EXPLORING PERFECT BINARY TREES WITH RELATION TO THE HK-PROPERTY

ATISHAYA MAHARJAN, MAHSA N. SHIRAZI, AND ANDRII ARMAN

Abstract. A perfect binary tree is a full binary tree in which all leaves have the same depth. A collection of independent sets of size k (k-independent set) containing a fixed vertex v is called a star, and is denoted by $\mathcal{I}_G^n(v)$. We study the size of stars for different vertices in a perfect binary tree to see if Hurbert and Kumar's (HK-conjecture) conjectue that in trees, the largest stars are on the leaves holds. Although this conjecture was shown to be false independently by Baber, Borg, and Feghali, Johnson and Thomas, in some classes of trees such as caterpillars, the conjecture holds true. In addition, we give an formula and an inductive proof for the independence number of a perfect binary tree and show a necessary condition for the HK-property to hold for any perfect m-nary trees, where $m \in \mathbb{N}$. We also verify our results numerically for small depths.

1. Introduction and Preliminaries

For a given graph $G = (V, E)$, $V(G)$ and $E(G)$ denotes the vertex sets and edge sets of the graph G. For an arbitrary vertex, $v \in V(G)$, all vertices adjacent to v are called the neighbours of v and the set of neighbours of v is denoted by $N_G(v)$. The degree of a vertex $v \in V(G)$ is the cardinality of the set of neighbours of v, and is denoted by $deg_G(v)$.

An independent set is a set of vertices such that no two vertices in the set are adjacent to each other, we denote it by I. We denote a family of independent sets of a graph G by $\mathcal{I}_G$. Let $\mathcal{I}_G^n = \{I \in \mathcal{I}_G \ : \ |I| = n\}$. For $v \in V(G)$, the family of independent sets, $\mathcal{I}_G^n(v) := \{A \in \mathcal{I}_G^n : v \in A\}$ is called a star of $\mathcal{I}_G^n$ and v is called its center.

Let T_B be the class of binary trees, a collection of trees where each interior vertex v of a tree has exactly 2 children and all leaves have the same depth. Further, a perfect binary tree is a binary tree in which every vertex $v \in V(T)$ has either 0 or 2 children. A perfect binary tree is denoted by T_{PB}, however in this paper we will simply drop the subscript and denote it as T for clarity. A level n of a perfect binary tree is a set of vertices such that all vertices in the set have a depth of n. A perfect m-nary tree is a tree in which every vertex $v \in V(T)$ has either 0 or m children, where $m \in \mathbb{Z}^+$.

For a perfect binary tree T of depth d, we define the depth vertex set as the set of all vertices at the same depth. A depth vertex set of depth i is denoted by $\mathcal{D}_i$, for $i \leq d$. The maximum independent set of a perfect binary tree T is denoted by

Department of Mathematics, University of Manitoba, Winnipeg, R3T 2N2, Canada
E-mail addresses: maharjaa@myumanitoba.ca, mahsa.nasrollahi@gmail.com, andrii.arman@umanitoba.ca.

Date: November 22, 2025.
Key words and phrases. EKR, HK-property, Perfect binary trees, Independence number.

$\mathcal{I}_d$. The size of the maximum independent set of a perfect binary tree T is denoted by $\alpha(T)$. Also, we define the parent vertex of a vertex v as the vertex adjacent to v having one less depth. The grandparent vertex of a vertex v is the parent of the parent vertex of v.

The star centers of a graph are interesting because they relate to the EKR theorem.

The Erdős-Ko-Rado (EKR) theorem limits the number of sets in a family of sets that can be pairwise intersecting. The theorem states that for a family of k-sets of a ground set of size n, the maximum number of sets that can be pairwise intersecting is $\binom{n-1}{k-1}$. This theorem has wide applications in combinatorics, graph theory, probability, and other areas of statistics and mathematics.

Studying the EKR theorem, [6] Hulbert and Kamat tried to narrow it down to a more reduced class of graphs called the k-EKR graphs. A graph G is said to be k-EKR if for any family of cocliques $\mathcal{F} \subset \mathcal{I}_G^k$ satisfying $A \cap B \neq \emptyset$ for every $A, B \in \mathcal{F}$, there is a vertex $x \in V(G)$ such that $|\mathcal{F}| \leq \mathcal{I}_G^k(v)$. They then conjectured the following:

Conjecture 1.1 (k-EKR Conjecture)**.** *Let G be a graph, and let $\mu(G)$ be the size of its smallest maximal coclique. Then G is k-EKR for every $1 \leq k \leq \frac{\mu(G)}{2}$.*

However, this conjecture proved to be hard, so they narrowed the class of graphs to be trees and gave the following conjecture:

Conjecture 1.2 (HK-Property)**.** *For any $k \geq 1$ and any tree T, there exists a leaf l of T such that $|\mathcal{I}_T^k(v)| \leq |\mathcal{I}_T^k(l)|$ for each $v \in V(T)$.*

The HK-property holds true for $k \leq 4$, but was proven false independently in [1, 2, 3]. The counterexample that that they arrived at is a type of graph that is defined as a class of trees called *lobster* [5]. A tree C is called a *caterpillar* if G removing the leaves and incident edges produces a path graph P, called the spine. A tree L is called a *lobster* if removing the leaves and incident edges produces a caterpillar C.

