COMPUTING VALUATED TREES

DONNA BEERS, ROGER HUNTER, FRED RICHMAN, and ELBERT A. WALKER

Abstract In [3] it was pointed out that (equivalence classes of) valued trees are natural generalizations of ordinals, and play a fundamental role in Ulm's theorem for simply presented valued p-groups. In this paper we show how to compute the distributive lattice of finite valued trees with values not exceeding the nonnegative integer n. For n ≤ 6, this computation was carried out completely. Unexpectedly, the lattices turned out to be almost self-dual. We explain this duality in the last section.

1. THE LATTICE OF FINITE VALUATED TREES

All trees in this paper will be rooted and have no elements of exponent 0 (see [3] for terminology). If T₁ and T₂ are valued trees, then we write T₁ ≤ T₂ if there is a map from T₁ to T₂. For n a nonnegative integer, the (isomorphism classes of) finite valued trees with values not exceeding n form a quasi-ordered set Sₙ under this ordering.

An ideal I in a quasi-ordered set Q is a subset I of Q such that if t ≤ s ∈ I, then t ∈ I. The set Q of ideals in Q is a complete distributive lattice. If every finite subset of Q has an infimum, then the intersection of two finitely generated ideals in Q is finitely generated.
By a branch of tree we mean a node \( x \) just above the root, together with all nodes above \( x \). The next theorem shows that to compare trees in \( S_{r+1} \), we compare their root values, and we compare the ideals in \( S_n \) generated by their branches.

1.1 THEOREM. Define the map \( \varphi : S_{r+1} \rightarrow \{0, \ldots, n+1\} \times S_n \) by \( \varphi(T) = (r, I) \) where \( r \) is the value of the root of \( T \), and \( I \) is the ideal generated by the branches of \( T \). Then \( T_1 \leq T_2 \) if and only if \( \varphi(T_1) \leq \varphi(T_2) \).

PROOF. If there is a map from \( T_1 \) to \( T_2 \), then by [3: 3.1] we can find a map from \( T_1 \) to \( T_2 \) that takes root to root, and maps each branch of \( T_1 \) into some branch of \( T_2 \). Thus \( \varphi(T_1) \leq \varphi(T_2) \). Conversely if \( \varphi(T_1) \leq \varphi(T_2) \), then each branch of \( T_1 \) maps into some branch of \( T_2 \), so we can define a map from \( T_1 \) to \( T_2 \) by piecing together these maps, and taking root to root.

The image \( \varphi(S_n) \) is the set of all \( (r, I) \in \{0, \ldots, n\} \times S_n \) such that \( r \) exceeds any value occurring in a tree of \( I \). By (1.1), this image is isomorphic to the partially ordered set of equivalence classes of elements of the quasi-ordered set \( S_n \).

1.2 COROLLARY. The set of ideals \( S_n \) is finite.
PROOF. The set $\mathcal{F}_0$ has exactly one element: note that $\mathcal{F}_n$ is not finite unless $n = 0$. There is a natural one-to-one correspondence between the ideals in a quasi-ordered set and the ideals in the corresponding partially ordered set: so $\mathcal{F}_{n+1} \equiv \mathcal{F}(\mathcal{F}_n)$. But $\mathcal{F}(\mathcal{F}_{n+1}) \subset (0, \ldots, n+1) \times \mathcal{F}_n$ is finite by induction, so $\mathcal{F}_{n+1}$ is finite. \(\square\)

1.3 THEOREM. $\mathcal{F}(\mathcal{F}_{n+1})$ is a sublattice of the distribution lattice $\{0, \ldots, n+1\} \times \mathcal{F}_n$.

