198	()(())(())(((())))	199	(()((())(())))
200	()()()((()))((()))				

For example, the ordered rooted tree



is represented by the reduced term $p(\mathbf{1})\otimes p(p(\mathbf{1})\otimes p(\mathbf{1}))$ which yields the string ( ) (( ) ( )), while the string ((( )) ( )) is interpreted as $p(p(p(\mathbf{1}))\otimes p(\mathbf{1}))$, representing the ordered rooted tree:



But this 1–1 correspondence between ordered trees and legal strings of parentheses is just the well-known one of Cayley, described for example in [12].

If in addition we insist on completely reduced terms, we recover a version of Read's coding of rooted trees [13], with some minor differences due to the fact that the ordering of strings of parentheses interpreted as binary digits differs from the ordering $\prec$ established above.

A major advantage of coding the algebra $\mathcal{T}$ by strings of parentheses is the ease with which it can be implemented on the digital computer. For example, Table 1 is a list of the first 200 rooted trees, coded as described above.

1. Applications

Suppose $n = p(n_1)^{k_1}p(n_2)^{k_2}\cdots p(n_r)^{k_r}$ is a positive integer. The number of ways of writing n as a product of primes is the number of ways of arranging in a row $k_1+k_2+\cdots+k_r$ coloured balls of which k_i have colour i for $i = 1, 2, \ldots, r$ and balls of the same colour are indistinguishable. But this is just the multinomial coefficient

$$\binom{k_1+k_2+\cdots+k_r}{k_1, k_1, \ldots, k_r} = \frac{(k_1+k_2+\cdots k_r)!}{k_1!\,k_2!\cdots k_r!} \qquad [1, \text{p. } 32].$$

Interpreting this trivial observation in $\mathcal{T}$ we find:

Theorem 1. *For each positive integer n, let $N(n)$ be the number of ordered rooted trees isomorphic to a given rooted tree $\phi(n)$.*

Then $N(n)$ is computed recursively as follows:
(1) $N(1)=1$
(2) *If* $n=p(n_1)^{k_1}p(n_2)^{k_2}\cdots p(n_r)^{k_r}$, *then*

$$N(n)=\binom{k_1+k_2+\cdots+k_r}{k_1,k_2,\ldots,k_r}N(n_1)N(n_2)\cdots N(n_r).$$

For applications, one frequently needs a catalogue of all rooted trees satisfying certain properties, or in our representation, a list of positive integers corresponding to such trees. Thus we need to translate the tree properties into number theoretic properties. The first step is to translate them into properties of completely reduced terms: the following characterization follows immediately from the definitions:

Theorem 2. *Let* $\mathbf{u}$ *be a completely reduced term in* $\mathcal{T}$. *Then*
(1) *The number of occurrences of p in* $\mathbf{u}$ = *the number of edges of* $\mathbf{u}$.
(2) *The number of occurrences of* **1** *in* $\mathbf{u}=1+$(*the number of occurrences of* $\otimes$ *in* $\mathbf{u}$) = *the number of leaves of* $\mathbf{u}$.
(3) *Each pair of matched brackets contains a completely reduced term of* $\mathcal{T}$, *so represents a rooted sub-tree of* $\mathbf{u}$.

Let P be any tree-theoretic property, and $T(P)$ the set of rooted trees having property P. Here is a general procedure for enumerating $T(P)$ which is what Read [13] calls an orderly algorithm:
(1) Find a lower bound m and an upper bound M for the set $\{n\in N:\phi(n)\in T(P)\}$.
(2) For each n between m and M, express n in completely reduced form and check whether $\phi(n)$ has property P.

Of course the success of this procedure depends on finding tight bounds for m and M and on translating P into properties of completely reduced terms. It turns out that in many common situations, both these tasks are feasible. Some powerful number theoretic results are needed to establish the following lemma.

Lemma 1. *Let n and k be positive integers.*
Then $p(kn)\geq kp(n)$ *iff* $(n,k)\neq(1,2),(1,3)$ or $(1,4)$.

Proof. The lemma is trivially true for $k=1$, and for $n=1$ and $k>4$.

