



POMAX GAMES – A FAMILY OF INTEGER-VALUED PARTIZAN
GAMES PLAYED ON POSETS

Erik Järleberg¹

*Department of Mathematics, Royal Institute of Technology
SE-100 44 Stockholm, Sweden
erikjar@kth.se*

Jonas Sjöstrand²

*Department of Mathematics, Royal Institute of Technology
SE-100 44 Stockholm, Sweden
jonass@kth.se*

Received: 2/11/14, Revised: 2/8/15, Accepted: 1/29/16, Published: 2/29/16

Abstract

We introduce the following class of partizan games, called *pomax games*. Given a partially ordered set whose elements are colored black or white, the players Black and White take turns removing any maximal element of their own color. If there is no such element, the player loses. We prove that pomax games are always integer-valued and for colored tree posets and chess-colored Young diagram posets we give a simple formula for the value of the game. However, for pomax games on general posets of height 3 we show that the problem of deciding the winner is PSPACE-complete and for posets of height 2 we prove NP-hardness. Pomax games are just a special case of a larger class of integer-valued games that we call *element-removal games*, and we pose some open questions regarding element-removal games that are not pomax games.

1. Introduction

A *pomax game* is played as follows. Given a finite poset P whose elements are colored black or white, the players Black and White take turns removing any maximal element of their own color. When a player cannot make a legal move, he loses the game. As an example, the pomax game in Figure 1 is a zero game (that is, a second player win): If Black starts he must remove z and White can counter by removing

¹This work was performed at KTH in Stockholm where the first author was writing his Master's thesis [7], laying the foundation for this paper.

²The second author was supported by a grant from the Swedish Research Council (621-2009-6090).

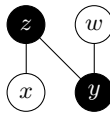


Figure 1: A pomax game of value zero.

x , leaving Black with no legal move. If White starts he must remove w , Black must remove z , White removes x and finally Black removes the last element y .

With the convention that Black is the left (positive) player and White is the right (negative) player, one may ask for the game value of a pomax game in general. As we will show in Section 3, pomax games are always integer-valued – a very rare property among combinatorial games.

Since the birth of modern combinatorial game theory in the 1970s, hundreds of two-player games with perfect information have been invented (or discovered) and analyzed. Many of them are impartial and thus have nimber values by the Sprague-Grundy Theorem. Among the properly partizan games, some are always numbers – Hackenbush restrained being the most prominent example [2] – but, to the best of our knowledge, essentially only one game studied in the literature is always integer-valued, namely Cutcake [1, pp. 24–27 and p. 51]. This game comes in two flavors, Cutcake and Maundy Cutcake, both of which have a very regular structure that admits a complete analysis.

Despite being integer-valued, pomax games have a sufficiently rich structure so that it is PSPACE-complete to decide the winner of the game, as we will see in Section 7. However, in some special cases the game is computationally tractable, and in Sections 4 and 5 we give simple formulas for the value of the pomax game played on colored tree posets and chess-colored Young diagram posets.

Many combinatorial games have been found to be PSPACE-complete, including common board games like Checkers, Hex and Reversi [4, 8, 6], but also more fundamental games like General Geography. Recently, Grier showed that poset games are PSPACE-complete in general [5].

A poset game is an impartial game played on a poset, where a legal move consists of removing any element along with all greater elements. Examples include the games Nim (where the poset is a disjoint sum of chains) and Chomp (where the poset is a product of chains). In a wide sense, pomax games are a partizan variant of poset games, but, being partizan, they have a quite different role to play in the abelian group of games.

For an exposé over computational complexity results for combinatorial games, we refer to [3].

Pomax games are just a special case of a larger class of games that we call

element-removal games, and when possible we will state our results in this more general setting.

The starting position of an element-removal game is a finite set X whose elements are colored black or white, and in each move the player (Black or White) removes an element of his own color. However, not all elements are removable at any stage, but the set of removable elements is a function of the set A of elements that are still present. Once an element becomes removable, it may never lose this status until it is removed. Formally, the removability function $\rho: 2^X \rightarrow 2^X$ has the property that

$$\rho(B) \cap A \subseteq \rho(A) \subseteq A$$

for any $A \subseteq B \subseteq X$.

Pomax games are the special case where ρ maps A to the maximal elements of the subposet induced by A .