PROOF. Suppose $\mathcal{F}(T_1) = (r_1, I_1)$ and $\mathcal{F}(T_2) = (r_2, I_2)$. If $T$ is the tree obtained by identifying the root of $T_1$ with the root of $T_2$, and valuating the resulting root with the maximum of $r_1$ and $r_2$, then $\mathcal{F}(T)$ is the supremum of $\mathcal{F}(T_1)$ and $\mathcal{F}(T_2)$. If $T$ is the tree whose root has value the minimum of $r_1$ and $r_2$, and whose branches are a generating set of $I_1 \cap I_2$, then $\mathcal{F}(T)$ is the infimum of $\mathcal{F}(T_1)$ and $\mathcal{F}(T_2)$. \(\square\)

We will identify the set $\mathcal{F}_n$ with the set of equivalence classes of $\mathcal{F}_n$, which is isomorphic to $\mathcal{F}(\mathcal{F}_n)$. As $\mathcal{F}_n$ is finite, it satisfies the descending chain condition. Any partially ordered set satisfying the descending chain condition is partitioned in a natural way into levels. Our algorithm for generating $\mathcal{F}_n$ generates it one level at a time.
1.4 THEOREM. Let \( P \) be a partially ordered set satisfying the descending chain condition. Then there exists a unique function \( L \) from \( P \) to the ordinals such that
\[
L(s) = \sup\{L(t)+1 : t < s\}.
\]

PROOF. To show uniqueness, suppose that \( L' \) is another such function and that \( L'(t) = L(t) \) whenever \( L(t) \leq \alpha \). If \( L(s) = \alpha \), then
\[
L(s) = \sup\{L(t)+1 : t < s\} = \sup\{L'(t)+1 : t < s\} = L'(s).
\]
To show that such a function exists, define \( P_\beta \) inductively to be the set of minimal elements of \( \bigcap_{\alpha \leq \beta} P_\alpha \) and let \( L(s) = \beta \) if \( s \in P_\beta \). \( \Box \)

We say that \( t \) is an immediate successor of \( s \) if \( s < t \) and there is no \( r \) such that \( s < r < t \). Our algorithm for generating \( P_n \) generates level \( n+1 \) as the set of all immediate successors of elements of level \( n \). This is justified by the following theorem.

1.5 THEOREM. If \( P \) is a lower semimodular lattice satisfying the descending chain condition, and \( t \) is an immediate successor of \( s \) in \( P \), then \( L(t) = L(s) + 1 \).

PROOF. We use the term lower semimodular in the sense of Crawley and Dilworth [1; p.23]; substitute modular if you don't like this usage—in any event, our lattices are
distributive. It suffices to show that if \( r < t \), then \( L(r) \preceq L(s) \). As \( t \) is an immediate successor of \( s \), either \( r \vee s = s \), or \( r \vee s = t \). If \( r \vee s = s \), then \( r \preceq s \), so \( L(r) \preceq L(s) \). If \( r \vee s = t \), then, by lower semimodularity, \( r \) is an immediate successor of \( r \wedge s \). Hence by induction on \( L(r) \) we have \( L(r) = L(r \wedge s) + 1 \preceq L(s) \).

We will need to construct immediate successors in \( S^n \).

We do this as follows.

**1.6 Theorem.** Let \( T_1 \) and \( T_2 \) be trees in \( S^n \). Then \( T_2 \) is an immediate successor of \( T_1 \) if and only if \( \psi(T_2) \) is an immediate successor of \( \psi(T_1) \) in \( \psi(S^{n-1}) \).

**Proof.** The "if" part is clear. Let \( (r_1, I_1) = \psi(T_1) \), and suppose that \( T_2 \) is an immediate successor of \( T_1 \). Then \( r_1 \leq r_2 \) and \( I_1 \subseteq I_2 \). If \( r_1 < r_2 \), then \( (r_1, I_1) = \psi(T) \), so \( T_1 \preceq T \preceq T_2 \), so \( r_2 = r_1 + 1 \). If \( r_1 = r_2 \), and \( I_1 \subseteq I_1 \subseteq I_2 \), then \( (r_1, I_1) = \psi(T) \), so \( T_1 \preceq T \preceq T_2 \); therefore \( I_2 \) is an immediate successor of \( I_1 \).

From Theorem 1.6 we see that there are two ways to construct immediate successors of a tree \( T \) in \( S^n \). One is to increase the value of the root of \( T \) by one, without changing the values on the branches. The other is to leave the root of \( T \) alone, and replace the set \( B \) of branches of \( T \) by a
generating set for an immediate successor of the ideal generated by \( B \). Thus we must know how to construct the immediate successors of an ideal, which we do as follows.

