Enumerating Pyranim Isomorphisms

Yoav Binyamin[1], Evan Kniffen[1]

[1]Department of Mathematics, Texas A&M University

Abstract

We study a Nim-adjacent poset game, where chips are arranged in a pyramid and two players take turns removing chips that do not currently support chips to the upper-left and upper-right; upon removing this chip, all supported chips above are also removed. Play continues until the final chip is removed. We use a bijection between the game states, Dyck paths, and unit interval graphs to count the number of possible game states up to isomorphism.

Keywords: Dyck words, unit interval graphs, combinatorial games, Catalan numbers.

Contents

1 Introduction **2**
- 1.1 Setup . . . 2
- 1.2 Isomorphisms . . . 3

2 Structural Bijections **4**
- 2.1 Dyck Paths & Words . . . 4
- 2.2 Dyck Correspondence . . . 4
- 2.3 Unit Interval Graphs . . . 5
- 2.4 Unlabeled Unit Interval Graph Correspondence . . . 5

3 Isomorphism Equivalence **6**
- 3.1 The Forward Implication . . . 6
- 3.2 Reconstruction . . . 7
  - 3.2.1 Connected Reconstruction . . . 8
  - 3.2.2 Disconnected Reconstruction . . . 10
- 3.3 The Converse Implication . . . 11

4 Formulaic Results **12**

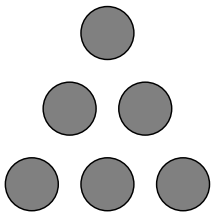
5 Conclusion **14**
- 5.1 Summary . . . 14
- 5.2 Other Findings . . . 14

1 Introduction

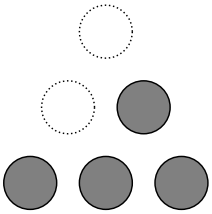
Pyramid Nim is defined in [8], studied as a combinatorial game played on a directed acyclic graph. However, the authors (i.e. Binyamin) came up with **Pyranim** independently in its more specific play on triangular starting positions. Historically, both formulations independently used "pyramid" rather than "triangle" to convey the stability of the positions in the game.

Our formulation of this 2-player impartial Nim game **Pyranim** is as follows: you start with an n-tier triangle of chips (consisting of $\frac{n(n+1)}{2}$ chips total). Player 1 goes first, and can take any of the chips on the outer layer of the pyramid. By "outer layer," we mean every chip that does not support two chips on the row directly above it; that is, every chip which can be seen observing the game from above. Upon doing so, it will also remove all chips supported by the removed chip. Player 2 does the same with the modified pyramid and play continues until the final chip is taken, and the last to take a chip is the winner. We will continue with an example game to describe the proceedings:

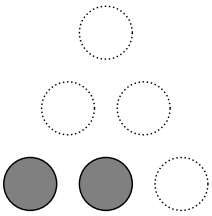
Example. For a 3-tier game with six total chips, the game starts with Player 1 being able to remove any chip other than the middle chip on the third layer.



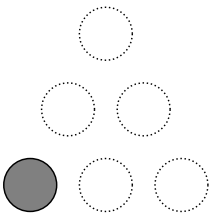
1. At this step, Player 1 *cannot* remove the bottom middle chip, but *can* remove any other chip. Player 1 removes the left chip on the second tier, taking the apex ball with it. This leaves the three bottom chips and the right chip on the second tier.



2. Player 2 removes the lower right chip leaving two chips on the lowest row.



3. Player 1 removes one chip, leaving the final chip to Player 2 who wins the game.



1.1 Setup

To formally define the game, we will treat each chip as a discrete interval containing the chips supporting it. Let $\mathcal{P}_n := \{[i,j] : 1 \le i \le j \le n\}$ be ordered by containment:

$$[a,b] \le [i,j] \iff i \le a \le b \le j.$$

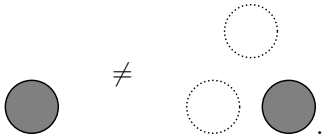
Write $c_{i,j} = [i, j]$. The bottom chips are $c_{i,i}$; the immediate supports of $c_{i,j}$, $i < j$, are $c_{i,j-1}$ and $c_{i+1,j}$. This structure will allow us to wield a multitude of wonderful algebraic and combinatorial tools.

Definition 1.1. A **stable position** X is an order ideal in $\mathcal{P}_n$ under containment:

$$c_{i,j} \in X, \quad i \le a \le b \le j \implies c_{a,b} \in X.$$

Let $\mathcal{T}_n$ be all stable positions and $\mathcal{S}_n$ those with $c_{1,1}, \ldots, c_{n,n} \in X$. Set $\mathcal{T}_0 = \mathcal{S}_0 = \{\emptyset\}$.

We interpret $\mathcal{T}_n$ as the set of all of the legal games generated by normal play and $\mathcal{S}_n$ as the set of all of the legal games with a full bottom row of chips ($\mathcal{S}_n \subseteq \mathcal{T}_n$). This is an important distinction when considering raw states generated by legal play on different pyramid sizes, since



Definition 1.2. A surviving chip $c_{i,j} \in X$ is **visible** if its two possible immediate upper chips $c_{i-1,j}$ and $c_{i,j+1}$ are not both present (indices outside $\mathcal{P}_n$ are absent).

