2012

Exploring the On-line Partitioning of Posets Problem

Leah F. Rosenbaum
Scripps College

Exploring the On-line Partitioning of Posets Problem

by
Leah Rosenbaum

Submitted to Scripps College in partial fulfillment of the degree of bachelor of arts

Professor Shahriari
Professor Chaderjian

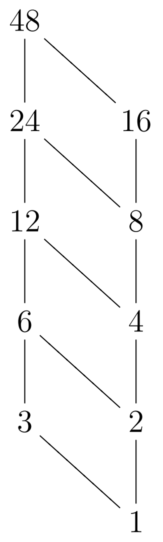
March 9, 2012

Contents

Chapter 1

Introduction

1.1 Terminology



The above graph represents a *partially ordered set*, or *poset*. Here the set is the divisors of 48, and the partial ordering is division. The edges connect elements that are comparable under the partial ordering. The transitivity and reflexivity of this comparability are assumed, so 48 is comparable to 2 even though they are not directly connected.

Definition: A *chain* is a subset of the poset in which every element is comparable to one another, that is, every element can be reached by traveling the edges up. The subset $\{1, 4, 8, 24\}$ is a chain.

Definition: An *anti-chain* is a subset of the poset in which every element is incomparable to one another, that is, cannot be reached by traveling the edges up. The subset $\{3, 4\}$ is an anti-chain. Note that $\{3, 4, 8\}$ is not an anti-chain because, while 3 is incomparable to both 4 and 8, 4 and 8 are comparable to one another.

Definition: The *width* of a poset is the number of elements in the largest anti-chain. The example poset above has width 2. Much of this thesis focuses on posets of width 2.

Definition: A poset is said to be $(t + t)$*-free* if it does not contain two chains of t points in which all pairs of points, one from each chain, are incomparable. The example from above contains a $(2+2)$ (the chains $3, 6$ and $4, 8$) but does not contain a $(3+3)$, so it is $(3+3)$-free.

The example poset is also *graded*, meaning that the journey from the bottom element, here 1, to the top element, here 48, always contains the same number of points no matter the specific path traveled.

1.2 On-line Partitioning

One problem related to posets is that of partitioning. To *partition a poset*, we divide the poset's elements into chains, usually seeking to use the fewest number of chains.

Dilworth's Theorem (1950): Let $(P, \leq)$ be a finite poset. Then width of P is the minimum number of chains needed to partition P (not in a on-line fashion). The example poset has width 2, as mentioned above, and it can clearly be partitioned into the chains $\{1, 2, 4, 8, 16\}$ and $\{3, 6, 12, 24, 48\}$.

The term *on-line* describes a situation in which the elements of an unknown poset are not presented all at once, as in the example poset above, but are instead presented one at a time according to some predetermined order.

On-line partitioning of posets is often described as a game between two opponents: the Spoiler and the Algorithm. The Spoiler gives the Algorithm one element of the poset at a time (the on-line part) along with that element's comparability to all previously given elements. Without knowledge of any further points to come, the Algorithm must then irrevocably assign that element to some chain of the previously given elements or start a new chain with that new element, always aiming to use the fewest number of chains to partition the poset. Assignments are also discussed in the language of coloring, where all elements with the same color must form a chain.

We define $val(w) = k$ to be the minimum number k such that there exists a strategy for an Algorithm to partition a poset of width less than or equal to w into k chains. In other words, there exists an Algorithm that can partition a poset of width at most w into exactly k chains.

1.2.1 On-line partitioning of width 2 posets

Establishing a specific value for $val(2)$ occurred in a few stages. First, lower and upper bounds were found for $val(2)$, specifically that $5 \leq val(2) \leq 6$, though this paper will focus only on the lower bound [3]. It is important to note, this lower bound was established regardless of the Algorithm used, that is, the proof develops a Spoiler than can force any Algorithm to use at least 5 chains in an on-line partitioning of a width 2 poset.

Next, an Algorithm was developed to on-line partition any width 2 poset into no more than 5 chains [2]. This development, together with the previously established bounds, conclusively set $val(2) = 5$.

Other work explored bounds on partitions using Algorithms other than the optimal one developed in [2]. Work with the Greedy Algorithm proved that, with the proper Spoiler, the Greedy Algorithm could be forced to use an infinite number of chains to partition a width 2 poset [1].

1.2.2 Finite width, $(t + t)$-free posets

While the Greedy Algorithm could be forced to use an infinite number of chains to partition a general width 2 poset, researchers sought conditions on the poset under which there does exist a bound on the number of chains the Greedy Algorithm would use [1].

It was found that for finite width, $(t + t)$-free posets, there does exist an upper bound on the number of chains the Greedy Algorithm will use [1].

Though not explicitly related to on-line partitioning, the final chapter of this thesis develops a tight upper bound on the number of incomparabilities a given point can have in a finite width, $(t + t)$-free, graded poset.

Chapter 2

Background

Definition: Let P be a set and $\sim$ a relation on the elements of P. Then $\sim$ is called a *partial order* if for all x, y, z in P,

- $x \sim\ x$ (reflexive)
- if $x \sim y$ and $y \sim z$, then $x \sim z$ (transitive)
- if $x \sim y$ and $y \sim x$, then $x = y$ (anti-symmetric)

When $\sim$ fulfills these criteria, $\leq$ is used instead of $\sim$.

The set P together with its partial ordering $\leq$ is called a *partially ordered set* or *poset*.

For a poset $(P, \leq)$ and any elements x, y in P, it could be that $x \leq y$, $y \leq x$, or neither. If either $x \leq y$ or $y \leq x$, then x and y are *comparable*. Otherwise, they are *incomparable*. Incomparability is often denoted $x|y$.

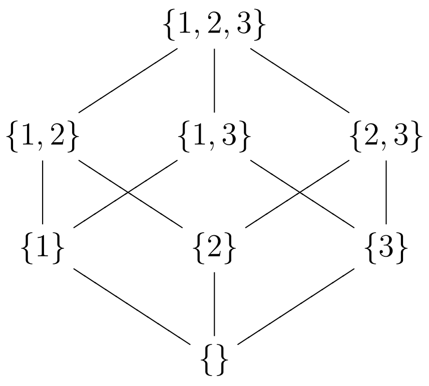
If every element in a poset is comparable to every other element, that poset is called a *total order* or a *linear order*. A totally ordered subset of any poset is called a *chain*, and if a subset of a poset contains only pairwise incomparable elements, that subset is called an *anti-chain*.

For a poset $(P, \leq)$ with x, y in P and $x \leq y$, we say that y *covers* x or x *is covered by* y if there is no z in P such that $x \leq z \leq y$.

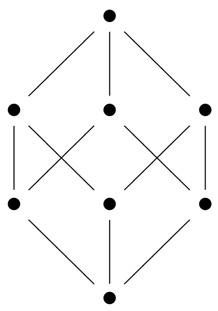
For a poset $(P, \leq)$, its *Hasse diagram* is a graph with:

- vertices are the elements of P
- for x, y in P, an edge connects x and y if and only if y covers x
- if $x \leq y$, then x is drawn lower than y

The Hasse diagram of the poset of subsets of $\{1, 2, 3\}$ with the inclusion partial ordering is:



or simply



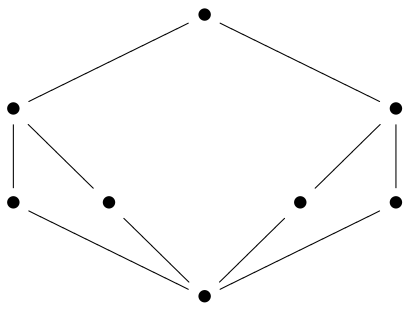
where $\{\}, \{1\}, \{1,2,3\}$ is a chain, $\{1\}, \{2\}, \{3\}$ is an anti-chain, and $\{2\}, \{1,2\}$ is a covering.

For a poset $(P, \leq)$, the chain (or anti-chain)

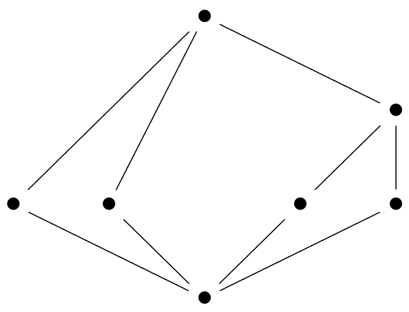
$$x_1 \leq x_2 \leq \ldots x_k$$

has *size* k and *length* $k-1$. We can think of the length as the number of links needed to make the chain (or anti-chain), the number of edges in the Hasse diagram of that chain (or anti-chain).

Note that in the Hasse diagram above, all maximal chains have length 3. Additionally, any chain from $\emptyset$ to a fixed subset A has length $|A|$. If all maximal chains of a poset have the same length, that poset is called *graded*:



A poset that is not graded is called *not graded*:



If a poset is graded, each element can be assigned a *rank* according to the number of links from the lowest element. Elements of the same rank form *levels*, which partition the poset. The *poset rank* is the rank of the maximal elements. The number of elements in the kth-level of P is the kth *rank number* of P. In the diagram of the subset of $\{1, 2, 3\}$, there are four levels with rank numbers 1,3,3,1.

Some examples of finite graded posets are:

- subset of a finite set with the inclusion ordering
- divisors of a positive integer with the divisibility ordering

For a finite poset $(P, \leq)$, we can define the *height* of P as the size of the longest chain in P (height(P) = rank(P) + 1) and the *width* of P as the size of the longest anti-chain in P. So the poset of subsets from $\{1, 2, 3\}$ has height 4 and width 3.

Chapter 3

Lower Bound on $val(2)$

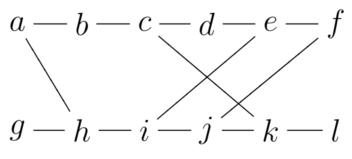
Theorem: $val(2) \geq 5$ [3].

Equivalently, the Spoiler can force the Algorithm to use at least 5 colors in on-line partitioning a width-2 poset. It is important to note that the proof of this theorem does not aim to identify one specific width-2 poset that cannot be partitioned into 4 chains. Rather, it illustrates how the Spoiler can respond to the Algorithm's coloring choices and build a poset that requires 5 colors. Here, the Algorithm is static, and the Spoiler is adaptable.

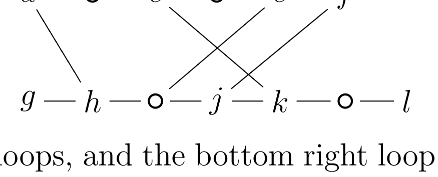
3.1 Notation

The proof as presented in [3] is heavily based on diagrams. Is these diagrams, points are connected with an edge if the left point is comparable to the right point under the relation*. The open circles or loops indicate possibly empty chains. A letter names each point, while a number indicates the color that the Algorithm assigned to that point.

So the uncolored poset



could be represented as

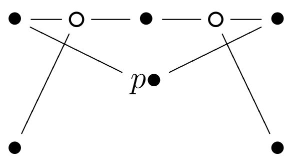


where b, d, i were put into loops, and the bottom right loop is empty. With respect to the points shown in the second diagram, the comparability relations are maintained from the first diagram.

*While convention for Hasse diagrams is bottom to top comparability, this left to right scheme is better suited to the size of these diagrams

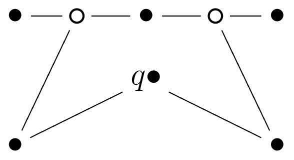
Before beginning the proof, consider some helpful plays for the Spoiler, that is, useful comparability relations that the Spoiler can build. Introducing a point with these comparability relationships would help The Spoiler force the Algorithm to use 5 colors on the poset.

$A(p)$ indicates introducing point p with comparability relationships



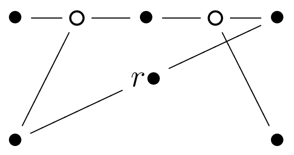
relative to the points around it.

$B(q)$ indicates introducing point q with comparability



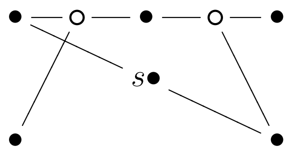
relative to the points around it.

$C(r)$ indicates introducing point r with comparability



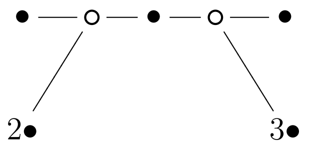
relative to the points around it.

$D(s)$ indicates introducing point s with comparability



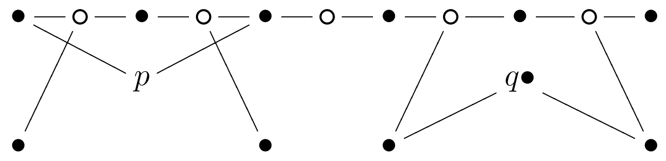
relative to the points around it.

Units of points are referred to as *blocks*, identified by the points sticking down. The following group of points

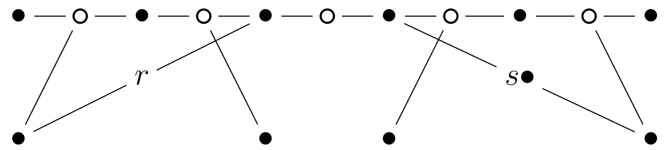


would be called a $2, 3$ *block*. Concatenated blocks are called *strings*.

Depending on the moves played, the most recently introduced points may or may not be comparable to one another. For example, if $A(p)$ and $B(q)$ are played in successive blocks, then $p|q$



but if $C(r)$ and $D(s)$ are played in successive blocks, then $r \leq s$ by transitivity.



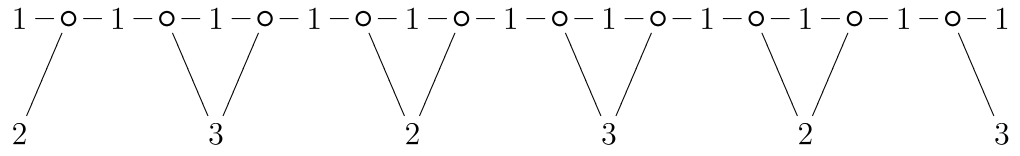
The linear order on the top of each diagram is (arbitrarily) colored with color 1. Any introduced element cannot be colored 1 because it is incomparable to the 1-colored element directly above it in the diagram.

3.2 Winning Strategies for the Spoiler

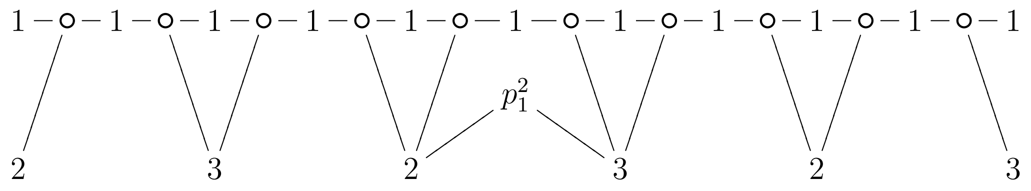
Newly introduced elements will be denoted p_n where n indicates the order of introduction (p_1 is the first point introduced, p_2 the second, etc.). After the Algorithm assigns a color, that color will appear as a superscript on the point. For example, p_3^4 indicates that the third element introduced was assigned color 4.