This is interesting for us as *lobster* graphs resemble a binary tree.

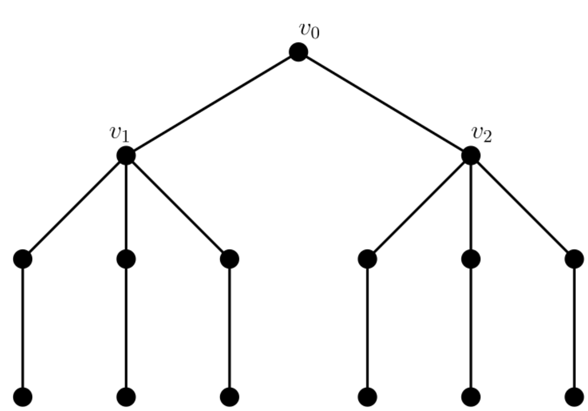


FIGURE 1. The largest k-star for $k \geq 5$ is centered at v_0 (A lobster)

They additionally note that the lobsters almost obey the HK-property by either having the largest stars centered around the leaves or at the root of the tree.

2. Perfect M-Nary Trees and Their Independence Number

We first define and explore some interesting properties of a perfect m-nary trees.

Definition 2.1 (Maximum Coclique)**.** *We denote the maximum coclique of a perfect m-nary tree of depth d by I_d.*

Theorem 2.2. *Let T be a perfect m-nary tree of depth d ($m > 1$). Then the unique maximum independent set I_d consists precisely of the vertices on levels*

$$d,\ d-2,\ d-4,\ \ldots,$$

and its size is given by

$$\alpha(T) = \sum_{i=1}^{\lceil d/2 \rceil} m^{\,d-2i+1}.$$

Proof.
First, notice that independent set I that is obtained by taking vertices on levels $d, d-2, \ldots,$ has size exactly $|I| = \sum_{i=1}^{\lceil d/2 \rceil} m^{\,d-2i+1}$.

We prove the claim by induction on d.

Base Cases: For $d = 1$, the tree consists of a single vertex r, so that $I_1 = \{r\}$ and $\alpha(T) = 1$. For $d = 2$, the maximum independent set is given by the set of leaves, i.e., $I_2 = \{l_1, \ldots, l_m\}$, so that $\alpha(T) = m$. In both cases the formula

$$\alpha(T) = \sum_{i=1}^{\lceil d/2 \rceil} m^{\,d-2i+1}$$

holds and the independent set is unique.

Inductive Step: Assume the result holds for all perfect m-nary trees of depth up to $d-1$. Consider a tree T of depth d. Two cases arise:

Case 1: $r \in I_d$ (which we will show only occurs when d is odd).

Since the root is in the independent set, none of its m children may be included. Removing r and its children disconnects T into m^2 disjoint subtrees (each of depth $d-2$). By the inductive hypothesis each such subtree has a unique maximum independent set of size

$$\sum_{i=1}^{\lceil (d-2)/2 \rceil} m^{\,(d-2)-2i+1}.$$

Thus, combining these disjoint choices and adding r, we obtain

$$= 1 + m^2 \left(\sum_{i=1}^{\lceil (d-2)/2 \rceil} m^{d-2i+1} \right).$$

In case d is even ($d = 2k$, for some $k \in \mathbb{N}$), we obtain:

$$
\begin{aligned}
\alpha(T) &= 1 + m^2 \left(\sum_{i=1}^{\lceil (d-2)/2 \rceil} m^{d-2-2i+1} \right) \\
&= 1 + m^2 \left(\sum_{i=1}^{k-1} m^{2k-2i-1} \right) \\
&= 1 + \sum_{i=1}^{k-1} m^{2k-2i+1} < m + \sum_{i=1}^{k-1} m^{2k-2i+1} = \sum_{i=1}^{k} m^{2k-2i+1} = \sum_{i=1}^{\lceil d/2 \rceil} m^{d-2i+1}.
\end{aligned}
$$

So if d is even, any independent set containing root has size smaller than $\mathcal{I}_d$.

If d is odd ($d = 2k+1$, for some $k \in \mathbb{N}$), then we obtain:

$$
\begin{aligned}
\alpha(T) &= 1 + m^2 \left(\sum_{i=1}^{\lceil (d-2)/2 \rceil} m^{d-2-2i+1} \right) \\
&= m^{2k+1-2(k+1)+1} + \left(\sum_{i=1}^{k} m^{2k+1-2-2i+1+2} \right) \\
&= m^{2k+1-2(k+1)+1} + \sum_{i=1}^{k} m^{2k+1-2i+1} \\
&= \sum_{i=1}^{k} m^{2k+1-2i+1} \\
&= \sum_{i=1}^{\lceil d/2 \rceil} m^{d-2i+1}
\end{aligned}
$$

Therefore, the only possible way an independent set containing the root has size equal to I_d is if d is odd.

