On-line Adaptive Chain Covering of Upgrowing Posets

Bartlomiej Bosek, Piotr Micek

On-line Adaptive Chain Covering of Upgrowing Posets

Bartłomiej Bosek[1] and Piotr Micek[2†]

Algorithmics Research Group
Faculty of Mathematics and Computer Science
Jagiellonian University
Gronostajowa 3, 30-387 Kraków, Poland
[1]*Bartlomiej.Bosek@tcs.ii.uj.edu.pl,* [2]*Piotr.Micek@tcs.ii.uj.edu.pl*

We analyze on-line chain partitioning problem and its variants as a two-person game. One person (Spoiler) builds an on-line poset presenting one point at time. The other one (Algorithm) assigns new point to a chain. Kierstead gave a strategy for Algorithm showing that width w posets can be on-line chain partitioned into $\frac{5^w-1}{4}$ chains. Felsner proved that if Spoiler presents an upgrowing poset, i.e., each new point is maximal at the moment of its arrival then there is a strategy for Algorithm using at most $\binom{w+1}{2}$ chains and it is best possible. An adaptive variant of this problem allows Algorithm to assign to the new point a set of chains and than to remove some of them (but not all) while covering next points. Felsner stated a hypothesis that in on-line adaptive chain covering of upgrowing posets Algorithm may use smaller number of chains than in non-adaptive version. In this paper we provide an argument suggesting that it is true. We present a class of upgrowing posets in which Spoiler has a strategy forcing Algorithm to use at least $\binom{w+1}{2}$ chains (in non-adaptive version) and Algorithm has a strategy using at most $O(w\sqrt{w})$ chains in adaptive version.

Keywords: on-line algorithm, chain partition, order, poset

Throughout this paper, we consider a *poset* (partially ordered set) $\mathcal{P}$ to be a pair $(P, \leqslant)$ where P is a set and $\leqslant$ is an order on P. We say that x is above y and y is below x when $x \leqslant y$. Points x, y are *comparable* in P when either $x \leqslant y$ or $y \leqslant x$ and we denote it by $x \geqslant y$. Otherwise, we say x and y are *incomparable* and write $x \parallel y$. For $X \subseteq P$ by *closed/open upset* of X in $\mathcal{P}$ we mean

$$\begin{aligned} X{\Uparrow_{\mathcal{P}}} &:= \{p \in P : \text{ there is } x \in X \text{ so that } x \leqslant p\}, \\ X{\uparrow_{\mathcal{P}}} &:= \{p \in P : \text{ there is } x \in X \text{ so that } x < p\}. \end{aligned}$$

Dually we define *downsets* $X{\Downarrow_{\mathcal{P}}}$ and $X{\downarrow_{\mathcal{P}}}$ of X in $\mathcal{P}$. If $X = \{x\}$, we prefer to write $x{\uparrow_{\mathcal{P}}}$ instead of $\{x\}{\uparrow_{\mathcal{P}}}$. The reference to the poset $\mathcal{P}$ is often omitted when $\mathcal{P}$ is clear from the context.

[†]The authors are a scholar of the project which is co-financed from the European Social Fund and national budget in the frame of The Integrated Regional Operational Programme.

A subset $\alpha \subseteq P$ is a *chain* in $\mathcal{P}$ if each two points in α are comparable in $\mathcal{P}$. Dually, $A \subseteq P$ is an *antichain* in $\mathcal{P}$ if each two distinct points are incomparable. The *width* of X, denoted by *width*(X), is the maximum size of an antichain in $X \subseteq P$. We say that $\alpha_1, \ldots, \alpha_n$ forms a *chain partition* of $\mathcal{P}$ if all α_i's are chains, $\alpha_1 \cup \ldots \cup \alpha_n = P$ and $\alpha_i \cap \alpha_j = \emptyset$ for $i \neq j$. The following, Dilworth's theorem connects the width of the poset with the minimal size of its chain partition: if $\mathcal{P} = (P, \leqslant)$ is a poset and *width*$(P) = w$, then there exists a chain partition $\alpha_1, \ldots, \alpha_w$ of $\mathcal{P}$.

An on-line chain partitioning algorithm receives as input an on-line poset. This means that the elements of the poset are taken one by one from some externally determined list. Being presented with a new element the algorithm learns the comparability status of previously presented elements to the new one. Based on this knowledge the algorithm assigns the new element to a chain in an irrevocable manner. The performance of an on-line chain partitioning algorithm is measured by comparing the number of chains used with the minimal size of a chain partition, i.e., with the width of the poset.

All on-line problems considered in this paper can be viewed as two-person games. We call the players Algorithm and Spoiler. Algorithm represents an on-line algorithm and Spoiler represents an adversary. The game is played in rounds. During each round Spoiler introduces a new point x to the poset and describes comparabilities between x and the points from all previous rounds. Algorithm responds by assigning x to a chain. The most important feature of on-line games is that Algorithm's previous moves (decisions) restrict his actual possibilities. For simplicity it is convenient to identify the chain partition generated by Algorithm with a coloring. In this terms Algorithm colors the points presented by Spoiler and each mono-colored set must be a chain.

The value of the on-line chain partitioning problem for width w posets is the largest integer *CP*(w) so that there is a strategy of presenting points that forces any algorithm to use *CP*(w) colors. Note that we may as well define *CP*(w) as the least integer so that there is an algorithm that never uses more colors. An unpublished argument of Szemerédi proves lower bound and Kierstead [3] showed upper bound in the following:

$$\binom{w+1}{2} \leqslant \mathit{CP}(w) \leqslant \frac{5^w - 1}{4}.$$

This is already a complicated result, and no progress has been made on this problem for the last 20 years. Felsner [2] introduced a variant of the on-line chain partitioning problem. He restricts possible inputs by the rule that the sequence in which elements are released is a linear extension of the poset, i.e., a comparability of a new element x to the former y's has to be of the form $y < x$. In other words, each new point is maximal at the moment of its arrival. On-line posets with this property are called upgrowing posets and the value of the chain partitioning problem of upgrowing width w posets, defined similarly to the previous one, is denoted by *CPU*(w). Felsner [2] proved that

$$\mathit{CPU}(w) = \binom{w+1}{2}. \tag{1}$$

This paper focuses on the on-line adaptive chain covering problem. We describe this problem in terms of an on-line game. During each round Spoiler, as previously, introduces a new point x and describes comparabilities between x and points already presented. Algorithm responds by assigning to x a non-empty set of colors $c(x)$. Moreover, Algorithm may remove some colors from $c(x)$ in subsequent moves but keeping $c(x) \neq \emptyset$. Additionally, for each color γ the set of points colored with γ forms a chain. Let *ACCU*(w) be the value of the on-line adaptive chain covering problem for width w, upgrowing posets.