Lemma: If a 232323 string is formed during the course of an on-line partitioning game on a width-2 poset, then the Spoiler can force the Algorithm to use 5 colors [3].

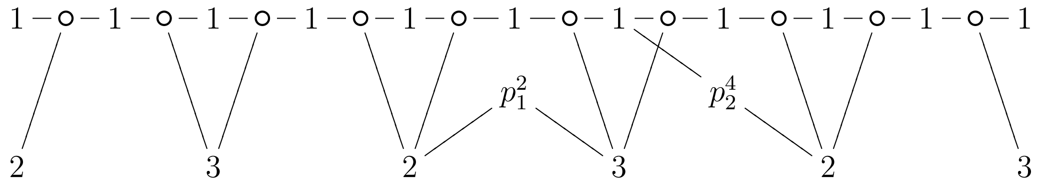
Proof: A 232323 string looks like:



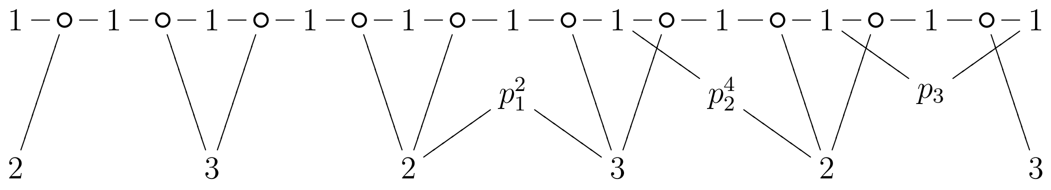
First, the Spoiler plays $B(p_1)$ at the third block. While the Algorithm could color p_1 any color of $2, 3, 4$, suppose the Algorithm chooses 2. We will consider the other cases later.



Next, the Spoiler plays $D(p_2)$ at the fourth block. Since p_2 is not comparable to p_1, it cannot be colored 2. p_2 is also not comparable to the 1-colored element above it or to the 3-colored element to its left, so it must be colored 4.

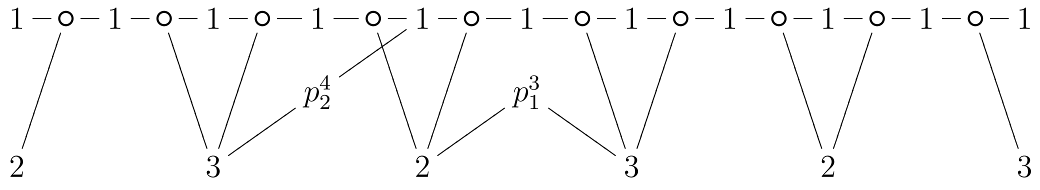

1 − o − 1 − o − 1 − o − 1 − o − 1 − o − 1 − o − 1 − o − 1 − o − 1 − o − 1 − o − 1
p_1^2
p_2^4
2
3
2
3
2
3

Finally, the Spoiler plays $A(p_3)$ at the fifth block.


1 − o − 1 − o − 1 − o − 1 − o − 1 − o − 1 − o − 1 − o − 1 − o − 1 − o − 1 − o − 1
p_1^2
p_2^4
p_3
2
3
2
3
2
3

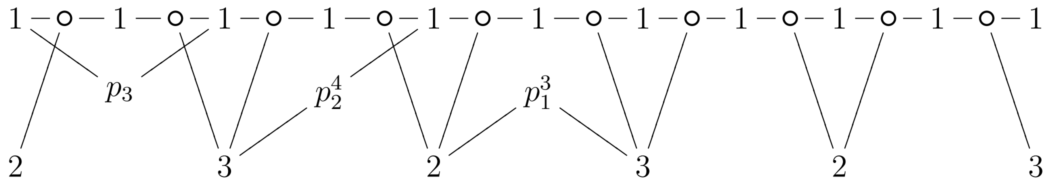
Since p_3 is incomparable to the adjacent 2- and 3-colored elements, it cannot receive either of those colors. p_3 is also incomparable to the element of the linear order above it and to p_2, so it cannot be colored 1 or 4; the Spoiler has forced the Algorithm to use a fifth color for p_3.

We did assume, though, that the Algorithm colored the first point, p_1, with color 2. We must consider the outcomes if the Algorithm has assigned either colors 3 or 4. If the Algorithm had colored p_1^3, then the Spoiler would play $C(p_2)$ at the second block, resulting in


1 − o − 1 − o − 1 − o − 1 − o − 1 − o − 1 − o − 1 − o − 1 − o − 1 − o − 1 − o − 1
p_2^4
p_1^3
2
3
2
3
2
3

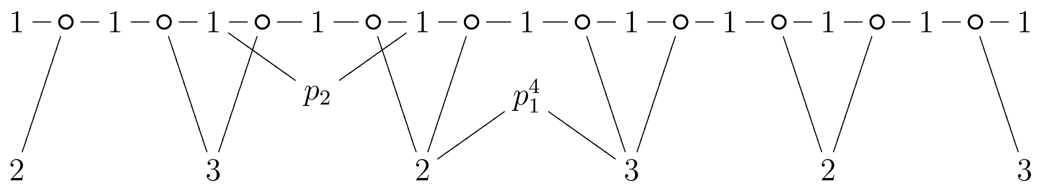
where the Algorithm had to assign color 4 to p_2 because p_2 is incomparable to the 2-colored element adjacent to it, the 1-colored element above it, and to p_1^3.

To win, the Spoiler plays $A(p_3)$ at the first block.


1 − o − 1 − o − 1 − o − 1 − o − 1 − o − 1 − o − 1 − o − 1 − o − 1 − o − 1 − o − 1
p_3
p_2^4
p_1^3
2
3
2
3
2
3

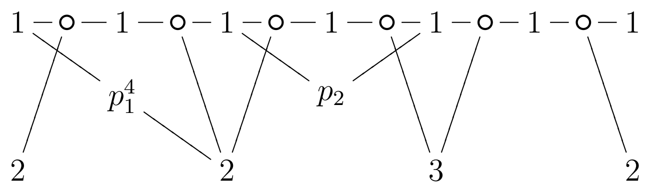
Since p_3 is incomparable to the 1-colored element above it, the adjacent 2- and 3-colored elements, and p_2^4, the Algorithm needs a fifth color for p_3. Notice the symmetry of these last two strategies; this symmetry is based on the 232323 pattern of the string.

Finally, if the Algorithm had colored p_1 with color 4, the Spoiler could win by playing $A(p_2)$ at the second block.

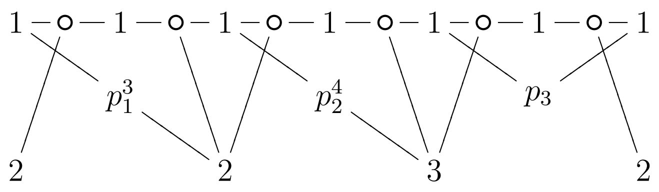


Lemma: If a 2232 string is formed during the course of an on-line partitioning game on a width-2 poset, then the Spoiler can force the Algorithm to use 5 colors [3].

Proof: Consider the following series of plays and colorings. The Spoiler begins by playing $D(p_1)$ at the first block. The Algorithm can color p_1 either 3 or 4. If the Algorithm chooses color 4 for p_1, the Spoiler can win by playing $A(p_2)$ at the second block.

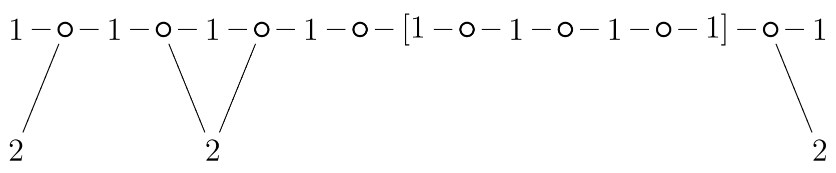


If the Algorithm chooses color 3 for p_1, the Spoiler can win by playing $D(p_2)$ at the second block followed by $A(p_3)$ at the third block.

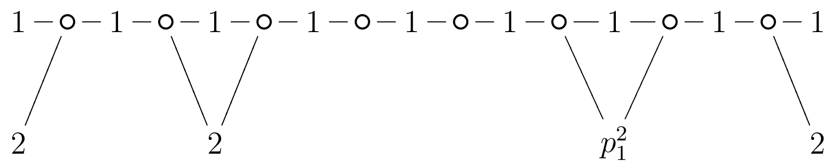


Lemma: If a 222 string in which the last loop contains a chain of size at least 4 is formed during the course of an on-line partitioning game on a width-2 poset, then the Spoiler can force the Algorithm to use 5 colors [3].

Proof: The condition of the size-4 chain translates to four 1-colored elements and looks like

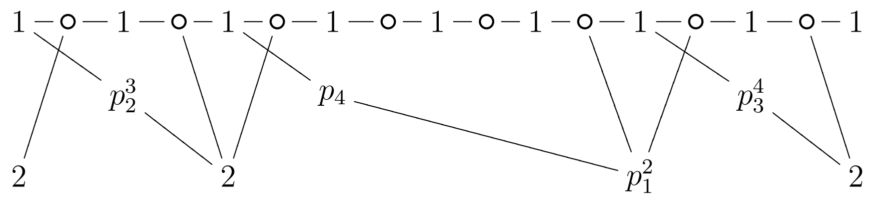


where the four 1-colored elements are bracketed for emphasis. The Spoiler begins by playing p_1 as indicated below.

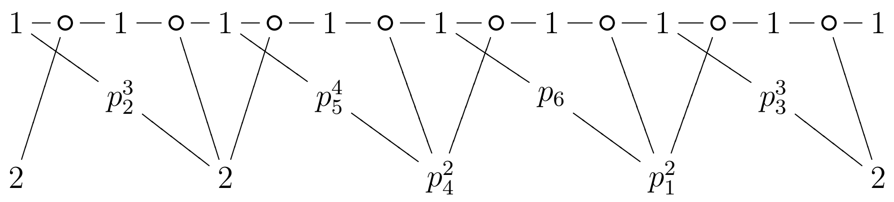


If the Algorithm were to color p_1 with either 3 or 4, the result would be a 2232 or 2242 string, which we just proved is a winning position for the Spoiler. The Algorithm must assign color 2 to p_1. From here, there are a few possible paths the game could take.

When the Spoiler plays $D(p_2)$ at the first block, the Algorithm must assign either 3 or 4 to p_2. This choice is not significant, as the following steps would symmetrically reflect the assigned color. Suppose the Algorithm assigns p_2^3. When the Spoiler plays $D(p_3)$ at the $p_1, 2$ block, the Algorithm must again assign either color 3 or 4. This choice is significant. If the Algorithm chooses color 4, then the Spoiler plays $D(p_4)$ at the second block to win.



If the Algorithm assigns color 3 to p_3, then the Spoiler plays p_4 as indicated below. To avoid a 2232 or 2242 string, the Algorithm must color p_4 with color 2. At the second block, the Spoiler then plays $D(p_5)$, which the Algorithm must color 4 by the incomparabilities. The Spoiler wins by playing $D(p_6)$ in the block after p_4.



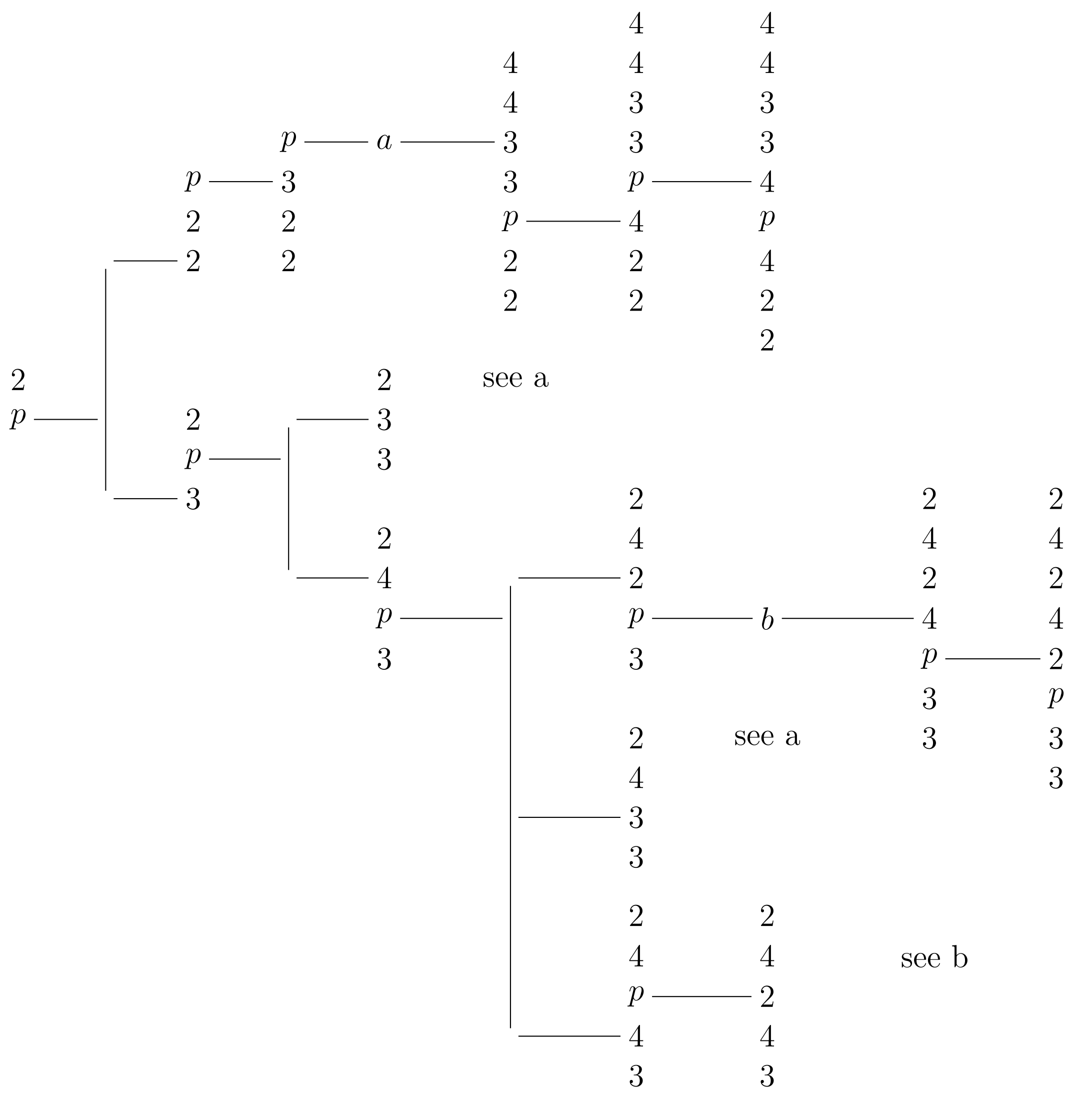
More generally, any string of the form xxx, $xxyx$, or $xyxyxy$ where x, y are in $\{2, 3, 4\}$ is a winning string for the Spoiler.

