A Constructive Proof that Tree are Well-Quasi-Ordered Under Minors (Detailed Abstract)

Arvind Gupta*
School of Computing Science
Simon Fraser University
Burnaby, Canada, V5A 1S6
email: arvind@cs.sfu.ca

Abstract

The recent proof of Robertson and Seymour that graphs are well-quasi-ordered under minors immediately implies that a number of intesting problems have polynomial time algorithms. However, partially because of their non-constructive nature, these proofs do not yield any information about the algorithms.

Here we present a constructive proof that trees are well-quasi-ordered under minors. This extends the results of Murty and Russell [MR90] who give a constructive proof of Higman's Lemma. Our proof is based on transforming finite sequences of trees to ordinals. We begin by describing a transformation which carries trees to finite strings of numbers and give an ordering on these strings which preserves the minor ordering on the underlying trees. We show that in our well-quasi-ordering argument, these strings are actually over a finite alphabet. This allows us to conclude the result. We require the well-ordering of the ordinal ε_0 for our proof.

1 Introduction

In 1965 K. Wagner conjectured, and Robertson and Seymour recently proved [RSb], that finite graphs are well-quasi-ordered under the minor relation. This immediately implies a finite characterization of any minor closed graph family namely the set of minimal graphs not in the family (called the obstructions of the family). Robertson and Seymour also found a polynomial time algorithm for the disjoint paths problem [RSa]. Then, for every fixed graph H, this algorithm can easily be converted into one which tests whether an input graph G contains H as a minor. Thus, every minor closed family of graphs has a polynomial time membership test.

*Research supported by the Natural Sciences and Engineering Research Council of Canada, the Centre for System Sciences and the President's Research Grant, Simon Fraser University. A detailed exposition of these results appear in the author's dissertation [Gup90].

One problem is that Robertson and Seymour's proofs are highly non-constructive and as such do not yield information about the obstructions for minor closed families. This is an important consideration since there are a large number of interesting minor closed graph families some of which were previously not even known to have recursive solutions [FL88].

Early work on well-quasi-orderings was performed by Higman [Hig52] who showed that for a well-quasi-order $(\mathcal{Q}, \preceq)$, the set of finite sequences of $\mathcal{Q}$ is well-quasi-ordered under a relation which essentially orders the sequences component-wise. In 1960, J. Kruskal [Kru60] showed that the family of finite trees is well-quasi-ordered under topological embeddings. This immediately implied that trees are well-quasi-ordered under minors. A short and elegant proof of both Higman's and Kruskal's results was given by Nash-Williams in 1963 [NW63].

The proofs of Higman, Kruskal and Nash-Williams all use non-constructive (classical) reasoning. They show that there are no infinite antichains but do not show how two comparable elements can be obtained from an infinite sequence. Of course, one can always scan through an infinite sequence to find two comparable elements. However, in a constructive proof there must be a measure of progress associated with the scanning of the sequence. Commonly this is a function which maps finite sequences to ordinals. When a sequence is extended by an incomparable element, the function value on the new sequence decreases. Since ordinals are well-ordered, the well-quasi-ordering of the set follows.

Informally, a proof of a Π_2^0 statement, $\forall x \exists y \phi(x,y)$, is constructive if, implicit in the proof, is a computable function f such that for each x_0, $\phi(x_0, f(x_0))$ holds. Friedman [Fri78] and Leivant [Lei85] have shown that for Π_2^0 statements with certain restrictions on $\phi(x,y)$, a classical proof in some proof system can be converted into a constructive proof in the (constructive version) of the same system. Since a statement of well-quasi-ordering is a Π_2^0 statement, constructive proofs of Wagner's Conjecture exist. However, these proofs are syntactic translations from which it seems difficult to obtain information about obstructions.

Notice that a constructive proof of well-quasi-ordering is not sufficient to generate obstructions of closed families. However, the hope is that the proof will help in giving alternate characterizations of closed families which can then be used to find obstructions. In [Gup91], this technique is used for finding obstructions for minor closed tree families.

Recently, Murthy and Russell have given an explicit constructive proof of Higman's Lemma [MR90]. Their proof takes a finite sequence and maps it to a description of the elements incomparable to those in the sequence. They define a well-founded ordering on these sequences which decreases as the sequence is extended with incomparable elements. This ordering can be translated into a function which maps finite sequences to ordinals.

In this paper we use the techniques of Murthy and Russell to find a constructive proof that trees are well-quasi-ordered under minors. We describe a map from finite sequences of trees to ordinals. Our mapping takes the empty sequence to the ordinal ε_0 and thus we require the well-ordering of ε_0.

Some ideas in our proof are motivated by a result of Beinstock, Robertson, Seymour and Thomas [BRST89] who show that for any tree T, a graph not containing T as a minor has a certain type of linear decomposition into subgraphs each no larger than T. We also use techniques from Thomas [Tho90] who shows that graphs which have a certain type of tree-decomposition are well-quasi-ordered. We find similar properties on our linear decompositions for trees.