Authors proved in [1] that

$$\limsup_{w\to\infty} \frac{ACCU(w)}{w} \geqslant 2.$$

Felsner stated in [2] a hypothesis that $ACCU(w)$ is substantially smaller than $CPU(w) = \binom{w+1}{2}$. In other words he hypothesized that the possibility of covering by sets of colors and afterwards removing some of them adaptively to Spoiler's moves helps Algorithm to use smaller number of colors.

We provide an argument suggesting that this hypothesis is true. We present a class of upgrowing posets on which Spoiler may force Algorithm to use $\binom{w+1}{2}$ colors (in non-adaptive version) on width w posets from this class. Thus, by (1) as much as on the whole class of upgrowing posets. On the other hand there will be presented a strategy for Algorithm using $\frac{2\sqrt{2}}{3}w\sqrt{w} + O(w)$ colors on width w posets from that class in adaptive version. To define our class recall one more classical theorem of Dilworth. The set of maximum (with respect to the size) antichains of the poset $\mathcal{P} = (P, \leqslant)$ forms a lattice. The order relation of this lattice is given by $A \leqslant B$ for maximum antichains A, B of $\mathcal{P}$ iff $B \subseteq A{\Uparrow_{\mathcal{P}}}$. We will use the notation $HMA(P)$ to denote the *highest maximum antichain* of $\mathcal{P}$, i.e., $HMA(P)$ is the unique maximal element of the lattice of the maximum antichains in $\mathcal{P}$. The poset $\mathcal{P}$ is said to be *flat* if $HMA(P)\uparrow$ is an antichain. Consider an on-line chain partitioning of upgrowing flat posets as a game. Spoiler inroduces new point x in an upgrowing way and keeps invariant the condition that the currently presented poset is flat. Algorithm responds by assigning to x a color in such a way mono-colored sets must form chains. Let $CPUF(w)$ be the value of this game for width w posets. Consider also an adaptive variant in which Algorithm assigns to x a non-empty set of colors and it may remove some colors from $c(x)$ in subsequent moves but keeping $c(x) \neq \emptyset$ invariant. Moreover, for every γ the set $\{x : \gamma \in c(x)\}$ must form a chain. Let $ACCUF(w)$ be the value of the on-line adaptive chain covering problem for width w, upgrowing posets. This paper contains the proofs of following theorems.

Theorem 1

$$CPUF(w) = \binom{w+1}{2}$$

Theorem 2

$$ACCUF(w) \leqslant \frac{2\sqrt{2}}{3}w\sqrt{w} + O(w)$$

Proof of Theorem 1: Obviously, $CPUF(w) \leqslant CPU(w) = \binom{w+1}{2}$. The proof of the lowerbound is a modification of Felsner's argument proving (1). We show a strategy for Spoiler forcing Algorithm to use at least $\binom{w+1}{2}$ colors on upgrowing flat width w poset. For a color (chain) α we define $top(\alpha)$ as the maximal element in α. If x is a maximal element of a poset partitioned into chains $private(x)$ is the set of colors α with $top(\alpha) \leqslant x$ and $top(\alpha) \not\leqslant y$ for all maximal elements $y \neq x$. We prove that for all $k \geqslant 1$ there is a strategy for Spoiler S_k such that

- first k points form an antichain and then the width of the poset remains k through the whole game,
- at each round the poset of the game is flat,
- S_k generates a poset with k maximal points $x_1, \ldots, x_k$ and $|private(x_i)| \geqslant i$.

As the sets *private*(x_i) are pairwise disjoint, the theorem is an immediate consequence of the existence of S_k's. Strategy S_1 is trivial. Spoiler presents only a single point and any assignment of a color to this point leads to the desired situation. The sructure S_{k+1}, constructed by induction, consists of following phases:

1 Spoiler presents an antichain with $k+1$ points $x_1, \ldots, x_{k+1}$.

2 Spoiler runs S_k and each point presented is above $x_1, \ldots, x_k$ and not above x_{k+1}. After this phase (by induction hypothesis) there are $k+1$ maximal points $y_1, \ldots, y_k, x_{k+1}$ and $|\mathit{private}(y_i)| \geqslant i$.

3 Now, Spoiler again runs S_k and each point presented is above $y_1, \ldots, y_{k-1}$, and x_{k+1}. After this phase there are again $k+1$ maximal points $z_1, \ldots, z_k, y_k$ and $|\mathit{private}(z_i)| \geqslant i$, $|\mathit{private}(y_k)| \geqslant k$.

4 Spoiler puts new point u above all already presented points. Algorithm colors this point with $c(u)$. Obviously, $c(u)$ may not belong to both *private*(y_k) and *private*(z_k). We assume that $c(u) \notin$ *private*(z_k), otherwise interchange the role of y_k and z_k in the remainder of the argument. Lastly, Spoiler runs S_k and each point presented is above $z_1, \ldots, z_{k-1}, y_k$. After all we obtain $k+1$ maximal points $w_1, \ldots, w_k, u$ and $|\mathit{private}(w_i)| \geqslant i$, $|\mathit{private}(u)| \geqslant k+1$.

It is easy to check that S_{k+1} generates width $k+1$ poset. To prove that generated poset is flat at each round of the game consider $\mathcal{P} = (P, \leqslant)$ as the poset after i-th round. If i-th round is performed in 1-st phase then $\mathcal{P}$ is an antichain which is clearly flat. In the case of 2-nd phase, S_k is runned generating $\mathcal{Q}$ a subposet of $\mathcal{P}$. By the induction hypothesis we know that $\mathcal{Q}$ is an antichain or *width*$(Q) = k$. In the first case $\mathcal{P}$ is clearly flat. Thus, suppose *width*$(Q) = k$. In this setting one can prove that *HMA*$(P) =$ *HMA*$(Q) \cup \{x_{k+1}\}$ and since $x_{k+1}\uparrow = \emptyset$ we have *HMA*$(P)\uparrow =$ *HMA*$(Q)\uparrow$. Combining this with the fact that $\mathcal{Q}$ is flat (by the induction hypothesis) we obtain that $\mathcal{P}$ is flat. The fact that poset of the game remains flat through phases 3 and 4 follows in a very similar way. □

Proving Theorem 2 we consider an order on the set of all antichains of $\mathcal{P}$ defined as follows: $A \leqslant B$ for antichains $A, B \subseteq P$ if $B \subseteq A\Uparrow$. An antichain is said to be *high* in $\mathcal{P}$ if for all antichains B such that $B \subseteq A\Uparrow$ and $B \neq A$ we have $|B| < |A|$. Note that A is high in $\mathcal{P}$ iff $A =$ *HMA*$(A\Uparrow)$. There are arguments where the chain determined by a color is handy. If $\mathcal{P}$ is colored (covered) by c with colors $\gamma_1, \ldots, \gamma_k$ then $c^{-1}(\gamma_i)$ is a chain in $\mathcal{P}$ colored with γ_i, i.e., $c^{-1}(\gamma) := \{p \in P : \gamma \in c(P)\}$.