Only visible chips are playable. Removing one chip forces the recursive removal of every chip that subsequently lacks a support. That is, if $s(x)$ is the maximal order ideal contained in $x \subseteq \mathcal{P}_n$, then the state that forms from removing $c_{i,j}$ is $s(X \setminus c_{i,j})$. Let the option set, $\mathrm{Opt}(X)$, be the set of distinct successor positions. That is,

$$\mathrm{Opt}(X) = \{s(X \setminus c_{i,j}) : c_{i,j} \text{ is visible}\}.$$

1.2 Isomorphisms

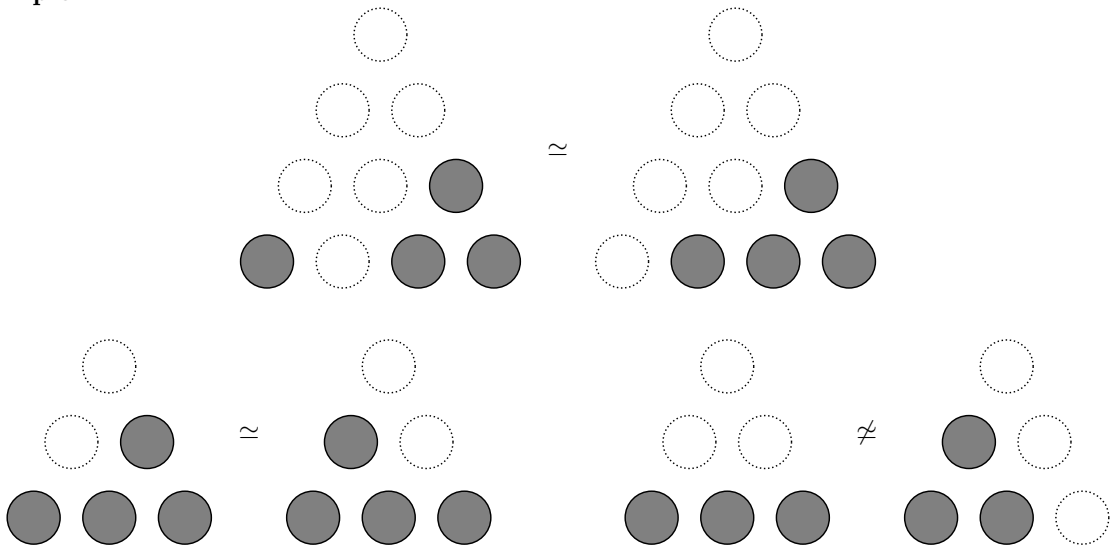
While some states may be *literally* different as defined, they play out the same *in practice* (an exmaple of which can be seen in the above figure). For example, reflections and translations of a game play out the exact same. Inspired to reduce the number of effective game states, we recursively define isomorphism.

Definition 1.3. Let X and Y be two states. Then X and Y are **isomorphic** ($X \simeq Y$) as games, if, given that the next legal moves from X are $\mathrm{Opt}(X) = \{X_1, \ldots, X_m\}$ and similarly for Y, then there is a bijection (called **isomorphism**)

$$\varphi : \mathrm{Opt}(X) \longrightarrow \mathrm{Opt}(Y) \quad \text{such that} \quad \forall i \in [m],\ \varphi(X_i) \simeq X_i.$$

Note that the empty state of 0 chips is isomorphic only to itself. Game isomorphism will sometimes be denoted $\simeq_{game}$ to differentiate with graph isomorphisms when unclear.

Example.



As expected, this isomorphism is an equivalence relation. Therefore, for all $n \geq 0$, put

$$\mathcal{K}_n = \mathcal{T}_n / \simeq_{game} \qquad \text{and} \qquad K_n = |\mathcal{K}_n|.$$

K_n represents the number of states on each n-pyramid up to isomorphism (with $K_0 = 1$). The rest of the paper focuses on finding a closed form of K_n and its ordinary generating function

$$K(x) = \sum_{n \geq 0} K_n x^n, \tag{1.1}$$

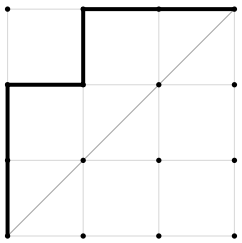
if possible.

2 Structural Bijections

In this section we will discuss the bijections present under normal state-counting as well as modulo isomorphism. In particular, the former bijects to Dyck paths and the latter bijects into unlabeled unit interval graphs.

2.1 Dyck Paths & Words

Recall that a Dyck path is a north-east lattice path from $(0, 0)$ to (n, n) that never goes strictly below the line $y = x$ (i.e. subdiagonal). The set of Dyck paths of length $2n$, and thus semilength n, will be denoted Dyck_{2n}. Dyck paths can also be written as words of the Dyck language, with up-steps written as U and down-steps written as D. We force that every prefix has at least as many U's as D's for a valid Dyck word.



$UUDUDD$

Definition 2.1. A Dyck path is **primitive** if it only returns to the diagonal at the final down-step. This can be extended to words—a word is primitive if no proper nonempty prefix is itself balanced in the number of U's and D's. Alternatively, removing the first U and last D results in another Dyck word.

We will use a well-known combinatorial result,

$$|\,\text{Dyck}_{2n}\,| = C_n = \frac{1}{n+1}\binom{2n}{n}, \tag{2.1}$$

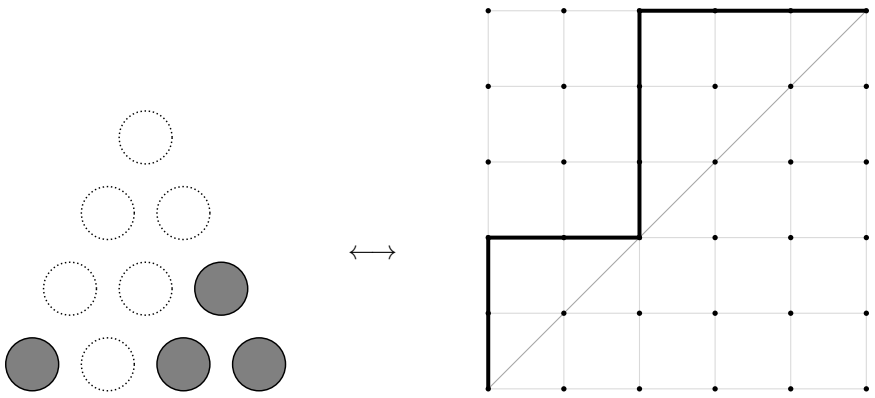
where C_n denotes the Catalan numbers [7].

2.2 Dyck Correspondence

Intuitively, we can see that our pyramids fit nicely into the full boxes generated by the area between the Dyck path and the diagonal.