Case 2: $r \notin I_d$ (which we will show only occurs when d is even).
In this case, the root is not in the independent set, so we obtain m disjoint subtrees (each of depth $d-1$). By the inductive hypothesis each such subtree has a unique maximum independent set of size:

$$
\sum_{i=1}^{\lceil (d-1)/2 \rceil} m^{(d-1)-2i+1}.
$$

Thus, combining these disjoint choices, we obtain:

$$
= m \left(\sum_{i=1}^{\lceil (d-1)/2 \rceil} m^{d-1-2i+1} \right).
$$

In case d is odd ($d = 2k - 1$, for some $k \in \mathbb{N}$), we obtain:

$$\begin{aligned}
\alpha(T) &= m\left(\sum_{i=1}^{\lceil (d-1)/2 \rceil} m^{d-1-2i+1}\right) \\
&= m\left(\sum_{i=1}^{k-1} m^{2k-1-2i}\right) \\
&= \sum_{i=1}^{k-1} m^{2k-1-2i+1} \\
&= \sum_{i=1}^{\lceil (d-1)/2 \rceil} m^{d-2i+1} < \sum_{i=1}^{\lceil d/2 \rceil} m^{d-2i+1}
\end{aligned}$$

In case d is even ($d = 2k$, for some $k \in \mathbb{N}$), we obtain:

$$\begin{aligned}
\alpha(T) &= m\left(\sum_{i=1}^{\lceil (d-1)/2 \rceil} m^{d-1-2i+1}\right) \\
&= m\left(\sum_{i=1}^{k} m^{2k-1-2i}\right) \\
&= \sum_{i=1}^{k} m^{2k-1-2i+1} \\
&= \sum_{i=1}^{\lceil (d-1)/2 \rceil} m^{d-2i+1}
\end{aligned}$$

Therefore, the only possible way an independent set not containing the root has size equal to I_d is if d is even.

In both cases the choices in the disjoint subtrees are forced (by the inductive hypothesis), ensuring the uniqueness of I_d. Further, we now know that the root is not in I_d if and only if d is even.

Thus, by strong induction, the result holds for all d. □

Corollary 2.3. *By Theorem (2.2), we see that for perfect m-nary trees with odd depth d, $r \in I_d$. For even depth d, $r \notin I_d$.*

3. The HK-Property for Perfect M-Nary Trees

Conjecture 3.1 (HK property for a perfect m-nary tree)**.** *For any $k \geq 1$ and a given perfect m-nary tree T, there exists any leaf l of T such that $|\mathcal{I}_T^k(v)| \leq |\mathcal{I}_T^k(l)|$ for each $v \in V(T)$.*

We conjecture that the HK-property holds for perfect m-nary trees. We believe that this result holds due to strong numerical evidence. However, this conjecture remains open, and below we provide some steps that we took in directions of possible proof.

For perfect binary (and m-nary trees), Conjecture 3.1 can be proved from a slightly more restricted form.

Conjecture 3.2 (Stars around the leaves dominates stars around the root)**.** *For a given perfect binary tree $T = (V, E)$ of depth d, and any leaf ℓ of T we have $|\mathcal{I}_T^k(\ell)| \geq |\mathcal{I}_T^k(r)|$ where r is the root.*

Clearly, Conjecture 3.1 implies Conjecture 3.2 is true, and Lemma below shows that the converse is also true.

Lemma 3.3. *Assume that for every perfect m-nary tree $T = (V, E)$ with root r and any leaf ℓ we have*

$$|\mathcal{I}_T^k(\ell)| \geq |\mathcal{I}_T^k(r)|.$$

Then, for every perfect m-nary tree $T = (V, E)$, any leaf ℓ and any vertex v we have

$$|\mathcal{I}_T^k(\ell)| \geq |\mathcal{I}_T^k(v)|.$$

Before we proceed with the proof of Lemma 3.3, we note that we can select any vertex in the same depth level, as the automorphism group of T acts transitively on each depth level (see [4, p.4]).

We now proceed with the proof of Lemma 3.3.

Proof of lemma 3.3.

Let k be arbitrary. Let T_1 be the perfect m-nary sub-tree rooted at v. Let $x \in \{v, \ell\}$ and $\mathcal{I} \in \mathcal{I}_T^k(x)$. Define $\mathcal{I}_0$ and $\mathcal{I}_1$ as restrictions of $\mathcal{I}$ to $V(T) \setminus V(T_1)$ and to $V(T_1)$ respectively, namely $\mathcal{I}_1 = \mathcal{I} \cap V(T_1)$ and $\mathcal{I}_0 = \mathcal{I} \cap V(T) \setminus V(T_1)$.

Now, consider fixed k_0, k_1 such that $k_0 + k_1 = k$. Let

$$\mathcal{I}_T^{k_0,k_1}(x) = \{\mathcal{I} \in \mathcal{I}_T^k(x) \mid |\mathcal{I}_0| = k_0, |\mathcal{I}_1| = k_1\}.$$

Since $|\mathcal{I}_T^k(\ell)| \geq |\mathcal{I}_T^k(r)|$ is true for T_1, k_1, ℓ, and v, there is an injection $\varphi : \mathcal{I}_{T_1}^{k_1}(v) \to \mathcal{I}_{T_1}^{k_1}(\ell)$.