1.7 THEOREM. Let \( I \) and \( J \) be ideals in a partially ordered set \( P \). Then the following are equivalent.

(i) \( J \) is an immediate successor of \( I \) in \( \mathcal{A} \)
(ii) \( J = I \cup \{ s \} \) for some minimal element \( s \in \mathcal{A} \)
(iii) \( I = \mathcal{A} \backslash \{ s \} \) for some maximal element \( s \in J \).

PROOF. Suppose (i). Choose an element \( s \in \mathcal{A} \). Then \( s \) is a minimal element of \( \mathcal{A} \), and \( I \cup \{ s \} \) is an ideal contained in \( J \) and properly containing \( I \), so \( J = I \cup \{ s \} \).

Now suppose (ii). Then clearly \( s \) is a maximal element of \( J \), and \( I = J \backslash \{ s \} \).

Finally suppose (iii). Clearly there are no ideals strictly between \( I \) and \( J \). 0

2. IRRETRACTIBLE TREES

In this section, we show that the equivalence class of a finite valued tree contains a canonical smallest tree, namely the unique irretractible tree in that class. The algorithms for generating \( \mathcal{T}_n \) described in the next section actually generates these irretractible trees.

A retraction of a valued tree \( T \) is a map \( r : T \to T \)
such that \( r^2 = r \). If \( r \) is a retraction of \( T \), then \( T \) and \( rT \) are certainly equivalent. A retraction is nontrivial if it is different from the identity map. A valuated tree is irretractible if it has no nontrivial retractions. From [2: Lemma 3] we see that a valuated tree is irretractible if and only if its branches are irretractible and incomparable. In this section we will allow our trees to have elements of exponent 0.

2.1 Lemma. Let \( T \) be a valuated tree. If \( T \) has an endomorphism different from the identity, then \( T \) has a nontrivial retraction, or an element of height \( \omega \).

Proof. Let \( f \) be an endomorphism of \( T \), and let \( x \in T \) be a node of minimal exponent such that \( f(x) \neq x \). If \( f(x) > x \), then \( h(x) = \omega \) as \( f \) cannot decrease height. If \( f(x) < x \), then \( T \) has an element of exponent 0 by [3: 3.1], in which case mapping \( T \) to that element is a retraction. Otherwise define \( r : T \to T \) by

\[
\begin{align*}
  r(y) &= f(y) \quad \text{if } x \leq y \\
  r(y) &= y \quad \text{otherwise}.
\end{align*}
\]

Clearly \( r \) is nontrivial. If \( x \leq y \), then \( r(y) = f(y) \geq f(x) \) cannot lie above \( x \), so \( r^2(y) = r(y) \), demonstrating that \( r \) is a retraction.
2.2 THEOREM. If $T_1$ and $T_2$ are irretractible valued trees such that $T_1 \leq T_2$ and $T_2 \leq T_1$, then $T_1$ and $T_2$ are isomorphic.

PROOF. If $f : T_1 \to T_2$ and $g : T_2 \to T_1$ are maps, then $gf$ and $fg$ are endomorphisms of $T_1$ and $T_2$, hence are trivial by Lemma 2.1, unless either $T_1$ or $T_2$ has an element of height $= 1$. In the latter case, as the $T_i$ are irretractible, they are either both isomorphic to the one-element tree whose node is of exponent 0, or they are both isomorphic to the tree whose order type is that of the positive integers (the tree whose corresponding p-group is $\mathbb{Z}^{\infty}$)).

2.3 COROLLARY. If $T$ is a finite valued tree, then $T$ has a unique irretractible retraction, which is the unique irretractible tree equivalent to $T$.

3. ALGORITHMS FOR GENERATING $S_n$

In this section we describe the algorithms that we used for generating trees on a computer. The first algorithm generates $S_n$ from the bottom up, while the second generates $S_n$ from the top down. Both algorithms are necessary for investigating lattices that are too large to generate in their entirety.