For $k=2$, the Prime Number Theorem [10] shows that $p(2n)>2p(n)$ for all n greater than some n_0. The size of n_0 depends on the Riemann Hypothesis in the following way: the larger the first zero of the zeta function for which the Riemann Hypothesis fails, the smaller is n_0 [9]. Fortunately, Rosser Schoenfeld and Yohe have computed that the Riemann Hypothesis holds for the first 3,500,000 zeros of $\zeta(s)$ and have shown that this is enough to prove that $p(2n)>2p(n)$ for all $n\geq 11$ [15]. A trivial computation concludes the case $k=2$.

For $k \geq 3$ *and* $2 \leq n \leq 20$, we use induction on k and a result of Schinzel [16] that

$$\pi(x+y) \leq \pi(x) + \pi(y) \qquad \text{for } \min\{x, y\} \leq 146$$

where $\pi(x)$ means the number of primes $\leq x$. First fix n with $2 \leq n \leq 20$, and check that for $k = 3$, $p(kn) > kp(n)$. Assume that for some $k \geq 3$, $p(kn) > kp(n)$. Then

$$\begin{aligned} \pi((k+1)p(n)) &\leq \pi(kp(n)) + \pi p(n) \qquad (\text{since } p(20) = 71 < 146) \\ &< \pi(p(kn)) + (p(n)) = (k+1)n, \end{aligned}$$

so for the $(k+1)n$th prime, $p((k+1)n) > (k+1)p(n)$, as required.

It remains to deal with the case $k \geq 3$ and $n > 20$, and for this we use a result of Rosser and Schoenfeld [14, Theorem 3]. This theorem states that for any $n > 20$,

$$n(\log n + \log\log n - \tfrac{3}{2}) < p(n) < n(\log n + \log\log n - \tfrac{1}{2}).$$

Hence $p(kn) > kp(n)$ if

$$kn(\log n + \log\log n - \tfrac{1}{2}) < kn(\log kn + \log\log kn - \tfrac{3}{2}),$$

if $\log\log n + 1 < \log k + \log\log kn$, and this is true since $k \geq 3$.

Finally, we check that in the exceptional cases,

$$p(2) < 2p(1) < p(3) < 3p(1) < p(4) < 4p(1).$$

In order to express the following theorem succinctly we introduce notation for the iterates of p and $\otimes$:

$$p^n(\mathbf{1}) = p(p(\cdots p(\mathbf{1}))) \quad [n \text{ times}]; \qquad \mathbf{u}^k = \mathbf{u} \otimes \mathbf{u} \otimes \cdots \otimes \mathbf{u} \quad [k \text{ times}].$$

Theorem 3. *Let $\phi(n)$ be a rooted tree with $e \geq 3$ edges, and let $e = 3i + j$, where $0 \leq j < 3$.*

(1) *If $j = 0$, then $5^i \leq n \leq p^{e-3}(8)$;*

(2) *If $j = 1$, then $9 \times 5^{i-1} \leq n \leq p^{e-3}(8)$;*

(3) *If $j = 2$, then $3 \times 5^{i-1} \leq n \leq p^{e-3}(8)$.*

In each case, upper and lower bounds can be achieved.

Proof. Suppose $\mathbf{u} = \phi(n)$ is represented by a completely reduced term in $\mathscr{T}$ containing e occurrences of p.

Firstly we check that the theorem is true for all trees with $e = 3, 4$ or 5. Now assume that $n = p(n_1)p(n_2) \cdots p(n_k)$ and that the theorem is true for each $\phi(n_i)$ that has at least 3 edges.

For the lower bound, assume n_1 is minimal, so $p(n_1) = p(p^2(1)^i \times p^3(1)^j)$ for some $i = 0, 1$ or 2 and $j \geq 0$. By Lemma 1, $p(n_1)$ can be diminished without altering the number of occurrences of p by changing it to

$$p^2(1)^{i-1} \times p^3(1)^{j+1} \quad \text{if } i \neq 0,$$

or

$$p^3(1)^{j-1} \times p^4(1) \qquad \text{if } i = 0.$$

The latter can be further diminished to $p^3(1)^{j-1} \times p^2(1)^2$. Similar reductions applied to the other factors $p(n_i)$ and to the terms with less than 3 edges yields a product of the form $p^2(1)^i \times p^3(1)^j$, in which factors of the form $p^2(1)^3$ can be diminished by replacing them by $p^3(1)^2$ until the required form is achieved.

