ssdnm
środowiskowe
studia doktoranckie
z nauk matematycznych

Adam Gągol

Uniwersytet M. Curie-Skłodowskiej w Lublinie

Pathwidth and nonrepetitive colourings of graphs

Praca semestralna nr 2

(semestr zimowy 2012/13)

Opiekun pracy: Jakub Kozik

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

PATHWIDTH AND NONREPETITIVE COLOURINGS OF GRAPHS

ADAM GAGOL, GWENAEL JORËT, JAKUB KOZIK AND PIOTR MICEK

1. Introduction

A *repetition* of length k is a sequence $(a_1, \ldots, a_k, a_1, \ldots, a_k)$. A graph is *nonrepetitively coloured* if there is no simple path for which the sequence of colors along the path form a repetition. The minimum number of colours in a nonrepetitive colouring of a graph G is *nonrepetitive chromatic number* (denoted by $\pi(G)$).

In 1906 Thue [4] proved that path of any length can be nonrepetitively coloured with just 3 colours. The argument of Thue does not seem to generalise to colourings from lists or colouring of more complex graphs. Recent applications of Lovasz Local Lemma together with its algorithmic version yielded new results for nonrepetitive chromatic number [2]. One of advantages of those proofs is that they easily generalise to list colourings. The minimum number n such that G can be nonrepetitively coloured from lists of size n is denoted by $\pi_l(G)$ and called nonrepetitive list chromatic number. The difference between π and π_l can be arbitrarily large. Simple modification of Thue's result shows that $\pi(T) \leqslant 4$ for every tree T while a result of Fiorenzi et al. [1] shows that $\pi_l(T)$ can be arbitrarily large. Micek and Kozik showed that for every ϵ there exists such c that$\pi_l(T) < c\delta(T)^\epsilon$ for any tree [3].

In this paper we investigate graphs with bounded path-width. We show that this condition is sufficient for trees to have bounded π_l (Section 2) and provide an example of graph family of path-width 2 and arbitrarily large π_l (Section 3).

2. Trees

The main theorem of the paper is:

Theorem 2.1. *There exists a function f such that if T is a tree with pathwidth w then it is possible to colour T nonrepetitively from any assignment of lists of length $f(w)$.*

The rest of this section is devoted to the proof of this theorem.

For technical reasons we generalize the definition of repetition. A *repetition of length n with gap t* is a sequence $(a_1, \ldots, a_n, b_1, \ldots, b_t, a_{n+1}, \ldots, a_{2n})$ where $a_i = a_{i+n}$. A *k-repetition* is a repetition of lenght k. Abusing the notation a path in a graph for which a sequence of the colors along the path forms a repetition is also called a repetition (the distinction will be always clear from the context). The *mid-edge* of a repetition of length k is an edge between k-th and $k+1$-th vertex of the path. The *twin* of l-th vertex in a repetition of length k is the vertex of the path with index $(l+k)(mod\, 2k)$.

We use the fact that path-width of a tree is bounded to decompose it into a tree of paths with bounded depth.

Lemma 2.2. *If T is a tree with path-width k then one can decompose T into meta-tree of paths of depth k, i.e. the set $\mathcal{P}$ of disjoint paths of T such that:*

1. *every vertex of T belongs to exactly one path of $\mathcal{P}$*
2. *a tree $\mathbb{T}$ of contracted paths $\mathcal{P}$ has diameter at most $2^{k+1} - 1$.*

Proof. The proof is an induction on k. The case of $k = 0$ is trivial. Suppose that $k > 0$. Since T has path-width k, it is a connected subgraph of interval graph with representation I of height $k+1$. Lets P_0 be the path from the vertex with leftmost beginning in I to the vertex with rightmost ending. We choose P_0 to be the root path of a meta-tree. After removing P_0 from the graph we are left with a forest of path-width at most $k-1$ (for $k = 0$ the remaining graph is empty). By induction every tree of the forest have can be decomposed into meta-tree of depth at most $k-1$. Let $\mathcal{P}$ be the sum of those meta trees and $\{P_0\}$. Clearly paths of $\mathcal{P}$ partition the vertices of T and the diameter of contracted tree is at most $2d+1$, where d is the maximum diameter of contracted meta-trees of the forest. □

Fix a graph T of path-width k. Let us fix some planar embedding, root of T, and meta-tree $\mathcal{P}$. Additionally we direct every path of $\mathcal{P}$ (in an arbitrary way). For every vertex v of T, *the level* of v is the distance from the path containing v to the path containing root of T within the contracted tree $\mathbb{T}$. A path in T is a *one-way* if one of its ends is on the lowest level attained by this path and the other on level that is higher. A part of path P consisting of vertices of the lowest level is a *stable* part of P. One-way path $(v_1, \ldots, v_l)$ with lowest end v_1 is *pointing to the left* if (v_2, v_1) is a directed edge in the path from $\mathcal{P}$ containing v_1. Otherwise the path is *pointing to the right* (even if v_1 and v_2 belong to different paths of $\mathcal{P}$).