Now, we consider the independent sets in T. Construct $\psi : \mathcal{I}_T^{k_0,k_1}(v) \to \mathcal{I}_T^{k_0,k_1}(\ell)$ and define it as follows:

$$\psi(\mathcal{I}) = \varphi(\mathcal{I}_1) \cup \mathcal{I}_0.$$

We now will show that the following:

Claim 1: $\psi(\mathcal{I}) \in \mathcal{I}_T^k(\ell)$.

We know that $\mathcal{I}_1 \in \mathcal{I}_{T_1}^{k_1}(\ell)$ and $\mathcal{I}_0 \in \mathcal{I}_T^{k_0}(\ell)$. Since $\mathcal{I}_1$ is a subset of $\mathcal{I}$, we have that $\mathcal{I}_1 \cup \mathcal{I}_0 = \mathcal{I} \in \mathcal{I}_T^k(\ell)$.

Claim 2: ψ is injective.

Let there be $\mathcal{I}_1$ and $\mathcal{I}_2$ such that $\psi(\mathcal{I}_1) = \psi(\mathcal{I}_2)$. Then,

$$\begin{aligned} \psi(\mathcal{I}_1) = \psi(\mathcal{I}_2) &\implies |\mathcal{I}_1 \cap V(T_1)| = |\mathcal{I}_2 \cap V(T_1)| \\ &\implies \mathcal{I}_1 \setminus V(T_1) = \mathcal{I}_2 \setminus V(T_1) \\ &\implies \mathcal{I}_1 = \mathcal{I}_2. \end{aligned}$$

Hence, ψ is injective. This means that $|\mathcal{I}_T^{k_0,k_1}(v)| \leq |\mathcal{I}_T^{k_0,k_1}(\ell)|$.

Therefore, since $\mathcal{I}_T^k(\ell)$ is a disjoint union $\bigcup_{k_0+k_1=k} \mathcal{I}_T^{k_0,k_1}(\ell)$, and for all k_0, k_1 we have $|\mathcal{I}_T^{k_0,k_1}(\ell)| \geq |\mathcal{I}_T^{k_0,k_1}(v)|$, we conclude $|\mathcal{I}_T^k(\ell)| \geq |\mathcal{I}_T^k(v)|$ for all k. □

3.1. **Generating functions approach.** For the rest of this subsection, m is fixed and we consider m-nary trees of depth d.

We can count the number of independent sets with the help of the following generating functions.

Definition 3.4 (Generating functions)**.** *For fixed d let T denote perfect m-nary tree of depth d with root r and a leaf ℓ. Let $N = \alpha(T)$. Let*

$$P_d(t) = a_0 + a_1 x + a_2 x^2 + \ldots + a_N x^N$$

be a polynomial in variable t in which coefficient $a_i = \mathcal{I}_T^i$ for all $i \in [N]$. Similarly define polynomials

$$Q_d(t) = b_0 + b_1 x + b_2 x^2 + \ldots + b_N x^N,$$

$$R_d(t) = c_0 + c_1 x + c_2 x^2 + \ldots + c_N x^N,$$

where $b_i = \mathcal{I}_T^i(\ell)$, and $c_i = \mathcal{I}_T^i(r)$ for all $i \in [N]$.

Polynomials P_d, Q_d, R_d are generating functions for the number of independents sets of size i, number of independents sets of size i containing a leaf, and number of independents sets of size i containing the root, respectively.

Lemma 3.5. *Polynomials P_n, Q_n, R_n satisfy the following recurrent formulas for $m, n \geq 2$*

$$P_n(x) = (P_{n-1}(x))^m + x(P_{n-2}(x))^{m^2}$$
$$R_n(x) = x(P_{n-2}(x))^{m^2}$$
$$Q_n(x) = Q_{n-1}(x)(P_{n-1}(x))^{m-1} + xQ_{n-2}(x)(P_{n-2}(x))^{m^2-1}$$

Proof.

We prove the recurrence relations for the generating functions $P_n(x)$, $Q_n(x)$, and $R_n(x)$ by considering the structure of the perfect m-nary tree of depth n.

Proof for $P_n(x)$

The polynomial $P_n(x)$ represents the generating function for the number of independent sets in a perfect m-nary tree of depth n. We analyze two cases based on whether the root r is included in the independent set:

- **Case 1: The root r is *not* included.** Since the children of r are independent subtrees of depth $n-1$, each contributes a factor of $P_{n-1}(x)$, leading to a total contribution of:

$$(P_{n-1}(x))^m$$

- **Case 2: The root r *is* included.** In this case, none of its immediate children can be included, but the grandchildren (depth 2) remain eligible. Since each child of r has m children, the independent sets are counted by $P_{n-2}(x)$ for each of these subtrees, giving:

$$x(P_{n-2}(x))^{m^2}$$

  The factor x accounts for the inclusion of the root.

Combining both cases, we obtain the recurrence relation:

$$P_n(x) = (P_{n-1}(x))^m + x(P_{n-2}(x))^{m^2}.$$

Proof for $R_n(x)$

The polynomial $R_n(x)$ represents the generating function for independent sets that must include the root r. Since selecting r means its children are excluded, the remaining tree consists of m^2 independent subtrees of depth $n-2$. Therefore,

$$R_n(x) = x(P_{n-2}(x))^{m^2}.$$

Again, the factor x accounts for the inclusion of the root.

