

# COUNTEREXAMPLES TO CONJECTURES ABOUT SUBSET TAKEAWAY AND COUNTING LINEAR EXTENSIONS OF A BOOLEAN LATTICE

ANDRIES E. BROUWER AND J. DANIEL CHRISTENSEN

ABSTRACT. We study the game known as Subset Takeaway (Chomp on a hypercube), and give unexpected answers to questions of Gale and Neyman. We show that the number of linear extensions of the lattice of a 7-cube is 630470261306055898099742878692134361829979979674711225065761605059-425237453564989302659882866111738567871048772795838071474370002961694720 (roughly  $6.3 \cdot 10^{137}$ ).

## 1. INTRODUCTION

Chomp is a game played on a partially ordered set  $P$  with smallest element 0. A move consists of choosing a nonzero element  $x$  of  $P$  and removing  $x$  and all larger elements from  $P$ . Whoever is unable to move loses. This game has been studied for several families of partially ordered sets. For some history and discussion, see the web page [2]. The name Chomp is due to Martin Gardner [6].

If  $P$  has a largest element 1, different from 0, then a trivial strategy-stealing argument shows that the first player wins. (If choosing 1 does not win, it is because the opponent has the devastating reply  $a$ . But in that case the first player wins by starting with  $a$ .) However, this argument is non-constructive: the winning move is unknown.

Subset Takeaway is the special case of Chomp, where  $P$  is chosen to be the Boolean lattice  $B_n$  of all subsets of an  $n$ -set. It was studied by Gale and Neyman [5]. Since there is a largest element, the first player wins. But how? Gale and Neyman asked whether taking the top element, the unique  $n$ -set, is the winning move, and this is true for  $n \leq 5$  ([5]) and for  $n = 6$  ([3]).

More generally, let  $P = P_{n,k}$  be the partially ordered set of all subsets of size at most  $k$  from an  $n$ -set. Gale and Neyman asked whether the first player loses in  $P_{n,k}$  if and only if  $(k+1) \mid n$ , and prove this for  $k \leq 2$ .

However, these conjectures are false. Indeed, using computer search, one finds fairly easily that the first player loses in  $P_{7,3}$  and with more effort that the first player wins in  $P_{7,6}$  (by choosing any 4-set). In particular, the top move in  $P_{7,7}$  is not a winning move, but a 4-set is.

In Section 2, we extend the above results by giving computations of Grundy values of  $P_{n,k}$  for some values of  $n$  and  $k$ , showing that a natural pattern fails to continue. We also list some positions with large Grundy values, and describe how a reduction technique using involutions can extend our results to some larger positions. In Section 3, we apply our methods to the problem of counting the number of linear extensions of the poset  $B_n$ , reproducing known results for  $n \leq 6$  and giving a new result for  $n = 7$ . Finally, in Section 4, we discuss the size of the search space, describe the techniques we used to efficiently search it, and give some sample running times and space requirements that could be useful for future work.

We thank Alejandro Morales for bringing our attention to the problem of counting linear extensions.

## 2. SUBSET TAKEAWAY

**2.1. Win/loss computation.** We computed that the only winning move in  $P_{7,7}$  is to choose a set of size 4. As we will explain below, the search tree is much too large to be analyzed by hand, but we can give the first two layers. A move of size 1 leads to  $P_{6,6}$ , in which the winning move is known to be the set of size 6. Symmetrically, the response to a set of size 6 is the complement. Similarly, the winning response to a move of size 2 or 5 is the complement; this follows from the symmetry exchanging the elements of the move of size 2, using the technique described in Subsection 2.4. However, the winning response to a move of size 3 is not the complement. For example, our programs show that a winning response to  $\{0, 1, 2\}$  is  $\{0, 1, 3, 4, 5\}$ . The remaining case is a move of size 4, which our programs show is a winning move.

*Date:* February 9, 2017.

*2010 Mathematics Subject Classification.* 91A46, 06A07.