The outline of the remainder of this paper is as follows. In §2 we give the necessary background material. In §3 we describe a transformation which maps rooted trees to finite sequences of pairs of numbers. In §4 we define a partial-order on the sequences of §3. In §5 we prove that the sequences defined in §3 are well-quasi-ordered under the ordering in §4. Finally, in §6 we discuss future work.

2 Preliminaries

2.1 Graph Theory

For T a tree, $V(T)$ and $E(T)$ are the vertex and edge sets of T respectively. We treat rooted trees as directed graphs where each edge is directed away from the root. The *complete binary tree of height h*, B_h, is a rooted tree such that every internal vertex has exactly 2 children and every path from the root to the leaves has length h.

A *planar planted* tree is a rooted tree with an ordering on the children of every internal vertex. We will normally write the children of a vertex as a sequence, say $(c_1, \ldots, c_k)$, to signify this ordering. A *planar planting* of a rooted tree T is a planar planted tree T' which, as a rooted tree, is isomorphic to T. Notice that B_h remains invariant under all planar plantings. For T a planar planted tree and $v \in V(T)$ suppose v has children $(c_1, \ldots, c_k)$ where $k > 1$. Then c_i is said to be to the *left* (*right*) of c_j if $i < j$ ($i > j$). More generally, a vertex v occurs to the left (right) of a vertex w if at least common ancestor of v and w, say z, the child of z which is an ancestor of v occurs to the left (right) of the child of z which is an ancestor of w.

If G is a graph and e is an edge of G then the *contraction* of e in G is formed by identifying the endpoints of e. That is, if $e = \{x, y\}$ then we delete x and y and add a new vertex z adjacent to every vertex w which was adjacent to x and y in G.

Definition: Let G and H be graphs. H is a *minor* of G ($H \leq_m G$) if there is a subgraph S of G such that H is isomorphic to a graph formed by a sequence of edge contractions on S.

An alternate characterization is given by the following lemma.

Lemma 2.1 *Let H and G be graphs. H is a minor of G if and only if there is an injective function $im : V(H) \to \{subgraphs\ of\ G\}$, such that*

1. *For every $v \in V(H)$, $im(v)$ is a connected non-null subgraph of G.*

2. *For $v, w \in V(H)$, if $v \neq w$ then $im(v) \bigcap im(w) = \emptyset$.*

3. *For $e \in E(H)$, $e = \{v, w\}$, there is an $x \in im(v)$ and $y \in im(w)$ such that $\{x, y\} \in E(G)$.*

We extend the definition of minor to rooted and planar planted trees.

Definition: Let T and T' be planar planted trees. Then T is a *rooted minor* of T' ($T \leq_{m^r} T'$) if $T \leq_m T'$ and for $v \in V(T)$, $v \neq root(T)$, all vertices in $im(v)$ are descendants of some vertex of the parent of v. T is a *planar planted minor* of T', ($T \leq_{m^p} T'$), if $T \leq_{m^r} T'$ and for every $u, v \in V(T)$, if u is to the left of v in T then $im(u)$ is to the left of $im(v)$ in T'.

2.2 Well-Quasi-Orders

Definition: Let Q be a set and $\preceq$ a binary relation on S. Then $(Q, \preceq)$ is a *quasi-order* if $\preceq$ is transitive and reflexive. A sequence $(q_i)_{i=1}^{\infty}$ on Q is an *antichain* if for every $i, j \in \mathbf{N}$, if $1 \leq i, j \leq k$ and $i \neq j$ then $q_i \not\preceq q_j$. Also, $(q_i)_{i=1}^{\infty}$ is a *descending chain* if for every i, $q_{i+1} \preceq q_i$ but $q_i \not\preceq q_{i+1}$. A quasi-order $(Q, \preceq)$ is a *well-quasi-order* if every infinite sequence is neither an antichain nor a descending chain.

Alternatively, a quasi-order $(Q, \preceq)$ is a well-quasi-order if and only if for every sequence $(q_i)_{i=1}^{\infty}$, there is an $i < j$ such that $q_i \preceq q_j$. The following are some basic facts about well-quasi-orders.

Lemma 2.2 *Let $(Q_1, \preceq_1)$, $(Q_2, \preceq_2)$ be well-quasi-orders. Then,*

1. *For $S \subseteq Q_1$, $(S, \preceq_1)$ is a well-quasi-order.*

2. *$(Q_1 \times Q_2, \preceq_\times)$ is a well-quasi-order where $(a, b) \preceq_\times (c, d)$ if $a \preceq_1 c$ and $b \preceq_2 d$.*

3. **(Higman's Lemma)** *Let SQ_1 be the set of finite sequences on Q_1. Let $X, Y \in SQ_1$, where $X = (x_1, \ldots, x_k)$ and $Y = (y_1, \ldots, y_\ell)$. Define $X \preceq^H Y$ iff there is a function $f : \{1, \ldots, k\} \to \{1, \ldots, \ell\}$ such that for all i, $f(i) < f(i+1)$ and $x_i \preceq_1 y_{f(i)}$. Then, $(SQ_1, \preceq^H)$ is a well-quasi-order.*

2.3 Miscellaneous

We assume a basic knowledge of ordinal numbers and ordinal arithmetic.