A tree is represented by a pair whose first element is
Given an irretractible tree $T$ at level $h-1$, the up-edges (immediate successors) are computed in two ways, according to Theorem 1.6. If the root of $T$ is less than $n$, then increasing the root value by one yields an immediate successor. The second way is more interesting, and occurs much more frequently. Let $(r,I) = φ(T)$, and note that, since $T$ is irretractible, the branches of $T$ are the maximal elements of $I$. The second kind of successors of $T$ are of the form $(r,I')$ where $I'$ is an immediate successor of $I$ in $\mathcal{F}_{n-1}$. The elements of $I$ are below level $h-1$, so their immediate successors are known. Thus, we can compute the minimal elements of $\mathcal{F}_{n-1} \setminus I$ directly from the part of the lattice we have already constructed. By Theorems 1.6 and 1.7, the trees with the same root value as $T$, and whose branches are the branches of $T$ together with a minimal element $M$ of $\mathcal{F}_{n-1} \setminus I$, are immediate successors of $T$. To get an irretractible immediate successor, we delete those branches of $T$ that are immediate predecessors of $M$.

Going Backwards.

Here we need portions of all of the previous lattices before we can start an $\mathcal{F}_n$. We compute the top level first, and compute level $h-1$ by computing the immediate predecessors of trees $T$ at level $h$. As in the forward algorithm, there are two cases. In one we decrease the value of the root of $T$. 
element trees; this gives \( n+2 \) levels. Let \( T_1, T_2, \ldots, T_N \) be
an enumeration of \( \mathcal{T}_n \) such that if \( i \leq j \), then the level of
\( T_i \) does not exceed the level of \( T_j \). Then for each \( j \) the set
\( I_j = \{ T_i : i \leq j \} \) is an ideal, and \( T_{j+1} \) a minimal tree not
in \( I_j \). Let \( S_j \) be the tree with root value \( n+1 \) and set of
branches \( I_j \). Then \( S_0 \) is the one-element tree with value
\( n+1 \), and \( S_j \) is an immediate successor of \( S_{j-1} \) for \( j = 1, \ldots, N \). Clearly \( S_N \) is the biggest tree in \( \mathcal{T}_{n+1} \), so is at
the highest level. \( n \)

The first few lattices are easily computed by hand.

Here is a picture of \( \mathcal{T}_3 \), which contains the smallest tree
with two branches.
4. PATHS AND SPARSE TREES.

As a check on the algorithm, we examined two classes of trees which can be counted without the algorithm.

A tree is called a path if it is linearly ordered. Each path in $\mathcal{S}_n$ is specified by its set of values, which is a nonempty subset of $\{0, \ldots, n\}$; so the number of paths in $\mathcal{S}_n$ is $2^{n+1} - 1$. The paths in $\mathcal{S}_n$, together with the empty tree, form a self-dual lattice, the duality taking a path corresponding to a subset $S$, to the path corresponding to the complement of $S$. This lattice is not a sublattice of $\mathcal{S}_n$. We had the computer go through $\mathcal{S}_n$ for $n \leq 6$, and count the paths; the counts agreed with the theoretical values, showing that the algorithm probably does something right.

Call a tree in $\mathcal{S}_n$ sparse if the set of values of its
nodes is not \( \{0, 1, \ldots, n\} \). The number of sparse trees in \( S_n \) can be computed from the numbers of trees in \( S_i \) for \( i \leq n \).

### 4.1 Theorem

Let \( t_n \) be the number of trees in \( S_n \) and let \( p_n \) be the number of non-sparse trees in \( S_n \). Then

\[
t_n = p_n + \binom{n+1}{n} p_{n-1} + \binom{n+1}{n-1} p_{n-2} + \cdots + \binom{n+1}{1} p_0.
\]

**Proof.** To each non-sparse tree \( T \) in \( S_{n-j} \) there corresponds \( \binom{n+1}{n+1-j} \) trees in \( S_n \) upon replacing the \( n+1-j \) values on \( T \) by the different subsets of \( \{0, \ldots, n\} \) of cardinality \( n+1-j \). Moreover each tree in \( S_n \) is constructed in this way from a unique \( j \), and tree \( T \) in \( S_{n-j} \). \( \square \)

We had the computer count the sparse trees in the lattices it generated; the counts were consistent with Theorem 4.1.