Proof of Theorem 2: To prove the theorem we provide a strategy for Algorithm, i.e. an on-line algorithm for on-line adaptive chain covering problem, using $\frac{2\sqrt{2}}{3} w\sqrt{w} + O(w)$ colors on all upgrowing flat width w posets. Our strategy maintains an auxiliary data stucture S that depends on an upgrowing flat poset presented (by Spoiler) as an input $\mathcal{P} = (P, \leqslant)$ as well as on the already built (by Algorithm) coloring c of $\mathcal{P}$. When $\mathcal{P}$ expands to $\mathcal{P}^+ = (P \cup \{x\}, \leqslant)$ by a new, maximal point x then Algorithm modifies S to get a new structure S^+ for $\mathcal{P}^+$. The set of colors assigned to x can be read from S^+. Following invariants describes S:

Invariants. Put $S = (P, \leqslant, k, L_1, \ldots, L_k, \Gamma_1, \ldots, \Gamma_k, \lambda_1, \ldots, \lambda_k, c)$, where

I0 $\Gamma_1, \ldots, \Gamma_k$ are pairwise disjoint sets of colors; c is a coloring of $\mathcal{P}$ using colors from $\bigcup_{i=1}^{k} \Gamma_i$, i.e.

- **a** $c : P \longrightarrow \{\alpha : \alpha \subseteq \bigcup \Gamma_i\} - \{\emptyset\}$,
- **b** $c^{-1}(\gamma)$ is a chain for all $\gamma \in \bigcup_{i=1}^{k} \Gamma_i$,

c if $\mathcal{P}^-$ is the poset of the game from the previous round and $S^- = (P^-, \leqslant, \ldots, c^-)$ is its auxiliary structure then $c(p) \subseteq c^-(p)$ for all $p \in P^-$,

I1 **a** $L_1, ..., L_k$ are high antichains in $\mathcal{P}$,

b $L_1 \leqslant ... \leqslant L_k$,

c $|L_1| = \mathit{width}(P)$,

d $|L_i| - |L_{i+1}| = i + 1$, for $i = 1, ..., k-1$,

e $|L_k| \leqslant k + 1$,

I2 for $i = 1, ..., k$, $\lambda_i : L_i \longrightarrow \Gamma_i$, is a bijection such that
$\mathit{statictop}(\lambda_i(y)) \leqslant y$, for all $y \in L_i$, where $\mathit{statictop}(\gamma) := \mathit{max}\Big(c^{-1}(\gamma) \cap \bigcup_{i=1}^{k} L_i\Downarrow\Big)$,

I3 for $i = 1, ..., k$, $y \in L_i\uparrow - L_{i+1}\Uparrow$ there is a function $\mathit{ind}^y : L_i \cap y\downarrow \longrightarrow \{1, ..., i\}$ such that
$c(y) = \big\{\lambda_{\mathit{ind}^y(z)}(z) : z \in L_i \cap y\downarrow\big\}$.

In further considerations it is handy to put $C_i := L_i\uparrow - L_{i+1}\Uparrow$, where $L_{k+1} := \emptyset$. We present several useful consequences of our invariants.

Fact 1 *I1a, I1c for S implies $L_1 = \mathit{HMA}(P)$.*

Fact 2 *I1a, I1b, I1c for S implies $C_i\downarrow \cap L_i \subseteq \bigcap_{j=1}^{i} L_j - L_{i+1}\Uparrow$ for all $i = 1, \ldots, k$.*

Fact 3 *I1a, I1b, I1c for S implies that $C_1, \ldots, C_k, \bigcup_{i=1}^{k} L_i\Downarrow$ forms a partition of $\mathcal{P}$.*

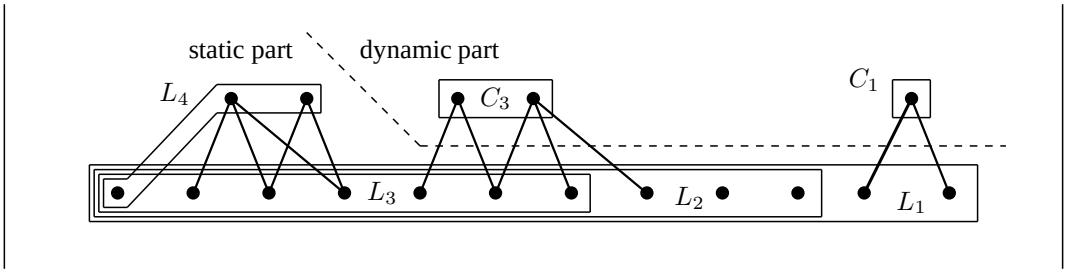
Fact 4 *I1b for S implies $(C_i\downarrow \cap L_i) \cap (C_j\downarrow \cap L_j) = \emptyset$ for all $1 \leqslant i \neq j \leqslant k$.*

Fact 5 *I0, I2 for S implies that for all $i, j \in \{1, \ldots, k\}$, $u \in L_i$, $w \in L_j$ we have $\lambda_i(u) = \lambda_j(w) \Leftrightarrow i = j$ and $u = w$.*

Fact 6 *I0, I1, I2, I3 for S implies that $c(C_i) \cap \bigcup_{l=1}^{j} \lambda_l(\bigcap_{m=1}^{j} L_m - L_{j+1}\Uparrow) = \emptyset$ for $1 \leqslant i \neq j \leqslant k$.*

Now, we give some intuitions for our invariants. The key role in the auxiliary structure S for $\mathcal{P}$ play high antichains $L_1, \ldots, L_k$ called the *levels* of $\mathcal{P}$. The antichain L_1 (the 1-st level) is $\mathit{HMA}(P)$ (see Fact 1). Consecutive levels lie above previous with respect to the order $\leqslant$ on antichains and their size decrease as I1d says. It will be proved that k - the number of levels - is $O(\sqrt{w})$. This will be important in determining the number of used colors.