For the upper bound assume n_1 is maximal, so $p(n_1) = p^i(8)$ for some i. Now if $k > 1$, n can be increased by Lemma 1, since

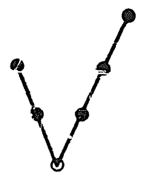
$$p^i(8) \times p^j(8) < p(p^{i-1}(8) \times p^j(8)) < \cdots < p^{i+j}(8).$$

Finally we note that no further reductions are possible since $p^2(1)$ is the least reduced term with 2 p's, $p^3(1)$ the least with 3, $p^2(1)^2$ the least with 4 and $p(1)^3$ the greatest with at most three.

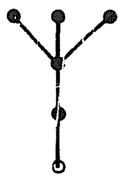
The lower bound is achieved by the rooted tree having $[\frac{1}{3}e]$ chains of length 3 and either 0, 1 or 2 chains of length 2, joined at the root.

The upper bound is achieved by the chain of length $e-3$ topped with a 3-leaved star.

Example. For trees with 5 edges, the minimum of 15 is achieved by



and the maximum of 67 by



Remark. The upper bound quickly becomes computationally infeasible since for eight edges we have $p^5(8) = 19\,577$. However things do not deteriorate so rapidly when extra conditions are imposed.

The proof of the following result is similar to that of Theorem 3:

Theorem 4. *Let $\phi(n)$ be a rooted tree with $e \geq 3$ edges and $l \geq 3$ leaves. Let $e = li + j$, where $0 \leq j < l$. Then*

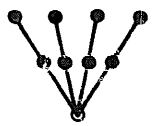
$$p^i(1)^{l-j} \times p^{i+1}(1)^j \leq n \leq p^{e-l}(2^l).$$

Both bounds are achieved.

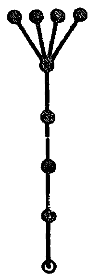
Table 2

L \ E	4	5	6	7	8
2	4	6	9	12	16
3	3	8	18	35	62
4		4	14	39	97
5			5	21	72
6				6	30
7					7

Note that the minimal tree consists of l chains, as near equal in length as possible, joined at the root. The maximal tree consists of a chain of length $e-l$, topped by an l-leaved star. For example for trees with 8 edges and 4 leaves, the minimum of 81 is achieved by



and the maximum of 12 763 by



Now a slight modification to the programme used to produce Table 1 will enumerate the trees having e edges and l leaves for fixed e and l. For example Table 2 presents a count of such trees for $e=4,\ldots,8$ and $l=2,\ldots,7$.

5. Simple presentations of finite abelian p-groups

Let p be a prime, and G a finite abelian p-group. Define subgroups

$$p^nG=\{p^nx : x\in G\} \quad \text{for all } n=0,1,\ldots$$

and

$$p^nG[p]=\{x\in p^nG : px=0\}.$$

There is a minimal e, called the *exponent* of G, for which $p^eG=0$. A non-zero $x\in G$ is said to have *height* n if $x\in p^nG\backslash p^{n+1}G$. Each $p^nG[p]$ admits the field with p elements as operators in the obvious way, and the dimension of the vector space $p^nG[p]$ is denoted v_n. The sequence $(v_0, v_1, \ldots, v_{e-1})$ is called the *height distribution* of G. It is a trivial consequence of the Fundamental Theorem of Finite Abelian Groups that the height distribution is a complete invariant of G, which means that two groups have the same height distribution if and only if they are isomorphic.

The height distribution is related to the better known Ulm invariant [4, p. 154] as follows: for $n=0,1,\ldots$ let u_n denote the dimension of the space $p^nG[p]/p^{n+1}G[p]$. Then the sequence $(u_0, u_1, \ldots, u_{e-1})$ is the Ulm invariant of G. In case G is infinite, this provides more information than the height distribution, but in our case, they are equally informative, since:

$$u_n = v_n - v_{n+1} \quad \text{for } n<e-1, \qquad u_{e-1} = v_{e-1},$$

and conversely, $v_n=\sum_{j=n}^{e-1} u_j$ for $n\leq e-1$. Thus $v_0\geq v_1\geq\cdots\geq v_{e-1}$, and to each such sequence there exists a finite abelian p-group G having height distribution $(v_0, v_1, \ldots, v_{e-1})$, unique up to isomorphism.