### 5. Empirical Duality in \( S_n \)

Our algorithm generates the trees in \( S_n \) level by level. As each level was generated, we printed out the number of trees in it; the resulting list of numbers exhibited a striking symmetry. For \( S_6 \) we get the following table of the number of trees in each level.

\[
\begin{array}{ccc}
\text{level} & \text{trees} & \text{level} & \text{trees} \\
1 & 223 & -1 & 221 \\
2 & 222 & -2 & \end{array}
\]
<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>220</td>
<td>-3</td>
</tr>
<tr>
<td>1</td>
<td>219</td>
<td>-4</td>
</tr>
<tr>
<td>1</td>
<td>218</td>
<td>-5</td>
</tr>
<tr>
<td>1</td>
<td>217</td>
<td>-6</td>
</tr>
<tr>
<td>2</td>
<td>216</td>
<td>-7</td>
</tr>
<tr>
<td>2</td>
<td>215</td>
<td>-8</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>213</td>
<td>-10</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>211</td>
<td>-12</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>210</td>
<td>-13</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>209</td>
<td>-14</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>208</td>
<td>-15</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>206</td>
<td>-17</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>7</td>
<td>205</td>
<td>-18</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>8</td>
<td>204</td>
<td>-19</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>9</td>
<td>203</td>
<td>-20</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>10</td>
<td>202</td>
<td>-21</td>
</tr>
<tr>
<td>11</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>11</td>
<td>201</td>
<td>-22</td>
</tr>
<tr>
<td>12</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>12</td>
<td>200</td>
<td>-23</td>
</tr>
<tr>
<td>13</td>
<td>13</td>
<td>13</td>
</tr>
<tr>
<td>13</td>
<td>199</td>
<td>-24</td>
</tr>
<tr>
<td>14</td>
<td>14</td>
<td>14</td>
</tr>
<tr>
<td>14</td>
<td>198</td>
<td>-25</td>
</tr>
<tr>
<td>:</td>
<td>:</td>
<td>:</td>
</tr>
<tr>
<td>92</td>
<td>572</td>
<td>572</td>
</tr>
<tr>
<td>93</td>
<td>579</td>
<td>576</td>
</tr>
<tr>
<td>94</td>
<td>574</td>
<td>574</td>
</tr>
<tr>
<td>95</td>
<td>577</td>
<td>577</td>
</tr>
<tr>
<td>96</td>
<td>571</td>
<td>571</td>
</tr>
<tr>
<td>:</td>
<td>:</td>
<td>:</td>
</tr>
<tr>
<td>105</td>
<td>503</td>
<td>503</td>
</tr>
<tr>
<td>106</td>
<td>496</td>
<td>496</td>
</tr>
<tr>
<td>107</td>
<td>498</td>
<td>498</td>
</tr>
</tbody>
</table>

In the missing portions of the table the numbers in the two middle columns are equal across and monotone up and down.

The equality of the number of trees in corresponding levels suggested a lattice duality. In fact from level 6 to level 208 the lattice 5_6 is self-dual, and this duality can be extended uniquely down to level 0 by omitting nodes from
levels 209 through 223. Thus by removing 17 of the 54049 trees we get a self-dual lattice.

We cannot generate the entire lattice \( \mathcal{I}_7 \), but we can generate the top and the bottom of it. Here is a table of the number of trees in each level at the top and bottom of \( \mathcal{I}_7 \).