Before we construct a final nonrepetitive colouring we do some preprocessing that shortens the lists but exclude possibility of the following types of repetitions:

(1) k-repetitions with gap t where $t < k$ with all vertices on the same level of meta-tree,
(2) one-way k-repetitions with arbitrarily large gap pointing to the right with the intersection of gap and stable part not longer than k,
(3) one-way k-repetitions with arbitrarily large gap pointing to the left with the intersection of gap and stable part not longer than k.

The preprocessing is described in the following three lemmas. Shortening the lists in a tree can be considered as a coloring of the tree with lists of smaller length. Every such coloring will be called an *l-coloring.* In our proofs all the shortened lists have the same lengths. A path with associated lists is called an *l-repetition* if it is possible to choose colours from lists of its vertices in such a way that the coloured path is a repetition (analogously we define l-repetition of type (1) etc.). The proofs follow the idea of Moser-Tardos' algorithmic proof of Lovasz local lemma. In the proof we assume that a naive algorithm that given an infinite sequence of numbers tries to colour T nonrepetitively, never stops and use this fact to compress initial segments of the sequence to a better extent than it is actually possible.

For technical reasons we need the following observation:

Observation 2.3. *For a sequence of nonnegative numbers* $(x_1, \ldots, x_n)$, *if* $\Sigma_{i=1}^{n} x_i = M$ *then* $\prod_{i=1}^{n} x_i \leqslant 4^M$.

Proof. Cauchy inequality implies that the product of n numbers with constant sum is largest when all the numbers are equal. We ask for the maximum value of expression x^n provided that $xn = M$. Note that for $x > 4$ we have $(\frac{x}{2})^2 > x$ and the product is greater if we choose $x' = x/2$. Therefore the maximum is attained for x's smaller than 4 and the product is smaller than 4^M. □

Lemma 2.4. *If T is a tree of path-width k with lists of size $N = 64n^2 + n - 1$ in each vertex, then it is possible to choose sublists of size n in such a way that there is no colouring from sublists with a repetition of type* (1).

Proof. Let T be a tree of path-width k with meta-tree $\mathcal{P}$ and $\{L_v\}_{v \in V(T)}$ be a list assignment with $|L_v| = 64n^2 + n - 1$ for each v. Suppose for contradiction that it is not possible to choose sublists $L'_v \subseteq L_v$ for all $v \in V(T)$ which don't form an l-repetition of type (1) such that $|L'_v| = n$. Since, by definition, every repetition of type (1) is contained it one meta-path it means that there exists meta-path $P = (v_1, \ldots, v_m) \in \mathcal{P}$ without such choice of sublists. We consider a randomized algorithm 1 that tries to build an l-coloring of P, with sublists of length n, that does not contain an l-repetition of type (1).

Algorithm 1: Elimination of l-repetitions of type (1) on a path

```
input: S : ℕ → [(N choose n)]
i ← 1, j ← 1, C ← ∅
while the path is not entirely l-coloured do
    C(v_j) ← the sublist of L_{v_j} of length n with index S(i)
    i++
    j++
    if there is an l-repetition R of type (1) ending in v_j then
        let P be the vertices of the repeated part of R containing v_j
        for v ∈ P do
            erase the value of C(v)
        end
        j ← index of the first point in P
    endif
end
return C
```

Note that by our assumption that apropriate l-coloring does not exist, the algorithm never stops. Let us fix some input sequence S and run the algorithm for M steps (i.e. M iterations of the main loop). With every such run we associate some structure describing the behaviour of the algorithm. Clearly such a structure depends only on M initial values of S. More importantly sequences S and S' which differ on at least one of M initial positions would produce different structures. The structure we use for description of a run of the algorithm is a tuple (P, F, G, R) where:

(1) $P = (p_1, \ldots, p_M)$ is a sequence of numbers such that p_i is the number of coloured vertices after i-th step (i.e. the number of vertices v for which $C(v)$ is defined),
(2) F is a (partial function) C after M steps of the algorithm,
(3) $G = (g_1, \ldots, g_s)$ is a sequence of numbers such that g_i is the length of gap in the i-th retracted l-repetition of type (1) (s is the number of retracted repetitions),
(4) $R = (r_1, \ldots, r_s)$ is a sequence of sequences of numbers such that each r_i describes the i-th repetition in the following way (again s is the number

of retracted repetitions). Suppose that the i-th retracted repetition had length k, gap t, it occurred on a path $(a_1, \dots, a_k, b_1, \dots, b_t, a_{k+1}, \dots a_{2k}$ and just before the retraction the lists assigned to these vertices were correspondingly $A_1, \dots, A_k, B_1, \dots, B_t, A_{k+1}, \dots, A_{2k}$. For $j \in \{1, \dots, k\}$ let α_j be the index of list A_{k+j} among all the sublists of $L(a_{k+j})$ of length n that have nonempty intersection with A_j. We put $r_i = (\alpha_1, \dots, \alpha_k)$.

Now we need to prove that this encoding of a prefix of S is loseless and that it is an actual compression for M big enough.

Compression. We are concerned with the asymptotic number of descriptions when M tends to infinity.

(1) Let us start with estimating the number of possible P's. Each sequence $(p_1, \dots, p_M)$ can be injectively transformed into a sequence of differences $(1, p_2 - p_1, p_3 - p_2, \dots, p_M - p_{M-1}$. All the differences in the sequence belong to the set $\{1, 0, -1, -2, \dots\}$. Next we transform the sequence of differences into yet another sequence by substituting each element k of $\{0, -1, -2, \dots\}$ of a sequence by a sequence $(1, -1, -1, \dots)$ of length $k+2$. It is easy to see that the second transformation is also an injection. The resulting sequence is a sequence of $\{-1, 1\}$ and its length is smaller than $2M$ (the number of -1 in the resulting sequence is exactly the number times when a value of $C(v)$ was erased for some v (line 10), while the number of 1 in the sequence is the number of times when a value $C(v)$ was assigned to some v (line 4)). The number of such sequences is $O(4^M)$ and since both transformations are injective the same bound is valid for the number of possible sequences P.
(2) Number of possible F's is constant with respect to M.
(3) Sum of elements of G is not greater than the sum of lengths of all retracted l-repetitions which in turn is smaller than M. The number of all retractions is not greater than M. Hence the number of possible sequences G is smaller than the number of sequences of $M+1$ nonnegative integers which sum up to M. That number is $\binom{2M}{M} = o(4^M)$.
(4) Now we need to estimate the number of possible R's. Recall that a number α_j belonging to some sequence r_i from R is an index of some n-sublist among all n-sublist of a list of size N having nonempty intersection with some fixed sublist of size at most n. That number is never greater than $n\binom{N-1}{n-1}$. If we take any sequence R from the description of a run of the algorithm and concatenate all its list we get a sequence of numbers from $\left[n\binom{N-1}{n-1}\right]$ of length at most M. On the other hand any sequence of length at most M can be splitted into a sequence of nonempty sequences whose concatenation gives the original sequence in at most 2^M ways. Then we get that the number of possible R's is $O((2n\binom{N-1}{n-1})^M)$.

Putting those estimation together we get that the number of possible tuples (P, F, R, B) describing the run of the algorithm of length M is

$$o(\left(2^5 n \binom{N-1}{n-1}\right)^M) = o(\left(2^5 n \frac{n}{N-n+1} \binom{N}{n}\right)^M) = o(\binom{N}{n}^M).$$

That shows that, for sufficiently large M, the number of possible descriptions is smaller than the number of possible initial segments of length M of an input sequence S. To obtain a contradiction we show that descriptions (P, F, G, R) are different for input sequences which differ on initial segment of length M.

Loselessness. We prove that it is possible to reconstruct the first M elements of the input sequence S from a tuple (P, F, R, G) constructed in M steps. Given (P, F, R, G) we are going to decode $S(M)$ and (P', F', R', G') - tuple constructed by algorithm running $M - 1$ steps on the same input sequence S. Then by simple iteration we will extract all values $S(i)$ for $i \in [M]$.

If $p_M = p_{M-1} + 1$ then no l-repetition was retracted during the last step so $S(M)$ is simply the index of the last sublist of F, P' and F' are one element shorter, $R' = R, B' = B$.