Definition: Define $\gamma_0 = 1$ and for $i > 0$ define $\gamma_i = \omega^{\gamma_{i-1}}$. Then $\varepsilon_0 = lim\{\gamma_0, \gamma_1, \ldots\}$.

Lemma 2.3 *For ordinals $\alpha_1, \ldots, \alpha_k, \beta_1, \ldots, \beta_\ell, \theta$ if $\alpha_1, \ldots, \alpha_k, \beta_1, \ldots, \beta_\ell < \theta$, then*

$$\omega^{\alpha_1} + \cdots + \omega^{\alpha_k} + \beta_1 + \cdots + \beta_\ell < \omega^\theta$$

It will be convenient to fix a representation of all ordinals up to ε_0. We will use the *Cantor normal form* [Can15] which gives a unique representation.

Lemma 2.4 *Suppose $\alpha < \varepsilon_0$. Then there are ordinals $\alpha_1, \ldots, \alpha_k < \varepsilon_0$ and natural numbers $n_1, \ldots, n_k \neq 0$ such that for each i, $\alpha_{i+1} < \alpha_i$ and*

$$\alpha = \omega^{\alpha_1} n_1 + \omega^{\alpha_2} n_2 + \cdots + \omega^{\alpha_k} n_k$$

Finally, if A and B are sequences then $A^\frown B$ will denote the concatenation of A and B, $|A|$ the length of A and $\pi_i(A)$ the i^{th} element of A.

3 A Linear Representation of Trees

In this section we describe a transformation of planar planted trees to finite sequences of pairs of numbers. We then describe a specific planar planting of rooted trees. Finally we show that if we consider rooted trees which exclude some fixed tree as a minor then the numbers in our finite sequences are bounded. Our transformation of planar planted trees is based on depth-first-search.

Definition: Let T be a planar planted tree and $(v_1, v_1, \ldots, v_n)$ be a (preorder) depth-first-search listing of the vertices of T. For $i \in \mathbb{N}$, $1 \leq i \leq n$ define

$$\alpha_i = \{v_j : j < i \text{ and there is a } k \geq i \text{ such that } v_k \text{ is a child of } v_j\} \cup \{v_i\}$$

For the remainder of this section, unless otherwise indicated, let T be a planar planted tree and $(v_1, \ldots, v_n)$ be the depth-first-search listing of T. The next lemma gives an alternate characterization of α_i.

Lemma 3.1 *Let $v_i, v_j \in V(T)$. Then $v_j \in \alpha_i$ if and only if either $v_j = v_i$, $v_j = parent(v_i)$ or $j < i$ and some child of v_j occurs to the right of v_i.*

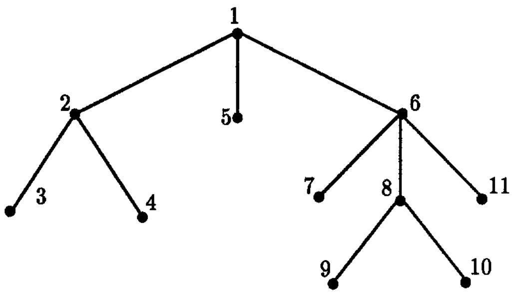
We now transform $(\alpha_1, \ldots \alpha_n)$ into the required sequence.

Definition: For $1 \leq i \leq n$, define p_i to be a sequence on $\mathbb{N} \times \{0, 1, 2, 3\}$ such that for $1 \leq i < n$

$$p_i = \begin{cases} ((|\alpha_i|, 0)) & \text{if } \alpha_i \subseteq \alpha_{i+1}; \\ ((|\alpha_i|, 1), (|\alpha_i \bigcap \alpha_{i+1}|, 0)) & \text{if } v_i \in \alpha_{i+1}, parent(v_i) \notin \alpha(v_i); \\ ((|\alpha_i|, 2), (|\alpha_i \bigcap \alpha_{i+1}|, 0)) & \text{if } v_i \notin \alpha_{i+1}, parent(v_i) \in \alpha(v_i); \\ ((|\alpha_i|, 3), (|\alpha_i \bigcap \alpha_{i+1}|, 0)) & \text{if } v_i, parent(v_i) \notin \alpha_{i+1}. \end{cases}$$

Also, if $n = 1$ then $p_n = ((1, 0))$ otherwise $p_n = ((2, 3))$. The sequence $\tau(T) = p_1{}^\frown p_2{}^\frown \cdots {}^\frown p_n$ is the *tree linearization of* T Suppose $\tau(T) = \gamma_1 \ldots \gamma_m$. Then, the *width* of $\tau(T)$, $width(\tau(T))$, is $\max\{\pi_1(\gamma_i) : 1 \leq i \leq m\}$ (see Figure 1).