Proposition 2.2. *Each stable position (i.e. legal state) for an n-pyramid game has a unique Dyck path of semilength $n+1$ and corresponding Dyck word of length $2(n+1)$. This process is reversible, inducing a clear bijection. Therefore by Equation 2.1,*

$$|\mathcal{T}_n| = C_{n+1} = \frac{1}{n+2}\binom{2n+2}{n+1}.$$



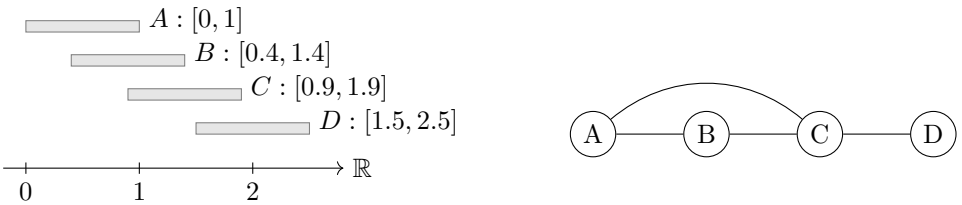
Both of these are represented by the Dyck word $UUDDUUUDDD$. This longer word can also be written in the form $U^2D^2U^3D^3$. A full [resp. empty] n-pyramid can be written as $U^{n+1}D^{n+1}$ [resp. $(UD)^{n+1}$].

2.3 Unit Interval Graphs

We now show that $\mathcal{K}_n$ corresponds to a subclass of known graphs.

Definition 2.3. A **unit interval graph** is a graph formed from a set of intervals on the real line, where each interval has a length of exactly 1. Each vertex represents an interval, and an edge connects two vertices if and only if their corresponding intervals overlap.

Example.



The following characterizations are equivalent to unit interval graphs: indifference graphs, proper interval graphs, and $K_{1,3}$-free interval graphs [3, 4]. We will use these terms interchangeably depending on what property we are trying to highlight. If the labels A, B, C, and D are removed, we get an **unlabeled unit interval graph** (or UUIG, for brevity). Permuting the labels of a labeled unit interval graph creates an isomorphic labeled unit interval graph, but these two graphs are *equal* as UUIGs.

2.4 Unlabeled Unit Interval Graph Correspondence

Definition 2.4. Construct the graph $\Gamma(X) = (V, E)$ by

$$V = \{i : c_{i,i} \in X\} \qquad \text{and} \qquad E = \{\{i, j\} : i < j,\, c_{i,j} \in X\}.$$

Therefore,

$$|X| = |V(\Gamma(X))| + |E(\Gamma(X))|. \tag{2.2}$$

Remark. We interpret this map as follows: map each game state X to its Dyck word. Remove the first U and last D. Let interval i be the interval with the ith U as the left endpoint and the ith D as the right endpoint, interpreting intervals where the right endpoint appears before the left endpoint as missing. Then $\Gamma(X)$ is the proper interval graph of these intervals.

Lemma 2.5. $\Gamma(X)$ *has the umbrella property*

$$i < j < k,\, \{i, k\} \in E \implies \{i, j\}, \{j, k\} \in E.$$

$\Gamma(X)$ *is therefore a proper interval graph; equivalently, a unit interval graph. Every unit interval graph with a chosen proper interval ordering* $1 < \cdots < n$ *yields one supported ideal [10]: retain each* $c_{i,i}$ *and retain* $c_{i,j}$ *if and only if* $\{i, j\} \in E$*. Thus supported states biject with properly ordered unit interval graphs, not with all arbitrarily labeled graphs.*

Proof. If $c_{i,k}$ survives, all its subintervals, including $c_{i,j}$ and $c_{j,k}$, survive. Conversely, the umbrella property, iterated over indices within $[i, j]$, makes every pair inside an edge $\{i, j\}$ an edge as well. The chips specified by an ordered graph are therefore an ideal. The proper–unit equivalence is proved in [3]. □

We can pass from *labeled* unit interval graphs to *unlabeled* unit interval graphs by looking at these constructions modulo isomorphism.

Theorem 2.6 (Isomorphism Equivalence)**.** *For $X, Y \in \mathcal{T}_n$,*

$$\Gamma(X) \cong \Gamma(Y) \iff X \simeq_{game} Y.$$

We will spend the majority of the paper proving this theorem. The benefits of this will become clear after taking note of known results regarding UUIGs.

Corollary 2.7. Let

$$\mathcal{S}_n^* = \mathcal{S}_n / \simeq_{game}$$

By Theorem 2.6, $\mathcal{S}_n^*$ bijects with UUIGs on n vertices (thus the two objects are equinumerous). Let u_n denote the counts of those graphs—corresponding to the sequence A005217 [12]. Then $|\mathcal{S}_n^*| = u_n$. As consequence of working modulo game isomorphism, we get

$$K_n = \sum_{k=0}^{n} u_k. \tag{2.3}$$

Indeed, we can consider states of different-size support separately; if a state X has k bottom chips, then one can trivially find an isomorphism between X and an element of $\mathcal{S}_k$.

3 Isomorphism Equivalence

3.1 The Forward Implication

Lemma 3.1. *For all visible $c_{i,j}$ in the order ideal X, removing that chip results in the position*

$$M_{c_{i,j}}(X) = X \setminus \{c_{a,b} \in X : a \le i \le j \le b\}.$$

Where distinct playable chips give distinct legal successor positions.