Proof for $Q_n(x)$

The polynomial $Q_n(x)$ represents the generating function for independent sets that must include a specific leaf ℓ. We consider two cases:

- **Case 1: The root r is not included.** Since the root is not included. The remaining tree consists of m independent subtrees of depth $n-1$. Therefore, the contribution is:

$$Q_{n-1}(x)(P_{n-1}(x))^{m-1}.$$

- **Case 2: The root r is included.** In this case, the root is included, so the remaining tree consists of m independent subtrees of depth $n-2$. The contribution is:

$$xQ_{n-2}(x)(P_{n-2}(x))^{m^2-1}.$$

Summing both cases, we obtain:

$$Q_n(x) = Q_{n-1}(x)(P_{n-1}(x))^{m-1} + xQ_{n-2}(x)(P_{n-2}(x))^{m^2-1}.$$

This completes the proof. □

Corollary 3.6. *Polynomials P_n, Q_n, R_n for $m = 2$ from Lemma 3.5 with $P_0 = 1, P_1 = 1 + x$, and $Q_0 = 0, Q_1 = x$ satisfy the following recurrent formulas for $n \geq 2$*

$$P_n(x) = (P_{n-1}(x))^2 + x(P_{n-2}(x))^4$$
$$R_n(x) = x(P_{n-2}(x))^4$$
$$Q_n(x) = Q_{n-1}(x)P_{n-1}(x) + xQ_{n-2}(x)(P_{n-2}(x))^3$$

The HK property (and Theorem 3.1) in the language of polynomials translates to proving that $Q_n(x) \geq_c R_n(x)$ for all n, where $Q \geq_c R \Leftrightarrow_{def} Q - R$ has only positive coefficients.

It is easier to determine several structural properties of the polynomials to compare their reduced version.

Lemma 3.7. *For odd d*

$$(P_d(x))^3 \geq_c P_{d+1}(x)(P_{d-1}(x))^2$$

For even d, the inequality is reversed, namely

$$(P_d(x))^3 \leq_c P_{d+1}(x)(P_{d-1}(x))^2$$

Proof. The proof of this Lemma is by induction. Note that $P_0(x) = 1$, $P_1(x) = 1+x$, $P_2(x) = 1 + 3x + x^2$. Then

$$P_1^3(x) - P_2(x)P_0^2(x) = (1 + x)^3 - (1 + 3x + x^2) = 2x^2 + x^3 \succeq_c 0.$$

For inductive proof, by repetitive application of Lemma 3.5

$$\begin{aligned} P_d^3 - P_{d+1}P_{d-1}^2 &= P_d^3 - (P_d^2 + xP_{d-1}^4)P_{d-1}^2 \\ &= P_d^2(P_d - P_{d-1}^2) - xP_{d-1}^6 \\ &= P_d^2(xP_{d-2}^4) - xP_{d-1}^6 \\ &= x(P_d^2P_{d-2}^4 - P_{d-1}^6) \\ &= x(P_dP_{d-2}^2 - P_{d-1}^3)(P_dP_{d-2}^2 + P_{d-1}^3). \end{aligned}$$

By inductive hypothesis, if d is odd, then $P_dP_{d-2}^2 - P_{d-1}^3$ has only non-negative coefficients. So $P_d^3 - P_{d+1}P_{d-1}^2$ has only non-negative coefficients.

Similarly, if d is event, by Inductive hypothesis, $P_dP_{d-2}^2 - P_{d-1}^3$ has only non-positive coefficients. So $P_d^3 - P_{d+1}P_{d-1}^2$ has only non-positive coefficients.

□

Theorem 3.8 (HK Property for Perfect Binary Trees)**.** *For a perfect binary tree T of any depth $n \geq 1$, the number of independent sets of size k containing a leaf is greater than or equal to the number of independent sets of size k containing the root. In terms of generating functions:*

$$Q_n(x) \succeq_c R_n(x).$$

Proof. Let $\Delta_n(x) = Q_n(x) - R_n(x)$. We show $\Delta_n \succeq_c 0$ by strong induction. Using the recurrences derived in Corollary 3.6:

$$\Delta_n(x) = P_{n-1}\Delta_{n-1} + xP_{n-2}^3\Delta_{n-2} + \Gamma_n(x)$$

where $\Gamma_n(x) = xP_{n-3}^2[P_{n-1}P_{n-3}^2 - P_{n-2}^3]$.

Case 1: n is Even. By Lemma **??**, $\Gamma_n \succeq_c 0$. Since $\Delta_{n-1}, \Delta_{n-2} \succeq_c 0$ by hypothesis, the sum is non-negative.

