Exploring perfect binary trees with relation to the HK-property

MXML Presentation

Atishaya Maharjan
Mahsa N. Shirazi

March 28, 2024

Outline

Introduction

- Providing an overview of the Erdős-Ko-Rado (EKR) theorem and its relevance to intersecting families of sets.
- Introducing perfect binary trees and their relation to the HK-property.
- Objectives of this presentation:
  - Exploring properties of perfect binary trees.
  - Discussing the HK-property.
  - Investigating potential connections between perfect binary trees and the HK-property.

EKR Theorem

- [1] The Erdős-Ko-Rado (EKR) theorem, named after mathematicians Paul Erdős, Chao Ko, and Richard Rado, is a fundamental result in extremal set theory.
- The theorem deals with intersecting families of sets, which are collections of sets that share a common non-empty intersection.
- Specifically, the EKR theorem provides conditions under which the size of the largest intersecting family of sets can be determined.
- [2] This result has applications in combinatorics, graph theory, probability and other areas of statistics and mathematics.

[1] Erdös, Ko, and Rado, "INTERSECTION THEOREMS FOR SYSTEMS OF FINITE SETS".
[2] Anderson, *Combinatorics of finite sets*.

EKR Theorem

Definition (Intersecting family)

A family of subsets $\mathcal{F}$ of some set is **intersecting** if any two members of $\mathcal{F}$ have a non-empty intersection.

- The **Erdős-Ko-Rado** theorem limits the number of sets in an intersecting family.

Theorem (EKR Theorem)

[a] *If $\mathcal{F}$ is an intersecting family of k-subsets of an n-set (cardinality of the set is n), then*

- $|\mathcal{F}| \leq \binom{n-1}{k-1}$
- *If equality holds, $\mathcal{F}$ consists of the k-subsets that contain i, for some i in the n-set.*

[a] Erdös, Ko, and Rado, "INTERSECTION THEOREMS FOR SYSTEMS OF FINITE SETS".

HK-property

Definition (Cocliques)

- A **coclique** in a graph is a set of vertices such that no two vertices in the set are adjacent.
- The maximum size of a coclique in a graph is called the **indepdence number** of the graph. For a graph G, it is denoted by $\alpha(G)$.

Definition (Stars and Stars Center)

- Let $G = (V, E)$ be a graph, and $v \in V$. The set of all cocliques of G of cardinality n is denoted as $\mathcal{I}_T^n$.
- For $v \in V$, the family $\mathcal{I}_G^n(v) = \{A \in \mathcal{I}_G^n : v \in A\}$ is called a **star** of $\mathcal{I}_G^n$ and v is called its **start center**.

Definition (k-EKR graph)

A graph is said to be k-EKR if for any family of cocliques $\mathcal{F} \in \mathcal{I}_G^n$ with the intersection of any two sets in $\mathcal{F}$ being non-empty, there is a vertex $v \in V$ such that $|\mathcal{F}| \leq \mathcal{I}_G^n(v)$.

HK-property

Studying the EKR theorem,[3] Holroyd and Talbot made the following two conjectures:

Conjecture (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 $1 \leq k \leq \frac{\mu(G)}{2}$.

Conjecture (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 $v \in V(T)$.

[3] Holroyd and Talbot, "Graphs with the Erdős–Ko–Rado property".

HK-property

- The HK-property was proven for $k \leq 4$, but the conjecture was shown to be false.[4][5][6]

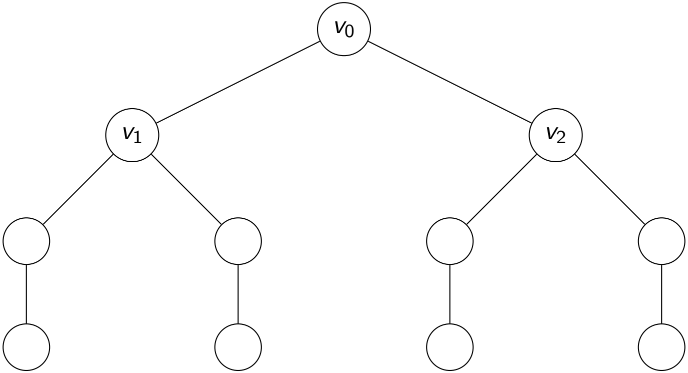


Figure 1: The largest k-star for $k \geq 5$ is centered at v_0

[4] Borg and Holroyd, "The Erdős-Ko-Rado properties of various graphs containing singletons".
[5] Borg, "Stars on trees".
[6] Baber, *Some results in extremal combinatorics*.