<table>
<thead>
<tr>
<th>level</th>
<th>trees</th>
<th>trees</th>
<th>level</th>
<th>level-54056</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>54056</td>
<td>-0</td>
<td>1</td>
<td>54056</td>
</tr>
<tr>
<td>1</td>
<td>54055</td>
<td>-1</td>
<td>1</td>
<td>54055</td>
</tr>
<tr>
<td>1</td>
<td>54054</td>
<td>-2</td>
<td>1</td>
<td>54054</td>
</tr>
<tr>
<td>1</td>
<td>54053</td>
<td>-3</td>
<td>1</td>
<td>54053</td>
</tr>
<tr>
<td>1</td>
<td>54052</td>
<td>-4</td>
<td>1</td>
<td>54052</td>
</tr>
<tr>
<td>1</td>
<td>54051</td>
<td>-5</td>
<td>1</td>
<td>54051</td>
</tr>
<tr>
<td>1</td>
<td>54050</td>
<td>-6</td>
<td>1</td>
<td>54050</td>
</tr>
<tr>
<td>1</td>
<td>54049</td>
<td>-7</td>
<td>2</td>
<td>54048</td>
</tr>
<tr>
<td>2</td>
<td>54047</td>
<td>-8</td>
<td>2</td>
<td>54047</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>54046</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>54045</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>54044</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>54043</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>54042</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>54041</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>6</td>
<td>6</td>
<td>54040</td>
</tr>
<tr>
<td>7</td>
<td>6</td>
<td>7</td>
<td>7</td>
<td>54039</td>
</tr>
<tr>
<td>8</td>
<td>7</td>
<td>7</td>
<td>8</td>
<td>54038</td>
</tr>
<tr>
<td>9</td>
<td>8</td>
<td>8</td>
<td>9</td>
<td>54037</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>54036</td>
</tr>
<tr>
<td>11</td>
<td>11</td>
<td>11</td>
<td>11</td>
<td>54035</td>
</tr>
<tr>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>54034</td>
</tr>
<tr>
<td>13</td>
<td>13</td>
<td>13</td>
<td>13</td>
<td>54033</td>
</tr>
<tr>
<td>14</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>54032</td>
</tr>
<tr>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>54031</td>
</tr>
</tbody>
</table>

Here is the top and bottom of the lattice \( \mathcal{I}_8 \); the level \( -k \) means level \( 8-k \) where \( N \) is the number of the top level.

<table>
<thead>
<tr>
<th>level</th>
<th>trees</th>
<th>trees</th>
<th>level</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-0</td>
<td>-0</td>
<td>-0</td>
</tr>
</tbody>
</table>
For each \( n > 3 \) there is a tree near the top that corresponds to the empty tree, while the other trees that do not participate in the duality form a principal filter. For \( \mathfrak{A}_4 \), the tree corresponding to the empty tree, and the tree generating the excluded principal filter, are respectively

\[
\begin{array}{cccc}
0 & 2 & 3 & 4 \\
1 & 1 & 2 & 3 \\
2 & 3 & 3 & 2 \\
3 & 4 & 4 & 3 \\
4 & 5 & 5 & 4 \\
5 & 6 & 6 & 5 \\
6 & 7 & 7 & 6 \\
7 & 8 & 8 & 7 \\
8 & 9 & 9 & 8 \\
9 & 11 & 11 & 10 \\
10 & 13 & 13 & 12 \\
11 & 15 & 15 & 13 \\
12 & 18 & 18 & 14 \\
13 & 22 & 22 & 15 \\
14 & & & \\
\end{array}
\]
For $\mathcal{F}_0$ they are

\[
\begin{array}{c}
0 \\
1 \\
2 \\
3 \\
4 \\
5 \\
6
\end{array}
, \quad
\begin{array}{c}
0 \\
1 \\
2 \\
3 \\
4 \\
5 \\
6
\end{array}
\]

For $\mathcal{F}_1$ they are

\[
\begin{array}{ccc}
0 & 0 & 0 \\
1 & 2 & 1 \\
3 & 3 & 3 \\
4 & 3 & 4 \\
5 & 5 & 5 \\
6 & 5 & 6 \\
7 & 7 & 7
\end{array}
, \quad
\begin{array}{ccc}
0 & 0 & 0 \\
1 & 2 & 1 \\
3 & 3 & 3 \\
4 & 3 & 4 \\
5 & 5 & 5 \\
6 & 5 & 6 \\
7 & 7 & 7
\end{array}
\]