Proof. Every superinterval of $[i, j]$ requires it as support through a chain of immediate supports and must disappear. Deleting the entire set of superintervals in $M_{c_{i,j}}(X)$ leaves an ideal: any subinterval of a surviving chip still survives, as it is unrelated to the original chip—or any chip removed thereafter. Thus no other chip can lose a support. To see that distinct chips $c_{i,j}$ and $c_{k,l}$ yield distinct positions, assume without loss of generality that $[k, l] \not\subseteq [i, j]$. Therefore, the interval $[i, j]$ is *not* a superinterval of $[k, l]$, meaning $c_{i,j}$ is preserved when $c_{k,l}$ is removed. Because $c_{i,j}$ is eliminated when being played, we have

$$c_{i,j} \in M_{c_{k,l}}(X) \setminus M_{c_{i,j}}(X),$$

so the successor positions are distinct. □

Lemma 3.2. *Graph components occur in consecutive runs of bottom indices and no retained chip spans two components. A chip move modifies only its own component.*

Proof. If consecutive vertices lack an edge, a crossing edge would force that edge by Lemma 2.5, so there are no crossing edges. If a bottom chip is missing, no chip containing that singleton can survive. A retained interval $[i, j]$ forces all consecutive edges between i and j, hence cannot span distinct components. Therefore $M_{c_{i,j}}(X)$ is restricted to one component. □

Proposition 3.3 (The Forward Implication)**.** *For $X, Y \in \mathcal{T}_n$,*

$$\Gamma(X) \cong \Gamma(Y) \implies X \simeq_{game} Y.$$

Proof. By Lemma 3.2, $\Gamma(X)$'s connected components' vertices occupy consecutive indices, and no surviving chip crosses from one connected component to another. Since $\Gamma(X) \cong \Gamma(Y)$, we may match the connected components of X with those of Y. Uniqueness of the primitive endpoint word says that the ordered chip patterns in a matched pair agree, possibly after reversing one connected component. Thus we obtain a bijection $F : X \to Y$ by moving entire connected components and, if necessary, reversing their local indices. On a connected component of length m, reversal is defined by the map

$$c_{ij} \longmapsto c_{m+1-j,\, m+1-i},$$

where i, j are interpreted as local indices.

We claim that this same map works throughout the game, even after bottom chips have been removed. For any stable subposition $Z \subseteq X$, put $\widehat{F}(Z) = \{F(c) : c \in Z\}$. Shifting or reversing a connected component preserves containment of intervals (since chips in different connected components cannot contain one another). Hence $\widehat{F}(Z)$ is stable. Moreover, the two possible chips immediately above c are sent to the two possible chips immediately above $F(c)$, with their roles exchanged under reversal. Note that any upper chip crossing a connected component boundary is absent on both sides. Consequently,

$$c \text{ is visible in } Z \iff F(c) \text{ is visible in } \widehat{F}(Z).$$

The cascade after playing c removes precisely the surviving chips whose intervals contain c. Since F preserves containment,

$$\widehat{F}\big(M_c(Z)\big) = M_{F(c)}\big(\widehat{F}(Z)\big).$$

Thus by Lemma 3.1, $\widehat{F}$ matches every legal option of Z with a legal option of $\widehat{F}(Z)$, and the inverse connected component map matches them back. Because $\widehat{F}$ is injective on positions, distinct actual successors remain distinct, even when they belong to the same game-isomorphism class. We keep this one connected component map for all descendants; we do not compress and relabel each successor anew.

Starting with $\widehat{F}(X) = Y$, the displayed identity gives a bijection of options at every subsequent position. Induction on the number of surviving chips therefore gives $X \simeq_{\text{game}} Y$. □

Remark. The above proposition relies on the known result that unit interval graphs are isomorphic if and only if they have isomorphic connected components, and connected unit interval graphs are isomorphic if and only if they were created from the same or reversed order of intervals [2]. In the game sense, this corresponds to translations of connected components, reversals within connected components, and permutations of the order of appearance of connected components. However, these clearly preserve isomorphism. The *converse implication*, however, says that these are in fact the *only* ways for two game states to be isomorphic; we prove the converse in Section 3.3.

3.2 Reconstruction

The converse implication of Theorem 2.6 means that for a state X, knowing the graphs of $\mathrm{Opt}(X)$, we can reconstruct the original $\Gamma(X)$ uniquely. However, we will prove an even *stronger* statement: rather than the entirety of $\mathrm{Opt}(X)$, one only needs the states where a *bottom chip* was removed from X. This is captured by the following definition and theorem.

Definition 3.4. For a unit interval graph G, choose a proper interval ordering within each connected component. Select its first and last vertices if it has at least two vertices, or its unique vertex if it is a singleton. The **component-end deck** is the multiset

$$\partial G = \{\{[G - v] : v \text{ is a selected component end}\}\},$$

where $[\cdot]$ denotes graph isomorphism class and $G - v$ is the graph gotten by removing v and all of the edges connected to it. An element of ∂G is called a **card**.

Theorem 3.5 (Reconstruction)**.** *Every unit interval graph G is determined up to isomorphism by*

$$(|V(G)|, |E(G)|, \partial G). \tag{3.1}$$

3.2.1 Connected Reconstruction

We'll first prove the theorem for connected $G = \Gamma(X)$, and then use the result for each of the connected components of a disconnected G. When G is connected, X's Dyck word is primitive.

Definition 3.6. Let w be a primitive Dyck word, and write $w = U^r Dz$ for z a word on U's and D's. Then

(i) $\rho(w)$ flips the order of the U and D, and then swaps every U to a D, and vice versa.

(ii) $[w] := \{w, \rho(w)\}$.

(iii) $L(w) := U^{r-1}z$ deletes the first U and first D.

(iv) $R(w) := \rho(L(\rho(w)))$ deletes the last U and last D.

(v) For $1 \le d \le r$, $P_d(w) := U^{d+1}DU^{r-d}Dz$.

Remark. We interpret this as follows:

(i) ρ flips the game. If one player sees w, then the player on the other end of the table sees $\rho(w)$.

(ii) This is the set of all primitive Dyck words with the same (connected) interval graph as w. As mentioned before, it is a known result that if the resulting graph is connected, then these are the only possible representatives for primitive Dyck words [2].

(iii) L removes the leftmost (bottom) chip.

(iv) R removes the rightmost (bottom) chip.

(v) P_d adds $d+1$ chips on the left. The original w had $r-1$ chips to add on top of, so $d \le (r-1)+1 = r$.