The paper is organized as follows. In Section 3 we show that element-removal games, and thus pomax games, are always integer-valued. In Section 4 we study *balanced* games, a special kind of element-removal game that is easy to analyze, and in Section 5 we give a formula for the value of any pomax game on a colored tree poset.

After that, we switch our focus to the computational complexity of pomax games. In Section 7 we show that pomax games are PSPACE-complete even when restricted to height-three posets. As a warm-up, we show NP-hardness in Section 6, a result of more than pedagogical value since it holds already for posets of height two.

Finally, in Section 8 we suggest some further research and pose some open questions.

2. Prerequisites

Here, we will briefly recall those parts of combinatorial game theory that will be used in the forthcoming sections. No proofs will be given, but everything follows easily from the comprehensive discussion in the book “On Games and Numbers” by Conway [2].

We will adopt standard notation and terminology for partizan games. Black will always be the left player and White the right player, and we will use curly-bracket notation $G = \{G^L \mid G^R\}$, where G^L and G^R are typical left and right options of the game G . The game $\{\mid\}$ is called *the zero game* and is denoted by 0, and the game $\{0 \mid\}$ is called 1.

Recall that there is an equivalence relation on games, denoted by an ordinary equality sign “=”, such that $G = 0$ if and only if the second player wins G (under optimal play). If $G = H$ we will simply say that G is equal to H .

The (disjunctive) sum $G + H$ and the negation $-G$ is defined for games, and the

equivalence classes of games form an abelian group under these operations, with the equivalence class of 0 as the zero element.

There is also a partial order on (equivalence classes of) games, denoted by “ \geq ”, such that $G \geq 0$ if and only if White wins as a second player. The order relation is compatible with the group structure.

A game is *integer-valued* if it is equal to a game of the form $1 + 1 + \dots + 1$ or its negation, and the equivalence classes of integer-valued games form a totally ordered abelian subgroup of the group of all games.

We will use the following sufficient condition for a game to be integer-valued, which is a simple consequence of the Simplicity Theorem [2, Th. 11].

Lemma 1. *A game with a finite number of options is integer-valued if its options are integer-valued and the difference $G^R - G^L$ between any right and left options is at least 2. In that case, the value of the game is the integer closest to zero that is strictly larger than any left option and strictly smaller than any right option.*

For posets we will write $x < y$ to denote that x is *covered* by y , that is, $x < y$ and there is nothing in between.

3. Element-removal Games are Integer-valued

Clearly, the class of element-removal games (and the class of pomax games) is closed under summation and negation, and negation just means inversion of the coloring so that white elements become black and vice versa – it does not affect the removability function.

Our first result is a structure theorem telling us that element-removal games are very simple objects from an algebraic point of view.

Theorem 1. *Any element-removal game (and thus any pomax game) is integer-valued.*

Proof. It suffices to show that, for any element-removal game G and any left (Black) option G^L , we have $G - G^L \geq 1$. By symmetry, this will imply that $G - G^R \leq -1$ and thus that $G^R - G^L \geq 2$, and by Lemma 1, and Conway induction, G will be integer-valued. So, if White starts playing the game $G - G^L - 1$ we must show that Black has a winning strategy.

Let X denote the set of elements of G and let $x \in X$ be the element that Black removed from G to obtain G^L .

Case 1 White removes an element y from the G component. Since y is removable from X , it is still removable from $X \setminus \{x\}$, and thus Black may reply by removing y in the $-G^L$ component. The resulting position is $G^R - G^{RL} - 1$,

where G^R is the game obtained from G by removing the white element y and G^{RL} is obtained from G^R by removing the black element x (which is removable from $X \setminus \{y\}$ since it is removable from X). By Conway induction, this game is nonnegative.

Case 2 White removes an element y in the $-G^L$ component. Then Black replies simply by removing x from the G component, and the resulting position is of the form $G^L - G^{LL} - 1$, where G^{LL} is obtained from G^L by removing y . Again, this is nonnegative by Conway induction.

Case 3 White consumes his single move in the -1 component. Then Black replies by removing x from the G component, and the resulting position is $G^L - G^L = 0$.

□

4. Balanced Games

As we will see in Section 7, it is very hard to compute the value of a pomax game in general (unless PSPACE = P). In this section, however, we will look at a class of particularly well-behaved element-removal games to which we give the attribute *balanced*. It turns out that the value of such a game is given simply by the number of black elements minus the number of white elements.