Some graphs that DO satisfy the HK-property[7]

- **Caterpillars:** A tree such that when you remove the leaves and incident edges, you get a path.
- **Lobsters*:** A tree such that when you remove the leaves and incident edges, you get a caterpillar.
- **Spiders:** A tree with exactly one vertex of degree greater than 2.

*Note: The graph shown in Figure 1 is a lobster. As such, it does not strictly satisfy the HK-property. Instead, the largest stars are centered at the vertices of degrees 1 or 2.

[7] Hurlbert and Kamat, "Erdős-Ko-Rado theorems for chordal graphs and trees".

Perfect Binary Tree

Definition (Depth of a vertex)

For a tree $T = (V, E)$ with a root vertex $r \in V$, the **depth** of a vertex $v \in V$ is defined as the length of the path from the r to v.

Definition (Binary Tree)

A **binary tree** is a tree in which each vertex has at most two children, referred to as the left child and the right child.

Definition (Perfect Binary Tree)

A **perfect binary tree** is a binary tree in which all the internal nodes have exactly two children and all the leaves are at the same depth.

Does a perfect binary tree satisfy the HK property?

- Probably.
- The lobster almost satisfies the HK property and the perfect binary tree has a close relation to the lobster.
- In addition, the perfect binary tree is very symmetric and has a lot of structure that we can maniputlate.

Idea 1: Expand the definition of the flip function to relate it to the perfect binary trees

- Estrugo and Pastine[8] proved the HK-property holds for caterpillars and partially lobsters by utilizing a ‘flip’ function alongside an ‘escape-path’ property.
- Why not expand the definition of the flip function to fit it to perfect binary trees?

[8]Hurlbert and Kamat, “Erdős-Ko-Rado theorems for chordal graphs and trees”.

Idea 1: Expand the definition of the flip function to relate it to the perfect binary trees

Definition (Escape Paths)

[a]Let $G = (V, E)$ be a graph and $P = v_1 v_2, \ldots, v_n$ a path of length n such that $P \subset G$. We say that P is an **escape path** from v_1 to v_n if $deg(v_n) = 1$ and $deg(v_i) = 2$ for every $2 \leq i \leq n-2$. If this is the case, we say that v_1 has an escape path to v_n.

[a]Hurlbert and Kamat, "Erdős-Ko-Rado theorems for chordal graphs and trees".

Definition (Flip function)

Let $G = (V, E)$ be a graph and $P = v_1 v_2, \ldots, v_n$ be an escape path from v_1 to v_n in G. Then the flip of P in G, $\text{flip}_P : V \to V$, is the function defined as follows:

$$\text{flip}_P(v) = \begin{cases} v & \text{if } v \notin V(P) \\ v_{n+1-i} & \text{if } v = v_i \in V(P) \end{cases}$$

Idea 1: Expand the definition of the flip function to relate it to the perfect binary trees

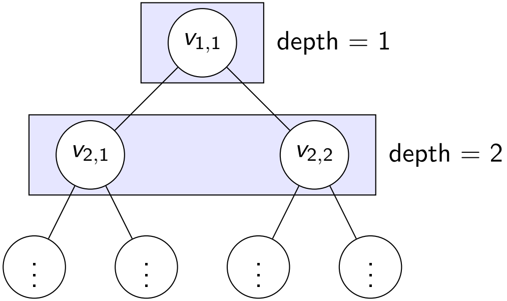
We need to address the following questions:

- How do we enumerate our vertices in the perfect binary tree so that we can use it in the flip function?
- How do we define a flip function so that it works for any vertex in the perfect binary tree?
- How do we define the escape paths in the perfect binary tree?