3.3 Forcing a Winning Position

Up to this point, we assumed that the game proceeded via one of the strings we have examined. We must also prove that the Spoiler can introduce points in such a way as to force the appearance of such a string in the on-line partitioning of any width-2 poset.

Lemma: In an on-line partitioning game on a width-2 poset, the Spoiler can force the appearance of either an xxx, an $xxyx$, or an $xyxyxy$ string where x, y are in $\{2, 3, 4\}$ [3].

Proof: consider the following diagram.



Each vertical, continuous string represents the situation given to the Algorithm where p is the newly introduced element that needs coloring. The assumption is that the Algorithm assigns colors so as to avoid an xxx, $xxyx$, or $xyxyxy$ pattern for as long as possible. This diagram is read from left to right with the different branches representing the different color assignments the Algorithm might make. All assignment possibilities, up to a permutation of the colors, are presented in the above diagram.

The line segments running through a and b represent situations in which the Algorithm can effectively stall. In a, the Spoiler will keep introducing points above the lowest 3-colored element. The Algorithm can assign color 4 to two of those points, but it must assign color 3 to the third (assigning color 2 would result in a 2232, and assigning color 4 would give a 444). At b, the Spoiler will again introduce elements above the bottom 3-colored element. The Algorithm can stall by assigning color 3 to the first of these introduced elements, but the Algorithm must color the next element there 4 in order to avoid a 333 or 2242.

To understand this diagram better, follow its bottom-most branch. First the Spoiler introduces an element, which the Algorithm colors 2[†]. The Spoiler introduces a new element below the first element (though it could also be above, by symmetry). The Algorithm colors the new element 3 or essentially not the same color as the first element. The Spoiler introduces a new element between 2 and 3, which the Algorithm colors 4 (essentially, the Algorithm chooses a color not yet used). The Spoiler introduces the new element below 4 and above 3. At this point, each color assignment leads to a different path. Suppose the Algorithm assigns the color 4 to the new point. The Spoiler plays between the two 4-colored points. The Algorithm cannot color this new point 4, so it must choose either 2 or 3, a symmetric choice in this case. The result becomes (a permutation) of the situation after point b. The Spoiler plays above the two bottom 3-colored elements. To avoid a 333 or a 4424, the Algorithm colors the new point 2. When the Spoiler again plays above the two bottom 3-colored elements, the Algorithm is stuck. Coloring the new point 4 results in a 242424; coloring the new point 3 results in a 333; and coloring the new point 2 results in a 2242. The Spoiler has achieved a winning position.

This diagram addresses, up to a permutation of colors, all possible assignments that the Algorithm could make given the Spoiler's introductions. As all those assignments lead to a winning position for the Spoiler, we can conclude that the Spoiler can always force the game in to a winning position for itself.

3.4 Completing the Proof

Theorem: $val(2) \geq 5$ [3].

Proof: The Spoiler begins the game by introducing a linear order of $4(6 \times 2^9 - 1) - 3 = 12,281$ elements. No matter its coloring scheme, the Algorithm will color at least $6 \times 2^9 - 1$ of these the same color, say 1. Let L be the set of those $6 \times 2^9 - 1$ elements. The Spoiler's $12,281 + i$ move for $0 < i \leq 9$ will consist of choosing an element a_i in L and then introducing a new element, b_i, that is incomparable to a_i but comparable to every other previously introduced element. These a_i will be chosen so that there are at least $6 \times 2^{9-i} - 1$ elements of L above and below a_i and separating a_i from any previous a_j. This requirement ensures the condition of the lemma that xxx is a winning position, namely, that there are always at least four 1-colored elements in any loop of the linear order.

The Algorithm must color the b_i with the colors 2,3,4. By the previous lemma, the Spoiler can force the game into a xxx, $xxyx$, or $xyxyxy$ string. Thus the Spoiler can always win the game, forcing the Algorithm to use at least 5 colors.

[†]If the Algorithm had chosen a different starting color, the colors in the diagram would simply be permuted.

3.5 Examples

To better understand this proof, consider some specific algorithms.

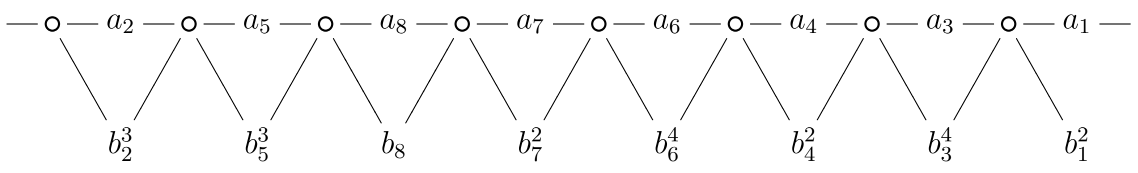
3.5.1 The "Stupid" Algorithm

The "stupid" algorithm uses as many of the four colors as possible, always using whichever color has been used the least. As the Spoiler introduces the 12,281 elements of the linear order, the stupid algorithm will color them in such a way that the resulting coloring can be divided into adjacent segments of size 4 where each segment contains each of the four colors exactly once. It might look something like:

$$\alpha\beta\delta\gamma|\delta\alpha\gamma\beta|\gamma\beta\alpha\delta|\ldots$$

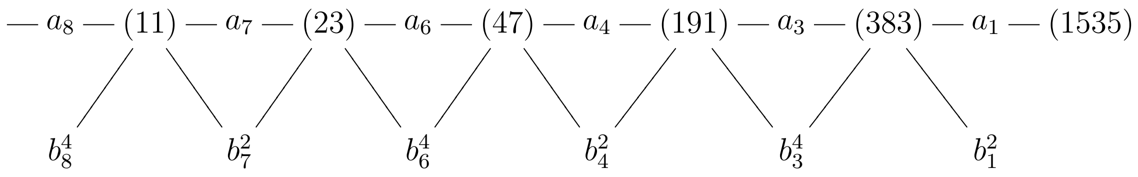
Since $12,281 = 4(6 * 2^9 - 1) - 3$, one of the colors has been used more than the other three. Call that color 1. That color has been used exactly $6 * 2^9 - 1$ times, and there are between 0 and 6 non-1-colored elements separating the 1-colored elements from each other. Consider just the $6 * 2^9 - 1$ elements of color 1. Since the algorithm uses a new color whenever possible, the Spoiler can set up the game through one of the bottom routes. Consider the route through b.

The poset would look like



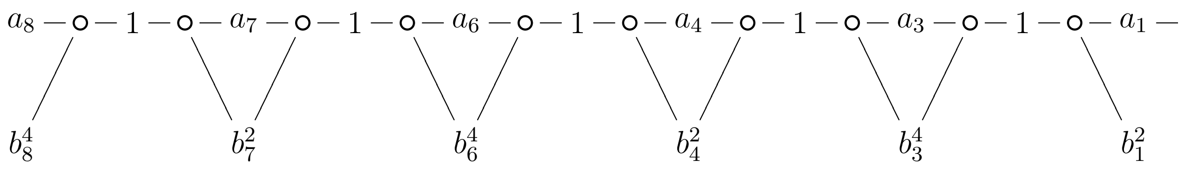
with the assigned colors indicated [‡].

For b_8, the stupid algorithm will use either 3 or 4 (2 has been used three times while 3 and 4 have only been used twice). For this case, suppose the algorithm assigns color 4, resulting in a 424242 pattern.

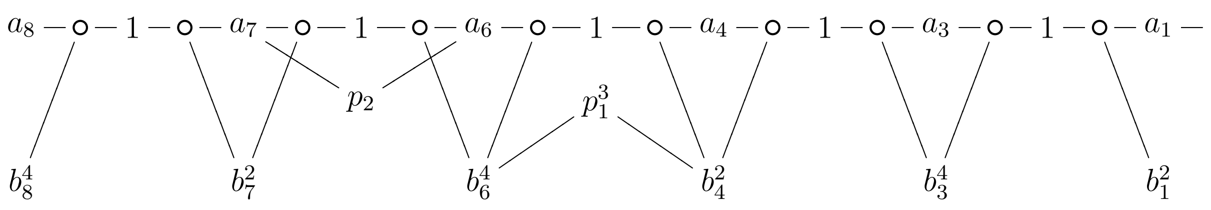


The numbers in parenthesis indicate the minimum number of 1-colored elements that each loop must contain according to the rules for selecting the a_i. Clearly, all of these numbers are greater than 1, so we can view the diagram as

[‡]These relationships are generated by following the path through b. The "stupid" algorithm would not always produce this coloring; it is simply an illustrative example.


a_8 — o — 1 — o — a_7 — o — 1 — o — a_6 — o — 1 — o — a_4 — o — 1 — o — a_3 — o — 1 — o — a_1 —
b_8^4
b_7^2
b_6^4
b_4^2
b_3^4
b_1^2

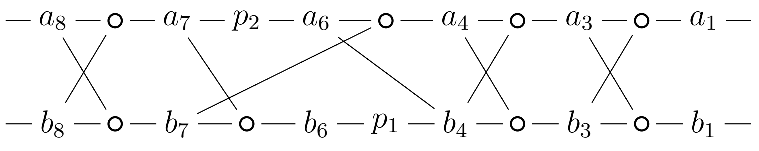
Following the steps in the lemma that *xyxyxy* is a winning position for the Spoiler, the Spoiler will first play $B(p_1)$ at the third block. Since 2 and 4 have been used three times, but 3 has only been used twice, the stupid algorithm will assign color 3 to p_1 (note that p_1 is comparable by transitivity to the 3-colored b_2 and b_5 which have been omitted from this view). The Spoiler can then play $A(p_2)$ at the second block to win.


a_8 — o — 1 — o — a_7 — o — 1 — o — a_6 — o — 1 — o — a_4 — o — 1 — o — a_3 — o — 1 — o — a_1 —
p_2
p_1^3
b_8^4
b_7^2
b_6^4
b_4^2
b_3^4
b_1^2

By considering the incomparabilities in this final diagram, we can roughly construct the poset that the Spoiler used to force the stupid algorithm to use at least 5 colors. The incomparabilities are:

a_8	a_7	p_2	p_2	a_6	a_6	a_4	a_4	a_3	a_1
b_8	b_7	b_7	b_6	b_6	p_1	p_1	b_4	b_3	b_1

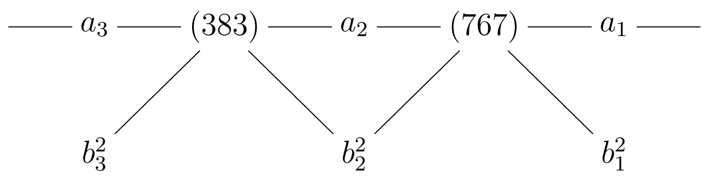
where each element is incomparable to the element directly above or below it. Using this list of incomparabilities and the comparabilities evident in the diagram above, we can construct the poset


— a_8 — o — a_7 — p_2 — a_6 — o — a_4 — o — a_3 — o — a_1 —
— b_8 — o — b_7 — o — b_6 — p_1 — b_4 — o — b_3 — o — b_1 —

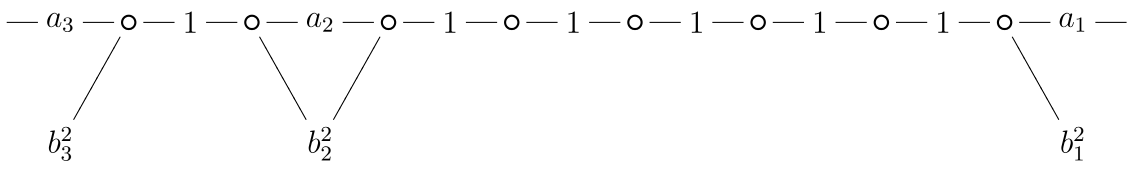
where the loops indicated here cannot be empty.

3.5.2 The Greedy Algorithm

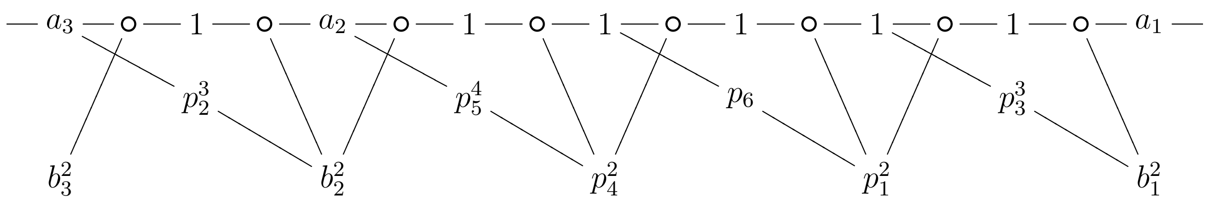
When the Algorithm is the greedy algorithm, it will color all 12,281 elements of the linear order with one color, say 1. When the Spoiler chooses a_1 and plays b_1, the greedy algorithm assigns color 2 to b_1. When the Spoiler chooses a_2, a_3 and plays b_2, b_3, the greedy algorithm with also color these b_2, b_3 with color 2, creating a 222 scenario.



Since there are certainly more than four 1-colored elements in the final loop, we can view the diagram as follows



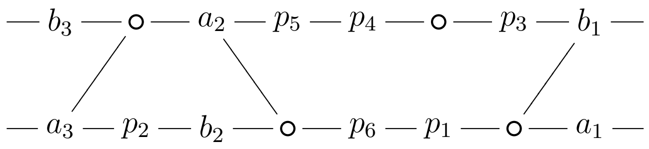
and follow the moves that the Spoiler plays to win. The Spoiler plays p_1, which the Algorithm will still color 2. Then the Spoiler plays $D(p_2)$ at the first block, making $p_2|b_3^2$, so p_2 is 3-colored. Next, the Spoiler plays $D(p_3)$ at the p_1 block. Since $p_3|p_1^2$, the Algorithm will assign color 3 to p_3. The Spoiler plays p_4 as indicated, and the Algorithm will color it 2. Then the Spoiler plays $D(p_5)$ at the second block, making $p_5|b_2^2$ and $p_5|p_2^3$, leading the Algorithm to assign color 4 to p_5. To win, the Spoiler plays $D(p_6)$ at the p_4 block.



Again, we can consider the incomparabilities and construct the poset. The incomparabilities, where each element is incomparable to the one below or above it, are:

a_3	p_2	p_2	b_2	b_2	p_6	p_6	p_6	a_1	a_1	p_1	p_1
b_3	b_3	a_2	a_2	p_5	p_5	p_4	p_3	p_3	b_1	p_3	p_4

so the poset looks like:



To further explore this Spoiler on the greedy algorithm, consider how large a poset is needed to force the greedy algorithm to use 5 colors.