Definition 1. An element-removal game is *balanced* if it has the following two properties.

- All options are balanced.
- If all removable elements are of the same color, then at least half of the total set of elements have that color.

For convenience, we say that a colored poset is balanced if its pomax game is.

Thus, a balanced colored poset cannot consist of millions of white elements covered by a few maximal black elements – there is always a maximal element of the majority color.

Proposition 1. *The value of a balanced game is the number of black elements minus the number of white elements, and the outcome of the game is independent of the players' strategies.*

Proof. Let G be a balanced game with b black elements and w white elements. Since all options of G are also balanced, by Conway induction, the value of any left option is $G^L = b - w - 1$ and the value of any right option is $G^R = b - w + 1$.

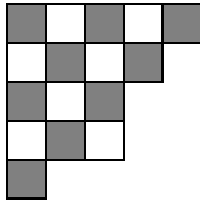


Figure 2: A chess-colored Young diagram with four outer corners.

If G has at least one left option and at least one right option it follows that $G = b - w$ by Lemma 1.

Suppose G has no right option. Then, since G is balanced, we have $b \geq w$ and thus $G = \{G^L \mid\} = \{b - w - 1 \mid\} = b - w$ by Lemma 1. The case where G has no left option is completely analogous.

Since the value of the game is a function of the number of black and white elements, the outcome does not depend on the strategies. \square

4.1. Balanced Pomax Games

If we are given a poset and want to color it in a way that will make it balanced, it seems natural to try a *chess coloring*, namely a coloring where no element covers an element of the same color. In this section, we show that this idea is successful at least for two kinds of posets: tree posets and Young diagram posets.

In a (non-empty) *tree poset* each element except one – the root – covers exactly one element. (For technical reasons, the empty poset is also considered to be a tree.)

Proposition 2. *The pomax game on a chess-colored tree poset is balanced.*

Proof. Suppose all maximal elements of the poset are white. Then, each black element can be paired with one of the white elements covering it. \square

A *Young diagram* (in English notation) is a finite collection of cells, arranged in left-justified rows, with the row lengths weakly decreasing. It can be interpreted as a poset by the rule that a cell covers the cell immediately to its left and the cell immediately above it (if those cells exist).³ The maximal cells are called *outer corners*. Figure 2 shows an example.

Proposition 3. *The pomax game on a chess-colored Young diagram is balanced.*

³Young diagram posets can be equivalently characterized as being the order ideals of a product of two finite chains.

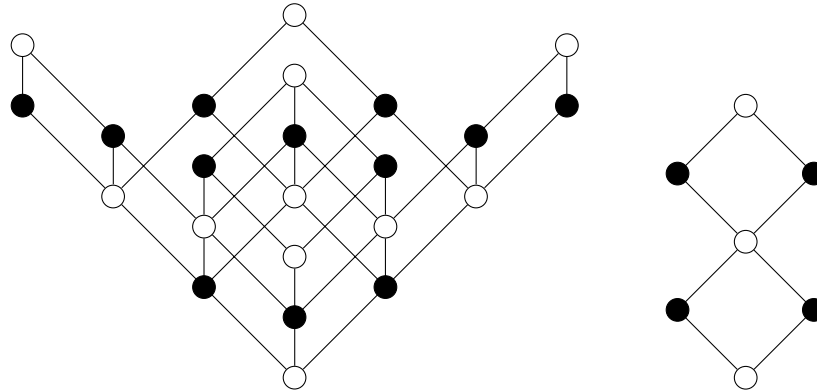


Figure 3: Chess-colored posets with more black than white elements but with all maximal elements white. Left: a plane partition poset, that is, an order ideal of a product of three chains. Right: a two-dimensional distributive lattice.

Proof. Suppose all outer corners of the Young diagram are white. Then, each row that ends with a black cell has a row of the same length immediately below it, and together these two rows have equally many black as white cells. A row that ends with a white cell has at least as many white cells as black ones. \square

Figure 3 shows that Propositions 2 and 3 cannot be extended to three-dimensional (plane partition) diagrams nor to two-dimensional distributive lattices. However, they can be extended to a larger class of colorings, namely those avoiding blocking triples.

Definition 2. A *blocking triple* in a colored poset is a triple of elements $x < y < z$ such that x and y are of the same color and z is of a different color.

Lemma 2. Let P be a colored poset without blocking triples and suppose that all maximal elements are white. Then, no black element is covered by a black element.