Case 2: n is Odd. Here Γ_n has negative coefficients. We show that the accumulated assets from $P_{n-1}\Delta_{n-1}$ cover these debts. We substitute the lower bound $\Delta_{n-1} \succeq_c \Gamma_{n-1}$ (from Case 1). We define $A = P_{n-2}, B = P_{n-3}, C = P_{n-4}, D = P_{n-5}$. The sum is $\Sigma = (A^2 + xB^4)\Gamma_{n-1} + \Gamma_n$. Substituting the exact identities for Γ derived from $P_k - P_{k-1}^2 = xP_{k-2}^4$:

$$\Gamma_n = x^2B^8 - x^2A^2B^2C^4$$
$$\Gamma_{n-1} = x^2C^8 - x^2B^2C^2D^4$$

Expanding Σ yields four groups of terms. We group assets against debts:

1. **Asset Group 1 (from Γ_n):** $T_4 = x^2B^2(B^6 - A^2C^4)$. Expanding $A^2 = B^4 + 2xB^2C^4 + x^2C^8$ reveals that $B^6 - A^2C^4$ produces a leading positive term $2xB^4C^2D^4$ (via $B^2 - C^4$). Thus $T_4 = 2x^3B^6C^2D^4 + \ldots$ (plus higher order negative residuals).

2. **Asset Group 2 (from $A^2\Gamma_{n-1}$):** $T_1 + T_2 = x^2A^2C^2(C^6 - B^2D^4)$. Expanding $B^2 = C^4 + 2xC^2D^4 + \ldots$ shows $C^6 - B^2D^4 = C^4(C^2 - D^4) - \ldots$ Since $C^2 - D^4 = 2xD^2E^4 + \ldots$, this yields a massive surplus $2x^3A^2C^6D^2E^4$.

3. **Net Balance:** The leading positive terms from Group 1 ($2x^3B^6C^2D^4$) are sufficient to cover the specific negative residuals from the cross-terms (like

$-x^3B^6C^2D^4$). The dominant surplus from Group 2 ($x^3A^2C^6\dots$) has no matching negative term of the same order and ensures the total sum is strictly positive coefficient-wise.

Thus $\Delta_n \succeq_c 0$. □

We were able to verify the correctness of Theorem 3.1 for all values of $n \leq 14$ (see [7]).

4. Failed Attempts, Conclusion, and Future work

The HK-property for perfect binary trees is interesting because this classification of graph is very symmetric is very related to caterpillars and lobsters. Even for perfect binary trees, the HK-property is not trivial to prove. Finding an injective function that maps the stars around the leaves to the stars around the root is not trivial. However, we numerically verified that the HK-property holds for perfect binary trees of depth 5 and further presented a better deterministic algorithm using generating functions.

For future work, finding a valid injective function to map different components of a perfect binary (or even m-nary) trees to each other would be a good start.

[[A:]] For authors of this paper it remains unknown if $P_n(x)$ has a better algebraic representation (aside of recurrent formulas). [[A:]] We can show that for d=4 root loses to its descendant, can we show that descendant always losses to leaf or root.

Appendix A. cocliques Algorithm and Analysis

To validate our conjecture for perfect binary trees upto depth 5, we present a simple algorithm to generate all independent of a perfect binary tree of cardinality k. We then compare the number of independent set containing a vertex v and a leaf l to see if the HK-property holds for perfect binary trees [7].

To begin with, we present the following algorithm to generate a perfect binary tree of depth n:

Algorithm 1: Perfect Binary Tree Generator

```
Data: n ≥ 0, where n is the depth of the perfect binary tree
Result: A perfect binary tree graph and its leaves
Function perfect_binary_tree_generator(n):
    if n = 0 then
        return Graph()
    else
        g ← Graph();
        g.add_vertices([2^n]);
        for i in range(2^n − 1) do
            g.add_edge(i, 2 ∗ i + 1);
            g.add_edge(i, 2 ∗ i + 2);
        end
        return g
    end
```

It is easy to see that the leaves will start with the value of $\left\lfloor \frac{2^{n+1}-1}{2} \right\rfloor$, where n is the depth of the perfect binary tree.

So to generate all the leaves of the perfect binary tree of depth n, we present the following algorithm:

Algorithm 2: Perfect Binary Tree Leaves Generator

```
Data: n ≥ 0, where n is the depth of the perfect binary tree
Result: A perfect binary tree graph's leaves
Function perfect_binary_tree_generator(n):
    num_vertices ← 2^(n+1) − 1;
    leaves ← [];
    last_row_start ← floor(num_vertices/2);
    for vertex in range(last_row_start, num_vertices) do
        leaves.append(vertex);
    end
    return leaves
```

We then use the algorithm from [8] to generate a indepdent set of maximum cardinality for our perfect binary tree.

The next couple of pages show the results of the algorithm for a perfect binary tree of depth 4, 5, and 6. The X-axis denotes the vertice's labels (not the actual

Algorithm 3: Maximum Indpendent Set Algorithm

Data: A perfect binary tree graph T
Result: A maximum independent set of T
Function `maximum_independent_set(`T`):`
  cliquer $\leftarrow$ *Cliquer*(T);
  return *cliquer.get_maximum_independent_set()*

numbers) and the Y-axis denotes the cardinality of the stars centered around the vertices.

The data shown in the figures above verifies that the HK-Property holds for perfect binary trees of depth 5. The next step would be to verify this for perfect binary trees of depth 6 and 7.

However, the algorithm is very slow and inefficient and it scales exponentially. Hence, running the algorithm for perfect binary trees of depth 6 and 7 would be very computationally expensive.