The greedy algorithm will color b_1, b_2, b_3 with the same color, so the Spoiler only needs to introduce three points to get to a winning position. Loosen the restriction on the placement of a_1, a_2, a_3 to require that there are $6 * 2^{3-i} - 1$ elements of L, the linear order, on either side of a_i and between a_i and any previous a_j. Using this restriction

$$\begin{tikzcd}
(11) \arrow[r, dash] & a_2 \arrow[r, dash] & (11) \arrow[r, dash] \arrow[dl, dash] \arrow[dr, dash] & a_1 \arrow[r, dash] & (5) \arrow[r, dash] \arrow[dl, dash] \arrow[dr, dash] & a_3 \arrow[r, dash] & (17) \\
& b_2^2 & & b_1^2 & & b_3^2 &
\end{tikzcd}$$

where there need to be at least 23 elements on either side of a_1, we see that the linear order must have size at least 47. With the strategy above, we also needed the three b_i and the six additional points p_1 to p_6, so the Spoiler could force the greedy algorithm to use 5 colors on the poset containing 56 elements.

Taking the spacing requirement even more loosely, since the algorithm is greedy and the *xxx* strategy can be easily employed, eliminate the left 11 elements, 10 of the 11 between a_2 and a_1, and the right 17 elements. The Spoiler could force the greedy algorithm to use 5 colors on a poset containing 18 elements.

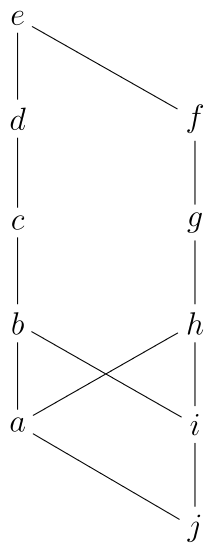
Chapter 4

Exact Value for $val(2)$

Knowing that $val(2) \geq 5$, we can consider an Algorithm that partitions any width-2 poset into exactly 5 chains [2]. For a poset $(P, \leq)$, consider the following series decomposition of P:

- This decomposition must be the finest decomposition such that any pair of incomparable elements of P belong to the same unit of the decomposition, the same *block*.
- Any block that contains more than 1 element is called *rigid*. Non-rigid blocks are called *singletons*.
- Any rigid block will have 2 maximal and 2 minimal elements. These at most 4 elements are called the *corners*.

To understand this decomposition, consider the following poset:



The elements e and j are comparable to every other element, so each can be in its own component without violating the rule that a pair of incomparable elements must belong to the same component. So there are e, j singletons. a is incomparable to i, but both a and i are comparable to every other element in the poset. Thus a, i form a rigid component where a and i are both top and bottom corners. Then there is an interior series of incomparabilities:

$$b|h, h|c, c|g, g|d, d|f$$

Since all pairs of incomparable elements must be in the same component, we get a big rigid component with bottom corners b, h, top corners d, f, and c, g in the middle. Thus the decomposition looks like

$$\{e\}\{j\}\{a,i\}\left\{\begin{array}{cc} d & f \\ c & g \\ b & h \end{array}\right\}$$

4.1 The Algorithm

Consider an on-line partitioning Algorithm is based on the above partitioning process. When on-line partitioning poset $(P, \leq)$, there are 5 possible actions that the new point p could induce [2].

1. p could form a new block on its own. That is, p is comparable to all of the previously introduced points. To be in a preexisting block, the newly introduced element must be incomparable to some element in that block.

2. p, together with two rigid blocks and possibly several singletons, could form a new rigid block. That is, p could be incomparable to elements that are comparable to each other.

3. p, together with some singletons, could form a new rigid block. Again, p could be incomparable to elements that are comparable to each other.

4. p and 1 rigid block and possibly some singletons could form a new rigid block with p as a corner. p could be smaller or larger than all the elements to which it is incomparable.

5. p could extend an existing rigid block but is not a corner. That is, p could be incomparable to some point in block R but would not be a maximum or minimum element for that block.

To assign the chains that partition P, the Algorithm assigns a color from $\{1, 2, 3, 4, g\}$ to each point as it is introduced. This coloring must not violate the following three properties:

A) If y is a corner of a rigid block, then y has an associated set of *virtual colors*, $vc(y)$ from $\{1, 2, 3, 4\}$.

B) Every color class forms a chain, that is, $\{z : \mathit{color}(z) = \gamma \text{ or } \gamma \in vc(z)\}$ is a chain for $\gamma = 1, 2, 3, 4, g$.

C) The sets of virtual colors for the top corners of a rigid block R and for the bottom corners of the next rigid block above R are different. Specifically, if t is a top corner of R and b is a bottom corner of the next block, then $|vc(t) \cap vc(b)| = 1$.

Preserving these three properties, the Algorithm assigns a color to each new point p according to the action it induces [2].

Case 1) If p induces action 1, the Algorithm assigns color g to p. p does not yet have a set of virtual colors.

Case 2) If p induces action 2, then it joins two rigid blocks. Let $\{t_1, t_2\}$ be the top corners of the lower block and $\{b_1, b_2\}$ be the bottom corners of the higher block of the two blocks combined by p. To avoid an anti-chain of size 3, p is comparable with exactly one t, say t_1, and exactly one b, say b_1. The Algorithm assigns to p the unique color of $vc(t_1) \cap vc(b_1)$. p does not yet have a set of virtual colors.

Case 3) If p induces action 3 (it generates a new block from singletons), then there may be a chain of $x_1, \ldots, x_h$ of g-colored points incomparable to p. Up to a permutation of colors, $vc(b_1, b_2 \text{ from block above } p\ ) = \{1,2\}, \{3,4\}$ and $vc(t_1, t_2 \text{ from block below } p\ ) = \{1,3\}, \{2,4\}$. Defining $vc(p) = \{1,4\}$ and $vc(x_1) = \ldots = vc(x_h) = \{2,3\}$ maintains the invariance of the three above properties. The Algorithm assigns a color to p from $vc(p)$.

Case 4) p becomes a new corner of block R. By duality, is it sufficient to consider the case of p as a new bottom corner. Let b_1, b_2 be the old bottom corners and x, y be the new bottom corners. Assume $p \leq b_1$ and $p|b_2$. Since $y|p$, y must be comparable to b_2 to avoid a size 3 anti-chain. We can define $vc(p) = vc(b_1)$, $vc(y) = vc(b_2)$, and give p a color from $vc(p)$.

Before considering Case 5, consider the following lemma.

Lemma: let R be a rigid block, and let C_1, C_2 be a chain partition of R. If p is a point extending R, then either $C_1 + p$ or $C_2 + p$ is a chain [2].

Proof: The incomparability graph of a rigid block is a connected bipartite graph. R and $R + p$ are rigid, so the unique bipartition of the incomparability graph of $R + p$ is obtained from the unique bipartition of the incomparability graph of R by extending one of the sides with p.

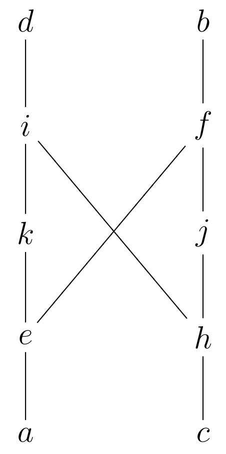
Case 5) p falls into the interior of R. Assume that $C_1 + p, C_2$ is the chain partition of $R + p$. Let y be the first element below p in C_1 that was ever a bottom corner, and let z be the first element above p in C_1 that was ever a top corner. y, z must exist since C_1 is bounded by the corners of R. Applying lemma 1, $C_1 + p$ is a chain, so $vc(y) \cap cv(z)$ is non-empty. Let $vc(p) = vc(y) \cap cv(z)$ and assign p a color from $vc(p)$.

Theorem: An on-line order of width 2 can be partitioned on-line into 5 chains [2].

Proof: applying the above cases and colorings produces no more than 5 colored chains that partition the on-line order.

4.2 Example Posets

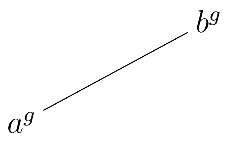
To better understand this partitioning Algorithm, try it out on some example posets. First, consider the poset



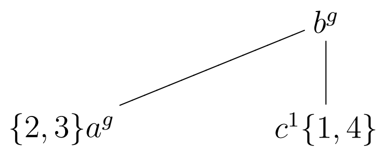
where the Spoiler introduces the points in alphabetical order.

First, the Spoiler introduces a. Since it is the first element, a it is trivially comparable to all other introduced elements, so it falls under case 1. The coloring rule for case 1 dictates that a receives color g.

Next, the Spoiler introduces b. Since b is comparable to a (through e and f in the diagram above), it also falls under case 1 and is colored g. At this point we have two singletons $\{a\}, \{b\}$, each colored g and having the comparability



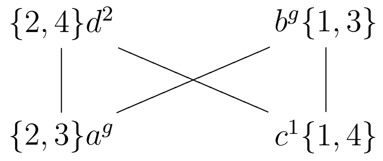
Now the Spoiler introduces c. Element c is comparable to b (via h, i, f), but it is not comparable to a. By the rules of the decomposition, c must be put into the same block as a. Element c becomes the second element in a's block, transforming a singleton into a rigid block under case 3. Closely follow the language of case 3 in terms of this poset. "If (in our case) c induces action 3, then there may be a chain $x_1, \ldots, x_h$ of g-colored points incomparable to c." Here, that chain of g-colored points is just a. "Up to a permutation of colors, $vc(b_1, b_2$ from component above c $) = \{1, 2\}, \{3, 4\}$ and $vc(t_1, t_2$ from component below c $) = \{1, 3\}, \{2, 4\}$." At this point, there is no component below c. Also, the component above c contains only b, which has a color but not a set of virtual colors. So continue with the rule. "Defining $vc(c) = \{1, 4\}$ and $vc(x_1) = \ldots = vc(x_h) = \{2, 3\}$ maintains the invariance of the three above properties. The Algorithm assigns a color to c from $vc(c)$." In this case, define $vc(c) = \{1, 4\}$, $vc(a) = \{2, 3\}$, so the Algorithm can assign color 1 to c. The decomposition has the $\{a, c\}$ rigid block and the $\{b\}$ singleton



Check that the invariance is obeyed. In adherence to rule A, a, c are corners of a rigid block and have an associated set of virtual colors. Element b does not have a set of virtual colors, but since it is only a singleton, it does not violate rule A. Certainly every color class forms a chain, satisfying rule B. There is only one rigid block so far, so rule C is trivially satisfied.

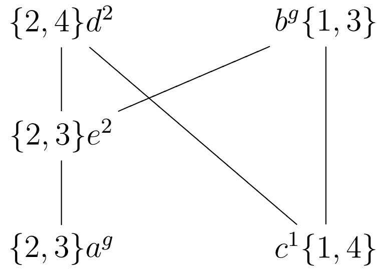
The Spoiler introduces d. Element d is comparable to a and c, but not to b. Thus d must be in the same block as b, expanding a singleton to a rigid block as in case 3. Recall the coloring rule for case 3 in terms of this poset and the newly introduced point d. Element d "induces action 3", and here the "chain of $x_1, \ldots, x_h$ of g-colored points incomparable to d" is simply element b. There is no block above d, but $vc(a, c$ from component below d $) = \{2,3\}, \{1,4\}$. Defining $vc(d) = \{2,4\}$ and $vc(b) = \{1,3\}$ maintains the invariance of the three above properties. The Algorithm can assign a color to d from $vc(d)$, say 2. Now the decomposition is the $\{a, c\}$ rigid block and the $\{d, b\}$ rigid block.

Again check that the invariance is obeyed. Each of the four introduced elements is a corner of a rigid block, and each has a set of virtual colors, satisfying A. The introduced elements are comparable as



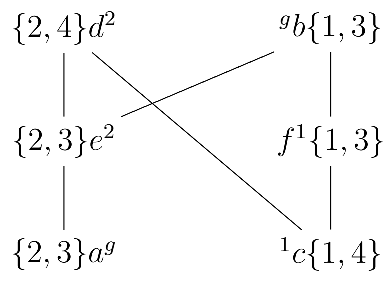
and inspection shows that each class of colors or virtual colors forms a chain, satisfying B. Inspection also reveals that the corners of adjacent blocks share exactly 1 virtual color, satisfying rule C.

Now the Spoiler introduces e. Element e is comparable to a, b, d and is incomparable to c, so by action 4, e forms a top corner of the rigid $\{a, c, e\}$ block. As this is a new case for this poset, carefully consider the rule's language. "e is a new corner of component $\{a, c, e\}$. By duality, is it sufficient to consider the case of e as a top corner. Let a, c be the old top corners and e, c be the new top corners. Assume $e \leq a$ and $e|c$. Since $a|c$, $a \leq a$ to avoid a size 3 anti-chain. Define $vc(e) = vc(a) = \{2,3\}$ and $vc(c) = vc(c)$ and give e a color from $vc(e)$," say 2. Now the decomposition consists of the rigid $\{a, c, e\}$ and $\{d, b\}$ blocks, and the comparability is

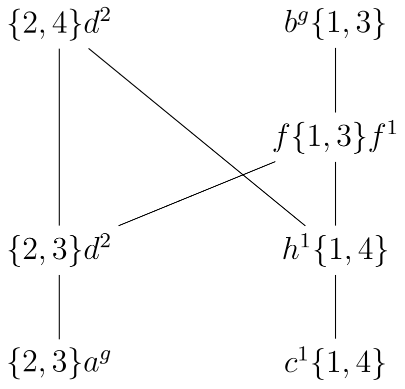


It is left to the reader to check that the invariance is obeyed.