Theorem 3.7 (Conjugacy)**.** *[11] Let A, B, z be words and $|A| = |B| = h > 0$. Then*

$$Az = zB$$

if and only if for some $t \ge 0$ and words p, q with $|p| < h$,

$$A = pq, \qquad B = qp, \qquad z = A^t p.$$

Proof. If $|z| \ge h$, then comparing the initial h letters, $z = Az_1$ for some word z_1, where $|z_1| = |z| - h$. Canceling, $Az_1 = z_1 B$. Continuing inductively, the process terminates because z has finite length. We get $z = A^t p$ and $Ap = pB$ for p some word and $|p| < h$. Inspecting the first h letters, $A = pq$ for some word q, and $B = qp$. The converse follows computationally from $(pq)^{t+1}p = (pq)^t p(qp)$. □

Lemma 3.8 (Nine-case Orientation)**.** *Let w be primitive, $d \ge 1$, and assume both $P_d(w)$ and $P_d(\rho(w))$ are defined. Then*

$$[R(P_d(w))] = [R(P_d(\rho(w))] \implies w = \rho(w).$$

In particular,

$$[R(P_d(w))] = [R(P_d(\rho(w))] \implies [P_d(w)] = [P_d(\rho(w))].$$

Proof. If $w = U^k D^k$, then it is self-reflecting, so the result is trivially true. Otherwise, write for some word z

$$w = U^r DzUD^s, \qquad \#U(z) - \#D(z) = s - r.$$

Then

$$\rho(w) = U^s D\rho(z)UD^r. \tag{3.2}$$

Put $\alpha = r - d \ge 0$ and $\beta = s - d \ge 0$. Then

$$a = R(P_d(w)) = U^{d+1}DU^{\alpha}DzD^{s-1},$$

$$b = R(P_d(\rho(w))) = U^{d+1}DU^{\beta}D\rho(z)D^{r-1}.$$

By assumption, $a = b$ or $a = \rho(b)$. If $a = b$ then $\alpha = \beta$, so $r = s$ and $z = \rho(z)$. Therefore, $w = \rho(w)$, and we are done.

If $a = \rho(b)$, then

$$U^{d+1}DU^{\alpha}DzD^{s-1} = U^{r-1}zUD^{\beta}UD^{d+1}. \tag{3.3}$$

Inspecting Equation 3.3, $d+1 \geq r-1$ and $d+1 \geq s-1$. Hence, $0 \leq \alpha \leq 2, 0 \leq \beta \leq 2$. This gives us the nine cases.

Let $A_\alpha = U^{2-\alpha}DU^{\alpha}D$ and $B_\beta = UD^{\beta}UD^{2-\beta}$. Then $A_\alpha z = zB_\beta$. Using Theorem 3.7, we arrive at the following conclusions as to what z is for each pair (α, β), where $t \geq 0$:

	$\beta = 0$	$\beta = 1$	$\beta = 2$
$\alpha = 0$	$***$ $(UUDD)^t$	$*$ N/A	$**$ $(UUDD)^tU$
$\alpha = 1$	$*$ N/A	$***$ $(UD)^t$	$*$ N/A
$\alpha = 2$	$**$ $(DUUD)^tD$	$*$ N/A	$***$ $(DUUD)^tDU$

- (*) For the cases marked with *, one *cannot* find p, q for which $|p| < 4$, $A_\alpha = pq$, and $B_\beta = qp$. This is checked by noting that the 4 possible p do not satisfy the relations. Therefore, these cases are impossible.

- (**) For the cases marked with **, one *can* find p, q for which $|p| < 4$, $A_\alpha = pq$, and $B_\beta = qp$. However, the number of U and D for w is not the same, so these cases are also impossible. Indeed,

  (i) $(0, 2)$: $p = U, q = UDD$ is the only option that works.

  $$\#U(z) - \#D(z) = s - r = \beta - \alpha = 2.$$

  However, z clearly has $\#U(z) - \#D(z) = 1 \neq 2$.

  (ii) $(2, 0)$: $p = D, q = UUD$ is the only option that works.

  $$\#U(z) - \#D(z) = s - r = \beta - \alpha = -2.$$

  However, z clearly has $\#U(z) - \#D(z) = -1 \neq -2$.

- (***) The cases marked with *** *are* possible. One can find p, q for which $|p| < 4$, $A_\alpha = pq$, and $B_\beta = qp$. Then, these choices imply that $r = s$, since $\alpha = \beta$. They also imply that $z = \rho(z)$. Indeed,

  (i) $(0, 0)$: We can choose any p of the 4, but only $p = \emptyset, q = UUDD$ satisfies

  $$\#U(z) - \#D(z) = \beta - \alpha = 0.$$

  Therefore, $z = (UUDD)^t\emptyset = (UUDD)^t$, so $z = \rho(z)$.

  (ii) $(1, 1)$: We can choose either $p = UD$, $q = UD$ or $p = \emptyset$, $q = UDUD$. In the former,

  $$z = (UDUD)^tUD,$$

  so there are *odd* many UD. Meanwhile, in the latter,

  $$z = (UDUD)^t\emptyset = (UDUD)^t,$$

  so there are *even* many UD. Combining the two cases, $z = (UD)^t$ for some $t \geq 0$. This satisfies

  $$\#U(z) - \#D(z) = \beta - \alpha = 0.$$

  Finally, $z = \rho(z)$.

  (iii) $(2, 2)$: $p = DU, q = UD$ is the only choice. $z = (DUUD)^tDU$,

  $$\#U(z) - \#D(z) = \beta - \alpha = 0,$$

  and $z = \rho(z)$.

Therefore, the only possible cases satisfy $r = s$ and $\rho(z) = z$. By inspecting Equation 3.2, $w = \rho(w)$. The "in particular" follows immediately, since $P_d(w) = P_d(\rho(w))$.

□

Proposition 3.9 (Connected Reconstruction)**.** *Every connected unit interval graph G is determined up to isomorphism by*

$$(|V(G)|, |E(G)|, \partial G).$$

Proof. The cases of $|V(G)| \leq 2$ are trivially true. Suppose $|V(G)| \geq 3$, and G, H are connected unit interval graphs with equal triples. Select a card $[A]$ from ∂G. Reflect either parent graph, if needed, so that $[A]$ is its left-end card. Since consecutive vertices in a connected proper interval ordering are adjacent, deleting an end preserves connectedness. Give A a primitive word w. The degree of the deleted vertex is the same in both parents:

$$d = |E(G)| - |E(A)| = |E(H)| - |E(A)|.$$

Since the parents are connected, $d \geq 1$. The deleted left-end vertex contributes the first U and first D of the parent word, with that D following exactly d of the remaining U's. Deleting these two letters recovers either w or $\rho(w)$.