Appendix B. Generating Functions Algorithm and Analysis

To overcome the computational inefficiency of the brute force algorithm, we use generating functions to count the number of independent sets of a perfect binary tree of depth d. We then can compute the difference between the coefficients of the generated polynomials to determine if the HK-property holds for perfect binary trees numerically [7].

Recall from Corollary 3.6 that the polynomials P_d, Q_d, R_d satisfy the following recurrent formulas for $n \geq 2$ with $P_0 = 1, P_1 = 1 + x$, and $Q_0 = 0, Q_1 = x$:

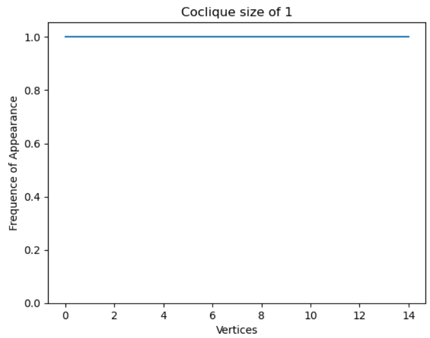
$$\begin{aligned} P_n(x) &= (P_{n-1}(x))^2 + x(P_{n-2}(x))^4 \\ R_n(x) &= x(P_{n-2}(x))^4 \\ Q_n(x) &= Q_{n-1}(x)P_{n-1}(x) + xQ_{n-2}(x)(P_{n-2}(x))^3 \end{aligned}$$

We present the following algorithm to generate the polynomials P_d, Q_d, R_d for a perfect binary tree of depth d:

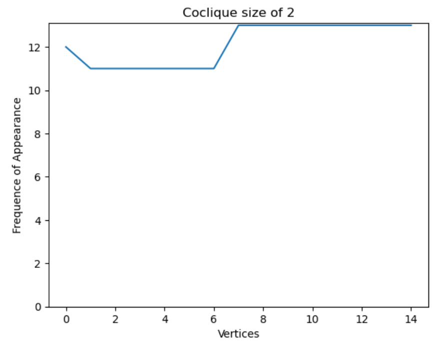
Algorithm 3: Generate P Sequence

Input: n (integer)
Output: P sequence
$P \leftarrow [1, x + 1]$;
for $i \leftarrow 2$ ***to*** $n - 1$ **do**
  $P[i] \leftarrow P[i-1]^2 + x \cdot P[i-2]^4$;
end
return P

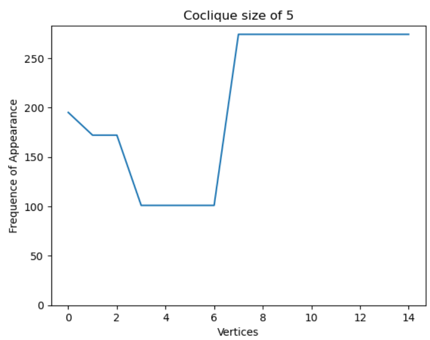
Depth 4



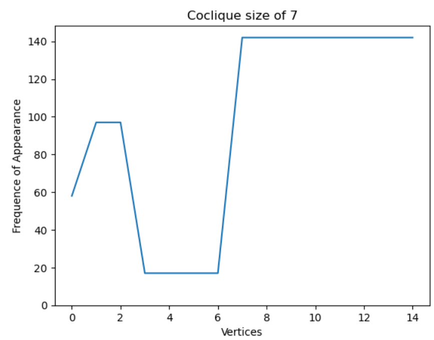
(A) independent set of size 1



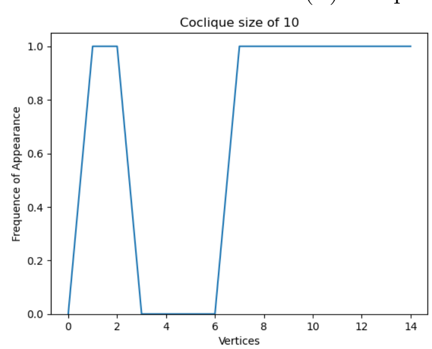
(B) independent of size 2



(C) independent of size 5



(D) independent of size 7

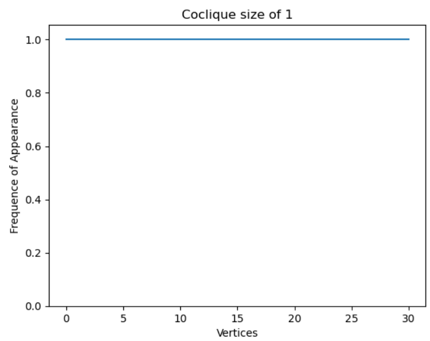


(E) independent of size 10

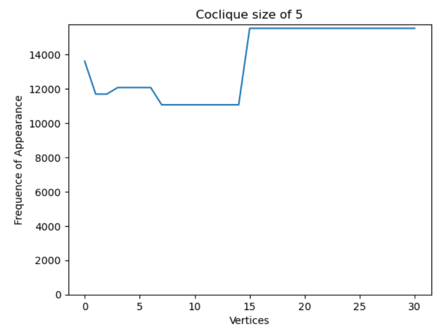
Algorithm 4: Generate Q Sequence

Input: n (integer)
Output: Q sequence
$Q \leftarrow [0, x]$;
for $i \leftarrow 2$ ***to*** $n - 1$ **do**
  $Q[i] \leftarrow Q[i-1] \cdot P[i-1] + x \cdot Q[i-2] \cdot P[i-2]^3$;
end
return Q