In our strategy Algorithm splits P into the static part and the dynamic part. The static part is formed by $\bigcup_{i=1}^{k} L_i\Downarrow$ and dynamic part is the whole rest of P. By Fact 3 the dynamic part equals $\bigcup_{i=1}^{k} C_i$.



All points in the static part are colored by a single color. This means that their color remains unchanged through the rest of the game. On the other hand points in the dynamic part are always colored with two or more colors. Intuitively, Algorithm is not sure yet which color is most aproppiate and which ones should be removed. The levels $L_1, \ldots, L_k$ are high antichains in $\mathcal{P}$ (by I1a). Every new point presented by Spoiler may cause that some L_i's stop to be high in an enlarged poset. Thus, new levels of $\mathcal{P}^+$ poset may intuitively "move up" and some points from dynamic part may jump to static. Of course, when point "jumps" Algorithm removes almost all of its colors but this will be discussed later.

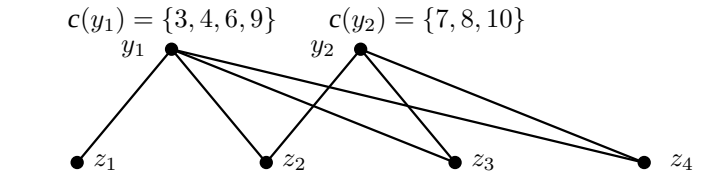
Each level L_i has assigned a set of colors Γ_i such that $|L_i| = |\Gamma_i|$ and there is a bijection $\lambda_i : L_i \longrightarrow \Gamma_i$ assigning to each point $y \in L_i$ a color from Γ_i. We stress that $\lambda_i(y)$ is not necessarily a color of y, i.e. $\lambda_i(y) \notin c(y)$ may be true. The connection of λ_i's with coloring function c is determined by I2 which says that $\mathit{statictop}(\lambda_i(y)) \leqslant y$ where $\mathit{statictop}(\gamma)$ is the maximal point colored with γ in the static part of $\mathcal{P}$. Loosely speaking, this condition tells us that if Spoiler presents a new point x above the point y Algorithm may use $\lambda_i(y)$ to color x. Indeed, even if color $\lambda_i(y)$ is used in the dynamic part of the poset it may be removed from there cause as we mentioned all points in the dynamic part are colored with at least two colors.

Spoiler introducing new points to the poset keeps it flat, i.e. $L_1\uparrow$ is an antichain. This together with $L_i \subseteq L_{i-1}\Uparrow, L_{i-1} \subseteq L_{i-2}\Uparrow, \ldots, L_2 \subseteq L_1\Uparrow$ (by I1b) and the definition of C_i gives Fact 2, i.e., $C_i\downarrow \cap L_i \subseteq L_1 \cap \ldots \cap L_i - L_{i+1}\Uparrow$. Consider $y \in C_i$ and let $z_1, \ldots, z_m$ be all predeccesors of y in L_i. We have $\{z_1, \ldots, z_m\} = y\downarrow \cap L_i \subseteq L_1 \cap \ldots \cap L_i$. Thus, all z_j's are in the domains of $\lambda_1, \ldots, \lambda_i$. The Invariant I3 tells us that $c(y) = \{\lambda_{i_1}(z_1), \ldots, \lambda_{i_m}(z_m)\}$ for some $1 \leqslant i_1, \ldots, i_m \leqslant i$, i.e., each predeccessor in L_i sends to $c(y)$ exactly one of his λ-colors. Consider following example:

$z_1, z_2, z_3, z_4 \in L_i$ for $i = 1, 2, 3$

$z_1, z_2, z_3, z_4 \notin L_i$ for $i \geqslant 4$

$C_3 = \{y_1, y_2\}$



$\lambda_1(z_1) = 1$	$\lambda_1(z_2) = 2$	$\lambda_1(z_3) = 3$	$\lambda_1(z_4) = 4$
$\lambda_2(z_1) = 5$	$\lambda_2(z_2) = 6$	$\lambda_2(z_3) = 7$	$\lambda_2(z_4) = 8$
$\lambda_3(z_1) = 9$	$\lambda_3(z_2) = 10$	$\lambda_3(z_3) = 11$	$\lambda_3(z_4) = 12$

Suppose that the poset of the game $\mathcal{P} = (P, \leqslant)$ is already presented by Spoiler and on-line adaptively colored by Algorithm. Now, Spoiler introduces a new maximal point x enlarging $\mathcal{P}$ to $\mathcal{P}^+ = (P \cup \{x\}, \leqslant)$. Algorithm's strategy modifies $S = (P, \leqslant, k, L_1, \ldots, L_k, \Gamma_1, \ldots, \Gamma_k, \lambda_1, \ldots, \lambda_k, c)$ for $\mathcal{P}$ to $S^+ = (P^+, \leqslant, k^+, L_1^+, \ldots, L_k^+, \Gamma_1^+, \ldots, \Gamma_k^+, \lambda_1^+, \ldots, \lambda_k^+, c^+)$ for $\mathcal{P}^+$. We are going to prove that all invariants are satisfied and this will imply that Algorithm uses $\frac{2\sqrt{2}}{3}w\sqrt{w} + O(w)$ colors on width w poset. Before describing our strategy we note that all segments of the form $X\downarrow, X\Downarrow, X\uparrow, X\Uparrow$ are considered in $\mathcal{P}^+$. Whenever we need to refer to an upset in $\mathcal{P}$ we write $X\uparrow \cap P$, while for $X \subseteq P$ downsets $X\downarrow$ are the same in $\mathcal{P}$ and $\mathcal{P}^+$. In particular $C_i = L_i\uparrow - L_{i+1}\Uparrow$ refers to $\mathcal{P}^+$.

Algorithm.