Proof. Let B be the set of black elements that are covered by a black element. Suppose B is not empty, and let x be an element that is maximal in B . Then x is covered by some black element y not in B which must be covered by some element z since no black element is maximal in P . Since y does not belong to B , the element z must be white, but this is impossible since $x < y < z$ form a blocking triple. \square

Proposition 4. Any colored tree poset without blocking triples is balanced.

Proof. Suppose all maximal elements are of the same color, say white. Then, by Lemma 2, each black element is covered by some white element. This pairing shows that there are at least as many white as black elements.

□

Proposition 5. *Any colored Young diagram without blocking triples is balanced.*

Proof. Suppose all outer corners are of the same color, say white. We want to show that at least half of the cells are white.

In light of Lemma 2 it is easy to see that any row in the Young diagram has at most one more black cell than white cells, and this happens only if the row both starts and ends with a black cell. Furthermore, a row both starting and ending with a white cell has an excess of white cells.

Any row both starting and ending with a black cell must have a row immediately below starting and ending with a white cell.

□

4.2. Other Balanced Element-removal Games

The pomax games considered above have several cousins which are element-removal games but not pomax games. Some of these variants can be shown to be balanced by the same argument as we used for pomax games.

First, we consider a variant called *min-max-removal games*. It is an element-removal game played on a poset, but we let not only the maximal elements but also the minimal elements be removable.

Starting with a colored tree poset, playing the min-max-removal game will soon result in a poset consisting of several disjoint trees, so we ought to formulate our results for such *forest posets*. The blocking triples turn out to be the right tool in this situation too.

Proposition 6. *The min-max-removal game on any colored forest poset without blocking triples is balanced.*

Proof. Identical to the proof of Proposition 4.

□

Starting with a Young diagram poset and playing the min-max-removal game will soon result in a *skew Young diagram poset*, that is, a Young diagram with a smaller Young diagram deleted from its upper-left corner.

Proposition 7. *The min-max removal game on any colored skew Young diagram poset without blocking triples is balanced.*

Proof. Identical to the proof of Proposition 5.

□

Now, let us throw the whole poset overboard for a while and consider a couple of element-removal games with a different ground structure.

Given a tree (in the graph-theoretical sense) whose vertices are colored black or white, the *leaf-removal game* is an element-removal game on the vertices of the

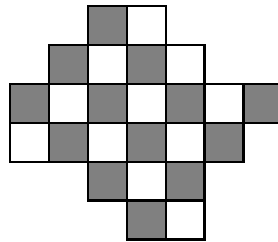


Figure 4: A chess-colored truncated square diagram with 11 corners.

tree, where the leaves are the removable elements. By a *chess coloring* we mean a black-white vertex coloring where adjacent vertices have different colors.

Proposition 8. *The leaf-removal game on any chess-colored tree is balanced.*

Proof. We can think of our tree as a chess-colored tree poset by choosing any root vertex (unique minimal element) and letting all edges (covering relations) be directed from the root. Then, the proof of Proposition 2 applies. \square

Finally, let us consider the *corner-removal game*, which is an element-removal game where the ground set is an $n \times n$ array of colored cells and where a cell is removable if it is a *corner*, that is, if it has at most one neighboring cell in the same row and at most one neighboring cell in the same column. We introduce the term *truncated square diagrams* for the cell diagrams obtained by iteratively removing corners from an $n \times n$ cell array. Figure 4 shows an example.

Cells are *neighbors* if they have a common side, and, as always, by a *chess coloring* we mean a black-white coloring where neighbors have different colors.

Proposition 9. *The corner-removal game on any chess-colored truncated square diagram is balanced.*

Proof. Identical to the proof of Proposition 5, except that there is no need for Lemma 2. \square

5. Tree Posets

In Section 4 we saw that it is easy to compute the value of the pomax game on a colored tree poset without blocking triples: Just take the number of black elements minus the number of white elements. In this section we give a complete analysis of pomax games on tree posets.