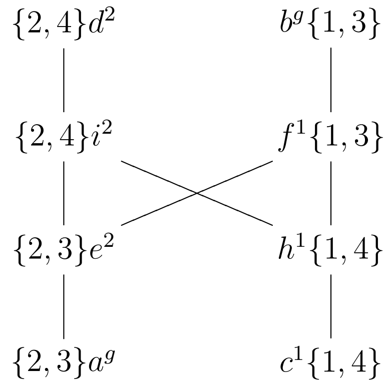
The Spoiler introduces f. Element f falls under case 4, as e did, except that it forms a new bottom corner of the $\{d, b, f\}$ block. Elements d, b were the old bottom corners and $f \leq b$, so by the coloring rules for case 4, the Algorithm will let $vc(f) = vc(b) = \{1, 3\}$ and assign a color from $vc(f)$, say 1. Then the decomposition is the rigid $\{a, c, e\}$ and $\{d, b, f\}$ blocks, and the comparability is



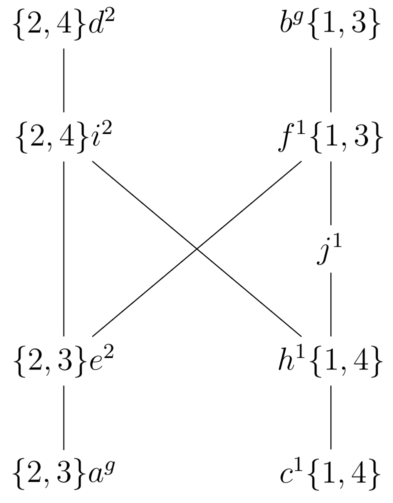
Element h, like element e, forms a new top corner of the rigid $\{a, c, e, h\}$ block under case 4. Since h is comparable to c, the old top corner, $vc(h) = vc(c) = \{1, 4\}$ and the Algorithm can assign color 1 to h. The decomposition is the rigid blocks $\{a, c, e, h\}$ and $\{d, b, f\}$ and the comparability is



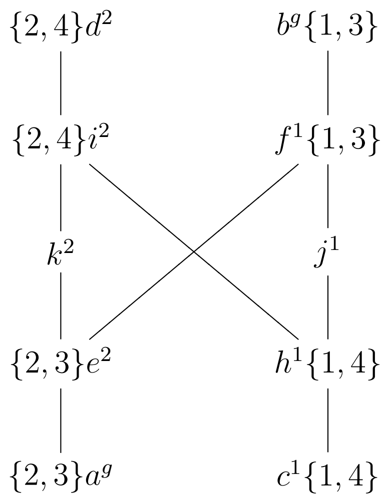
Element i acts like element h under case 4, except that it forms a new bottom corner of rigid block $\{d, b, f, i\}$ instead of a new top corner of $\{a, c, e, h\}$. Since i is comparable to older bottom corner d, the Algorithm will let $vc(i) = vc(d) = \{2, 4\}$ and assign color 2 to i. The decomposition is $\{a, c, e, h\}$ and $\{d, b, f, i\}$ with comparability



Now the Spoiler introduces j. Element j is incomparable to a, e, i, d and comparable to c, h, f, b. Since pairs of incomparable elements must be in the same block, j unites the two preexisting blocks and goes somewhere in the middle of the newly formed block, falling under case 2. Again, read this new case slowly. "If j induces action 2, then let $\{t_1, t_2\}$ be the top corners of the lower block and $\{b_1, b_2\}$ be the bottom corners of the higher block of the two blocks combined by j." Here, those top corners are $\{e, h\}$ and the bottom corners are $\{i, f\}$. "To avoid an anti-chain of size 3, j is comparable with exactly one t, say t_1, and exactly one b, say b_1." j is comparable to h and f. "Then $color(j)$ = the unique color of $vc(t_1) \cap vc(b_1)$." By rule C, there must be 1 color in the intersection of the virtual color sets of h and f. That color is 1, so the Algorithm assigns color 1 to j. j does not receive a set of virtual colors. Note that rule A is not broken, as j is not a corner of a rigid block. The decomposition is now one big rigid block with connectivity



Finally, the Spoiler introduces k. Element k is incomparable to c, h, j, f, b, so it must be put into their block. As k does not form a corner, it falls under case 5. Again, this a new case, so read the rules closely. "Assume that $C_1 + k, C_2$ is the chain partition of $\{a, b, c, d, e, f, h, i, j\} + k$. Let y be the first element below k in C_1 that was ever a bottom corner, and let z be the first element above k in C_1 that was ever a top corner." For this poset, $e$and i take on these respective roles. Applying lemma 1, $C_1 + k$ is a chain, so $vc(e) \cap cv(i) = \{2\}$ is non-empty, and the Algorithm can assign color 2 to k. The final comparability and color assignment is



For this poset, a greedy approach* to this Algorithm used 3 colors: 1,2, and g. Inspection of the diagram reveals that the Algorithm could have used any of the colors from the virtual color sets and used no more than five colors.

4.3 Performance Against Specific Spoilers

Now consider how this Algorithm would perform against the Spoiler developed in the previous chapter, the Spoiler intended to force the use of at least 5 colors.

The Spoiler begins the game by introducing a linear order of $4(6 \times 2^9 - 1) - 3 = 12,281$ elements. Because they form a linear order, each of the 12,281 elements forms a singleton, which the Algorithm colors g. It is perhaps easiest to follow the interaction of Spoiler and Algorithm as a series of actions and responses.

1) Spoiler: choose a_1 and introduce b_1.

   Algorithm: $a_1|b_1$ but b_1 is comparable to every other element in the linear order, so b_1 forms a rigid block with a_1 where b_1 is a corner (Case 4). The Algorithm assigns virtual color sets $vc(a_1) = \{1,2\}$ and $vc(b_1) = \{3,4\}$ up to a permutation of these colors. Without loss of generality, assume the Algorithm assigns color 3 to b_1.

2) Spoiler: choose a_2 below a_1 and introduce b_2.

   Algorithm: $a_2|b_2$ but b_2 is comparable to all other elements in the linear order and to b_1. Under case 4, b_2 forms a rigid block with a_2, and the Algorithm assigns virtual color sets $vc(a_2) = \{1,3\}$ and $vc(b_2) = \{2,4\}$. The Algorithm assigns b_2^2.

3) Spoiler: choose a_3 above a_2 and below a_1 and introduce b_3.

   Algorithm: $a_3|b_3$ but b_3 is comparable to all other elements in the linear order and to b_1 and b_2. Under case 4, b_3 forms a rigid block with a_3, and the Algorithm

*This was a greedy approach in that, when given a choice of colors from an element's virtual color set, the Algorithm only used colors that were already used, whenever possible.

assigns virtual color sets $vc(a_3) = \{1, 4\}$ and $vc(b_3) = \{2, 3\}$. The Algorithm assigns b_3^2.

4) Spoiler: choose a_4 above a_1 and introduce b_3.

   Algorithm: As in previous moves, b_4 induces a case 4, forming a rigid block with a_4. The Algorithm assigns virtual color sets $vc(a_4) = \{1, 4\}$ and $vc(b_4) = \{2, 3\}$. The Algorithm assigns b_4^3.

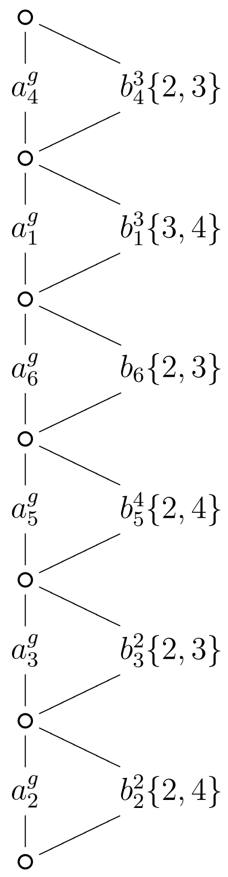
5) Spoiler: choose a_5 above a_3 and below a_1 and introduce b_5.

   Algorithm: As before, b_5 induces a case 4, forming a rigid block with a_5. The Algorithm assigns virtual color sets $vc(a_5) = \{1, 3\}$ and $vc(b_5) = \{2, 4\}$. If the Algorithm assigns b_5^2, the Spoiler wins by the 222 formed. Assume Algorithm assigns b_5^4.

6) Spoiler: choose a_6 above a_5 and below a_1 and introduce b_6.

   Algorithm: b_6 forms a rigid block with a_6. The Algorithm assigns virtual colors sets $vc(a_6) = \{1, 4\}$ and $vc(b_6) = \{2, 3\}$. No matter the assignment, either a 333 or a 2242 will be formed; the Spoiler wins.

Using this Algorithm, is it easier for the Spoiler to force a winning position because the color choices are limited to 2 colors (instead of 4) by the set of virtual colors. That said, this Algorithm cannot be forced to use more than 5 colors on any poset.



Chapter 5

On-Line Partitioning using the Greedy Algorithm

5.1 Forcing the Use of Infinite Colors on a Width-2 Poset

Theorem: There exists a poset $(P, \leq)$ and a presentation of P such that the Greedy Algorithm uses infinitely many chains in a on-line partition of P [1].

Proof: First we will describe the poset and give the Spoiler's on-line order for it. Then we will show that the Greedy Algorithm uses infinitely many chains to partition it.

Let P be composed of two disjoint chains, α_0 and α_1. To start, let $\alpha_1 = x_1$ and let $\alpha_0 = \emptyset$. For each m in $\mathbb{N}$, G_m is a linear order of m points. If m is odd, every element of G_m is comparable to every element of α_1, and the elements of G_m are referred to as $x_m, x_{m-1}, \ldots, x_1$. If m is even, every element of G_m is comparable to every element of α_0, and the elements of G_m are referred to as $y_m, y_{m-1}, \ldots, y_1$. While the following holds for both even and odd m, assume for the sake of illustration that m is even. The points of G_m are presented $y_m, y_{m-1}, \ldots, y_1$ and satisfy the following conditions:

- $y_k > G_1 \cup \ldots \cup G_{m-2} \cup \{x_{m-1}, x_{m-2}, \ldots, x_k\} \cup \{y_m, y_{m-1}, \ldots, y_{k+1}\}$
- $y_k || x_{k-1}, x_{k-2}, \ldots, x_1$

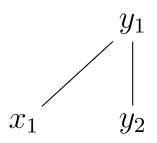
To understand this presentation, follow the introduction of a few G_m. The Spoiler starts by introducing $G_1 = \{x_1\}$

$$x_1$$

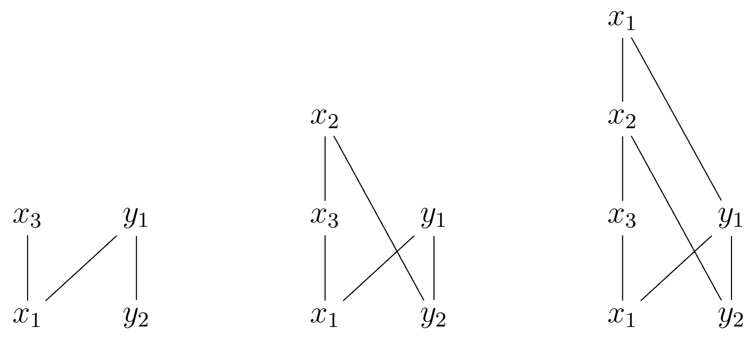
Then the Spoiler introduces the elements of $G_2 = \{y_2, y_1\}$ one at a time, first y_2, then y_1. Because $y_2|x_1$, the poset looks like

$$x_1 \qquad y_2$$

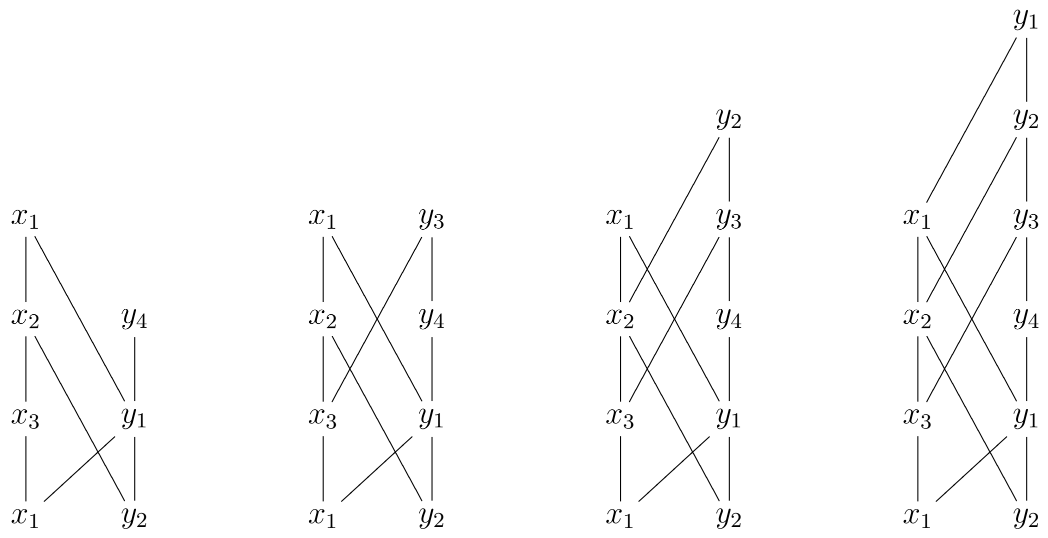
Then the Spoiler introduces y_1, which is comparable to y_2 and x_1 (since $y_k > x_k$). The poset is



Next the Spoiler introduces the elements of $G_3 = \{x_3, x_2, x_1\}$ one at a time.



Finally, the Spoiler introduces the elements of $G_4 = \{y_4, y_3, y_2, y_1\}$ one at a time.



Now go back and consider the colors that the Greedy Algorithm would assign when P is presented in this way.

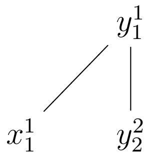
When the Spoiler introduces $G_1 = \{x_1\}$, the Greedy Algorithm will assign color 1 to x_1.

$$x_1^1$$

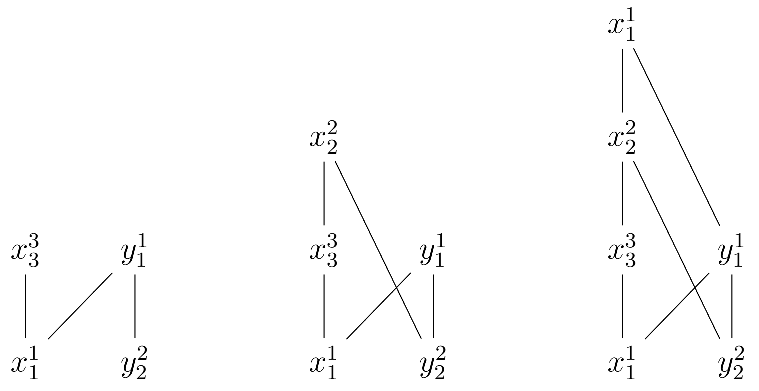
When the Spoiler introduces $G_2 = \{y_2, y_1\}$, the Greedy Algorithm will assign color 2 to y_2 because it is incomparable to x_1^1.

$$x_1^1 \qquad y_2^2$$

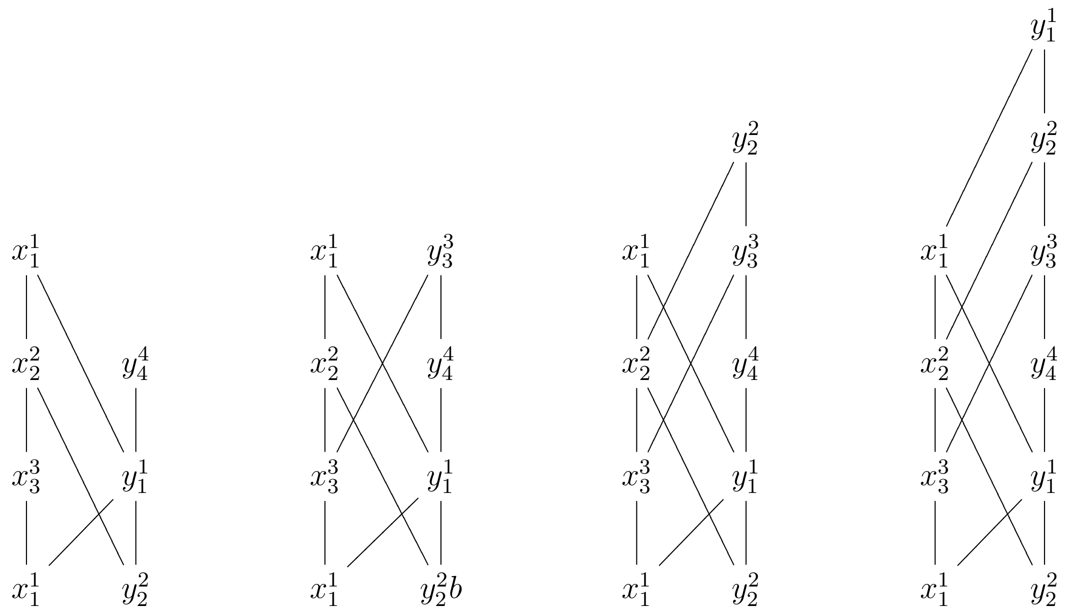
When the Spoiler introduces y_1, the Greedy Algorithm will color it 1. While y_1 is comparable to y_2^2 and x_1^1, 1 is the lower of the available colors.