Therefore, each parent has a word which is either $P_d(w)$ or $P_d(\rho(w))$. If the operation is defined for only one of w or $\rho(w)$, then the two parents G, H are represented by the same word, and are therefore in the same isomorphism class. If the operation is defined for both w and $\rho(w)$, then they are either both represented by the same word, or one of G, H is represented by $P_d(w)$ and the other by $P_d(\rho(w))$.

In the latter case, remove one occurrence of $[A]$ from each of the equal two-entry end decks. The remaining entries are the right-end cards, so

$$[R(P_d(w))] = [R(P_d(\rho(w)))].$$

This cancellation is valid even when both entries of an end deck equal $[A]$. Hence, by Lemma 3.8,

$$[P_d(w)] = [P_d(\rho(w))],$$

so that G, H are isomorphic. □

3.2.2 Disconnected Reconstruction

We'll prove Theorem 3.5.

Proof. We argue by strong induction on $|V(G)|$. The cases of $|V(G)| \leq 2$ are determined by the vertex and edge counts. If G has q nonsingleton components and s singleton components, then

$$|\partial G| = 2q + s.$$

Thus, for $|V(G)| \geq 3$, a connected graph has two cards, whereas a disconnected graph has at least three. The triple therefore determines whether G is connected.

Since the statement is proven for connected G by Proposition 3.9, let G be disconnected. If G is edgeless, this is trivially true. Otherwise, let M be the order of the largest connected component of G, in which case $M \geq 2$. Removing an end of a connected proper interval graph preserves connectedness and reduces its order by 1 if it is not a singleton, and deletes it if it is a singleton. Indeed, consecutive vertices in a connected proper interval ordering are adjacent; otherwise, Lemma 2.5 would forbid every edge crossing their gap.

Since G has at least two connected components, there is a card of G for which a component of order M is maintained. Consequently, M is the largest component order occurring in any card of ∂G.

For the following, let T denote an isomorphism class of connected proper interval graphs on M vertices. For a card $F \in \partial G$, let $b_T(F)$ be the multiplicity of T as a component of F. Write the vector

$$b(F) = (b_T(F))_T,$$

indexed by T.

If $b(F) = b(F')$ for every $F, F' \in \partial G$, then every component is in the same class T. Indeed, a smaller component would allow a deletion preserving every order-M component, while deletion in an order-M component decreases one coordinate. Two different order-M types would likewise allow deletions

decreasing different coordinates. Either possibility would give unequal vectors. Since G has at least two components, a copy of T survives in every card, so its type is determined. Thus, $[G]$ is the class of $|V(G)|/M$ copies of T.

Otherwise, if the vectors $b(F)$ are not all equal, then for each T there is an original component other than a copy of T. Deleting an end in that component retains all copies of T, and no deletion creates a new component of order M. Therefore, for each T, the number of components of class T in G is

$$m_T := \max_{F \in \partial G} b_T(F).$$

Therefore, we know all of the components of G of order M up to isomorphism. Now let Q be their disjoint union, and consider the subgraph $R = G - Q$ made by excluding these components. If R is empty, then $G = Q$ is already determined.

Otherwise, we know ∂R exactly from the cards $F \in \partial G$ with

$$b_T(F) = m_T \qquad \text{for every } T.$$

Indeed, deletion in an order-M component decreases one of these coordinates, while deletion in a smaller component leaves all of them unchanged. Thus, these cards are precisely the end vertex removals in components with fewer than M vertices. Removing Q from each selected card, while preserving every occurrence, gives the multiset ∂R. Here, $F - Q$ means the class F with all of its components of order M removed.

We also know

$$|V(R)| = |V(G)| - |V(Q)|, \qquad |E(R)| = |E(G)| - |E(Q)|.$$

Hence, the entire triple (3.1) for R is determined. Since R has fewer vertices than G, the inductive hypothesis determines R up to isomorphism. Knowing all of the other components, G is determined up to isomorphism. □

3.3 The Converse Implication

We will now define some invariants that occur between games. Let $X \simeq_{\text{game}} Y$. Then the number of chips at the bottom (labelled c_{ii}) and the total number of chips of X and Y are the same. Additionally, if $\ell_{\max}(X)$ is the maximum number of moves possible on X before the null position, and $\ell_{\min}(X)$ is the minimum number of moves possible on X before the null position, then

(i) $\ell_{\max}(X) = |V(\Gamma(X))| + |E(\Gamma(X)| = |X|$ is the total number of chips of X, and

(ii) $\ell_{\min}(X) = |V(\Gamma(X))|$ is the total number of bottom chips of X.

Consequently,

$$\ell_{\max}(X) = \ell_{\max}(Y), \qquad \ell_{\min}(X) = \ell_{\min}(Y),$$
$$|V(\Gamma(X))| = |V(\Gamma(Y))|, \qquad |E(\Gamma(X))| = |E(\Gamma(Y))|.$$

(Indeed, the invariance of $\ell_{\max}$ is a trivial inductive argument on the number of chips.

The invariance of $\ell_{\min}$ follows from the fact that if $X \simeq_{\text{game}} Y$ and $Z \subseteq X$ is stable, then there is a stable $W \subseteq Y$ with $Z \simeq_{\text{game}} W$. Taking Z as all of the bottom chips of X,

$$\#\text{ of bottom chips of } X = \ell_{\max}(Z) = \ell_{\max}(W) \leq \#\text{ of bottom chips of } Y.$$

Repeating the argument for Y gives the reverse inequality.

The rest of the equalities are obvious.

Note that the options $Z \in \text{Opt}(X)$ for which $\ell_{\min}(Z) = \ell_{\min}(X) - 1$ are exactly the options obtained from removing visible bottom chips, which are the component-end vertices of $\Gamma(X)$.