Let us begin with a simple example, namely the colored tree poset in Figure 5, which is just a chain. The pomax game on that poset is clearly a zero game: If

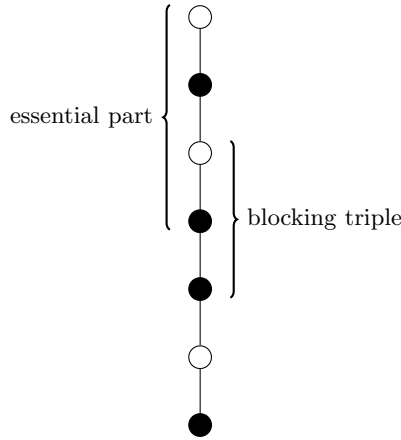


Figure 5: A chain poset with a blocking triple.

Black starts he loses immediately, and if White starts he will lose when the four topmost elements are removed. Note that the two elements at the bottom do not affect the value of the game at all. They are “blocked” by the blocking triple above.

Our example suggests the following definition.

Definition 3. For any colored tree poset P , its *essential part*, denoted by $\text{ess } P$, is the (unique) maximal upper set that does not contain any blocking triple.

We will refer to the elements of the essential part as *essential* elements.

From now on, we will let $\text{Po}(P)$ denote the pomax game on the colored poset P .

As the following theorem shows, all non-essential elements might be thrown away without affecting the value of the game, and since the essential part is balanced its game value is easy to compute.

Theorem 2. For any colored tree poset P , the game equality $\text{Po}(P) = \text{Po}(\text{ess } P)$ holds.

For the proof of Theorem 2 we need the following lemma.

Lemma 3. Let P be a black-rooted colored tree poset with at least one white element but no blocking triple. Let m be the (integer) game value of $\text{Po}(P)$. Then, in the game $\text{Po}(P) - m$, if Black starts White can win before Black gets an opportunity to remove the root of P .

Proof. By Propositions 4 and 1, White will win $\text{Po}(P) - m$ when Black starts, no matter what strategies they use. If White removes all white elements in the $-m$ component (if m is positive) before making any move in the $\text{Po}(P)$ component, she

will never have to remove all white elements in the $\text{Po}(P)$ component, and thus the root will never be removable for Black. \square

Proof of Theorem 2. We assume that $\text{ess } P \neq P$; otherwise there is nothing to prove.

The essential part consists of a disjoint union of trees $\text{ess } P = T_1 \cup T_2 \cup \dots \cup T_k$ and the non-essential part $P \setminus \text{ess } P$ is a tree. For $i = 1, \dots, k$, let m_i be the value of T_i (which is just the number of black minus the number of white elements since T_i does not contain any blocking triple). We want to show that the game $\text{Po}(P) - m_1 - m_2 - \dots - m_k$ is a win for the second player. By symmetry, it suffices to show that White will win if Black starts.

Note that, by construction of the essential part and by our assumption that $\text{ess } P \neq P$, none of the trees T_1, \dots, T_k is unicolored. Thus, by Lemma 3, if Black starts White can win $\text{Po}(\text{ess } P) - m_1 - m_2 - \dots - m_k$ without ever giving Black an opportunity to remove a minimal element of $\text{ess } P$. By adopting this strategy to the game $\text{Po}(P) - m_1 - m_2 - \dots - m_k$, White can win without removing any non-essential element. Black will not get the chance to remove any non-essential element, because each black maximal element x of $P \setminus \text{ess } P$ is covered by some black minimal element of $\text{ess } P$ – otherwise x would have been essential. \square

6. Pomax Games of Height 2 are NP-hard

Up to this point all of our results have been about the *simplicity* of pomax games: They are integer-valued and their values are easy to compute in some cases, in particular if the poset is a tree. In this and the forthcoming section, however, we will show that in general it is very hard to find the winner of a pomax game, even for very shallow posets. (All this is under the assumption that $\text{PSPACE} \neq \text{P}$.)

By the *height* of a poset we mean the length of its longest chain.

Theorem 3. *The problem of deciding whether a given pomax game equals zero is NP-hard even if the height of the colored poset is restricted to two.*

Proof. Recall that a Boolean formula is in *conjunctive normal form (CNF)* if it is a conjunction of clauses, where each clause is a disjunction of literals, each literal being a variable or the negation of a variable. If every clause has exactly three literals, it is a *3CNF-formula*. An example is $(x_1 \vee \neg x_2 \vee \neg x_4) \wedge (x_2 \vee \neg x_3 \vee x_4)$.

We will make a reduction from the canonical NP-complete problem 3-SAT.