$\alpha_1 = (1)$
$\alpha_2 = (1,2)$
$\alpha_3 = (1,2,3)$
$\alpha_4 = (1,2,4)$
$\alpha_5 = (1,5)$
$\alpha_6 = (1,6)$
$\alpha_7 = (6,7)$
$\alpha_8 = (6,8)$
$\alpha_9 = (6,8,9)$
$\alpha_{10} = (6,8,10)$
$\alpha_{11} = (6,11)$



$p_1 = ((1,0))$
$p_2 = ((2,0))$
$p_3 = ((3,2),(2,0))$
$p_4 = ((3,3),(1,0))$
$p_5 = ((2,2),(1,0))$
$p_6 = ((2,1),(1,0))$
$p_7 = ((2,2),(1,0))$
$p_8 = ((2,0))$
$p_9 = ((3,2),(2,0))$
$p_{10} = ((3,3),(1,0))$
$p_{11} = ((2,3))$

Figure 1: The tree linearization $\tau(T)$.

Intuitively, we obtain the first component of the elements of $\tau(T)$ from T by performing 2 steps at each stage of a depth-first-traversal of T. At the root, the first component has value 1. Suppose we are part way through a depth-first-traversal and the first component has value r. The first step is to visit the next vertex in the traversal and increase the value to $r+1$. At the second step, we set it to be the number of vertices on the path from the current vertex back to the root which have unvisited children. This will decrease the value by at most 2. It is the second component which allows us to determine how much the value decreases by.

We next define a planar planting of rooted trees.

Definition: The *binary-tree-number* of a planar planted tree T, $\beta(T)$, is $\max\{h : B_h \leq_m T\}$. Then, T is *binary-planar-planted* if, for every vertex $v \in V(T)$ with children $(c_1,\ldots,c_t)$, $\beta(T_1) \leq \beta(T_2) \leq \cdots \leq \beta(T_t)$ where T_i is the maximal subtree of T rooted at c_i.

It is not difficult to find a planar planting of a rooted tree which is binary-planar-planted:

Lemma 3.2 *There is an $O(n^2 \log n)$ algorithm which takes a rooted tree T and outputs a binary-planar-planted tree T' such that T and T' are isomorphic as rooted trees.*

We call the planar planting given in Lemma 3.2 a *binary-planar-planting.* The last result of this section is:

Theorem 3.3 *Let T and T' be rooted trees such that $T \not\leq_{m^r} T'$. Let T'' be the binary-planar-planting of T'. Then, $width(\tau(T'')) \leq |V(T)|$.*

The basic idea in the proof of Theorem 3.3 is to find the smallest binary tree which contains T as a minor. We begin by bounding the height of this binary tree in terms of $|V(T)|$.

Lemma 3.4 *For every rooted tree T, $T \leq_{m^r} B_{|V(T)|-1}$.*

It is easy to obtain a tight bound on the height of the binary tree required in Lemma 3.4 but for our purposes $|V(T)|-1$ will do. Now, to prove Theorem 3.3, it suffices to prove the following lemma.

Lemma 3.5 *Let T be a rooted tree and $h \geq 0$ such that $B_h \not\leq_{m^r} T$. If T' is the binary-planar-planting of T then $width(\tau(T')) \leq h+1$.*

The proof of Lemma 3.5 follows immediately from the following two lemmas.

Lemma 3.6 *Let T be binary-planar-planted. Then, $width(\tau(T)) \leq \beta(T)+2$.*

Lemma 3.7 *For T binary-planar-planted and $h > 0$, if $B_h \not\leq_{m^p} T$ then $\beta(T) \leq h-1$.*

To prove Lemma 3.6 we use lexicographical induction on the triple $(\beta(T), |V(T)|, c(T))$ where $c(T)$ is the number of children of the root of T. Lemma 3.7 follows from Lemma 3.4.

For the remainder of the paper, for a rooted tree T, $\tau(T)$ will be used to denote the tree linearization of the binary-planar-planting of T.

4 Ordering Tree Linearizations

In this section we define a generalization of tree linearizations and an ordering on these. We show that two linearizations are related under this ordering if and only if the underlying trees are related under the minor ordering.

Definition: For $n \in \mathbf{N}$, we denote the set $\{0,\ldots,n\}$ by $\bar{n}$. For $n,k \in \mathbf{N}$, $\mathcal{I}_{n,k}$ is the set of finite sequences on $\bar{n} \times \bar{k}$, that is, $\mathcal{I}_{n,k} = (\bar{n} \times \bar{k})^*$.

Notice that for any rooted tree T, $\{\tau(T') : T' \text{ a rooted tree}, T \not\leq_{m^r} T'\} \subseteq \mathcal{I}_{|V(T)|,3}$. We now define an ordering on $\mathcal{I}_{n,k}$.