Idea 1: Expand the definition of the flip function to relate it to the perfect binary trees

Notation

Let $\mathcal{V}_k \in V(T)$ be the set of vertices of depth k. We call $\mathcal{V}_k$ as the depth vertex set of depth k. Index all vertices in $\mathcal{V}_k$ from left to right as $v_{k,i}$, where k is the depth of the vertex and i is the index of the vertex in $\mathcal{V}_k$ such that $1 \leq i \leq 2^{k-1}$.



Idea 1: Expand the definition of the flip function to relate it to the perfect binary trees

To show that our flip function can work on any vertex, we approach it by defining a helper flip function on every depth vertex set.

Definition (Flip Function on Depth Vertex Set)

Let T be a perfect binary tree and $\mathcal{V}_k$ be the depth vertex set of depth k. Then, the flip of $\mathcal{V}_k$ in T, denoted by $flip_{\mathcal{K}} : \mathcal{V}_k(G) \to \mathcal{V}_k(G)$, is the function defined as follows:

$$flip_{\mathcal{K}}(v_{(k,i)}) = \begin{cases} v_{(k,2^{k-1}+1-i)} & \text{if } N_T(v_{(k,2^{k-1}+1-i)}) \not\subseteq \mathcal{I}_T(v_{(k,2^{k-1}+1-i)}) \\ v_{(k,2^{k-1}+1-i)}, & \text{if } N_T(v_{(k,2^{k-1}+1-i)}) \subseteq \mathcal{I}_T(v_{(k,2^{k-1}+1-i)}) \\ flip_N(v_{(k,i)}), & \text{if } N_T(v_{(k,2^{k-1}+1-i)}) \subseteq \mathcal{I}_T(v_{(k,2^{k-1}+1-i)}) \end{cases}$$

where, $flip_N$ is defined as:

$$flip_N(v_{(k,i)}) = flip_{\mathcal{K}}(u), \text{ for all } u \in N_T(v_{(k,i)})$$

Idea 1: Expand the definition of the flip function to relate it to the perfect binary trees

Then utilizing the flip function on the depth vertex set, we tried to define the flip function diagonally on the perfect binary tree.

Our approach is as follows:

- Pick any leaf ℓ.
- Utilize the $\text{flip}_{\mathcal{K}}$ function on the depth vertex set of ℓ to flip it with either $\ell_{(k,2^{k-1}+1)}$ or $\ell_{(k,0)}$ (The leaves at the very extremes).
- Take the path P from the root to l.
- Apply a 'diagonal' flip function on P. (Note: This is still a work in progress).

Idea 1: Expand the definition of the flip function to relate it to the perfect binary trees

We have now addressed 2 of the 3 questions:

- How do we enumerate our vertices in the perfect binary tree so that we can use it in the flip function?
- How do we define a flip function so that it works for any vertex in the perfect binary tree?
- How do we define the escape paths in the perfect binary tree?

Idea 1: Expand the definition of the flip function to relate it to the perfect binary trees

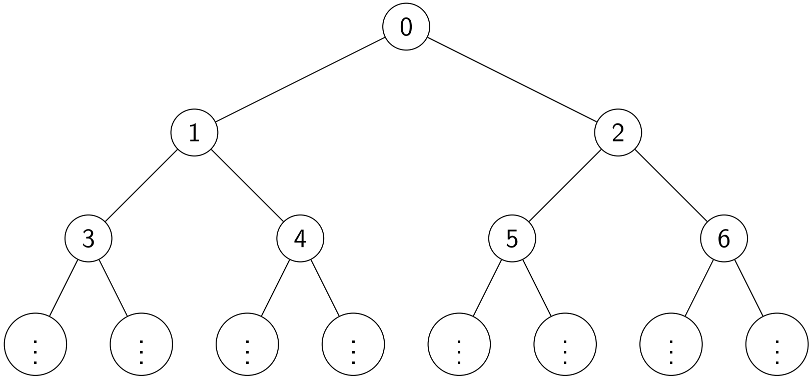
Unfortunately, we were unable to expand the definition of the escape path to the perfect binary tree. This is because it does not satisfy the core idea that we can generate a single path aside from the 2 vertices that we seek to flip. This is primarily due to the fact that all internal vertices (Excluding the root) have degree 3.