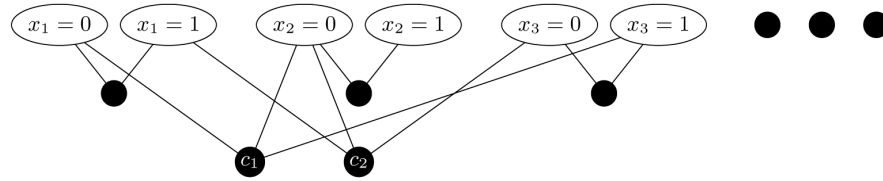


Figure 6: The colored poset constructed from the 3CNF-formula $(x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3)$.

3-Satisfiability (3-SAT)

Input A 3CNF-formula.

Output “Yes” if and only if the formula is true for some assignments of the variables.

Given a 3CNF-formula we will construct a colored poset (in polynomial time) whose pomax game is zero precisely if the formula is true.

For each variable x_i in the formula we put two white *assignment elements* in the poset, one called “ $x_i = 0$ ” and one called “ $x_i = 1$ ” (where 0 and 1 should be interpreted as “false” and “true”, respectively). Also, for each clause C_j in the formula we put a black *clause element* c_j in the poset and we let it be covered by exactly those assignment elements that would make the clause false. For instance, the clause element corresponding to $x_1 \vee x_2 \vee \neg x_3$ would be covered by the assignment elements “ $x_1 = 0$ ”, “ $x_2 = 0$ ” and “ $x_3 = 1$ ”.

We want the removal of an assignment element “ $x_i = \alpha$ ” during play to correspond to actually assigning the value α to the variable x_i , so we need some mechanism to prevent White from cheating by removing both “ $x_i = 0$ ” and “ $x_i = 1$ ”. This is accomplished by letting “ $x_i = 0$ ” and “ $x_i = 1$ ” cover a black *candy element* so that White cannot cheat without uncovering candy for his opponent.

Finally, we put as many black isolated elements in the poset as there are Boolean variables, so that Black has something to eat while White is trying to satisfy the formula. Figure 6 shows an example of our construction.

If White starts she cannot win, because Black has an isolated element for each pair of white assignment elements, and if White cheats Black gets candy.

If Black starts, White will win unless some of the black clause elements are uncovered during the game. Clearly, White can avoid uncovering a clause element precisely if the 3CNF-formula is satisfiable. □

7. Pomax Games of Height 3 are PSPACE-complete

Since the number of moves during a pomax game is bounded by the size of the poset, its outcome can be determined by an algorithm using only a polynomial amount of space. In this section we show that pomax games are in fact PSPACE-complete.

Theorem 4. *The problem of deciding whether a given pomax game equals zero is PSPACE-complete even if the height of the colored poset is restricted to three.*

Proof. We will make a reduction from the following archetypical PSPACE-complete problem.

Quantified boolean formula problem (QBF)

Input A QBF-formula, that is, a formula of the type

$$\forall x_1 \exists x_2 \forall x_3 \exists x_4 \cdots \forall x_{n-1} \exists x_n \phi(x_1, \dots, x_n),$$

where ϕ is a CNF-formula $C_1 \wedge C_2 \wedge \cdots \wedge C_m$. The number n of variables is even.

Output “Yes” if and only if the QBF-formula is true.

We will think of QBF as the problem of deciding the winner of a two-player game where the players, let us call them Black and White, assign truth values to the variables x_i . Black assigns variables with odd indices and White assigns variables with even indices. Furthermore, Black must assign x_1 first and then White, with knowledge of the value of x_1 , must assign x_2 , and so on. When all n variables have been assigned, White wins if the CNF-formula ϕ becomes true.

Given a QBF-formula as above we will construct a colored poset (in polynomial time) whose pomax game is zero precisely if the formula is true. Let us build this poset step by step, initially focusing on the main picture and taking care of the details as we go along.

As in the proof of Theorem 3, for each variable x_i in the formula we put two *assignment elements* in the poset, one called “ $x_i = 0$ ” and one called “ $x_i = 1$ ”. But now we color the elements black if i is odd and white if i is even.

Again following the proof of Theorem 3, for each clause C_j in the formula we put a black *clause element* c_j in the poset and we let it be covered by exactly those assignment elements that would make the clause false.