(A1) **if** $\mathit{width}(P) < \mathit{width}(P^+)$ **then Case** A

(A2) **if** *width*$(P) =$ *width*(P^+) **then begin**
(A3) $i_0 :=$ *max* $\{i \in N : x \in L_i\uparrow\}$
(A4) **if** L_{i_0} is high in P^+ **then Case** B
(A5) **else Case** C
(A6) **end**
(A7) **Case** A
(A8) {γ_1, ..., γ_{k+1} *are the brand new colors*}
(A9) **for** $i := 1$ **to** k **do begin**
(A10) $L_i^+ := L_i \cup \{x\}$ $\Gamma_i^+ := \Gamma_i \cup \{\gamma_i\}$ $\lambda_i^+ := \lambda_i \cup \{(x, \gamma_i)\}$
(A11) **end**
(A12) **if** $|L_k| = k + 1$ **then begin**
(A13) $k^+ := k + 1$
(A14) $L_{k+1}^+ := \{x\}$ $\Gamma_{k+1}^+ := \{\gamma_{k+1}\}$ $\lambda_{k+1}^+ := \{(x, \gamma_{k+1})\}$
(A15) **end**
(A16) **else** $k^+ := k$
(A17) $c^+|_{\mathcal{P}} := c$
(A18) $c^+(x) := \{\gamma_1\}$
(A19) **Case** B
(A20) $k^+ := k$
(A21) **for** $i := 1$ **to** k **do begin**
(A22) $L_i^+ := L_i$ $\Gamma_i^+ := \Gamma_i$ $\lambda_i^+ := \lambda_i$
(A23) **end**
(A24) $c^+|_{\mathcal{P}} := c$
(A25) $c^+(x) := \left\{ \mathit{repr}\left(\left(\bigcup_{i=1}^{i_0} \{\lambda_i(y)\} \right) - c(C_{i_0} - \{x\}) \right) : y \in (x\downarrow \cap L_{i_0}) \right\}$
(A26) {*repr*() *retrieves an element of a set provided on input. The proof of the fact that the set on input*
(A27) *is non-empty will be presented.*}
(A28) **Case** C
(A29) {*To simplify, Case C is split into two parts. First we construct a new structure*
(A30) $\overline{S} = (P, \leqslant, k, L_1, \ldots, L_k, \overline{\Gamma}_1, \ldots, \overline{\Gamma}_k, \overline{\lambda}_1, \ldots, \overline{\lambda}_k, c)$ *for* $\mathcal{P}$*and afterwards, basing on* $\overline{S}$ *a structure* S^+ *for* $\mathcal{P}^+$ *is constructed.*}
(A31) {*Generation* $\overline{S}$ *for* $\mathcal{P}$*:*}
(A32) **for** $i := 1$ **to** k **do** choose bijection $\psi_i : L_i \longrightarrow \mathit{HMA}(L_i\Uparrow)$,
(A33) such that $y \leqslant \psi_i(y)$ for $y \in L_i$
(A34) **for** $i := 1$ **to** i_0 **do** $\Delta_i := \{y \in L_{i_0} \cap C_{i_0}\downarrow : \psi_{i_0}(y) \in C_{i_0}$ & $\lambda_i(y) \in c(\psi_{i_0}(y))\}$
(A35) **for** $i := 1$ **to** $i_0 - 1$ **do** $\overline{\Gamma}_i := \Gamma_i - \lambda_i(\Delta_i) \cup \lambda_{i_0}(\Delta_i)$
(A36) $\overline{\Gamma}_{i_0} := \left(\Gamma_{i_0} - \bigcup_{i=1}^{i_0} \lambda_{i_0}(\Delta_i) \right) \cup \bigcup_{i=1}^{i_0} \lambda_i(\Delta_i)$
(A37) **for** $i := i_0 + 1$ **to** k **do** $\overline{\Gamma}_i := \Gamma_i$
(A38) **for** $i := 1$ **to** $i_0 - 1$ **do** $\overline{\lambda}_i := \lambda_i|_{L_i - \Delta_i} \cup \lambda_{i_0}|_{\Delta_i}$
(A39) $\overline{\lambda}_{i_0} := \lambda_{i_0}|_{L_{i_0} - \bigcup_{i=1}^{i_0} \Delta_i} \cup \bigcup_{i=1}^{i_0} \lambda_i|_{\Delta_i}$
(A40) **for** $i := i_0 + 1$ **to** k **do** $\overline{\lambda}_i := \lambda_i$
(A41) {*Now basing on* $\overline{S}$ *for* $\mathcal{P}$ *Algorithm generates* S^+ *for* $\mathcal{P}^+$}

(A42) $k^+ := k$
(A43) **for** $i := 1$ **to** k **do begin**
(A44) $\quad L_i^+ := \mathit{HMA}(L_i\Uparrow) \qquad \Gamma_i^+ := \overline{\Gamma}_i \qquad \lambda_i^+ := \overline{\lambda_i} \circ \psi_i^{-1}$
(A45) **end**
(A46) **for all** $y \in C_{i_0} \cap \mathit{HMA}(L_{i_0}\Uparrow)$ **do** $c^+(y) := \{\overline{\lambda}_{i_0}(\psi_{i_0}^{-1}(y))\}$
(A47) **for all** $y \in C_{i_0} - \mathit{HMA}(L_{i_0}\Uparrow)$ **do** $c^+(y) := c(y) \cap \bigcup_{j=1}^{i_0} \overline{\lambda}_j\,(y{\downarrow} \cap \mathit{HMA}(L_{i_0}\Uparrow))$
(A48) **for all** $y \in P - C_{i_0}$ **do** $c^+(y) := c(y)$

As we can see there are three cases dependent on how the new point x enlarges $\mathcal{P}$ to $\mathcal{P}^+$. If x increases the width of the poset (Case A) then Algorithm obtains k new chains for future use. It turns out that a simple enlargement of all levels L_i's by x works. Only when the last old level L_k had already $k+1$ points Algorithm does some twist adding a completely new level to hold the condition I1e (see (A12)-(A15)). If the width of the poset is unchanged and L_{i_0} remains high in $\mathcal{P}^+$ then Algorithm runs Case B. We will see that in this case in fact all L_i's remain high in $\mathcal{P}^+$. In this setting Algorithm leaves all levels untouched, i.e. $L_i^+ = L_i$ (see (A22)). Coloring the new point x Algorithm uses one color for each predecessor of x in L_{i_0} (see (A25)). This routine may be interpreted as securing all edges between $L_{i_0}^+$ and $C_{i_0}^+$. As we said Algorithm is not able to decide which color is appropiate for points in the dynamic part of the poset. Thus, all potentially candidates for a good color are used. This will guarantee the condition I3. If, in turn, L_{i_0} is not high in $\mathcal{P}^+$ then Algorithm runs Case C. This is the moment when some levels moves up. They move through the edges of the bijections ψ_i's stated in (A32), (A33). As a result some points from a dynamic part of $\mathcal{P}$ jump to the static part of $\mathcal{P}^+$. For each such point y Algorithm chooses a color from $c(y)$ securing the edge of the bijection and this color will be the only one in $c^+(y)$ (see (A46)). The set of colors of points in the dynamic part of $\mathcal{P}^+$ may be also reduced to these securing edges from $L_{i_0}^+ = \mathit{HMA}(L_{i_0}\Uparrow)$ (see (A47)) The middle step in Case C (lines (A34)-(A40)), constructing the structure $\overline{S}$, is only for the sake of clarity of presentation. Algorithm, already knowing ψ_i's, rearranges the colors in Γ_i's getting $\overline{\Gamma_i}$'s to achieve the additional condition: $\overline{\lambda}_{i_0}(y) \in c(\psi_{i_0}(y))$, for all $y \in \psi_{i_0}^{-1}(C_{i_0} \cap L_{i_0}^+)$.