In case G is expressed as a direct sum of cyclic subgroups, say

$$G=[a_1]\oplus[a_2]\oplus\cdots\oplus[a_k],$$

where a_i has order $p^{n(i)}$, then for each order p^j, the number of summands of order p^j is the jth term u_{j-1} of the Ulm invariant.

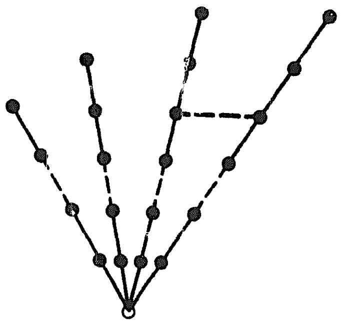
In 1969, Crawley and Hales [2] characterized an important class of abelian p-groups, which includes the finite ones, called the *simply presented groups*. These are the abelian p-groups having a presentation of the form $G=\langle X; R\rangle$, where X is an irredundant set of generators such that, if $x\in X$, then $px=0$ or $px\in X$, and all relations in R are of the form $px=0$ or $px=y$.

For example the group G defined above has a simple presentation $G=\langle X; R\rangle$, where

$$X=\{a_1, pa_1, \ldots, p^{n(1)-1}a_1; a_2, pa_2, \ldots, p^{n(2)-1}a_2; \ldots; a_k, pa_k, \ldots, p^{n(k)-1}a_k\}.$$

For any simply presented group G, G is finite if and only if X is, and in this case, a simple presentation $\langle X; R\rangle$ defines a unique rooted tree $T(\langle X; R\rangle)$ as follows: the vertices are $X\cup\{0\}$, 0 is the root, and there is an arrow from vertex x to vertex y if and only if $px=y$. Conversely, a rooted tree $\mathbf{u}$ defines a simple presentation $P(\mathbf{u})=\langle X; R\rangle$ in which X is the set of non-root vertices, the root is the group identity 0, and $px=y$ is a relation in R if and only if there is an arrow from vertex x to vertex y. Here we are considering a rooted tree as a directed graph with all arrows directed towards the root.

For example, in the case of G defined above, $T(\langle X; R\rangle)$ has the form



where the ith chain has length $n(i)$. A reduced term representing this tree is $p^{n(1)}(\mathbf{1})\otimes p^{n(2)}(\mathbf{1})\otimes\cdots\otimes p^{n(k)}(\mathbf{1})$.

A finite abelian p-group has in general many simple presentations, and in 1971, Hales [8] posed the following problem: find the set of all different rooted trees which correspond to simple presentations of a given finite abelian p-group G.

It follows from the first paragraph of this section that this is equivalent to finding all the rooted trees $\mathbf{u}$ such that $P(\mathbf{u}) = \langle X; R\rangle$ represents a group G with a given height distribution. Now it is a consequence of Theorems 3.4 and 3.5 of [2] that for all $x\in X$, $x\in p^nG$ if and only if there is a path of length n in $\mathbf{u}$ which terminates at x, and the exponent of G is the length of the longest path in $\mathbf{u}$. So let us define the *height* of a vertex x of $\mathbf{u}$ to be the length of the longest path in $\mathbf{u}$ terminating at x, the *exponent* e of $\mathbf{u}$ to be the height of the root, and the *height distribution* of $\mathbf{u}$ to be the sequence $(v_0, v_1, \ldots, v_{e-1})$, where v_i is the number of vertices of height i. Note that $v_0 \geq v_1 \geq \cdots \geq v_{e-1}$.

Finally, it is an easy consequence of Theorems 3.4 and 3.5 of [2] that the dimension of $p^nG[p]$ is precisely the number of vertices of $\mathbf{u}$ whose height is n. In other words, the height distribution of G is just the height distribution of any rooted tree $\mathbf{u}$ such that $P(\mathbf{u})$ is a simple presentation of G, and Hales' problem boils down to finding all rooted trees with a given height distribution.

The effect of the operations in $\mathcal{T}$ on height distributions is clear: If $\mathbf{u}_1$ has height distribution $d_1 = (v_0, v_1, v_2, \ldots, v_k)$, and $\mathbf{u}_2$ has height distribution $d_2 = (w_0, w_1, w_2, \ldots, w_j)$ with $j \leq k$, then $\mathbf{u}_1\otimes\mathbf{u}_2$ has height distribution $d_1 + d_2 = (v_0 + w_0, \ldots, v_j + w_j, v_{j+1}, \ldots, v_k)$, and $p(\mathbf{u}_1)$ has height distribution $(v_0, v_1, \ldots, v_k, 1)$.