When the Spoiler introduces the elements of $G_3 = \{x_3, x_2, x_1\}$, the Greedy Algorithm will assign 3 to x_3 because it is incomparable to y_2^2 and y_1^1. The algorithm will color x_2^2 and x_1^1.



Finally, the Spoiler introduces the elements of $G_4 = \{y_4, y_3, y_2, y_1\}$, which the Greedy Algorithm colors as $y_4^4, y_3^3, y_2^2, y_1^1$.



Stated more formally, the inverse numbering in each G_m, that $(g_m \leq g_{m-1} \leq \ldots \leq g_1)$, is intended to force the following behavior of the Greedy Algorithm:

$$c(g_k) = k \text{ for } k = 1 \text{ to } m$$

where $c(g_k)$ is the color of the g_k. So the Greedy Algorithm will assign color m to the m^{th} element of G_m, the behavior exhibited above. This can be proved in the general case by inducting on m. Certainly for G_1, $c(x_1) = 1$. Assume the behavior holds up to G_{m-1}. Then for G_m consisting of $y_m < \ldots < y_1$ and G_{m-1} consisting of $x_{m-1} < \ldots < x_1$, the fact that y_k is incomparable to $x_1, x_2, \ldots, x_{k-1}$ together with the behavior $c(g_k) = k$ for $k = 1 \ldots m-1$ means that the colors $1, 2, \ldots, k-1$ are unavailable to point y_k. However, since y_k is comparable to $G_1 \cup G_2 \cup \ldots \cup G_{m-2} \cup (G_{m-1} - \{x_1, x_2, \ldots, x_{k-1}\})$, color k is available to y_k. Since the Algorithm is greedy, $c(y_k) = k$.

5.2 Upper Bound for $(t+t)$-Free Posets

Theorem: For a $(t+t)$-free poset $(P, \leq)$ with finite width, there exists an upper bound on the number of colors that the Spoiler can force the Greedy Algorithm to use in on-line coloring poset P [1].

Proof: The proof of this theorem involves new notation and a lemma. The proof of the lemma is more involved that the proof of the theorem, so we will first explore the notation and lemma, discuss it with examples, and use it to prove the theorem. Later, the proof of the lemma will be presented using a specific poset.

For any $x, y \in P$,

$$\begin{aligned}(x,y] &= \{z \in P : x < z \leq y\}\\ [x,y] &= \{z \in P : x \leq z \leq y\}\\ [x,y) &= \{z \in P : x \leq z < y\}\\ (x,y) &= \{z \in P : x < z < y\}\end{aligned}$$

and for a chain C in P and points $x, y \in C$,

$$dist_C(x,y) = |[x,y] \cap C|$$

the number of points between and including x, y in C.

The Lemma

Lemma: Let the poset $(P \leq)$ be $(t+t)$-free and of width w. Then there exists a partition of P into sets* $P = P_{w-1} \cup P_{w-2} \cup \ldots \cup P_0$ such that for each $x \in P_l$,

$$|inc(x, P_l \cup P_{l-1} \cup \ldots \cup P_0)| \leq (4t-2)(w-1) - 2tl.$$

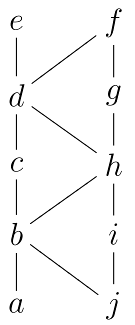
That is, the number of points incomparable to x in the sets including and above x's set is bounded above by $(4t-2)(w-1) - 2tl$ [1].

With that upper bound, the number of points incomparable to x decreases as x is of P_l with increasing l. Equivalently, those elements that are compared to more sets have fewer elements incomparable to them that those elements compared to a fewer number of sets. So likely the P_l for low l contain mostly points incomparable to one

*This partition is into sets not chains.

another or points incomparable to a high number of elements in the whole poset. As l increases, the sets P_l become increasingly chain-like.

Consider the following poset of width 2 and $(2+2)$-free.



By the lemma, there exists a partition of the poset into sets P_0 and P_1 such that,

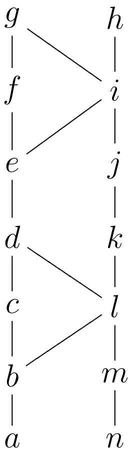
$$\begin{aligned} |inc(x,P_0)| &\leq (4t-2)(w-1)-2tl = (8-2)(1)-2\times 2\times 0 = 6 \text{ for } x \in P_0 \\ |inc(x,P_1 \cup P_0)| &\leq (4t-2)(w-1)-2tl = (8-2)(1)-2\times 2\times 1 = 2 \text{ for } x \in P_1. \end{aligned}$$

A simple way to create this partition is to place any elements incomparable to more than two other elements into P_0 and all other elements into P_1. The incomparabilities

point	a	b	c	d	e	f	g	h	i	j
$\|inc(x,P)\|$	2	1	3	1	2	1	3	1	3	1

could lead to the partition $P_0 = \{c,g,i\}$ and $P_1 = \{a,b,d,e,f,h,j\}$. In fact, any partition that has c,g,i in P_0 will meet the lemma's conclusions.

Consider also this example of a width 2, $(3+3)$-free poset.



For this poset, the partition is the sets P_0, P_1 with the incomparability requirements

$$\begin{aligned} |inc(x,P_0)| &\leq (4t-2)(w-1)-2tl = (12-2)(1)-2\times 3\times 0 = 10 \text{ for } x \in P_0 \\ |inc(x,P_1 \cup P_0)| &\leq (4t-2)(w-1)-2tl = (12-2)(1)-2\times 3\times 1 = 4 \text{ for } x \in P_1. \end{aligned}$$

Based on the incomparabilities within the poset

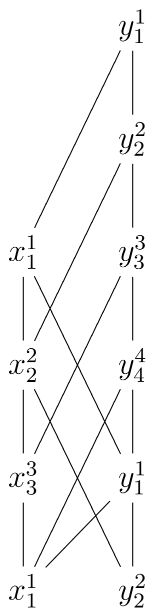
point	a	b	c	d	e	f	g	h	i	j	k	l	m	n
$\lvert inc(x, P)\rvert$	2	2	5	2	2	4	1	2	1	4	4	1	3	3

any partition that has c in P_0 would meet the lemma's conclusions.

To better understand the lemma, explore the condition that the poset be $(t+t)$-free. If a poset is not $(t+t)$-free, why is the lemma's conclusion not met? Consider the poset and its presentation used above to prove that the Spoiler can force the Greedy Algorithm to use infinitely many colors on a width 2 poset. In order to defy the lemma's conclusion, there must not exist a partition of P into sets P_1, P_0 such that

$$\begin{aligned} |inc(x, P_0)| &\leq (4 \times t - 2)(2-1) - 2 \times 2 \times 0 = 4t && \text{for } x \in P_0 \\ |inc(x, P)| &\leq (4 \times t - 2)(2-1) - 2 \times 2 \times 1 = 4t - 4 && \text{for } x \in P_1 \end{aligned}$$

First try introducing G_1, G_2, G_3, G_4.



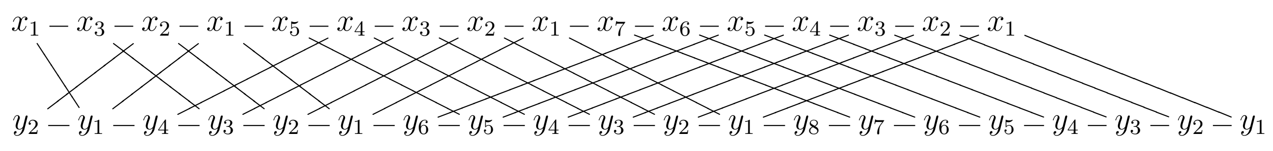
Here there is a (2+2) between x_3, x_2, y_1, y_4, but looking at the list of incomparabilities

point	x_1	x_3	x_2	x_1	y_2	y_1	y_4	y_3	y_2	y_1
$\lvert inc(x, P)\rvert$	1	3	3	3	2	2	3	2	1	0

there are not enough incomparabilities to violate the lemma's conclusion. Since all points are incomparable to fewer than 6 other points, they could all be put in P_0. To counter the lemma's conclusion, we need at least 1 element incomparable to 7 or more other elements.

Notice that the elements of G_1, G_2, G_3 above are incomparable to 1,2, and 3 elements, respectively; the elements of G_4 are incomparable to 3,2,1 other elements. This trend holds generally. That is, when the groups G_1 to G_m are introduced, the elements of G_k for $1 \leq k \leq m-1$ are incomparable to k elements. The elements g_i for $1 \leq i \leq m$ of G_m are incomparable to $i-1$ elements.

To get elements incomparable to 7 or more other elements, we must introduce groups at least up to G_8 (introducing only G_7 would give elements incomparable to 6 others in G_6 and elements with incomparabilities 6,5,4,3,2,1,0 in G_7). For the sake of space, $G_1 - G_8$ will be presented horizontally.


$x_1 - x_3 - x_2 - x_1 - x_5 - x_4 - x_3 - x_2 - x_1 - x_7 - x_6 - x_5 - x_4 - x_3 - x_2 - x_1$
$y_2 - y_1 - y_4 - y_3 - y_2 - y_1 - y_6 - y_5 - y_4 - y_3 - y_2 - y_1 - y_8 - y_7 - y_6 - y_5 - y_4 - y_3 - y_2 - y_1$

Notice that y_8 and the elements of G_7 are all incomparable to 7 elements. Specifically, y_8 is incomparable to every element in G_7. At most, y_8 and 6 of the elements of G_7 could be put in P_0. The seventh element would have to be put in P_1 and would be incomparable to 7 elements in P, violating the lemma's conclusion. If all seven of the G_7 elements were put in P_0, then y_8 would have to go in P_1, where is would still be incomparable to 7 elements in P.

Using the presentation of the G_m groups of a width 2 poset, the Spoiler would need to introduce G_1 to G_N where $N = 4t$ to ensure that the conclusion of the lemma cannot be met.

The Theorem

Theorem: The Greedy Algorithm uses at most $(3t-2)(w-1)w+w$ chains in a on-line partition of a $(t+t)$-free poset of width w [1].

Proof: Partition the poset according to the lemma and induct on s for $x \in P_{w-s}$. Let $M(w-s) = (4t-2)(w-1) - 2t(w-s)$, so M gives the number of points incomparable to a point in P_{w-s} as determined by the lemma. For $s = 1$, $x \in P_{w-1}$ so

$$|inc(x, P)| \leq M(w-1) = (4t-2)(w-1) - 2t(w-1)$$

by the lemma. In order for a color to be unavailable to x, it must have already been assigned to a point to which x is incomparable. Since x is incomparable to at most $M(w-1)$ points, there must exist a color in $[1, M(w-1)+1]$ that has not been used to color a point incomparable to x, so the Greedy Algorithm assigns to x a color no greater than $M(w-1)+1$.

Assume that the Greedy Algorithm used fewer than

$$(M(w-1)+1) + (M(w-2)+1) + \ldots + (M(w-(s-1))+1)$$

colors to color the points in $P_{w-1} \cup P_{w-2} \cup \ldots \cup P_{w-(s-1)}$. Then for $x \in P_{w-s}$, the colors forbidden to it must have been used to color those points incomparable to x in $P_{w-1} \cup P_{w-2} \cup \ldots \cup P_{w-(s-1)}$.

By induction, the Greedy Algorithm used no more than

$$[1, (M(w-1)+1) + (M(w-2)+1) + \ldots + (M(w-(s-1))+1)]$$

colors in coloring those points. And by the lemma, there are no more than $M(w-s)$ points incomparable to x in $P_{w-s} \cup P_{w-s-1} \cup \ldots \cup P_0$.

Therefore, the Greedy Algorithm assigns to x a color no greater than

$$(M(w-1)+1) + (M(w-2)+1) + \ldots + (M(w-(s-1))+1) + (M(w-s)+1).$$

To color all points in P, the Greedy Algorithm uses no more than

$$\begin{aligned}(M(w-1)+1) + \ldots + (M(0)+1) &= \sum_{l=0}^{w-1}((4t-2)(w-1) - 2tl + 1) \\ &= w(4t-2)(w-1) - 2t\sum_{l=0}^{w-1}(l) + w(1) \\ &= (4t-2)(w-1)w - 2t\frac{(w-1)w}{2} + w \\ &= (3t-2)(w-1)w + w\end{aligned}$$

colors.

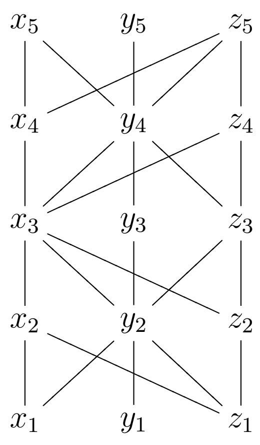
Returning to the Lemma: Proof

Lemma: Let the poset $(P \leq)$ be $(t+t)$-free and of width w. Then there exists a partition of P into sets $P = P_{w-1} \cup P_{w-2} \cup \ldots \cup P_0$ such that for each $x \in P_l$,

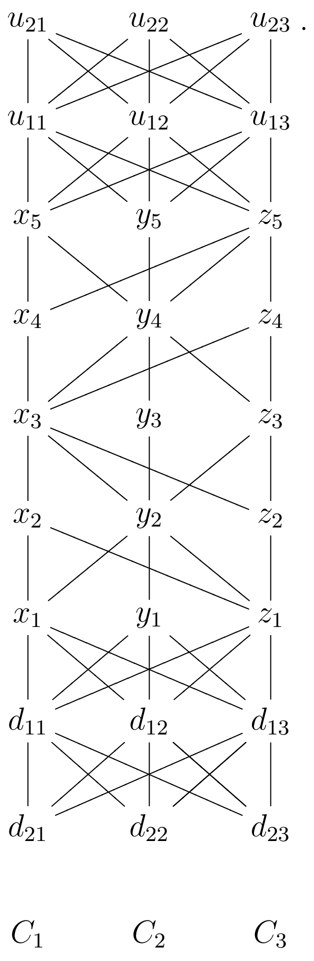
$$|inc(x, P_l \cup P_{l-1} \cup \ldots \cup P_0)| \leq (4t-2)(w-1) - 2tl.$$

The proof of this lemma is fairly involved, requiring the proof of 5 claims. We will cover the general proofs for each claim as presented in [1] and show how each claim can be applied to an example poset and hopefully better understood.