The proof that S^+ for $\mathcal{P}^+$ holds invariants I0-I3 is split into cases of presented strategy.

Case A: In this case the point x increases the width of the poset of the game (see (A1)). By the fact that a single point may increase the width of the poset at most by one we get $\mathit{width}(P^+) = \mathit{width}(P) + 1$.

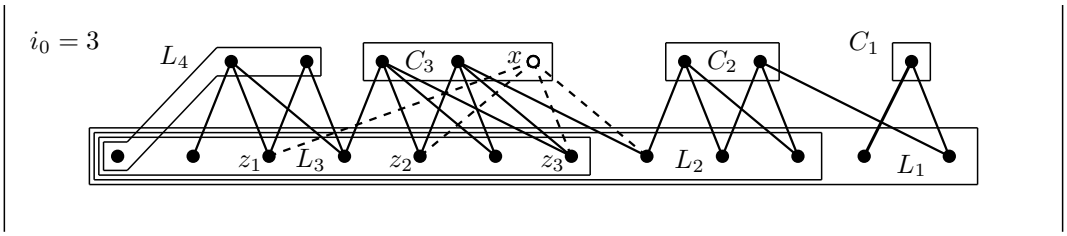
Claim. *Let $\mathcal{Q}^+$ be a poset obtained from $\mathcal{Q} = (Q, \leqslant)$ by adding a new element q in such a way that q is maximal in $\mathcal{Q}^+$. Then $\mathit{width}(Q) < \mathit{width}(Q^+)$ implies $\mathit{HMA}(Q^+) = \mathit{HMA}(Q) \cup \{q\}$.*

Using this claim we get that $\mathit{HMA}(P^+) = \mathit{HMA}(P) \cup \{x\}$. Thus, $x \notin L_1\Uparrow$ and therefore by I1b for S we have $x \notin L_i\Uparrow$ for all i's. All of these with some trivial observations suffice to prove that in Case A the structure S^+ holds all our invariants.

Case B: Since $x \notin L_i\Uparrow$ for $i > i_0$ (by (A3)) we trivially get that L_i is high in $\mathcal{P}^+$. By (A4) the level L_{i_0} is high in $\mathcal{P}^+$. The fact that $L_1, \ldots, L_{i_0-1}$ are also high in $\mathcal{P}^+$ simply follows from the next claim.

Claim. *Let $\mathcal{Q}^+$ be a poset obtained from $\mathcal{Q} = (Q, \leqslant)$ by adding a new element q in such a way that q is maximal in $\mathcal{Q}^+$. Let $A, B \subseteq Q$ be antichains such that A is high in $\mathcal{Q}$ and B is high in $\mathcal{Q}^+$. Then $A \leqslant B$, $x \in B{\uparrow}$ implies that A is high in $\mathcal{Q}^+$.*

The new point x enlarges the set C_{i_0}. This means that according to invariant I3 all predeccessors of x in L_{i_0}, say $z_1, \ldots, z_m$, have to dispose colors into $c^+(x)$. As we already discussed $\{z_1, \ldots, z_m\} = x{\downarrow} \cap L_{i_0} \subseteq L_1 \cap \ldots \cap L_{i_0}$, thus all these points are in the domains of $\lambda_1, \ldots, \lambda_{i_0}$.



Claim. *Let A, B be high antichains in a flat poset $\mathcal{Q}$ such that $A < B$. Then $|A\uparrow - B\Uparrow| < |A| - |B|$.*

Using this claim we obtain $|C_{i_0}| = |L_{i_0}\uparrow - L_{i_0+1}\Uparrow| < |L_{i_0}| - |L_{i_0+1}| \leqslant i_0 + 1$. Thus, $|C_{i_0}| \leqslant i_0$. Summing up, each z_j has i_0 colors to dispose on points in C_{i_0} (exactly one color on each point) and $|C_{i_0}| \leqslant i_0$. This proves $\bigcup_{i=1}^{i_0} \lambda_i(z_j) - c(C_{i_0} - \{x\})$ is non-empty for all $1 \leqslant j \leqslant m$. Thus, the function *repr*() in (A25) is applied to non-empty sets and therefore I3 for S^+ holds.

We also need to prove that c^+ is a legal coloring, i.e. I0 for S^+. Since $c^+|_P = c$ and c is legal for $\mathcal{P}$ it suffices to show that colors used in $c^+(x) = \{\lambda_{i_1}(z_1), \ldots, \lambda_{i_m}(z_m)\}$ form chains. First we are going to observe that $\lambda_{i_j}(z_j)$ is not used in the dynamic part of $\mathcal{P}$. According to (A25) $\lambda_{i_j}(z_j) \in \bigcup_{i=1}^{i_0} \lambda_i(\{z_1, \ldots, z_m\}) - c(C_{i_0} - \{x\})$, thus $\lambda_{i_j}(z_j)$ is not used by c on $C_{i_0} - \{x\}$. On the other hand applying $z_j \in \bigcap_{i=1}^{i_0} L_i - L_{i_0+1}\Uparrow$ (by Fact 2) to Fact 6 we obtain $\lambda_{i_j}(z_{i_j}) \notin c(C_i)$ for $i \neq i_0$. Now, the fact that $c^{+-1}\big(\lambda_{i_j}(z_j)\big)$ is a chain follows immediately from *statictop*$\big(\lambda_{i_j}(z_j)\big) \leqslant z_j \leqslant x$ (by I2 for S).

Case C: The new levels $L_i^+ = HMA(L_i\Uparrow)$ are high in $\mathcal{P}^+$ (by the definition), i.e. I1a for S^+ holds. The facts that $L_1^+ \leqslant \ldots \leqslant L_k^+$ and $|L_i| = \big|L_i^+\big|$ are obtained from following claims.