6. Transformations of rooted trees

We now present an algorithm which will determine the height distribution of a rooted tree from its completely reduced form, and then show how completely reduced terms with the same height distribution are related. Both results are accomplished by means of transformations of completely reduced terms.

6.1. *Absorption*

Let $\mathbf{u}$ be a rooted tree with height distribution $(v_0, v_1, \ldots, v_k)$. We have previously noted that v_0 = number of leaves in $\mathbf{u}$; v_1, being the number of vertices of height 1 in $\mathbf{u}$, is the number of vertices of height 0 in the rooted tree $A(\mathbf{u})$ obtained by deleting all the leaves of $\mathbf{u}$; for $i = 2, \ldots, e-1$, v_i is the number of vertices of height 0 in the rooted tree $A^i(\mathbf{u})$ obtained by deleting all the leaves of $A^{i-1}(\mathbf{u})$.

Therefore we define the operation absorption on completely reduced terms as follows:

$A(n)$ is the completely reduced term obtained from n by replacing each occurrence of $p(1)$ by 1, and then using identity property to rewrite the resulting term in completely reduced form.

For example, if $n = p(p(1) \times p(p(1)) \times p(p(p(1))))$, then

$$A(n) = p(p(1) \times p(p(1))), \quad \text{and} \quad A^2(n) = p(p(1)).$$

Since this operation corresponds to deleting the leaves of $\phi(n)$, the height distribution of $\phi(n)$ is $(v_0, v_1, \ldots, v_{e-1})$, where v_i = number of occurrences of 1 in $A^i(n)$.

Table 3 is a list of height distributions of the first 100 trees, produced by a computer program using the algorithm just described.

6.2. *Normalization*

In order to apply the orderly algorithm to list the rooted trees with a given height distribution, we need upper and lower bounds. We next define a transformation which replaces each rooted tree by a smaller one having the same height distribution.

Each reduced term $n > 1$ has the form

$$n = p^m(p^{i_1}(n_1)p^{i_2}(n_2) \cdots p^{i_k}(n_k))$$

where the n_j are reduced terms, $m \geq 0$ and k and the $i_j \geq 1$. If $m \neq 0$, i.e. if the root of $\phi(n)$ has degree 1, rearrange the terms inside the main bracket so that $i_1 = \max\{i_1, i_2, \ldots, i_k\}$.

The normalization of n is the reduced term

$$N(n) = p^{m-1}(p^{i_1+1}(n_1)p^{i_2}(n_2) \cdots p^{i_k}(n_k)).$$

Remarks. (1) By Lemma 1, $N(n) < n$; the exceptional cases of Lemma 1 cannot occur since $i_1 = \max\{i_1, i_2, \ldots, i_k\}$, so the term in the main bracket is at least 4.

(2) The effect of normalization on the corresponding rooted tree is to pick out a longest path as the main trunk and to slide the rest of the tree one edge down the trunk.

(3) After normalization, the terms in the main bracket can be rearranged to restore the completely reduced form.