Definition: Let $n,k \geq 0$. Let $P,Q \in \mathcal{I}_{n,k}$, where $P=(p_1,\ldots,p_r)$ and $Q=(q_1\ldots,q_s)$. Define P to be an *immersion* of Q ($P \leq_{\mathcal{I}} Q$) if there are sequences $Q_0,\ldots,Q_{r+1} \in \mathcal{I}_{n,k}$ such that the following holds:

1. $Q = Q_0{}^\frown Q_1{}^\frown \cdots {}^\frown Q_{r+1}$.
2. for $1 \leq i \leq r$, $p_i \in Q_i$.
3. for $1 \leq i \leq r$ and for every $q \in Q_i$, $\pi_1(p_i) \leq \pi_1(q)$.

We call $(Q_0,\ldots,Q_{r+1})$ an *immersion decomposition* of Q by P.

Obviously $(\mathcal{I}_k, \leq_{\mathcal{I}})$ is a partial-ordering.

Theorem 4.1 *For T_1 and T_2 planar planted, if $\tau(T_1) \leq_{\mathcal{I}} \tau(T_2)$ then $T_1 \leq_{m^P} T_2$.*

Proof (outline): Suppose $\tau(T) = (p_1, \ldots, p_r)$ and $\mathcal{Q} = (Q_0, \ldots, Q_{r+1})$ is an immersion decomposition of $\tau(T')$ by $\tau(T)$. Then we can show that $\mathcal{Q}$ can be transformed such that for every j, $1 \leq j \leq r$, p_j is the first and last element of Q_j. Now, for each i, we consider the subtree of T defined on those vertices visited before p_i was defined, say T_i. If we now restrict $\tau(T')$ to $(Q_0, \ldots, Q_i)$ and consider the subtree of T' induced on this subsequence, say T_i', then we can show by induction on i that $T_i \leq_{m^P} T_i'$. **QED**

5 Well-Quasi-Ordering Sequences under Immersions

In this section we show that for every $n, k \in \mathbf{N}$, $(\mathcal{I}_{n,k}, \leq_{\mathcal{I}})$ forms a well-quasi-order. This immediately implies that trees are well-quasi-ordered under minors: Given an infinite sequence of trees we begin by arbitrarily rooting them. Then, we assume that the first tree is not contained in any of the others as a minor, otherwise we are done. Now, the width of the tree linearizations of all the trees is bounded and, by the results in this section, there must be two comparable elements, the first an immersion of the second. But then, by Theorem 4.1, the first tree is a rooted minor of the second.

We begin by defining a restricted class of regular expressions. Let $\Sigma = \{(i,j) : i \in \bar{n}, j \in \bar{k}\}$. We denote both the empty string and the regular expression with language the empty string by ϵ. We also define a generalization to the star operator in regular expressions. For $\mathcal{E}$ a regular expression and $S \subseteq \Sigma$, $S \neq \emptyset$, $\mathcal{E}^S$ is the regular expression with language $\{\mathcal{L}(\mathcal{E}) \cdot S\}^* \cdot \mathcal{L}(\mathcal{E})$. We call $\mathcal{E}^S$ a *starred* expression.

Definition: The class of *immersion regular expressions* and a subclass of these, the *prime immersion regular expressions*, with associated atom-set (*atom*) are a subclass of all regular expressions defined inductively as follows:

1. ϵ is a prime immersion regular expression where $atom(\epsilon) = \emptyset$.

2. Let $\mathcal{E}$ be an immersion regular expression, $x \in \bar{n}$, $S \subseteq \bar{k}$ and $S \neq \emptyset$. If for every $(u,v) \in atom(\mathcal{E})$, $x < u$ then $\mathcal{E}' = \mathcal{E}^{\{x\} \times S}$ is a prime immersion regular expression with $atom(\mathcal{E}') = atom(\mathcal{E}) \bigcup (\{x\} \times S)$.

3. Let $\mathcal{E}_1, \mathcal{E}_2$ and $\mathcal{E}_3$ be prime immersion regular expressions, $(x,y) \in \Sigma$, $S \subseteq \bar{k}$ and $S \neq \emptyset$. Suppose that for every $(u,v) \in atom(\mathcal{E}_1) \bigcup atom(\mathcal{E}_2) \bigcup atom(\mathcal{E}_3)$, $x \leq u$, and $(x,y) \notin atom(\mathcal{E}_2)$. Then, the following are prime immersion regular expressions.

   (a) $\mathcal{E}' = \mathcal{E}_1 \cdot (x,y) \cdot \mathcal{E}_2 \cdot (x,y) \cdot \mathcal{E}_3$ where $atom(\mathcal{E}') = \bigcup_{i=1}^{3} atom(\mathcal{E}_i) \bigcup \{(x,y)\}$.

   (b) $\mathcal{E}' = \mathcal{E}_1 \cdot (x,y) \cdot \mathcal{E}_2$ where $atom(\mathcal{E}') = atom(\mathcal{E}_1) \bigcup atom(\mathcal{E}_2) \bigcup \{(x,y)\}$.