*Key words and phrases.* Combinatorial game, impartial game, Grundy number, partially ordered set, total order.

**2.2. Grundy values.** Any impartial game, such as Subset Takeaway, is equivalent in the sense of combinatorial game theory to the game of Nim (cf. [4]). Thus to each game position  $P$  one can associate a nonnegative integer  $g$  such that  $P$  is equivalent to a Nim heap of size  $g$ . This  $g$  is called the *Grundy value* of  $P$ , and is the smallest nonnegative integer that is not the Grundy value of one of the positions that can be reached by one move. The position is lost for the first player if and only if the Grundy value is 0. We give the Grundy values of  $P_{n,k}$  for  $0 \leq k \leq n$  in small cases.

$n \setminus k$	0	1	2	3	4	5	6	7	8
0	0								
1	0	1							
2	0	0	2						
3	0	1	0	3					
4	0	0	1	0	<b>1</b>				
5	0	1	2	1	0	<b>2</b>			
6	0	0	0	2	<b>2</b>	0	<b>3</b>		
7	0	1	1	<b>0</b>	<b>3</b>	1	<b>5</b>	<b>6</b>	
8	0	0	2						

In [7], the Grundy values of complete multipartite graphs are determined. In particular, they show that the Grundy value of  $P_{n,2}$  is  $n$  modulo 3, agreeing with the  $k = 2$  column of the table. Note that, without the results in bold, one might conjecture that the Grundy value of  $P_{n,k}$  is  $n$  modulo  $k + 1$ . We believe that most of the results for  $k > 2$  are new, except for the zeros when  $k = n - 1$  and  $n \leq 6$ . Other values are unknown.

We have seen already that  $P_{n,n}$  is a first player win for all  $n > 0$ . We shall see below that  $P_{8,3}$ ,  $P_{8,5}$ , and  $P_{9,3}$  are first player wins (and hence have a positive Grundy value).

**2.3. Large Grundy values.** Some positions in these games have rather large Grundy values  $g$ . In the table below we give  $g$  and the maximal sets of a simplicial complex with this Grundy value, for various positions with  $n = 7$ . We write  $ab$  instead of  $\{a, b\}$ , etc.

$g$	antichain
9	01, 02, 03, 04, 05, 06, 12, 13, 14, 15, 23, 24, 35, 46
23	012, 013, 014, 015, 023, 024, 025, 026, 034, 045, 056, 123, 126, 135, 136, 145, 236, 245, 346, 456
37	0124, 0134, 0234, 1234, 0125, 0235, 1235, 0145, 0245, 1245, 0345, 2345, 0126, 0136, 0236, 1236, 0146, 0246, 1246, 0156, 1356
44	012345, 01236, 01246, 01346, 2346, 01256, 1356, 2356, 0456

**2.4. Involutions.** Suppose  $P$  is a poset and  $\phi : P \rightarrow P$  an automorphism of order 2 such that  $P_0 = \{x \in P \mid \phi(x) = x\}$  is downwards closed. Then the games on  $P$  and  $P_0$  have the same outcome. (Indeed, if  $P_0$  is a second player win, then the second player can answer each move  $x$  outside  $P_0$  with  $\phi(x)$ , and each move inside  $P_0$  with the winning strategy for  $P_0$ . If  $P_0$  is a 1st player win, with winning move  $x_0$ , then the 1st player can play  $x_0$  and then apply the previous sentence.) More generally,  $P$  and  $P_0$  have the same Grundy value (cf. [7]).

It is easy to see that  $P_{8,3}$  is a first player win: just pick a singleton, reducing to  $P_{7,3}$  which is a first player loss. This involution argument allows one to see that  $P_{8,5}$  is also a first player win: pick a 2-set, say  $\{6, 7\}$ , and apply the transposition (67) to reduce to  $P_{6,5}$  which is a first player loss. More generally, we see that if  $P_{n,k}$  is a first player loss, then  $P_{n+1,k}$  is a first player win if  $k > 0$ , and  $P_{n+2,k}$  is a first player win if  $k > 1$ . So,  $P_{9,3}$  is also a first player win.