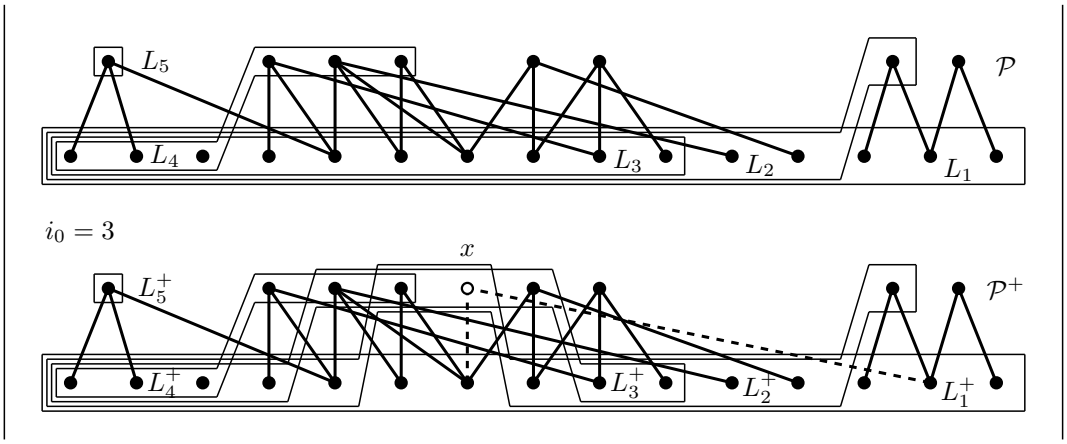
Claim. *Let $\mathcal{Q}^+$ be a poset obtained from $\mathcal{Q} = (Q, \leqslant)$ by adding a new element q such that q is maximal in $\mathcal{Q}^+$. Let $A, B \subseteq Q$ be high antichains in $\mathcal{Q}$. Then $A \leqslant B$ implies HMA$(A\Uparrow_{\mathcal{Q}^+}) \leqslant$ HMA$(B\Uparrow_{\mathcal{Q}^+})$.*

Claim. *Let $\mathcal{Q}^+$ be a poset obtained from $\mathcal{Q} = (Q, \leqslant)$ by adding a new element q in such a way that q is maximal in $\mathcal{Q}^+$. Let A be a high antichain in $\mathcal{Q}$. Then $|A| = |$HMA$(A\Uparrow_{\mathcal{Q}^+})|$.*

This proves I1b, I1c, I1d, I1e for S^+. In fact, dependencies between L_i's and new L_i^+'s are more complicated. We formulate them below. The proofs of these relationships are highly combinatorial and omitted in this paper. Afterwards an example ilustrating some of them is presented.

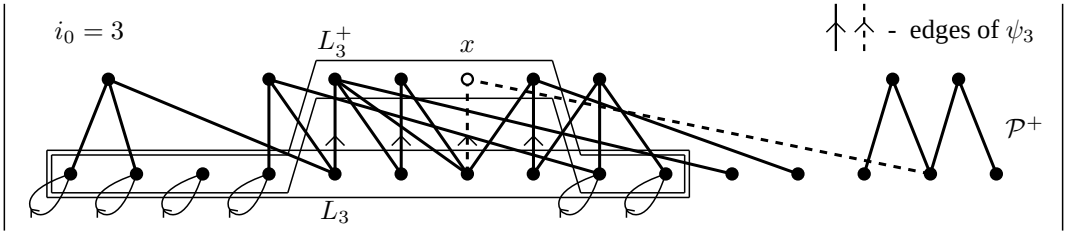
$$L_k = L_k^+, \quad \ldots, \quad L_{i_0+1} = L_{i_0+1}^+, \qquad \begin{array}{l} \emptyset \neq L_{i_0}^+ - L_{i_0} \supseteq L_{i_0-1}^+ - L_{i_0-1} \supseteq \ldots \supseteq L_1^+ - L_1, \\ \emptyset \neq L_{i_0} - L_{i_0}^+ \supseteq L_{i_0-1} - L_{i_0-1}^+ \supseteq \ldots \supseteq L_1 - L_1^+, \end{array}$$

$$\bigcup_{i=1}^{k} L_i^+\Downarrow = \bigcup_{i=1}^{k} L_i\Downarrow \cup (C_{i_0} \cap L_{i_0}^+), \qquad C_{i_0}^+ = C_{i_0} - L_{i_0}^+, \qquad C_i^+ = C_i \quad \text{for all } i \neq i_0. \tag{2}$$



Lines (A32), (A33) put ψ_i to be a bijection from L_i to L_i^+ such that $y \leqslant \psi_i(y)$. Following claim argues that such functions exist.

Claim. *Let A be a maximum with respect to the size antichain in a poset $\mathcal{Q} = (Q, \leqslant)$ and B an antichain in $\mathcal{Q}$. Then there is an injection ψ such that $\psi : B \longrightarrow A$ where $\psi(b) \geqslant a$ for all $b \in B$.*



Since L_{i_0} is high in $\mathcal{P}$ and not high in $\mathcal{P}^+$ we have $x \in L_{i_0}^+$. Note that by (2) points in $C_{i_0} \cap L_{i_0}^+$ (except the new point x) jumps from the dynamic part of $\mathcal{P}$ to the static part of $\mathcal{P}^+$.

The bijection ψ_{i_0} determines edges of $\mathcal{P}^+$ along which L_{i_0} moves to $L_{i_0}^+$. The condition I3 for S guarantees that all edges between L_{i_0} and C_{i_0} are secured by a color, i.e., for all $y \in L_{i_0}$, $z \in C_{i_0}$, $y \leqslant z$ there is $1 \leqslant s \leqslant i_0$ such that $\lambda_s(y) \in c(z)$. The idea of our strategy for Algorithm is that at the moment that bijection ψ_{i_0} is known a color securing edge $(y, \psi_{i_0}(y))$ becomes the only color of $\psi_{i_0}(y)$,i.e. $c^+(\psi_{i_0}(y)) = \{\lambda_s(y)\}$, for all $y \in \psi_{i_0}^{-1}(C_{i_0} \cap L_{i_0}^+)$. However, while the bijection is not determined, Algorithm secures all potential edges, i.e. all edges between L_{i_0} and C_{i_0}.

For the sake of clarity before the construction of S^+ Algorithm, in our strategy, permutes colors between Γ_i's constructing a new auxiliary structure $\overline{S}$ for $\mathcal{P}$ holding all conditions I0-I3 invariants and satisfying an additional condition

$$\overline{\lambda}_{i_0}(y) \in c(\psi_{i_0}(y)) \text{, for all } y \in \psi_{i_0}^{-1}(C_{i_0} \cap L_{i_0}^+). \tag{3}$$

In other words all colors securing ψ_{i_0}'s edges to $C_{i_0} \cap L_{i_0}^+$ are from $\overline{\Gamma}_{i_0}$.