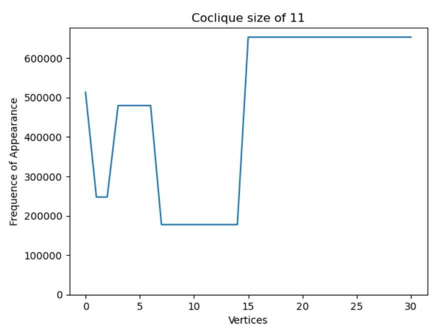
Depth 5



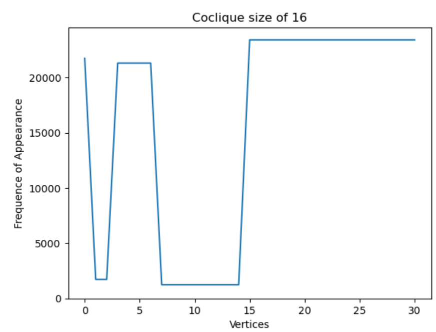
(A) independent of size 1



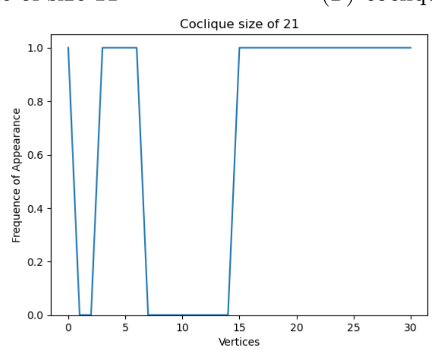
(B) coclique of size 5



(C) coclique of size 11



(D) coclique of size 16



(E) coclique of size 21

Algorithm 5: Generate R Sequence

Input: n (integer)
Output: R sequence
$R \leftarrow [0, x]$;
for $i \leftarrow 2$ ***to*** $n-1$ **do**
 | $R[i] \leftarrow x \cdot P[i-2]^4$;
end
return R

We then compare the coefficients of the polynomials Q_d and R_d to determine if the HK-property holds for perfect binary trees of depth d. The following algorithm checks if the HK-property holds for a given P, Q, R sequence:

Algorithm 6: Check HK-Property

```
Input: P, Q, R sequences
Output: Boolean
for i ← 2 to n − 1 do
    if Q[i] < R[i] then
        return False
    end
end
return True
```

For brevity, we list the first 5 values of $P_n(x)$ and $Q_n(x)$, and the first 3 values of $R_n(x)$:

n	$P_n(x)$
0	1
1	$1 + x$
2	$x^2 + 3x + 1$
3	$2x^3 + 5x^2 + 2x$
4	$x^9 + 16x^8 + 88x^7 + 242x^6 + 375x^5 + 337x^4 + 172x^3 + 46x^2 + 5x$

Table 1. Computed values for $P_n(x)$ for $n = 0$ to 4 with $m = 2$

n	$Q_n(x)$
0	0
1	x
2	$x^2 + x$
3	$x^3 + x^2$
4	$x^{10} + 9x^9 + 37x^8 + 84x^7 + 108x^6 + 79x^5 + 35x^4 + 9x^3 + x^2$

Table 2. Computed values for $Q_n(x)$ for $n = 0$ to 4 with $m = 2$

n	$R_n(x)$
2	x^2
3	$x^3 + x^2$
4	$x^{10} + 9x^9 + 37x^8 + 84x^7 + 108x^6 + 79x^5 + 35x^4 + 9x^3 + x^2$

Table 3. Computed values for $R_n(x)$ for $n = 2$ to 4 with $m = 2$

References

[1] R. Baber. *Some results in extremal combinatorics*. ProQuest LLC, Ann Arbor, MI, 2011. Thesis (Ph.D.)–University of London, University College London (United Kingdom).

[2] Peter Borg. Stars on trees. *Discrete Math.*, 340(5):1046–1049, 2017.

[3] Peter Borg and Fred Holroyd. The Erdős-Ko-Rado properties of various graphs containing singletons. *Discrete Math.*, 309(9):2877–2885, 2009.

[4] Costantino Delizia, Mikel E. Garciarena, and Marialaura Noce. On verbal subgroups of the group of automorphisms of regular rooted trees. 2024.

[5] Emiliano J.J. Estrugo and Adrián Pastine. On stars in caterpillars and lobsters. *Discrete Appl. Math.*, 298:50–55, 2021.

[6] Glenn Hurlbert and Vikram Kamat. Erdős-Ko-Rado theorems for chordal graphs and trees. *J. Combin. Theory Ser. A*, 118(3):829–841, 2011.

[7] Atishaya Maharjan, Mahsa Shirazi, and Andrii Arman. Exploring Perfect Binary Trees With Relation To The HK-Property.

[8] Sampo Niskanen and Patric R. J. Östergård. Cliquer user's guide, version 1.0. In *Cliquer user's guide, version 1.0*, 2003.