Consider the poset $P =$



P has width 3 and is $(2+2)$-free. Define P^* to be


u_{21} u_{22} u_{23} .
u_{11} u_{12} u_{13}
x_5 y_5 z_5
x_4 y_4 z_4
x_3 y_3 z_3
x_2 y_2 z_2
x_1 y_1 z_1
d_{11} d_{12} d_{13}
d_{21} d_{22} d_{23}
C_1 C_2 C_3

where the anti-chains U_1, U_2, D_1, D_2 were added so that $D_2 < D_1 < P < U_1 < U_2$. By Dilworth's theorem, there exists a partition of P^* into three chains, C_1, C_2, C_3.

For $x \in C_k$, define $\overline{x}, \underline{x} \in C_k$ and $\overline{x}_i, \underline{x}_i \in C_i$ for $i \neq k$ as:

- $dist(\underline{x}, x) = dist(x, \overline{s}) = t$
- $\underline{x}_i$ is the lowest point in C_i that is above $\underline{x}$

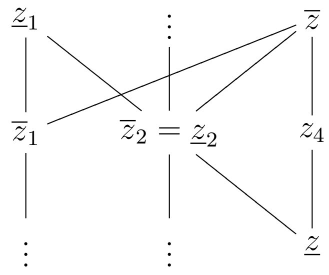
- $\overline{x}_i$ is the highest point in C_i that is below $\overline{x}$

So for $z_4 \in C_3$,

$$\begin{array}{lll} \overline{z} = z_5 & \overline{z}_1 = x_4 & \overline{z}_2 = y_4 \\ \underline{z} = z_3 & \underline{z}_1 = x_5 & \underline{z}_2 = y_4 \end{array}$$

Note that $\overline{x}_i, \underline{x}_i$ need not exist in P, but U_1, U_2, D_1, D_2 were introduced so that $\overline{x}_i, \underline{x}_i$ do exist in P^*.

We say that $x \in C_k$ *has a cross* on C_i for $i \neq k$ if and only if $\overline{x}_i < \underline{x}_i$. In the example poset, $z_4 \in C_3$ has a cross on C_1 but not on C_3 ($\overline{z}_2 = \underline{z}_2$, and the cross requires a strict $<$).



Claim: If we partition P^* according to the rule,

> put $x \in C_k$ in set P_l if the set $\{i : x \text{ has a cross on } C_i, 1 \leq i \leq w, i \neq k\}$ has exactly l elements

then this partition satisfies the lemma [1].

In the example P^*, the following elements have crosses:

element	x_1	x_2	x_3	x_4	x_5	y_1	y_2	y_3	y_4	y_5	z_1	z_2	z_3	z_4	z_5
has a cross on	C_2	C_3	C_2, C_3	C_3	C_2	$\emptyset$	C_1, C_3	$\emptyset$	C_1, C_3	$\emptyset$	C_1, C_2	C_1	C_2	C_1	C_1, C_2

So $P_0 = \{y_1, y_3, y_5\}, P_1 = \{x_1, x_2, x_4, x_5, z_2, z_3, z_4\}, P_2 = \{x_3, y_2, y_4, z_1, z_5\}$. This partition is such that

$$\begin{aligned} |inc(x, P_0)| &= 0 \text{ for } x \in P_0 \\ |inc(x, (P_1 \cup P_0)| &\leq 3 \text{ for } x \in P_1 \\ |inc(x, (P_2 \cup P_1 \cup P_0)| &\leq 2 \text{ for } x \in P_2 \end{aligned}$$

These counts meet the those of the lemma, which sets incomparability maxima of 12,8,4 respectively.

Proof: For a general poset, this lemma is proved by proving five subclaims, ABCDE.

Claim A: For any $x \in C_k$ and $y \in C_i$ with $i \neq k$, if $\overline{x}_i < y$ and $dist(\overline{x}_i, y) \geq t+1$, then $x < y$ [1].

Example: In our poset, consider $x_1 \in C_1$. $\overline{x} = x_2$ and $\overline{x}_2 = d_{12}$. Points y_2, y_3, y_4, y_5 (and u_{12}, u_{22}) satisfy $dist(\overline{x}_2, y) \geq t+1 = 3$. Indeed $x < y$ holds for all these y (also for u_{12}, u_{22}, but U_1, U_2 added so that $P < U_1 < U_2$).

Proof: It is sufficient to show that $x < y$ for any $y \in C_i$ such that $dist(\overline{x}_i, y) = t+1$. First assume that $x > y$. Then, in order to preserve the definition of $\overline{x}_i$ as the highest point in C_i below $\overline{x}$, $\overline{x}_i > y$. This contradicts $dist(\overline{x}_i, y) = t+1$.

Next, assume that x is incomparable to y. Define $z_1 \in ([x, \overline{x}] \cap C_k)$ and $z_2 \in ([\overline{x}_i, y] \cap C_i)$, and consider the pair (z_1, z_2). If $z_1 > z_2$, $\overline{x}_i$ is no longer the highest point in C_i below $\overline{x}$. If $z_1 < z_2$, then $x < y$, so both cases lead to contradiction.

Both assumptions are invalid, so $x < y$ for any $y \in C_i$ with $\overline{x}_i < y$.

Claim B: For any $x \in C_k$ and $y \in C_i$ with $i \neq k$, if $y < \underline{x}_i$ and $dist(y, \underline{x}_i) \geq t+1$, then $y < x$ [1].

Example: In our poset, consider $y_5 \in C_2$. Then $\underline{y} = y_4$ and $\underline{y}_1 = x_3$. $x_1 < x_3$ and $dist(x_1, x_3) = 3$. We can see that $x_1 < y_5$.

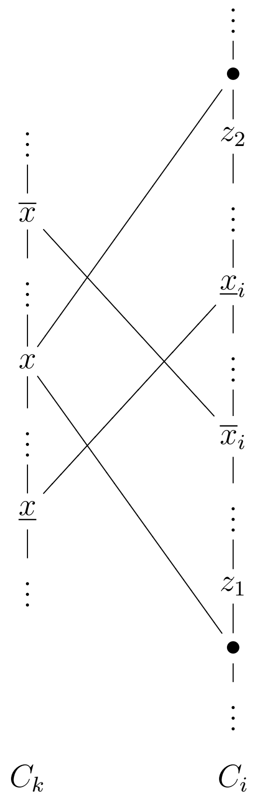
Proof: The proof of Claim B is symmetric to that of Claim A.

Claim C: If $x \in C_k$ has a cross on C_i, then

$$|inc(x, C_i)| \leq 2(t-1)[1]$$

Example: Consider $x_2 \in C_1$. Overall, x_2 is incomparable to five elements $\{y_1, y_2, y_3, z_2, z_3\}$. x_2 has a cross on C_3 and, indeed, is incomparable to only two elements in C_3.

Proof: Define $z_1 \in C_i$ as the lowest element incomparable to x and $z_2 \in C_i$ as the highest element incomparable to x (see diagram).



Since x has a cross on C_i, $z_1 < \overline{x}_i < \underline{x}_i < z_2$. Yet $z_1|x$ and $x|z_2$, so by Claims A and B, $dist(z_1, \underline{x}_i)$ and $dist(\overline{x}_i, z_2)$ are less than $t+1$. Therefore, the sets $[z_1, \underline{x}_i) \cap C_i$ and $(\overline{x}_i, z_2] \cap C_i$ each have at most $t-1$ elements. Also, note that

$$[z_1, z_2] \cap C_i = \{[z_1, \underline{x}_i) \cup (\overline{x}_i, z_2]\} \cap C_i = \{[z_1, \underline{x}_i) \cap C_i\} \cup \{(\overline{x}_i, z_2]) \cap C_i\}$$

since $\overline{x}_i < \underline{x}_i$ (x has a cross on C_i). Since the elements in C_i incomparable to x are contained within $[z_1, z_2]$, the above equalities give

$$|inc(x, C_i)| = |\{[z_1, \underline{x}_i) \cap C_i\} \cup \{(\overline{x}_i, z_2]) \cap C_i\}| \leq 2(t-1).$$

Claim D: If $x \in C_k$ does not have a cross on C_i for $i \neq k$, then for any y such that $dist(z_1, y) > 2t-1$ and $dist(y, z_2) > 2t-1$,

1) if x has a cross of C_j for $j \neq k$ and $j \neq i$, then y has a cross on C_j.

2) y has a cross on C_k. [1]

Example: Consider $x_1 \in C_1$. x_1 does not have a cross on C_3. z_1 looks to be in the appropriate position, and z_1 has a cross on C_1.

Proof: Define z_1 and z_2 as in Claim C. Note that, since $z_2|x$ and $z_2 > \overline{x}_i$, $dist(\overline{x}_i, z_2) < t+1$ by Claim A. Also, since $z_1|x$ and $\underline{x}_i > z_1$, $dist(z_1, \underline{x}_i) < t+1$ by Claim B. Since y was chosen to be at least $2t$ elements away from z_1 and z_2, the points $\overline{y}, \underline{y}$ (the points t away from y) follow $\overline{y} \leq \overline{x}_i$ and $\underline{x}_i \leq \underline{y}$.

D1). Assume that y does not have a cross on C_j, that is, $\underline{y}_j \leq \overline{y}_j$. By transitivity ($\overline{x} > \overline{x}_i \geq \overline{y} > \overline{y}_j$ and $\underline{x} < \underline{x}_i \leq \underline{y} < \underline{y}_j$), $\overline{x} > \overline{y}_j$ and $\underline{x} < \underline{y}_j$. Since $\overline{x}_j$ and $\underline{x}_j$ are defined to be the highest (lowest) points in C_j below (above) $\overline{x}$ and $\underline{x}$, it follows that $\overline{x}_j \geq \overline{y}_j$ and $\underline{x}_j \leq \underline{y}_j$. Therefore, $\underline{x}_j \leq \underline{y}_j < \overline{y}_j \leq \overline{x}_j$, indicating that $\underline{x}_j < \overline{x}_j$. This contradicts the given that x has a cross on C_j, so y must also have a cross on C_j.

D2). Note that $\overline{y}$ and $\underline{y}$ are within $[z_1, z_2]$, so $\overline{y}|x$ and $\underline{y}|x$. These relations indicate that $\overline{y}_k < x$ and $x < \underline{y}_k$. Therefore, $\overline{y}_k < \underline{y}_k$, and y has a cross on C_k.

Claim E: If $x \in P_l$ does not have a cross on C_i, then

$$|inc(x, (P_l \cup \ldots \cup P_0) \cap C_i)| \leq 4t - 2[1]$$

Example: $x_1 \in P_1$ does not have a cross on C_3. $(P_1 \cup P_0) \cap C_3 = \{z_2, z_3, z_4\}$. Of these points, x_1 is incomparable only to z_2, certainly fewer than $4(2) - 2 = 6$ elements.

Proof: $x \in P_l$ and does not have a cross on C_i, so by the claims D1 and D2, any point y in C_i further than $2t - 1$ from both z_1 and z_2 will have at least $l + 1$ crosses. By D1, y has as many crosses as x, and by D2, y has at least one more cross than x in that y has a cross on x's chain. Thus any such y lies in $P_{l+1} \cup \ldots \cup P_{w-1}$. So the points in C_i incomparable to x from $P_l \cup \ldots \cup P_0$ fall within $2t - 1$ of either z_1 or z_2. There are $4t - 2$ such points.

Now we can put all these claims together to prove the lemma.

Lemma: Let the poset $(P \;\leq)$ be $(t + t)$-free and of width w. Then there exists a partition of P into sets $P = P_{w-1} \cup P_{w-2} \cup \ldots \cup P_0$ such that for each $x \in P_l$,

$$|inc(x, P_l \cup P_{l-1} \cup \ldots \cup P_0)| \leq (4t - 2)(w - 1) - 2tl[1]$$

Proof: Consider x in P_l and in C_k. Let $i_1 = k$. Then x has a cross on C_{i_j} if and only if $2 \leq j \leq l + 1$. Since $x \in P_l$, it must have a cross on exactly l chains, and it cannot have a cross on its own chain. Then

$$\begin{aligned}|inc(x, P_l \cup \ldots P_0)| \leq{}& |inc(x, (P_l \cup \ldots P_0) \cap (C_{i_2} \cup \ldots \cup C_{i_{l+1}}))| \\ &+ |inc(x, (P_l \cup \ldots P_0) \cap (C_{i_{l+2}} \cup \ldots \cup C_{i_w}))| \text{ (the triangle inequality)}\end{aligned}$$

The first term counts the incomparabilities in the chains on which x has a cross. By Claim C,

$$|inc(x, C_i)| \leq 2(t - 1)$$

when x has a cross on C_i, and since there are l such C_i,

$$|inc(x, (P_l \cup \ldots P_0) \cap (C_{i_2} \cup \ldots \cup C_{i_{l+1}}))| \leq 2(t - 1)l$$

The second term counts the incomparabilities in the chains on which x does not have a cross. By Claim E,

$$|inc(x, C_i)| \leq 4t - 2$$

when x does not have a cross on C_i, and since there are $w - l - 1$ of these C_i,

$$|inc(x, (P_l \cup \ldots P_0) \cap (C_{i_{l+2}} \cup \ldots \cup C_{i_w}))| \leq (4t - 2)(w - l - 1)$$

Substituting these two values back into the inequality above yields

$$\begin{aligned} |inc(x, P_l \cup \ldots P_0)| &\leq 2(t-1)l + (4t-2)(w-l-1) \\ &= (4t-2)(w-1) - 2tl \end{aligned}$$

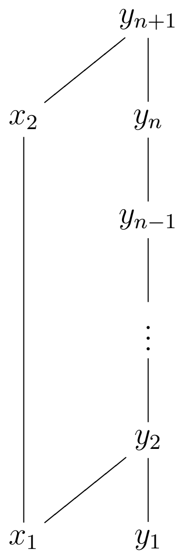
Chapter 6

Tight Bound on Incomparabilities in Graded, (t+t)-free Posets

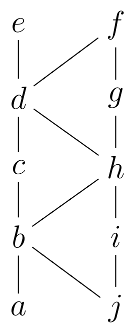
Notice that in exploring the previous lemma, none of the example posets contained a point with the maximum number of incomparabilities. No point had exactly the maximum incomparabilities allowed for a point in P_0. How many incomparabilities could a point have in a poset with a finite width and $(t+t)$-free?

It is important to establish that, within a given chain, the set of points incomparable to given point, x_0, must appear in a continuous interval. If $x_0 < y$ for any y, then x_0 is comparable to all points above y by transitivity. Similarly if $z < x_0$ for any z, then x_0 is comparable to all points below z by transitivity. Thus the points to which x_0 is incomparable must form continuous intervals within their respective chains.

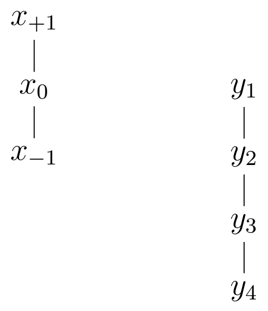
A trivial way of allowing a point be incomparable to arbitrarily many points is to build a non-graded poset. The poset below is width 2, $(2+2)$-free, and x_2 can be incomparable to n elements for any n in $\mathbb{N}$.