For $\mathcal{F}_2$ they are

\[
\begin{array}{cccc}
0 & 0 & 0 & 0 \\
1 & 2 & 1 & 2 \\
3 & 3 & 3 & 3 \\
4 & 3 & 4 & 4 \\
5 & 5 & 5 & 5 \\
6 & 5 & 6 & 6 \\
7 & 7 & 7 & 7
\end{array}
, \quad
\begin{array}{cccc}
0 & 0 & 0 & 0 \\
1 & 2 & 1 & 2 \\
3 & 3 & 3 & 3 \\
4 & 3 & 4 & 4 \\
5 & 5 & 5 & 5 \\
6 & 5 & 6 & 6 \\
7 & 7 & 7 & 7
\end{array}
\]

Note the relationship between the tree corresponding to the empty tree in $\mathcal{F}_n$ and the tree generating the excluded principal filter in $\mathcal{F}_{n+1}$.

Here is an example of a tree as our computer prints it out.
A node $y$ is an immediate successor of a node $x$ exactly when

(1) $y$ is in the row just above the row of $x$.

(2) $y$ is above or to the right of $x$.

(3) each node in the row of $x$, and to the right of $x$, is also to the right of $y$.

This tree is in level -16 of $\mathcal{F}_8$, and corresponds, under the duality on $\mathcal{F}_8$, to the tree $\frac{1}{3}$ in level 5.

There are two levels in $\mathcal{F}_6$ that contain 577 trees.

Each subset of such a level, when stuck on top of a node of value 7, gives a distinct tree in $\mathcal{F}_7$. Thus there are at least $2^{577}$ trees in $\mathcal{F}_7$. Moreover, there is a tree in $\mathcal{F}_7$ with 577 branches. Is 577 the size of the largest antichain (set of incomparable elements) in $\mathcal{F}_6$?

6. EXPLAINING THE DUALITY IN $\mathfrak{F}$

Here is the graph of the lattice $\mathfrak{F}_4$. The trees participating in the duality are labeled by their root values, the others are labeled $a,b,c,d$ or $\ast$. 

\begin{verbatim}
0 0 0 0 0 0
1 2 3 4 5 5
2 2 2 2 2 2
3 4 4 4 4 4
5 5 6 6 6 6
8 8 8 8 8 8
\end{verbatim}
We denote the lattice of elements participating in the duality by \( \mathfrak{d}_4 \), and the duality map by \( \delta_4 \). The root values of the trees in \( \mathfrak{d}_4 \setminus \mathfrak{n}_4 \) are all 4, except for the lowest \( \mathfrak{m} \) which has root value 3. For \( n > 4 \), all trees in \( \mathfrak{d}_n \setminus \mathfrak{n}_n \) have
root value \( n \).

Referring to the picture of \( S_4 \), denote the five top \( \star \)'s by \( t^0, t^1, t^2, t^3 \) and \( t^4 \), and the four \( \star \)'s coming down from \( c = c^0 \) by \( c^1, c^2, c^3, \) and \( c^4 \). The trees in \( S_4 \) are

\[
\begin{array}{cccccc}
0 & 1 & 0 & 0 & 0 & 0 \\
2 & 2 & 2 & 12 & 12 & 12 \\
3 & 3 & 3 & 3 & 3 & 3 \\
4 & 4 & 4 & 4 & 4 & 4 \\
\end{array}
\]

\[
\begin{array}{cccccc}
0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 2 & 1 & 2 \\
3 & 23 & 23 & 23 & 23 & 2 \\
4 & 4 & 4 & 4 & 4 & 3 \\
\end{array}
\]

The trees in \( S_4 \) and their duals are

\[
\begin{array}{cccc}
1 & 3 & 0 & 4 \\
0 & 4 & 32 & 1 \\
2 & 4 & 4 & 1 \\
3 & 2 & 23 & 0 \\
4 & 3 & 3 & 4 \\
0 & 0 & 0 & 1 \\
0 & 2 & 12 & 3 \\
4 & 4 & 4 & 4 \\
1 & 2 & 1 & 2 \\
4 & 3 & 6 & 4 \\
\end{array}
\]

We need some notation. Let \( \mathfrak{I}(x) \) denote the ideal
generated by $x$, and fix the ideal generated by the branches of $x$. If $I$ is a set of trees in $\mathcal{I}_{n-1}^*$, then let $n[I]$ denote the tree with root value $n$ and branches $I$. We will extend this notation by writing $n[I,t]$ for $n[I \cup \{t\}]$ and $n[I,t]$ for $n[I \cup \{t\}]$. Note that $n[I]$ and $n[I,t]$ are equivalent trees, hence equal as elements of $\mathcal{I}_{n}^*$, if and only if $I$ and $J$ generate the same ideal.