   (c) $\mathcal{E}' = \mathcal{E}_2 \cdot (x,y) \cdot \mathcal{E}_3$ where $atom(\mathcal{E}') = a(\mathcal{E}_2) \bigcup atom(\mathcal{E}_3) \bigcup \{(x,y)\}$.

4. Let $\mathcal{E}_1, \ldots, \mathcal{E}_h$ be prime immersion regular expressions. Then, $\mathcal{E}' = \mathcal{E}_1 + \ldots + \mathcal{E}_h$ is an immersion regular expression where $atom(\mathcal{E}') = atom(\mathcal{E}_1) \bigcup \ldots \bigcup atom(\mathcal{E}_h)$.

The basic form of prime immersion regular expressions is given by the following:

Lemma 5.1 *Suppose $\mathcal{E}$ is a prime immersion regular expression. Then there are $a_1, \ldots, a_h \in \Sigma$ and prime immersion regular expressions $\mathcal{E}_0, \mathcal{E}_1, \ldots, \mathcal{E}_h$ with each $\mathcal{E}_i$ either starred or ϵ such that $\mathcal{E} = \mathcal{E}_0 \cdot a_1 \cdot \mathcal{E}_1 \cdot a_2 \cdot \mathcal{E}_2 \ldots a_h \cdot \mathcal{E}_h$.*

We want to define a mapping from immersion regular expressions to ordinals. In order to do this, we first define a new addition and multiplication operator on ordinal numbers.

Definition: For $h > 0$ and ordinals $\alpha_1, \alpha_2, \ldots, \alpha_h$, let $\beta_1, \beta_2, \ldots, \beta_h$ be an ordering of $\alpha_1, \ldots, \alpha_h$ such that for $1 \leq i < h$, $\beta_i \geq \beta_{i+1}$. Then,

$$\begin{aligned} \alpha_1 \oplus \ldots \oplus \alpha_h &= \beta_1 + \ldots + \beta_h \\ \alpha_1 \odot \ldots \odot \alpha_h &= \beta_1 \cdot \ldots \cdot \beta_h \end{aligned}$$

We are now ready to define a mapping from immersion regular expressions to ordinals.

Definition: Let $\mathcal{E}$ be an immersion regular expression. Then, $ord(\mathcal{E})$ is an ordinal defined inductively as follows:

1. If $\mathcal{E} = \epsilon$ then $ord(\mathcal{E}) = \omega^{\omega}$.

2. If $\mathcal{E} = \mathcal{E}_1^S$ where $S \subseteq \Sigma$, $S \neq \emptyset$, and $\mathcal{E}_1$ is an immersion regular expression, then $ord(\mathcal{E}) = \omega^{\omega^{ord(\mathcal{E}_1^{S'})}}$ where $S' \subseteq S$ is such that $|S'| = |S| - 1$.

3. If $\mathcal{E} = \mathcal{E}_0 \cdot a_1 \cdot \mathcal{E}_1 \ldots a_h \cdot \mathcal{E}_h$ where each $\mathcal{E}_i$ is either ϵ or a starred immersion regular expression and each $a_i \in \Sigma$ then $ord(\mathcal{E}) = ord(\mathcal{E}_0) \odot ord(\mathcal{E}_1) \odot \ldots \odot ord(\mathcal{E}_h)$.

4. If $\mathcal{E} = \mathcal{E}_1 + \mathcal{E}_2 + \ldots + \mathcal{E}_h$ where each $\mathcal{E}_i$ is a prime immersion regular expression then $ord(\mathcal{E}) = ord(\mathcal{E}_1) \oplus \ldots \oplus ord(\mathcal{E}_h)$.

The main theorem is:

Theorem 5.2 *Let $\mathcal{E}$ be an immersion regular expression and $P \in \mathcal{L}(\mathcal{E})$. Then there is an immersion regular expression $\mathcal{E}(P)$ such that:*

1. *For every $Q \in \mathcal{L}(\mathcal{E})$, if $P \not\leq_I Q$ then $Q \in \mathcal{L}(\mathcal{E}(P))$.*

2. *$ord(\mathcal{E}(P)) < ord(\mathcal{E})$.*

Proof (outline): Let $\mathcal{E}$ be an immersion regular expression and $P \in \mathcal{L}(\mathcal{E})$. The proof proceeds by induction on $(|\mathcal{E}|, |P|)$.

We consider three different cases depending upon the structure of $\mathcal{E}$. We assume $\mathcal{E} \neq \epsilon$, otherwise clearly no P can exist.

1. Suppose $\mathcal{E} = \mathcal{E}_1 + \mathcal{E}_2 + \ldots + \mathcal{E}_h$ where $h > 1$ and for $1 \leq i \leq h$, $\mathcal{E}_i$ is prime. Then $P \in \mathcal{L}(\mathcal{E}_i)$ for some i, $1 \leq i \leq h$, say $i = h$. Suppose $\mathcal{E}_h(P) = \mathcal{F}_1 + \ldots + \mathcal{F}_\ell$ where the $\mathcal{F}_i$ are prime. Define $\mathcal{E}(P) = \mathcal{E}_1 + \ldots + \mathcal{E}_{h-1} + \mathcal{F}_1 + \ldots + \mathcal{F}_\ell$.