Within each connected component, consecutive vertices are adjacent: otherwise, Lemma 2.5 would forbid every edge crossing their gap. Thus, an interior bottom chip has both immediate upper chips present, while a component-end bottom chip lacks at least one. Hence, the visible bottom chips are exactly the component ends, with a singleton contributing one such move.

For a component end i, every surviving chip containing i has i as an endpoint, since all bottom indices of that chip belong to the same component. The cascade therefore deletes precisely vertex i and its incident edges, so the resulting option graph is $\Gamma(X) - i$. Different bottom moves give distinct successor positions, since each removes its own bottom chip and retains every other bottom chip.

The associated graph options, counted with multiplicity, form the multiset $\partial\Gamma(X)$. Equivalently,

$$\partial\Gamma(X) = \{\{[\Gamma(Z)] : Z \in \text{Opt}(X),\ \ell_{\min}(Z) = \ell_{\min}(X) - 1\}\}.$$

Proposition 3.10 (The Converse)**.** *For X, Y stable positions,*

$$X \simeq_{game} Y \implies \Gamma(X) \simeq \Gamma(Y).$$

Proof. Induct on the common chip number $\ell_{\max}(X)$.

The empty case (no chips) is trivially true.

Suppose $X \simeq_{\text{game}} Y$ and the claim holds for all smaller-chip game-isomorphic positions. Then the option bijection between $\text{Opt}(X)$ and $\text{Opt}(Y)$ restricts to a bijection between the options where a bottom chip was removed. However, by the inductive hypothesis, the graphs of the isomorphic states there are isomorphic. Hence, $\partial\Gamma(X) = \partial\Gamma(Y)$. However, as explained before, we also know that

$$|V(\Gamma(X))| = |V(\Gamma(Y))|, \quad |E(\Gamma(X))| = |E(\Gamma(Y))|.$$

Hence,

$$(|V(\Gamma(X))|, |E(\Gamma(X))|, \partial\Gamma(X)) = (|V(\Gamma(Y))|, |E(\Gamma(Y))|, \partial\Gamma(Y)).$$

By Theorem 3.5, $\Gamma(X) \simeq \Gamma(Y)$.

□

4 Formulaic Results

Since we have proven both directions of the argument, Theorem 2.6 is true and we can conclude Corollary 2.7. Thus we get Equation 2.3:

$$K_n = \sum_{k=0}^{n} u_k.$$

Define $K(x)$ as in Equation 1.1, and similarly

$$U(x) = \sum_{n\geq 0} u_n x^n,$$

with $u_0 = 1$. Thus by convolution of ordinary generating functions,

$$K(x) = \sum_{n\geq 0} K_n x^n = \left(\sum_{j\geq 0} u_j x^j\right)\left(\sum_{k\geq 0} x^k\right) = \frac{U(x)}{1-x}. \tag{4.1}$$

Now we will use known results of types of unit interval graphs on n vertices, most of which were proven by Hanlon in [1]. Firstly,

$$U(x) = \exp\left(\sum_{k\geq 1} \frac{\Psi(x^k)}{k}\right) \tag{4.2}$$

where $\Psi(x)$ is the ordinary generating function for the number of connected unit interval graphs on n vertices up to isomorphism, which we will denote c_n. Thus,

$$\Psi(x) = \sum_{n\geq 1} c_n x^n,$$

and by Burnside's Lemma (accounting for reflections) [6],

$$c_m = \frac{1}{2}\left(C_{m-1} + \binom{m-1}{\lfloor (m-1)/2 \rfloor}\right).$$

So,

$$\Psi(x) = \sum_{n\geq 1} \frac{1}{2}\left(C_{n-1} + \binom{n-1}{\lfloor (n-1)/2 \rfloor}\right) x^n = \frac{1}{4}\left(\frac{1+2x}{\sqrt{1-4x^2}} - \sqrt{1-4x}\right).$$

Therefore by Equations 4.1 and 4.2,

$$K(x) = \frac{1}{1-x}\exp\left(\sum_{k\geq 1} \frac{1}{4k}\left(\frac{1+2x^k}{\sqrt{1-4x^{2k}}} - \sqrt{1-4x^k}\right)\right). \tag{4.3}$$

Which can be written as the Euler product

$$K(x) = \frac{1}{1-x} \prod_{n \geq 1} (1 - x^n)^{-c_n},$$

which provides a clear generating function for K_n. Thus we can write it in expanded form as a formal power series:

$$K(x) = 1 + 2x + 4x^2 + 8x^3 + 17x^4 + 38x^5 + 93x^6 + 244x^7 + 691x^8 + 2080x^9 + \cdots .$$

Proposition 4.1. *$K(x)$ is not algebraic.*

Proof. By Equation 4.3, the analytic continuation has a square-root branch point at

$$x = \frac{1}{\sqrt[n]{4}}, \qquad \text{for every odd } n \geq 1.$$

At each of these points, only the nth summand adds that particular singularity, so it is still there after exponentiation. Infinitely many branch points imply $K(x)$ is transcendental [5]. □

While we cannot find a closed algebraic form for $K(x)$, we *can* get a nicer form for K_n.

Theorem 4.2. *For all $n \geq 0$,*

$$K_n = \sum_{\substack{k_1, \ldots, k_n \geq 0 \\ \sum_{m=1}^{n} m k_m \leq n}} \prod_{m=1}^{n} \binom{c_m + k_m - 1}{k_m}.$$

Proof. Here, k_m is the number of components with m vertices. So a choice $(k_1, \ldots, k_n)$ describes a graph with

$$k_1 + 2k_2 + \cdots + nk_n$$

vertices. The condition $\sum_{m=1}^{n} m k_m \leq n$ allows us to count every graph of order at most n.