The statement and proof of the following theorem are slightly complicated by the fact that $\mathcal{I}_3$ is not contained in $\mathcal{I}_4$, although $\mathcal{I}_{n-1} \subseteq \mathcal{I}_n$ for $n > 4$. However we prefer to start the induction at 4, which can be checked by hand fairly easily, rather than at 5, which is more tedious to check by hand.

6.1 Theorem. If $n \geq 4$, then the top $n+1$ levels of the lattice $\mathcal{I}_n$ form a chain of trees $t_n^0 > t_n^1 > \cdots > t_n^n$. The next two levels look like

```
      a_n
      |   |
      |   b_n
      |   |
      c_n
      |   |
      |   d_n
```

The tree $c_n$ is at the top of a saturated chain of trees $c_n = c_n^0 > c_n^1 > c_n^2 > \cdots > c_n^n$, with $\mathcal{I}_{n-1} \subseteq \mathcal{I}(t_n^i)$. Removing these 2n+5 trees leaves a self-dual lattice $\mathcal{I}_n = \{ x \in \mathcal{I}_n : x < d_n \}$ contained in $\mathcal{I}(c_n)$, with a duality map $\delta_n$ such that $\delta_n(\mathcal{I}_{n-1} \cap \mathcal{I}_n) = \mathcal{I}(c_n^i)$ for $i = 0, \ldots, n$. If $n > 4$, then $\mathcal{I}_{n-1} \subseteq \mathcal{I}_n$. 
Proof. Note that the theorem holds for $n = 4$; we shall show that if it holds for $n$, then it holds for $n+1$. Define

\[ t^i_{n+1} = n+1[t^i_n], \quad i = 0, \ldots, n \]
\[ a^i_{n+1} = n+1[c^i_n, b^i_n] \]
\[ b^i_{n+1} = n+1[c^i_n, d^i_n] \]
\[ d^i_{n+1} = n+1[a^i_n] \]
\[ c^i_{n+1} = n+1[c^i_n, d^i_n], \quad i = 0, \ldots, n \]
\[ c^{n+1}_{n+1} = n+1[d^i_n] \]

Clearly the $2(n+1) + 5$ trees $c^i_{n+1}, c^i_{n+1}, d^i_{n+1}, c^i_{n+1}, d^i_{n+1}$ satisfy the conditions of the theorem. The set $\mathcal{G}_{n+1}$ of the remaining trees consists of all elements of $\mathcal{G}_{n+1}$ that lie strictly below $d^i_{n+1} = n+1[a^i_n]$, that is, $\mathcal{G}_{n+1}$ is $\mathcal{G}_n$ together with those trees $x \in \mathcal{G}_{n+1} \setminus \mathcal{G}_n$ such that $\beta x \in \mathcal{S}(c_n)$. Clearly

\[ \mathcal{G}_{n+1} \subseteq \mathcal{S}(c_{n+1}) \]

as required. There are three types of elements of $\mathcal{G}_{n+1}$:

(i) trees $x \in \mathcal{G}_{n+1} \setminus \mathcal{G}_n$ such that $\beta x \subseteq \mathcal{G}_n$, that is, all branches of $x$ lie strictly below $d^i_n$.

(ii) trees $x \in \mathcal{G}_{n+1} \setminus \mathcal{G}_n$ such that $c^i_n \in \beta x \in \mathcal{S}(c_n)$.

(iii) elements of $\mathcal{S}_{n+1} \setminus \mathcal{S}_n$ for $i = 0, \ldots, n$. 

REFERENCES


3. ______________. Ulm's theorem for simply presented valued $p$-groups. This proceedings.