Now, we present an argument showing that c^+ is a legal coloring, i.e. I0 for S^+. The set $c^+(y)$ is trivially non-empty for $y \notin C_{i_0} - L_{i_0}^+$ (by (A46),(A48)). To prove it for $y \in C_{i_0} - L_{i_0}^+$ observe that since $C_{i_0}^+ = C_{i_0} - L_{i_0}^+$ (see (2)) we have that $y\downarrow \cap L_{i_0}^+$ is non-empty. Moreover, one can prove that

$y{\downarrow} \cap L_{i_0}^+ \subseteq L_{i_0}$. Combining $y \in C_{i_0}$, $\emptyset \neq y{\downarrow} \cap L_{i_0}^+ \subseteq L_{i_0}$ with $c(y) = \left\{\overline{\lambda}_{\overline{ind}^y(z)}(z) : z \in L_{i_0} \cap y{\downarrow}\right\}$ (see I3 for $\overline{S}$) we have that $c(y) \cap \bigcup_{j=1}^{i_0} \overline{\lambda}_j(y{\downarrow} \cap L_{i_0}^+)$ is non-empty, which finishes the proof of I0a for S^+. The fact that $c^+(y) \subseteq c(y)$ for all $y \in P$ follows from (A46), (A47), (A48) and (3). It remains to show that $c^{-1}(\gamma)$ is a chain for all $\gamma \in \bigcup \Gamma_i$. It is not trivial only for the color of x, i.e. $\overline{\lambda}_{i_0}(\psi_{i_0}^{-1}(x))$ (by (A46)). To prove that $c^{+-1}\big(\overline{\lambda}_{i_0}(\psi_{i_0}^{-1}(x))\big)$ is a chain we first need that $\overline{\lambda}_{i_0}(\psi_{i_0}^{-1}(x))$ is not used by c^+ in the dynamic part of $\mathcal{P}$. By (A46), (A47) we know that $\overline{\lambda}_{i_0}(\psi_{i_0}^{-1}(x))$ is not used in C_{i_0}. To prove that it is not used in other C_i's we need following claim.

Claim. *Let $\mathcal{Q}^+$ be a flat poset obtained from a flat poset $\mathcal{Q} = (Q, \leqslant)$ by adding a new maximal element q such that q is maximal in $\mathcal{Q}^+$ and width(Q^+) = width(Q). Then HMA$(Q){\uparrow}$ is an antichain in $\mathcal{Q}^+$.*

This together with $L_{i_0} \subseteq L_{i_0-1}{\Uparrow}$, …,$L_2 \subseteq L_1{\Uparrow}$ gives $C_{i_0}{\downarrow} \cap L_{i_0} \subseteq \bigcap_{m=1}^{i_0} L_m$. Applying $\psi_{i_0}^{-1}(x) \in C_{i_0}{\downarrow} \cap L_{i_0} \subseteq \bigcap_{m=1}^{i_0} L_m - L_{i_0+1}{\Uparrow}$ to Fact 6 we obtain that $\overline{\lambda}_{i_0}(\psi_{i_0}^{-1}) \notin c(C_j)$ for all $j \neq i_0$. Thus, the whole chain determined by $\overline{\lambda}_{i_0}(\psi_{i_0}^{-1}(x))$ except x is contained in the static part of $\mathcal{P}$. The fact that *statictop*$\big(\overline{\lambda}_{i_0}(\psi_{i_0}^{-1}(x))\big) \leqslant \psi_{i_0}^{-1}(x) \leqslant x$ finishes the proof.

To see that *statictop*$^+\big(\lambda_i^+(y)\big) \leqslant y$, i.e. I2 for S^+, recall that Spoiler presenting x enlarges the static part by points from $C_{i_0} \cap L_{i_0}^+$ and according to (A46) we have $c^+(y) = \big\{\overline{\lambda}_{i_0}(\psi_{i_0}^{-1}(y))\big\}$ for all $y \in L_{i_0}^+ \cap C_{i_0}$. Since *statictop* function concerns only on the static part of $\mathcal{P}^+$ we get

$$\begin{aligned} \textit{statictop}^+\big(\lambda_i^+(y)\big) &= \textit{statictop}\big(\overline{\lambda}_i(\psi_i^{-1}(y))\big) \leqslant \psi_i^{-1}(y) \leqslant y \qquad \text{for } i \neq i_0,\ y \in L_i^+, \\ \textit{statictop}^+\big(\lambda_{i_0}^+(y)\big) &= \textit{statictop}\big(\overline{\lambda}_{i_0}(\psi_{i_0}^{-1}(y))\big) \leqslant \psi_{i_0}^{-1}(y) \leqslant y \qquad \text{for } y \in L_{i_0}^+ - C_{i_0}. \end{aligned}$$

By (A44),(A46) in turn, we get *statictop*$^+\big(\lambda_{i_0}^+(y)\big) = \textit{statictop}\big(\overline{\lambda}_{i_0}(\psi_{i_0}^{-1}(y))\big) = y$ for $y \in L_{i_0}^+ \cap C_{i_0}$.

Combining many previous observations with a little effort one can prove that functions

$$\textit{ind}^{+y} := \overline{\textit{ind}}^y \quad \text{for } i \neq i_0,\ y \in C_i^+ \qquad \text{and} \qquad \textit{ind}^{+y} := \overline{\textit{ind}}^y|_{y{\downarrow} \cap L_{i_0}^+} \quad \text{for } y \in C_{i_0}^+,$$

witness I3 for S^+ where $\overline{\textit{ind}}^y$ witnesses I3 for $\overline{S}$ for appropiate y.

We sketched the proofs of all invariants in all cases. Having them satisfied we know that presented strategy builds an on-line adaptive coloring of poset presented by Spoiler. It remains to prove that Algorithm uses $\frac{2\sqrt{2}}{3} w\sqrt{w} + O(w)$ colors. Using I1c and I1d one can prove $\sum_{i=1}^{k} |L_i| = \frac{1}{3}k^3 - \frac{1}{3}k + k\,|L_k|$ and $k \leqslant 1 + \sqrt{2w}$. This together with I1e gives

$$\left|\bigcup_{i=1}^{k} \Gamma_i\right| = \sum_{i=1}^{k} |L_i| \leqslant \frac{1}{3}k^3 - \frac{1}{3}k + k(k+1) = \frac{2\sqrt{2}}{3} w\sqrt{w} + O(w).$$

□

References

[1] Bartłomiej Bosek and Piotr Micek. On-line dimension of upgrowing posets. *submitted to Theoretical Computer Science*.

[2] Stefan Felsner. On-line chain partitions of orders. *Theoret. Comput. Sci.*, 175(2):283–292, 1997. Orders, algorithms and applications (Lyon, 1994).

[3] Henry A. Kierstead. An effective version of Dilworth's theorem. *Trans. Amer. Math. Soc.*, 268(1):63–77, 1981.