2. Suppose $\mathcal{E} = \mathcal{E}_1^{\{x\}\times S}$ where $\mathcal{E}_1$ is an immersion regular expression, $S \subseteq \bar{k}$, $S \neq \emptyset$, $x \in \bar{n}$ such that for every $(u, v) \in a(\mathcal{E}_1)$, $x < u$.

   Case 1: Suppose there is no $(u, v) \in P$ such that $x = u$. Let $\mathcal{E}(P) = (\mathcal{E}_1(P))^{\{x\}\times S}$. Now, to show that for $Q \in \mathcal{L}(\mathcal{E})$ if $P \not\leq_{\mathcal{I}} Q$ then $Q \in \mathcal{L}(\mathcal{E}(P))$, we must consider whether or not $(x, v) \in Q$ for some $v \in S$.

   Case 2: Suppose there is a $(u, v) \in P$ such that $x = u$. Then, by definition, $v \in S$. Let $S = \{i_1, \ldots, i_\ell\}$ and suppose that $v = i_1$. Let $S' = \{i_2, \ldots, i_\ell\}$ (notice that S' may be $\emptyset$ in which case $\mathcal{E}_1^{\{x\}\times S'} = \mathcal{E}_1$). Let $P = P_1 {}^\frown (u, v) {}^\frown P_2$.

   Let $\mathcal{E}(P_1) = \mathcal{A}_1 + \ldots + \mathcal{A}_a$, $\mathcal{E}_1^{\{x\}\times S'} = \mathcal{B}_1 + \ldots + \mathcal{B}_b$ and $\mathcal{E}(P_2) = \mathcal{C}_1 + \ldots + \mathcal{C}_c$ where each $\mathcal{A}_i$, $\mathcal{B}_i$ and $\mathcal{C}_i$ is a prime immersion regular expression. Then, define

$$\begin{aligned}
\mathcal{E}(P) \quad = \quad & \sum_{i=1}^{a}\sum_{j=1}^{b}\sum_{h=1}^{c} \mathcal{A}_i \cdot (u,v) \cdot \mathcal{B}_j \cdot (u,v) \cdot \mathcal{C}_h + \sum_{i=1}^{a}\sum_{j=1}^{b} \mathcal{A}_i \cdot (u,v) \cdot \mathcal{B}_j + \\
& \sum_{j=1}^{b}\sum_{h=1}^{c} \mathcal{B}_j \cdot (u,v) \cdot \mathcal{C}_h + \sum_{j=1}^{b} \mathcal{B}_j
\end{aligned}$$

   Again to see that for $Q \in \mathcal{L}(\mathcal{E}(P))$ we consider whether or not $(x, v) \in Q$ for some $v \in S$.

3. Finally, consider expressions given by case 3 in the definition of immersion regular expression. We will only deal with expressions of the form $\mathcal{E} = \mathcal{E}_1 \cdot (x, y) \cdot \mathcal{E}_2 \cdot (x, y) \cdot \mathcal{E}_3$; the others are very similar. Since $P \in \mathcal{L}(\mathcal{E})$, there are $P_1, P_2, P_3 \in \mathcal{I}_{n,k}$ such that $P = P_1 {}^\frown (x, y) {}^\frown P_2 {}^\frown (x, y) {}^\frown P_3$ where $P_1 \in \mathcal{L}(\mathcal{E}_1)$, $P_2 \in \mathcal{L}(\mathcal{E}_2)$ and $P_3 \in \mathcal{L}(\mathcal{E}_3)$. We will assume that $P_1, P_2, P_3 \neq \epsilon$, the case where any is ϵ can be handled in a similar manner.

   By induction $\mathcal{E}_1(P_1) = \mathcal{A}_1 + \ldots + \mathcal{A}_a$, $\mathcal{E}_2(P_2) = \mathcal{B}_1 + \ldots + \mathcal{B}_b$, and $\mathcal{E}_3(P_3) = \mathcal{C}_1 + \ldots + \mathcal{C}_c$ where each $\mathcal{A}_i, \mathcal{B}_i$ and $\mathcal{C}_i$ is a prime immersion regular expression. Now

$$\begin{aligned}
\mathcal{E}(P) \quad = \quad & \sum_{i=1}^{a} \mathcal{A}_i \cdot (x,y) \cdot \mathcal{E}_2 \cdot (x,y) \cdot \mathcal{E}_3 + \sum_{i=1}^{b} \mathcal{E}_1 \cdot (x,y) \cdot \mathcal{B}_i \cdot (x,y) \cdot \mathcal{E}_3 + \\
& \sum_{i=1}^{c} \mathcal{E}_1 \cdot (x,y) \cdot \mathcal{E}_2 \cdot (x,y) \cdot \mathcal{C}_i
\end{aligned}$$

For $Q \in \mathcal{L}(\mathcal{E}(P))$ such that $P \not\leq_{\mathcal{I}} Q$, $Q = Q_1 {}^\frown (x,y) {}^\frown Q_2 {}^\frown (x,y) {}^\frown Q_3$. We can show that for some i, $P_i \not\leq_{\mathcal{I}} Q_i$ from which it follows that $Q \in \mathcal{E}(P)$.