### 3. COUNTING LINEAR EXTENSIONS OF A BOOLEAN LATTICE

The machinery we developed for studying Subset Takeaway on an  $n$ -set can be used more generally to recursively compute a function whose value on a poset is defined in terms of its values on smaller posets. Another example of such a function is  $e(P)$ , the number of linear extensions of the poset  $P$ . A *linear extension* of a poset  $P$  is a total order on  $P$  which is compatible with the given partial order. We can

compute the number of linear extensions of  $P$  in the following way. The maximum element of a total order must be one of the maximal elements of  $P$ , so  $e(P)$  is given by

$$e(P) = \begin{cases} 1, & \text{if } P = \emptyset, \\ \sum_m e(P \setminus \{m\}), & \text{otherwise,} \end{cases}$$

where the sum is over the maximal elements  $m$  of  $P$ . The values of  $e(P)$  when  $P$  is the Boolean lattice  $B_n$  of subsets of an  $n$ -set were known for  $n \leq 6$  (see OEIS A046873 [10]). We also give the value for  $n = 7$ .

$n$	$e(B_n)$
0	1
1	1
2	2
3	48
4	1680384
5	14807804035657359360
6	141377911697227887117195970316200795630205476957716480
7	630470261306055898099742878692134361829979979674711225065761605059425- 237453564989302659882866111738567871048772795838071474370002961694720

The value 70016 of  $e(B_4)/4!$  was given by Weinrich [11]. The value of  $e(B_5)$  was given by Morton [8], and  $e(B_6)$  was found by Wienand [12]. Wienand reports that the computation of  $e(B_6)$  took less than 16 hours on a compute server. For us that same computation took less than 1 second, while the computation of  $e(B_7)$  took less than 2.5 hours. Asymptotic bounds on  $e(B_n)$  were given in [1, 9].

#### 4. COMPUTATION

**4.1. Counting antichains.** How much work is it to analyze  $P_{n,n}$ ? The full game tree consists of all downwards closed families of sets on  $n$  points, or, equivalently (by taking maximal elements), of all antichains on  $n$  points. The number of antichains on  $n$  points is known as the  $n$ -th Dedekind number  $M(n)$ , cf. OEIS A000372 [10]. A rough lower bound is found by taking arbitrary families of sets of size  $\lfloor n/2 \rfloor$ , and we see that  $\log_2 M(n) \geq \binom{n}{\lfloor n/2 \rfloor}$ . For  $0 \leq n \leq 8$  the values of  $M(n)$  are 2, 3, 6, 20, 168, 7581, 7828354, 2414682040998, 56130437228687557907788. (Here both the empty antichain and the antichain consisting of just the empty set are counted.)

The amount of work can be reduced by using the action of the symmetric group  $\text{Sym}(n)$  of order  $n!$ . This saves at most a factor of  $n!$ , and for  $n = 6, 7$ , and 8 we have that  $M(n)/n!$  is approximately 10873, 479103580, and 1392123939203560464. The number of antichains on  $n$  points up to isomorphism, or, equivalently, of unlabeled simplicial complexes with at most  $n$  vertices, is given in OEIS A003182 [10]. For  $0 \leq n \leq 7$  the values are 2, 3, 5, 10, 30, 210, 16353, and 490013148. We see that our approximation  $M(n)/n!$  gets better as  $n$  increases, and so we expect something like  $1.4 \cdot 10^{18}$  antichains up to isomorphism for  $n = 8$ .

We see that in order to answer questions about  $P_{n,n}$  for  $n = 7$  it suffices to look at at most 490013148 posets, which is not too much for modern computers, while a brute force approach for  $n = 8$  requires looking at more than  $10^{18}$  posets, which is infeasible today.