Idea 2: Use an enumeration approach with computer algorithms to verify the results

```
Data: $n \geq 0$, where $n$ is the depth of the perfect binary tree
Result: A perfect binary tree graph's leaves
Function perfect_binary_tree_generator($n$):
    $num\_vertices \leftarrow 2^{n+1} - 1$;
    $leaves \leftarrow []$;
    $last\_row\_start \leftarrow floor(num\_vertices/2)$;
    for vertex in range(last_row_start, num_vertices) do
        $leaves$.append($vertex$);
    end
    return leaves
```

Idea 2: Use an enumeration approach with computer algorithms to verify the results

The algorithm will generate a perfect binary tree of this form:



Idea 2: Use an enumeration approach with computer algorithms to verify the results

We then use the algorithm from Niskanen and Östergård, "Cliquer user's guide, version 1.0" to generate a coclique of maximum cardinality for our perfect binary tree.

```
Data: A perfect binary tree graph $T$
Result: A maximum coclique of $T$
Function maximum_coclique($T$):
    $cliquer \leftarrow Cliquer(T)$;
    return cliquer.get_maximum_coclique()
```

Idea 2: Use an enumeration approach with computer algorithms to verify the results

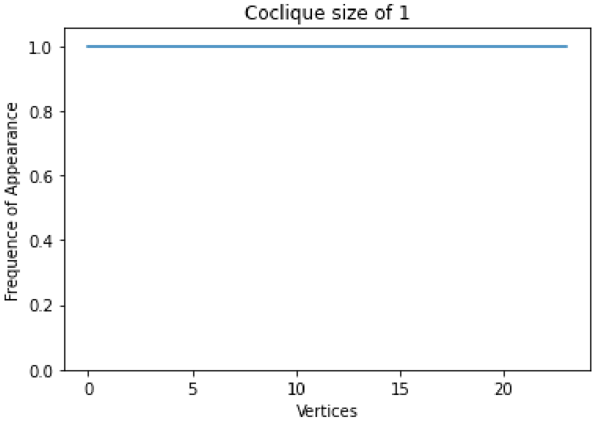
```
Data: A perfect binary tree graph $T$
Result: All cocliques of $T$
Function enumerate_cocliques($T$):
    $cocliques \leftarrow []$;
    $cocliques$.append($\emptyset$);
    for $vertex$ in $T$ do
        $new\_cocliques \leftarrow []$;
        for $coclique$ in $cocliques$ do
            for $neighbor$ in $vertex.neighbors$ do
                if $neighbor \notin coclique$ then
                    $new\_coclique \leftarrow coclique \cup \{neighbor\}$;
                    $new\_cocliques$.append($new\_coclique$);
                end
            end
        end
        $cocliques \leftarrow new\_cocliques$;
    end
    return $cocliques$
```

Idea 2: Use an enumeration approach with computer algorithms to verify the results

- We can then use the results from the algorithm to verify the HK-property for perfect binary trees.
- Do note that the algorithm is a brute-force approach and is not efficient for large trees. Hence, in the future we plan to optimize the algorithm.

Idea 2: Use an enumeration approach with computer algorithms to verify the results

This is the data for a perfect binary tree of depth 5.



Idea 2: Use an enumeration approach with computer algorithms to verify the results

- The results do indeed verify that the HK-property holds for perfect binary trees of depth 5. With the pattern, we do believe it holds for any depth perfect binary tree.
- However, it is clearly still not a proof.

Idea 3: An inductive proof

Conjecture (Maximum coclique size of perfect binary trees contains all the leaves)

For a given perfect binary tree T of depth d, the

Thank You!

Summary

- We explored the Erdős-Ko-Rado theorem and its relevance to intersecting families of sets.
- Introduced perfect binary trees and their relation to the HK-property.
- Discussed the HK-property and its relevance to intersecting families of sets.
- Demonstrated the use of computer algorithms to verify the HK-property for perfect binary trees of depth 5.
- Presented three ideas to investigate the HK-property for perfect binary trees.