Table 3

2 (1)	3 (1, 1)	4 (2)
5 (1, 1, 1)	6 (2, 1)	7 (1, 1)
8 (3)	9 (2, 2)	10 (2, 1, 1)
11 (1, 1, 1, 1)	12 (3, 1)	13 (2, 1, 1)
14 (3, 1)	15 (2, 2, 1)	16 (4)
17 (2, 1, 1)	18 (3, 2)	19 (3, 1)
20 (3, 1, 1)	21 (3, 2)	22 (2, 1, 1, 1)
23 (2, 2, 1)	24 (4, 1)	25 (2, 2, 2)
26 (3, 1, 1)	27 (3, 3)	28 (4, 1)
29 (2, 1, 1, 1)	30 (3, 2, 1)	31 (1, 1, 1, 1, 1)
32 (5)	33 (2, 2, 1, 1)	34 (3, 1, 1)
35 (3, 2, 1)	36 (4, 2)	37 (3, 1, 1)
38 (4, 1)	39 (3, 2, 1)	40 (4, 1, 1)
41 (2, 1, 1, 1)	42 (4, 2)	43 (3, 1, 1)
44 (3, 1, 1, 1)	45 (3, 3, 1)	46 (3, 2, 1)
47 (2, 2, 1, 1)	48 (5, 1)	49 (4, 2)
50 (3, 2, 2)	51 (3, 2, 1)	52 (4, 1, 1)
53 (4, 1)	54 (4, 3)	55 (2, 2, 2, 1)
56 (5, 1)	57 (4, 2)	58 (3, 1, 1, 1)
59 (2, 1, 1, 1)	60 (4, 2, 1)	61 (3, 2, 1)
62 (2, 1, 1, 1, 1)	63 (4, 3)	64 (6)
65 (3, 2, 2)	66 (3, 2, 1, 1)	67 (3, 1, 1)
68 (4, 1, 1)	69 (3, 3, 1)	70 (4, 2, 1)
71 (3, 1, 1, 1)	72 (5, 2)	73 (3, 2, 1)
74 (4, 1, 1)	75 (3, 3, 2)	76 (5, 1)
77 (3, 2, 1, 1)	78 (4, 2, 1)	79 (2, 1, 1, 1, 1)
80 (5, 1, 1)	81 (4, 4)	82 (3, 1, 1, 1)
83 (2, 2, 1, 1)	84 (5, 2)	85 (3, 2, 2)
86 (4, 1, 1)	87 (3, 2, 1, 1)	88 (4, 1, 1, 1)
89 (4, 1, 1)	90 (4, 3, 1)	91 (4, 2, 1)
92 (4, 2, 1)	93 (2, 2, 1, 1, 1)	94 (3, 2, 1, 1)
95 (4, 2, 1)	96 (6, 1)	97 (2, 2, 2, 1)
98 (5, 2)	99 (3, 3, 1, 1)	100 (4, 2, 2)

(4) Normalization preserves height distributions.

We now define a completely normalized tree to be one of the form

$$\mathbf{u} = p^{i_1}(\mathbf{1}) \otimes p^{i_2}(\mathbf{1}) \otimes \cdots \otimes p^{i_k}(\mathbf{1}).$$

Clearly $\mathbf{u}$ consists of k chains of lengths $i_1, i_2, \ldots, i_k$ joined at the root.

It is evident that any rooted tree can be transformed to a unique completely normalized tree with the same height distribution by successive applications of normalization to its corresponding number in completely reduced form.

Theorem 5. (1) *For each rooted tree* $\mathbf{v}$ *there exists a unique completely normalized tree* $\mathbf{u}$ *with the same height distribution.*

(2) *Of all trees with a given height distribution, the completely normalized tree is the minimum.*

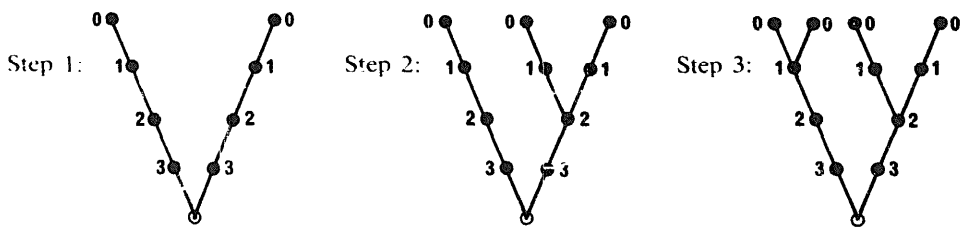
Having found a canonical tree and a tight lower bound on values of trees with a given height distribution, we must now find a tight upper bound. First, note that if $\mathbf{u}$ has height distribution $d = (v_0, v_1, \ldots, v_k)$, then $\mathbf{u}$ has $\sum_{i=0}^{k} v_i$ edges and v_0 leaves, so Theorem 4 gives a coarse upper bound, and the orderly algorithm can in principle proceed. One would like a better upper bound, however. Unfortunately, absorption and normalization are not in general reversible, so a recursive procedure is required to find the maximum.