As before, we need some mechanism to prevent players from cheating by removing both “ $x_i = 0$ ” and “ $x_i = 1$ ”, and this is accomplished by letting “ $x_i = 0$ ” and “ $x_i = 1$ ” cover some *candy elements* of the opposite color so that a player cannot

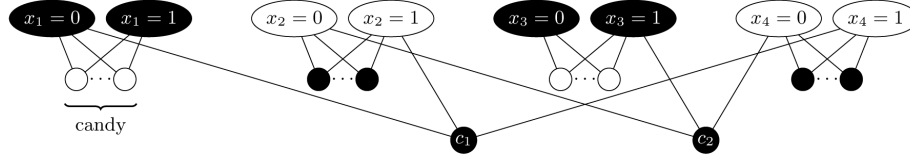


Figure 7: The assignment, clause and candy elements of the colored poset constructed from the QBF instance $\forall x_1 \exists x_2 \forall x_3 \exists x_4 (x_1 \vee \neg x_2 \vee \neg x_4) \wedge (x_2 \vee \neg x_3 \vee x_4)$.

cheat without uncovering lots of candy for his opponent. From now on we assume that there is enough candy to make sure that no player will ever cheat. (Obviously, if cheating uncovers more candy elements than the total number of non-candy elements in the poset, there will be no cheating. A more careful analysis shows that it suffices to have $m + 1$ white candy elements for each variable with odd index and one single black candy element for each variable with even index.) Figure 7 shows an example of a colored poset as constructed so far.

The idea is that Black would start the game and assign a value to x_1 by choosing to remove either “ $x_1 = 0$ ” or “ $x_1 = 1$ ”. Then, White would remove either “ $x_2 = 0$ ” or “ $x_2 = 1$ ”, and Black would remove either “ $x_3 = 0$ ” or “ $x_3 = 1$ ”, and so on. Finally, White would remove either “ $x_n = 0$ ” or “ $x_n = 1$ ” and she will win the game if no clause element c_j has been uncovered, which is the case exactly if the CNF-formula ϕ is true. However, nothing in the present construction will force the players to make the assignments in the correct order from left to right.

For each $i \in \{1, \dots, n - 1\}$, to make sure that the player making the assignment of the variable x_{i+1} will not have to do that before the other player has assigned the previous variable x_i , we install a gadget consisting of six new elements called $a_i^0, a_i^1, b_i^{00}, b_i^{01}, b_i^{10}, b_i^{11}$, and the covering relations “ $x_{i+1} = \beta$ ” $\succ a_i^\beta \succ b_i^{\alpha\beta}$ and “ $x_i = \alpha$ ” $\succ b_i^{\alpha\beta}$ for $\alpha, \beta \in \{0, 1\}$. We color a_i^β black if i is odd and white if i is even, and $b_i^{\alpha\beta}$ white if i is odd and black if i is even.

This completes our construction, and the result is exemplified in Figure 8.

Note that, since no player cheats and uncovers candy for the opponent, for each i , any time during play at most one of the elements a_i^0 and a_i^1 is maximal and at most one of the elements $b_i^{00}, b_i^{01}, b_i^{10}$ and b_i^{11} is maximal.

A player, let us say White, does not gain anything from removing “ $x_{i+1} = \beta$ ” while the previous pair of assignment elements “ $x_i = 0$ ” and “ $x_i = 1$ ” are both still present, because the other player, Black, could answer immediately by removing the element a_i^β without uncovering any white element. Not until later when Black removes “ $x_i = \alpha$ ” for some $\alpha \in \{0, 1\}$, White is compensated by the uncovering of the white element $b_i^{\alpha\beta}$, so White could as well have waited for this to happen before she removed “ $x_{i+1} = \beta$ ”.

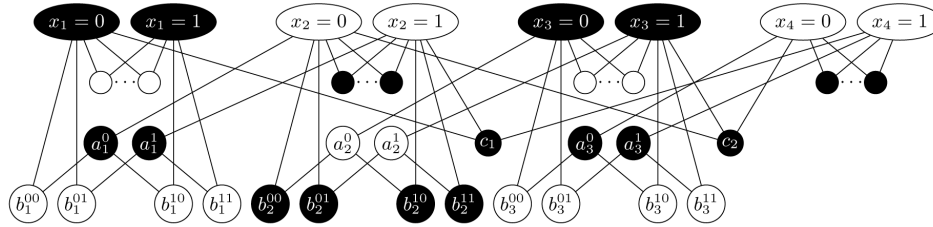


Figure 8: The colored poset constructed from the QBF instance $\forall x_1 \exists x_2 \forall x_3 \exists x_4 (x_1 \vee \neg x_2 \vee \neg x_4) \wedge (x_2 \vee \neg x_3 \vee x_4)$.