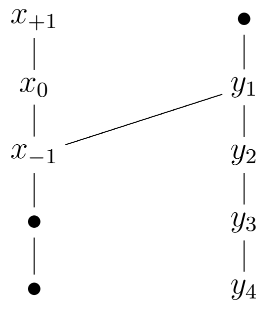
So to ask a better question, what is the largest number of points that can be incomparable to any given point in a graded poset of width 2 and $(t+t)$-free? Start with a graded poset of width 2 and $(2+2)$-free. In the poset below, points c, g, i are incomparable to 3 points. Can a point be incomparable to 4 points while still maintaining the poset as $(2+2)$-free?



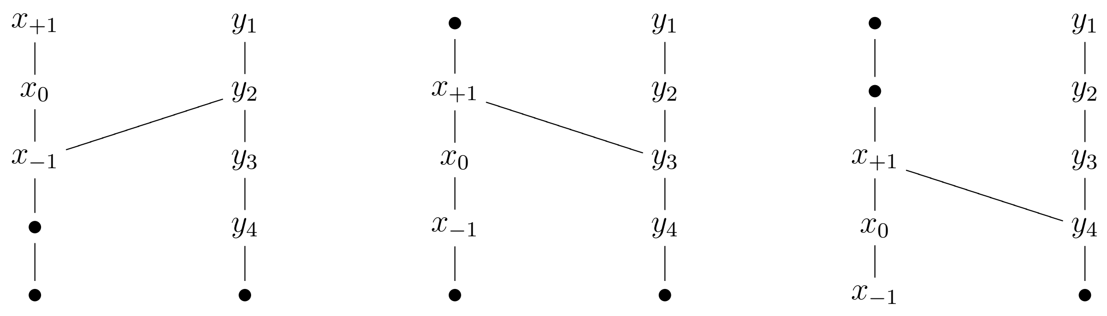
Suppose a point, x_0, is incomparable to 4 other points, y_1 to y_4. Certainly the point directly across from x_0 is one of those points. Suppose all the other y are below the level of x_0, so the points to consider are



where x_{+1}, x_{-1} are the points directly above and below x_0 in its chain. For x_0 to remain incomparable to y_4, x_{-1} cannot be comparable to y_3 or y_4. If x_{-1} is comparable to any of the y, it must only be comparable to y_1.



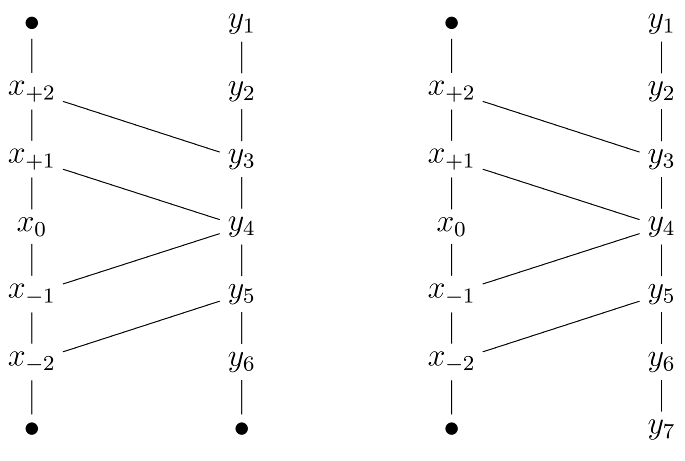
This set of comparabilities induces a $(2+2)$ between y_3, y_2 and x_{-1}, x_0, no matter the comparability of x_{+1} to any of the y's. Other possible arrangements for the four y are



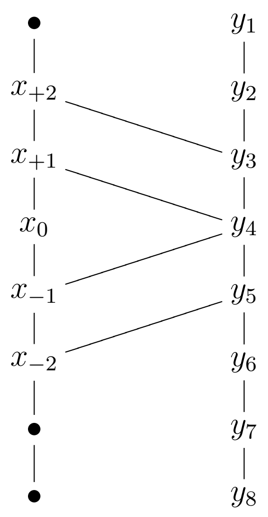
where, for the second and third in this group, x_{+1} cannot be comparable to y_1 or y_2.

There are always at least two y either below or above the level of x_0. If below, x_{-1} is incomparable to at least two of the y, creating a $(2+2)$ with x_0, x_{-1} and those two y. If above, x_{+1} is incomparable to at least two of the y, creating a $(2+2)$ with x_0, x_{+1} and those two y. No matter the orientation, there cannot be a graded, width-2, $(2+2)$-free poset containing an element incomparable to 4 elements. The maximum number of incomparabilities is 3.

Now consider a graded, width 2, $(3+3)$-free poset and evaluate the comparability relationships of x_{-2}, x_{-1} and x_{+1}, x_{+2}. To minimize the number of y incomparable to the x_-'s and x_+'s, the points should lay roughly symmetrically around x_0.

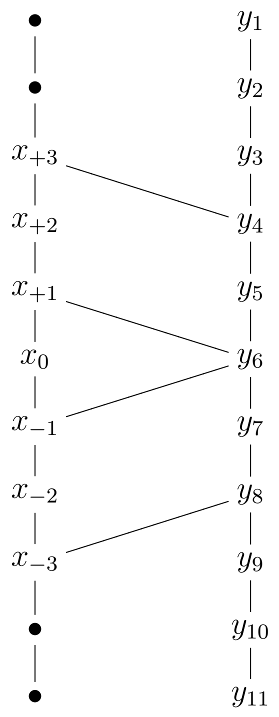


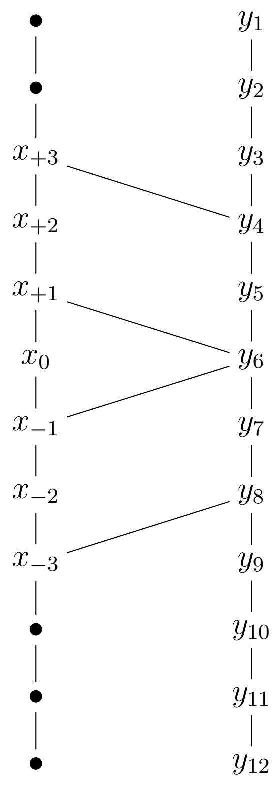
By the above diagrams, 6 and 7 are allowable incomparability numbers.



However when x_0 is incomparable to 8 points, a $(3+3)$ arises from x_{-2}, x_{-1}, x_0 and y_8, y_7, y_6. Notice that, had this 8th incomparable element been placed above y_1, a $(3+3)$ would have arisen between $x_0, x_{+1}x_{+2}$ and y_2, y_1, y_8. So the maximum allowed number of incomparabilities to a single element in a graded, width 2, $(3+3)$-free poset is 7.

On a graded, width 2, $(4+4)$-free poset, 11 incomparabilities are allowed but not 12. With 12 points incomparable to x_0, a $(4+4)$ arises between $x_{-3}, x_{-2}, x_{-1}, x_0$ and $y_{12}, y_{11}, y_{10}, y_9$.





This bound can be established generally.

Theorem: Let $(P, \leq)$ be a graded, $(t+t)$-free poset of finite width w. Then for all x_0 in P,

$$|inc(x_0, P)| \leq (4t-5)(w-1)$$

where $inc(x, p)$ is the set of points in P incomparable to x_0.

Proof: Induct on the width, w, of the poset.

Case: Let P_1 be any graded, $(t+t)$-free poset of width $w = 1$. Since the width is one, this poset is a linear order. Since every point x_0 in P_1 is comparable to every other point,

$$|inc(x_0, P)| = 0 \leq (4t-5)(1-1) = 0$$

for all x_0 in P_1.

Case: Let P_2 be any graded, $(t+t)$-free poset of width $w = 2$.

Claim: In P_2, a given point x_0 cannot be incomparable to $4t-4$ or more elements. That is, x_0 cannot be incomparable to $y_{+(2t-2)}$ or $y_{-(2t-2)}$.

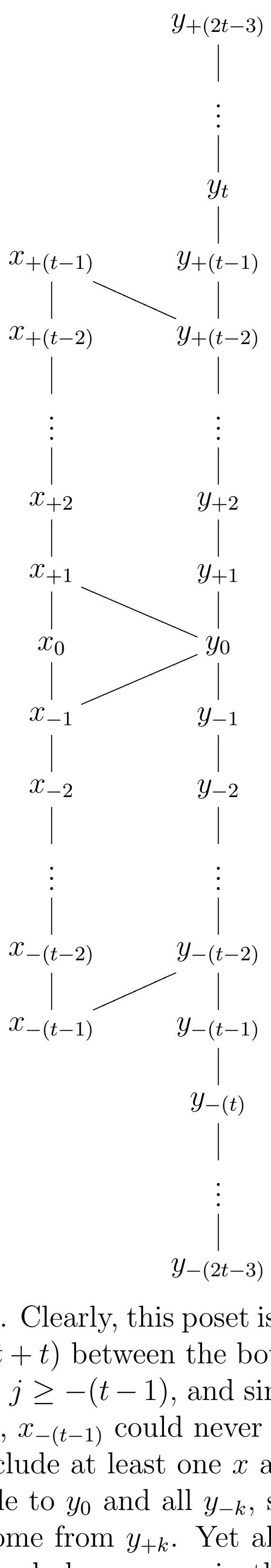
Proof: Assume x_0 is incomparable to $y_{+(2t-2)}$, that is, the point $2t-2$ levels above the level of x_0.

$$\begin{tikzcd}[row sep=small]
 & y_{+(2t-2)} \arrow[d, dash] \\
 & y_{+(2t-3)} \arrow[d, dash] \\
 & \vdots \arrow[d, dash] \\
 & y_{+t} \arrow[d, dash] \\
x_{+(t-1)} \arrow[d, dash] & y_{+(t-1)} \arrow[d, dash] \\
x_{+(t-2)} \arrow[d, dash] & y_{+(t-2)} \arrow[d, dash] \\
\vdots \arrow[d, dash] & \vdots \arrow[d, dash] \\
x_{+2} \arrow[d, dash] & y_{+2} \arrow[d, dash] \\
x_{+1} \arrow[d, dash] & y_{+1} \arrow[d, dash] \\
x_0 \arrow[d, dash] & y_0 \arrow[d, dash] \\
x_{-1} \arrow[d, dash] & y_{-1} \arrow[d, dash] \\
x_{-2} \arrow[d, dash] & y_{-2} \arrow[d, dash] \\
\vdots \arrow[d, dash] & \vdots \arrow[d, dash] \\
x_{-(t-2)} \arrow[d, dash] & y_{-(t-2)} \arrow[d, dash] \\
x_{-(t-1)} & y_{-(t-1)} \arrow[d, dash] \\
 & y_{-(t)} \arrow[d, dash] \\
 & \vdots \arrow[d, dash] \\
 & y_{-(2t-3)} \arrow[d, dash] \\
 & y_{-(2t-2)}
\end{tikzcd}$$

In order to keep x_0 incomparable to all y above the level of x_0, none of the x_+ can be comparable to any y between its own level and $y_{+(2t-1)}$. That is, they can't be comparable up. This restriction induces a $(t+t)$ between the size t chains $y_{+(t-1)}$ to $y_{+(2t-2)}$ and x_0 to $x_{+(t-1)}$.

Similarly, x_0 could not be incomparable to $y_{-(2t-2)}$ without inducing a $(t+t)$ between the size t chains $y_{-(t-1)}$ to $y_{-(2t-2)}$ and x_0 to $x_{-(t-1)}$.

Claim: In P_2, a given point x_0 can be incomparable to $4t-5$ elements.



Proof: Consider the above poset. Clearly, this poset is still width 2. Because $x_{-(t-1)} < y_{-(t-2)}$, there does not exist a $(t+t)$ between the bottom t y's and the bottom t x's. In fact, since $x_{-(t-1)} < y_j$ for all $j \geq -(t-1)$, and since a $(t+t)$ requires at least one y at or above the $-(t-2)$ level, $x_{-(t-1)}$ could never be used in a $(t+t)$.

An interval of t x's must include at least one x at or above the level of x_{+1}. Yet all x_{+k} for $k \geq 1$ are comparable to y_0 and all y_{-k}, so a $(t+t)$ could not include y_0 or y_{-k}. All t of the y's must come from y_{+k}. Yet all x_{-k} are comparable to y_0 and all y_{+k}, so a $(t+t)$ could not include any x_{-k} in the $(t+t)$. Thus all t of the x's must be x_0 to $x_{+(t-1)}$. Notice, however, that all y_{+m} for $m \leq (t-2)$ are comparable to $x_{+(t-1)}$, so none of those y could be included in the $(t+t)$. There are only $t-1$

remaining y from $y_{+(t-1)}$ to $y_{+(2t-3)}$, not enough to build a $(t+t)$ on this poset.

Thus x_0 can be incomparable to $4t-5$ elements, the y from $y_{-(2t-3)}$ to $y_{+(2t-3)}$, which include y_0.

Inductive Hypothesis: Let P_{w-1} be any graded, $(t+t)$-free poset of width $w-1$. Assume the upper bound holds for P_{w-1}, that is

$$|inc(x_0, P_{w-1})| \leq (4t-5)(w-2)$$

for all x_0 in P_{w-1}.

Let P_w be any graded, $(t+t)$-free poset of width w. Want to show that for any x_0 in P_w, the upper bound

$$|inc(x_0, P_w)| \leq (4t-5)(w-1)$$

is obeyed.

Proof: By Dilworth's Theorem, P_w can be partitioned into w distinct chains. Arbitrarily select $w-1$ of those chains. They form a P_{w-1}, so by the inductive hypothesis, any point x_0 in the $w-1$ chains is incomparable to at most $(4t-5)(w-2)$ other points in those $w-1$ chains. Now consider the incomparabilities between any x_0 in the $w-1$ chains and the wth chain of the partition. The chain containing x_0 forms a P_2 with the wth chain. By the case above, x_0 is incomparable to at most $4t-5$ points in this P_2. In P_w, then, x_0 from the $w-1$ chains is incomparable to at most $(4t-5)(w-2)+4t-5 = (4t-5)(w-1)$ points.

For any x_0 in the wth chain, consider the $w-1$ P_2's formed between the wth chain and the $w-1$ chains selected from the partition. Since x_0 in a P_2 is incomparable to at most $(4t-5)$ points, x_0 in the wth chain is incomparable to at most $(4t-5)(w-1)$ points in P_w.

Then, for any x_0 in P_w, x_0 is incomparable to at most $(4t-5)(w-1)$ other points.

Bibliography

[1] Bartlomiej Bosek, Tomasz Krawczyk, and Edward Szczypka, *First-fit algorithm for the on-line chain partitioning problem.* SIAM J. Discrete Math, Vol. 23, No. 4, pp. 1992-1999, 2010.

[2] Stefan Felsner, *On-line chain partitions of orders.* Theoretical Computer Science, Vol. 175 pp. 283-292, 1997.

[3] Henry A. Kierstead, *An effective version of Dilworth's theorem.* Transactions of the American Mathematical Society, Vol. 268 No. 1 pp. 63-77, 1981.