**4.2. Methods.** Both authors independently wrote and ran computer programs to verify the claims, and compared the programs on many test cases and against known positions.

As discussed above, one wants to use the action of  $\text{Sym}(n)$ . This is done by bringing each downwards closed set into a canonical form and saving already computed values in a hash table. Given a downwards closed set  $A$  on the set  $N = \{0, \dots, n-1\}$ , we consider the invariant that assigns to each  $i \in N$  the vector whose  $j$ th entry ( $j = 1, \dots, n$ ) is the number of  $j$ -sets in  $A$  containing  $i$ . We then apply a permutation to  $N$  that brings these invariants into some lexmin order. Finally, we consider all permutations in  $\text{Sym}(n)$  that preserve the invariants and apply the one that puts  $A$  in a lexmin form. Putting moves into canonical form is crucial for time and memory efficiency, and was also the bottleneck in our programs.

**4.3. Sample running times and memory usage.** The programs were written in C and were run on ordinary desktop hardware using a single core. Here we give sample computation times and posets stored. It would certainly be possible to improve these with more work, but they give an illustration of the resources we needed, which may help someone who wishes to reproduce or extend our work.

poset	computation	time (minutes)	posets stored
$P_{7,3}$	win/loss	5	9680539
$P_{7,7}$	win/loss	230	401448670
$P_{7,3}$	Grundy	22	15466911
$P_{7,7}$	Grundy	637	490013147
$P_{7,3}$	linear extensions	5	15466911
$P_{7,7}$	linear extensions	148	490013147

For all three types of computation,  $P_{6,6}$  took less than 1 second and stored at most 16352 posets in the hash table. All of the computations required less than 32 GB of virtual memory, except for the last one, counting the number of linear extensions of  $B_7$ . It required 38 GB, but it would not be hard to reduce this by ordering the search by the number of elements of the posets.

#### REFERENCES

- [1] G. R. Brightwell and P. Tetali, *The number of linear extensions of the Boolean lattice*, Order **20** (2003) 333–345.
- [2] A. E. Brouwer, *Chomp*, <http://www.win.tue.nl/~aeb/games/chomp.html>.
- [3] J. Daniel Christensen and Mark Tilford, *David Gale’s subset take-away game*, Amer. Math. Monthly **104** (1997) 762–766.
- [4] John Horton Conway, *On numbers and games*, Academic Press, London etc., 1976.
- [5] D. Gale and A. Neyman, *Nim-type games*, Internat. J. Game Theory **11** (1982) 17–20.
- [6] Martin Gardner, *Mathematical Games*, Scientific American, Jan. 1973, pp. 110–111.
- [7] Tirasan Khandhawit and Lynnelle Ye, *Chomp on Graphs and Subsets*, preprint, arXiv 1101.2718, Jan 2011.
- [8] Jason Ryder Morton, *Geometry of conditional independence*, Ph.D. thesis, University of California, Berkeley, Fall 2007.
- [9] Jichang Sha and D. J. Kleitman, *The number of linear extensions of subset ordering*, Discr. Math. **63** (1987) 271–278.
- [10] N. J. A. Sloane, *The On-Line Encyclopedia of Integer Sequences*, [A000372](#), [A003182](#) and [A046873](#).
- [11] D. M. Weinreich, *The rank ordering of genotypic fitness values predicts genetic constraint on natural selection on landscapes lacking sign epistasis*, Genetics **171** (2005) 1397–1405.
- [12] Oliver Wienand, *Algorithms for Symbolic Computation and their Applications*, Ph.D. thesis, Univ. Kaiserslautern, 2011.

*E-mail address:* [aeb@cw.nl](mailto:aeb@cw.nl)

DEPARTMENT OF MATHEMATICS, UNIVERSITY OF WESTERN ONTARIO, LONDON, ONTARIO, CANADA

*E-mail address:* [jdc@uwo.ca](mailto:jdc@uwo.ca)