We conclude that, if Black starts the game, White will win, and hence the game is ≤ 0 , if and only if the QBF-formula is true. If White starts the game, Black will win by simply removing a_i^β whenever White removes “ $x_{i+1} = \beta$ ”, so the game is always ≥ 0 . \square

8. Future Research and Open Questions

Theorems 3 and 4 leave us with an obvious open question.

Open Problem 1. Is it a PSPACE-complete problem to compute the outcome of a given pomax game even if the height of the colored poset is restricted to two?

Colored posets like the one in Figure 7 seem very hard to analyze, and though the players may cheat by assigning the variables in the wrong order, we would guess that games of this type are PSPACE-complete. There is also a theorem by Schaefer [9, Th. 3.8] that points in this direction.

The posets constructed in the proofs of Theorems 3 and 4 have small height but they might be quite high-dimensional. One could ask if it is possible to trade low height for low dimensionality while still maintaining the hardness of the problem.

Open Problem 2. How computationally hard is the problem of computing the outcome of a pomax game on a colored Young diagram poset?

In Section 4.2 we defined some particular element-removal games that are not pomax games, and we saw that they behave well if their underlying structure (poset, tree graph or cell diagram) is chess-colored. In particular min-max-removal games on forest posets and leaf-removal games might be possible to analyze for any coloring by essentially the same method we used for pomax games on tree posets in Section 5.

Open Problem 3. Find a formula for the value of the min-max-removal game on any colored forest poset.

Open Problem 4. Find a formula for the value of the leaf-removal game on any colored tree.

As mentioned in the introduction, pomax games are a partizan variant of poset games. But there is a more straightforward way to make a poset game partizan and that is simply to color the elements and let the active player choose any element of his own color and remove it along with all greater elements (even if some of those happen to be of the opposite color). The games so obtained – let us call them *partizan poset games* – seem to be related to Hackenbush restrained. For instance, it is easy to see that they equal numbers (by essentially the same argument as for Hackenbush restrained, see [2, p. 87]), and every restrained Hackenbush tree is obviously equivalent to a partizan poset game on a colored tree poset. This latter observation shows that partizan poset games are not integers but can take the value of any dyadic rational number. However, the similarity with restrained Hackenbush apparently disappears for more complex posets (or more complex Hackenbush graphs).

We think that a more thorough study of partizan poset games would be worthwhile.

References

- [1] E. R. Berlekamp, J. H. Conway, and R. K. Guy, *Winning Ways for Your Mathematical Plays*, vol. 1, A. K. Peters Ltd., Wellesley, Massachusetts, 2001.
- [2] J. H. Conway, *On Numbers and Games*, Academic press, Cambridge, England, 1976.
- [3] E. D. Demaine and R. A. Hearn. Playing games with algorithms: Algorithmic combinatorial game theory, *Games of No Chance 3* (M. H. Albert and R. J. Nowakowski, eds.), MSRI Publications, vol. 56, Cambridge University Press, 2009, 3–56.
- [4] A. S. Fraenkel. The complexity of Checkers on an $n \times n$ board – preliminary report, *Proceedings of the 19th Annual Symposium on Computer Science*, 1978, 55–64.
- [5] D. Grier. Deciding the winner of an arbitrary finite poset game is PSPACE-complete, *ICALP (1)* (F. V. Fomin, R. Freivalds, M. Z. Kwiatkowska, and D. Peleg, eds.), Lecture Notes in Computer Science, vol. 7965, Springer, 2013, 497–503.
- [6] S. Iwata and T. Kasai, The Othello game on an $n \times n$ board is PSPACE-complete, *Theor. Comp. Sci.* **123** (1994), 329–340.
- [7] E. Järleberg. Pomax games: A new class of partizan games played on posets, Master’s thesis, KTH, 2013, TRITA-MAT-E; 2013:46.
- [8] Stefan Reisch. Hex ist PSPACE-vollständig (Hex is PSPACE-complete), *Acta Inf.* **15** (1981), 167–191.
- [9] T. J. Schaefer. On the complexity of some two-person perfect-information games, *J. Comp. Sys. Sci.* **16** (1978), 185–225.