6.3. *Hashing*

Let $\mathbf{u}$ be a rooted tree with height distribution $d = (v_0, v_1, \ldots, v_{e-1})$. Then $\mathbf{u}$ contains v_{e-1} chains of length e, joined to the root. Label their vertices with their height. Now suppose the tree $\mathbf{u}$ has been partly constructed to form a tree $\mathbf{w}$ with height distribution $d' = (v_k, v_k, \ldots, v_k, v_{k+1}, \ldots, v_{e-1})$, where the last $e-k$ entries are the last $e-k$ entries of d, and the first k are constant, and suppose $\mathbf{u}'$ is the maximum tree with height distribution d'. Label the vertices of $\mathbf{u}'$ with their height.

Now take $w = v_{k-1} - v_k$ chains of length k and consider them attached to the v_k vertices of $\mathbf{u}'$ with label k in all possible ways: there are v_k^w such ways. Choose the maximum of all the resulting trees for the next step of the recursive construction of $\mathbf{u}$.

For example, suppose $d = (4, 3, 2, 2)$



We call this procedure *hashing*, and after e applications of hashing, the resulting tree is called *hashed*. We must now verify that hashing always produces the maximum rooted tree of given height distribution.

Lemma 2. *Let $\mathbf{u}'$ be the maximum rooted tree with height distribution $d' = (v_k, v_k, \ldots, v_k, v_{k+1}, \ldots, v_{e-1})$, where $k < e-1$.*

Let $\mathbf{u}$ be the tree obtained by hashing $\mathbf{u}'$. Then $\mathbf{u}$ is the maximum tree with height distribution $d = (v_{k-1}, v_{k-1}, \ldots, v_{k-1}, v_k, \ldots, v_{e-1})$.

Proof. Firstly note note that since hashing adds $w = v_{k-1} - v_k$ vertices of each height $0, 1, \ldots, k$, $\mathbf{u}$ does have height distribution d.

Now let $\bar{\mathbf{u}}$ be the unique maximum tree with height distribution d; $\bar{\mathbf{u}}$ can be obtained from a tree $\bar{\mathbf{u}}'$ of height distribution d' by attaching w chains of length k to vertices of height k, for if one were attached to a vertex of greater height, $\bar{\mathbf{u}}$ could be increased by moving this chain.

But since both $\bar{\mathbf{u}}'$ and $\mathbf{u}'$ have the same number of vertices of height $0, 1, \ldots, k$, the effect of attaching say s such chains to a vertex of height k is to transform a subtree $p^k(1)$ into $p^k(1)^s$.

Now let θ be a 1–1 correspondence between vertices of height k in $\bar{\mathbf{u}}'$ and vertices of height k in $\mathbf{u}'$. If in constructing $\bar{\mathbf{u}}$ from $\bar{\mathbf{u}}'$, s chains of length k were attached to vertex x, then attach s chains of length k to vertex $\theta(x)$ in $\mathbf{u}'$. Continue this procedure until $\bar{\mathbf{u}}$ is reconstructed, and let $\mathbf{u}''$ be the corresponding transformation of $\mathbf{u}'$.

If $\bar{\mathbf{u}}' < \mathbf{u}'$ before this procedure, then $\bar{\mathbf{u}} < \mathbf{u}''$ after the procedure. But this contradicts maximality of $\bar{\mathbf{u}}$, so $\bar{\mathbf{u}}' > \mathbf{u}'$; since $\mathbf{u}'$; since $\mathbf{u}'$ is the maximum tree of height distribution d', this means $\bar{\mathbf{u}}' = \mathbf{u}'$. But $\mathbf{u}$ itself is the maximum tree which can be obtained from $\mathbf{u}'$ by attaching w chains of length k, so $\mathbf{u} = \bar{\mathbf{u}}$.

This completes the proof of the following theorem:

Theorem 6. (1) *For each rooted tree* $\mathbf{v}$ *there exists a unique hashed tree* $\mathbf{u}$ *having the same height distribution.*

(2) *Of all trees with a given height distribution, the hashed tree* $\mathbf{u}$ *is the maximum.*

We can now present the algorithm for enumerating all rooted trees with given height distribution $d = (v_0, v_1, \ldots, v_{e-1})$.

Let $\phi(m)$ be the completely normalized tree, and $\phi(M)$ the hashed tree, with height distribution d.

For $n = m$ to M, $\phi(n)$ has height distribution d iff for $i = 0$ to $e - 1$, the number of occurrences of 1 in $A^i(n) = v_i$.