For a fixed m, there are c_m connected graph types available. We must choose k_m components from these, allowing for repeated copies and ignoring their order. If the types are $T_1, \ldots, T_{c_m}$ and a_i is the number of copies of T_i, then

$$a_1 + \cdots + a_{c_m} = k_m, \qquad a_i \geq 0.$$

Clearly, the number of choices is

$$\binom{c_m + k_m - 1}{k_m}.$$

For example, $c_3 = 2$: the connected 3-vertex types are the path P_3 and the triangle K_3. Choosing two such components gives

$$P_3 \sqcup P_3, \qquad P_3 \sqcup K_3, \qquad K_3 \sqcup K_3,$$

where $\sqcup$ denotes disjoint union. Accordingly, the factor for $m = 3$ and $k_3 = 2$ is

$$\binom{2 + 2 - 1}{2} = 3.$$

The product combines the independent choices for each component size. The sum then runs over all possible component counts with total order at most n. Since every graph has a unique decomposition into connected components, each graph isomorphism class is counted exactly once. In particular, the choice

$$k_1 = \cdots = k_n = 0$$

contributes 1: the empty graph. □

Example. For $n = 3$, the possible component sizes and their contributions are

Component sizes	# of choices
Empty graph	1
1	1
$1+1$	1
2	1
$1+1+1$	1
$2+1$	1
3	2

Hence,

$$K_3 = 1+1+1+1+1+1+2 = 8.$$

Remark. Replacing the inequality $\sum_{m=1}^{n} mk_m \leq n$ with the corresponding equality $\sum_{m=1}^{n} mk_m = n$ would instead give u_n.

5 Conclusion

5.1 Summary

Because the isomorphisms of unit interval graphs are completely characterized, we arrive at a complete characterization of game isomorphisms.

Corollary 5.1. For $X, Y \in \mathcal{T}_n$, the following are equivalent:

(i) $X \simeq_{\text{game}} Y$

(ii) $\Gamma(X) \simeq \Gamma(Y)$

(iii) Y is a combination of permutations of the order of supported components of X, translations of supported components of X, and reversals (flips) of supported components of X.

Proof. (i) $\iff$ (ii) is Theorem 2.6 and follows from Proposition 3.3 and Proposition 3.10.
The equivalence to (iii) follows from [2]. □

Remark. Unfortunately, game isomorphism is too simple to help studying the optimal strategy of Pyranim, which we discuss in another paper. However, game isomorphism *does* give a relatively natural sense of isomorphism for Dyck paths.

We also have the following results:

$$|\mathcal{T}_n| = C_{n+1}, \qquad |\mathcal{S}_n^*| = K_n - K_{n-1},$$

$$K(x) = \frac{U(x)}{1-x},$$

$$K_n = \sum_{\substack{k_1,\ldots,k_n \geq 0 \\ \sum_{m=1}^{n} mk_m \leq n}} \prod_{m=1}^{n} \binom{c_m + k_m - 1}{k_m}.$$

We call the sequence $(K_n)_{n=0}^{\infty}$ the **Kniffen knumbers** (the 'k' is silent), after one of our authors.

5.2 Other Findings

During our study, we also found other combinatorial objects that align with a representation of the unique game play of Pyranim—a few of which we have cataloged below.

1. Type-A positive-root ideals, where the number of remaining chips is equal to size of the root-ideal.
2. Nonnesting partitions, where each move splits one maximal interval into several.

3. 132-avoiding permutations, where each move corresponds to an increase in the number of inversions.
4. Nondecreasing parking functions, where the game raises parking preferences as chips disappear.
5. The positive roots of $\mathfrak{sl}_{n+1}$, where chip removal corresponds to adjoining a root vector and closing under Borel action.

There are plenty more interesting combinatorial objects to study that rely on the bijective properties of our game to the Catalan numbers and UUIGs. One of these is the directed acyclic graphs seen directly in the formulation of Pyramid Nim in [8], and other correspondences between these objects are discussed in [9, 10]. Clearly, this game contains some incredible algebraic properties, some of which have yet to be explored.

References

[1] Phil Hanlon. "Counting interval graphs". In: *Transactions of the American Mathematical Society* 272.2 (1982), pp. 383–426. DOI: 10.1090/S0002-9947-1982-0662044-8.

[2] Xiaotie Deng, Pavol Hell, and Jing Huang. "Linear-time representation algorithms for proper circular-arc graphs and proper interval graphs". In: *SIAM Journal on Computing* 25 (1996). URL: https://epubs.siam.org/doi/epdf/10.1137/S0097539792269095.

[3] Kenneth P. Bogart and Douglas B. West. "A short proof that "proper = unit"". In: *Discrete Mathematics* 201.1–3 (1999), pp. 21–23. DOI: 10.1016/S0012-365X(98)00310-0.

[4] Frederic Gardi. "The Roberts characterization of proper and unit interval graphs". In: *Discrete Mathematics* (2007). URL: https://www.sciencedirect.com/science/article/pii/S0012365X07000696.

[5] Philippe Flajolet and Robert Sedgewick. *Analytic Combinatorics*. Cambridge University Press, 2009.

[6] Toshiki Saitoh et al. "Random generation and enumeration of proper interval graphs". In: *IEICE Transactions on Information and Systems* E93-D.7 (2010), pp. 1816–1823. DOI: 10.1587/transinf.E93.D.1816.

[7] Richard P. Stanley. *Catalan Numbers*. Cambridge University Press, 2015.

[8] Stephen J. Curran et al. "Pyramid Nim". In: *Integers* (2022). URL: https://math.colgate.edu/~integers/wg2/wg2.pdf.

[9] Ben Adenbaum and Sergi Elizalde. "Rowmotion on 321-avoiding permutations". In: *The Electronic Journal of Combinatorics* 30.3 (2023), P3.5.

[10] Syu Kato. *A geometric realization of the chromatic symmetric function of a unit interval graph*. 2024. DOI: 10.48550/arXiv.2410.12231. arXiv: 2410.12231 [math.CO]. URL: https://arxiv.org/abs/2410.12231.

[11] Juhani Karhumäki. *Combinatorics of Words*. Lecture notes, University of Turku. URL: https://www.math.utu.fi/en/home/karhumak/pdf/combwo.pdf.

[12] OEIS Foundation Inc. *A005217: Number of unlabeled unit interval graphs with n nodes*. The On-Line Encyclopedia of Integer Sequences. URL: https://oeis.org/A005217.