To show that $ord(\mathcal{E}(P)) < ord(\mathcal{E})$, we use Lemma 2.3; further details are omitted. **QED**

6 Conclusions and Future Work

Our definition of *ord* in §5 maps immersion regular expressions to ordinals where the ordinal of the expression recognizing $\mathcal{I}_{n,k}$ is $\gamma_{2(n+1)(k+1)}$ (recall that γ_i is the ordinal $\omega^{\omega^{\omega^{\cdots}}}$ of height i). Since second order Heyting arithmetic ($\mathbf{HAS_0}$) can prove the well-ordering of all ordinals smaller than ε_0, we obtain

Theorem 6.1 *Let $n, k \in \mathbb{N}$. Then there is a proof in* $\mathbf{HAS_0}$ *that $(\mathcal{I}_{n,k}, \leq_{\mathcal{I}})$ is a well-quasi-order.*

However, for our proof that trees are well-quasi-ordered under minors we require that ε_0 be well-ordered, and therefore our proof gives an upper bound of ε_0 on the ordinal required to show well-quasi-ordering. Friedman [Fri90] has shown a lower bound of ε_0 on the ordinal required to show that binary trees are well-quasi-ordered under minors. Since this also yields a ε_0 lower bound for our theorem, this ordinal is a tight bound. In particular, since it not possible, in $\mathbf{HAS_0}$ to prove that ε_0 is well-ordered, a proof system stronger than $\mathbf{HAS_0}$ is required to prove the theorem.

One natural question is whether or not these types of arguments can be extended to give a constructive proof of Kruskal's Tree Theorem or more generally any of the Robertson and Seymour well-quasi-ordering theorems. The natural place to start this investigation is with graphs of bounded path-width. A graph is said to have bounded path-width if it is decomposable into a linear sequence of subgraphs each of bounded size such that every vertex appears in a consecutive sequence of subgraphs. Our results can be extended if we place a linking condition on the subgraphs (formally, the graphs have bounded linked path-decomposition), that is, if two subgraphs in the decomposition had the same size (say k) with every subgraph in between bigger than k, then there would be k vertex disjoint paths between the vertices of these two subgraph. However, other than trees, no natural class of graphs having bounded linked-path decompositions is known.

References

[BRST89] D. Bienstock, N. Robertson, P. Seymour, and R. Thomas. Quickly excluding a forest, 1989. in preparation.

[Can15] G. Cantor. *Contributions to the Founding of the Theory of Transfinite Numbers.* Dover Publications, New York, 1915. *Translated by Philip E.B. Jourdain.*

[FL88] M. Fellows and M. Langston. Nonconstructive tools for proving polynomial-time decidability. *Journal of the Association for Computing Machinery,* 35(3):727–739, July 1988.

[Fri78] H. Friedman. Classically and intuitionistically provably recursive functions. In G.H. Müller and D.S. Scott, editors, *Higher set theory,* volume 669, pages 21–28. Springer-Verlag, Berlin, 1978.

[Fri90] H. Friedman. Personal communication, 1990.

[Gup90] A. Gupta. *Constructivity Issues in Tree Minors.* PhD thesis, Dept. of Computer Science, University of Toronto, Toronto, Canada, 1990. Also Appears as Technical Report 244/90, Dept. of Computer Science, University of Toronto.

[Gup91] A. Gupta. A regular expression characterization of minor closed tree families, 1991. in preparation.

[Hig52] G. Higman. Ordering by divisibility in abstract algebra. *Proc. London Math. Soc.,* 3:326–336, 1952.

[Kru60] J. Kruskal. Well-quasi-ordering, the tree theorem, and Vázsonyi's conjecture. *Trans. Amer. Math Soc.,* 95:210–225, 1960.

[Lei85] D. Leivant. Syntactic translations and provably recursive functions. *The Journal of Symbolic Logic,* 50:682–688, 1985.

[MR90] C. Murthy and J. Russell. A constructive proof of Higman's lemma. In *Logic in Computer Science,* 1990.

[NW63] C.St.J.A. Nash-Williams. On well-quasi-ordering finite trees. *Proc. Cambridge Phil. Soc.,* 59:833–835, 1963.

[RSa] N. Robertson and P. Seymour. Graph Minors XIII. The disjoint paths problem. in preparation.

[RSb] N. Robertson and P. Seymour. Graph Minors XV. Wagner's conjecture. in preparation.

[Tho90] R. Thomas. A Menger-like property of tree-width. The finite case. *J. Combinatorial Theory (Ser. B),* 1990. to appear.