For example, let $d = (3, 2, 1)$. Then $m = 30$, $M = 73$ and the seven trees with height distribution d are:

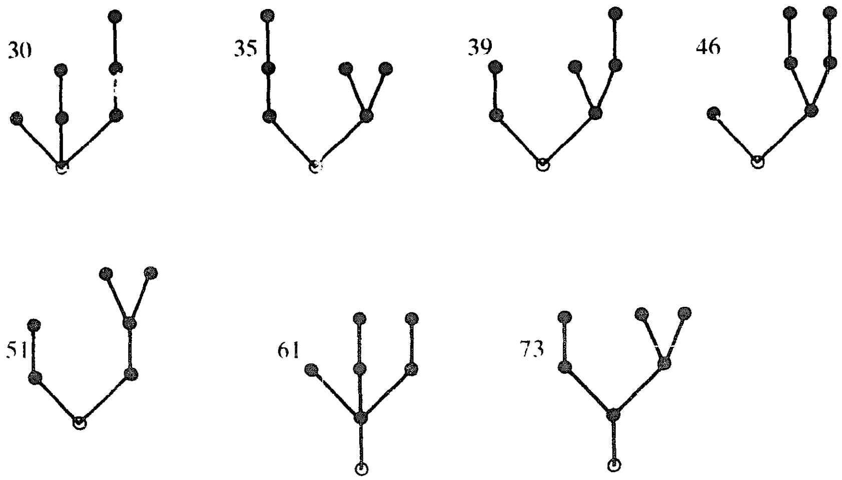


Table 4 is a count of the number of trees with various height distributions.

Table 4

Height distribution	No. of trees
(4, 2, 1)	17
(4, 2, 2)	9
(4, 3, 1)	11
(4, 3, 2)	9
(5, 3, 1)	29
(3, 2, 1, 1)	14
(4, 2, 1, 1)	41
(4, 3, 3, 1)	16

References

[1] C. Berge, Principles of Combinatorics (Academic Press, New York, 1971).

[2] P. Crawley and A.W. Hales, The structure of Abelian p-groups given by certain presentations, J. Algebra 12 (1969) 10-23.

[3] N. Dershowitz and S. Zaks, Enumerations of ordered trees, Discrete Math. 31 (1) (1980) 9–28.

[4] L. Fuchs. Infinite Abelian Groups, Vol. I (Academic Press, New York, 1970).

[5] F. Göbel, On a 1–1 correspondence between rooted trees and natural numbers, J. Combin. Theory (B) 29 (1980) 141–143.

[6] M. Gordon and W.B. Temple, The graph-like state of matter and polymer sciences, in: A.T. Balaban, ed., Chemical Applications of Graph Theory (Academic Press, New York, 1976) 300–332.

[7] M. Gordon and J. A. Torkington, Enumeration of coloured plane trees with a given type partition, Discrete Appl. Math. 2(3) (1980) 207–224.

[8] A.W. Hales, Combinatorial Representations of Abelian Groups, Proc. of Symposia in Pure Mathematics 19 (1971) 105–108.

[9] A.E. Ingham, The Distribution of Prime Numbers (C.U.P., 1932).

[10] E. Landau, Handbuch der Lehre van der Verteilung der Primzahlen, Vol. 1 (New York, 1953).

[11] J.W. Moon, in: F. Harary, ed., Graph Theory and Theoretical Physics (Academic Press, London, 1967).

[12] R.C. Read, The coding of various kinds of unlabeled trees, in: Graph Theory and Computing (Academic Press, New York, 1972) 153–183.

[13] R.C. Read, Every one a winner—or—How to avoid isomorphism search when cataloguing combinatorial configurations, in: B. Alspach, P. Hell and D.J. Miller, Algorithmic Aspects of Combinatorics, Ann. Discrete Math. 2 (North-Holland, Amsterdam, 1978) 107–120.

[14] J.B. Rosser and L. Schoenfeld, Approximate formulas for some functions of prime number theory, Illinois J. Math. 6 (1962) 69–94.

[15] J.B. Rosser, L. Schoenfeld and J.M. Yohe, Rigorous computation and the zeros of the Riemann Zeta Function, in: Information Processing 68 (North-Holland, Amsterdam, 1969) 70–75.

[16] A. Schinzel, Remarks on the paper "Sur certaines hypothèses concernant les nombres premiers", Acta Arith. 7 (1961) 1–8.