If $p_M = p_{M-1} - k + 1$ where $k > 0$ then in the last step there was a retraction of k elements that formed an l-repetition. The last element of G is the length t of the gap in this repetition. Let $(\alpha_1, \ldots, \alpha_r)$ be the last element of R. Let $a_1, \ldots, a_k, b_1, \ldots b_t$ be the last elements of F with assigned sublists and let $A_1, \ldots, A_k$ be the sublists assigned to $a_1, \ldots, a_k$. Let $a_{k+1}, \ldots, a_{2k}$ be the vertices following b_t on the path. Before the retraction these vertices were coloured and we want to reconstruct with which sublists. Let us consider a vertex a_{k+i}. We know its list $L(a_{k+i})$ (those lists does not change), we know the sublist A_i that is assigned to a_i and we know that the sublist previously assigned to a_{k+i} had index α_i among n-sublist of $L(a_{k+i})$ that have nonempty intersection with A_i. In that way we can reconstruct the partial l-coloring $\hat{F}$ from just before the retraction. Just like in the previous case $S(M)$ is the index of the last sublist of $\hat{F}$. Taking sequences P', F', R', B' that are one element shorter than corresponding sequences $P, \hat{F}, R, B$ we obtain a tuple constructed after $M - 1$ steps. □

Using similar arguments in the following lemmas we show that if lists are long enough then they can be shortened in such a way that every colouring from these list avoids repetitions of types (2) and (3).

For one-way path P an *entering point* is the first point on the stable part of P and for meta-path $\mathbb{P}$ a central point is the one with up-edge (clearly root-path have no central point).

For any point v from level k, the children of v are vertices from level $k+1$ which are adjacent to v. By $v\!\downarrow$ we denote the union of vertices of subtrees rooted in children of v together with v.

Proposition 2.5. *If T is a tree of path-width k with lists of size $N = 64n^2 + n - 1$ in each vertex and without repetitions of type* (1) *it is possible to choose sublists of size n in such a way that there is no colouring from sublists with a repetition of type* (2) *and stable part on the root path of the meta-tree.*

Proof. Within the proof l-coloring means coloring with sublists of length n and repetition of type $(2')$ means repetition of type (2) with stable part on the root-path. For a partial l-coloring C and vertex v let $d(C, v)$ be an l-colouring which extends C, colours all vertices of $v\!\downarrow$ and avoids repetitions of type (2'). Such an extension does not have to exists so d is a partial function. If there are many extensions we pick the least one wrt some fixed predefined order. We extend the function d in a natural way to the function that takes some set of vertices instead of just one vertex.

We analyse algorithm 2 that tries to find a total l-coloring that avoids l-repetitions of type (2'). We say that during the evaluation of the algorithm a value $C(v)$ is assigned *explicitly* if it is assigned by an instruction in line 5. Function `Next` used by the algorithm is defined on listing 3. For a vertex v from a (directed) meta-path $\mathbb{P}$ vertex `right(v)` is the next vertex on the path $\mathbb{P}$ and `left(v)` is the previous vertex of the path $\mathbb{P}$. Each of those functions is undefined on one end of each meta-path.

Algorithm 2: Elimination of l-repetitions of type (2)

1 **input:** $S : \mathbb{N} \to [\binom{N}{n}]$
2 `i` $\leftarrow 1$, $C \leftarrow \emptyset$
3 $v \leftarrow$ root of T
4 **while** *T is not entirely l-coloured* **do**
5     $C(v) \leftarrow$ the sublist of L_v of length n with index $S(i)$
6   `i++`
7   **if** *there is an l-repetition P of type (2')* **then**
8   **let** P' be the repeated part of P
9   **for** $v' \in P'\!\downarrow$ **do**
10   erase the value of $C(v')$
11   **end**
12         $v \leftarrow$ the point of P' that is closest to the root
13   **else**
14         $(C, v) \leftarrow \text{Next}(C, v)$
15   **endif**
16 **end**

Suppose that a tree T can not be l-coloured as claimed by the lemma. Let us run the algorithm for M steps. After every iteration of the main loop partial colouring C does not contain any l-repetition of size (2) so it can never be a total function. In particular it means that the algorithm never stops by itself. Moreover, whenever some l-repetition occurs, the currently coloured vertex v must be an end of the path containing the l-repetition. Note that at every moment of evaluation of the algorithm the vertices with explicitly assigned sublists form a path. Indeed the next vertex can change in two ways. If it is chosen by function `Next` then it is adjacent to v. On the other hand if some explicitly assigned sublists have been retracted, the retracted path is a suffix of explicitly assigned path, so the remaining explicitly assigned set still forms a path and the next vertex to be coloured is adjacent to its end. Observe also that at any moment of the evaluation of the algorithm the partial l-colouring of a tree is a determined by explicitly coloured vertices. Moreover i-th vertex of explicitly colored path $(v_1, \ldots, v_r)$ is determined by the sublists assigned to $(v_1, \ldots, v_{i-1})$ (v_1 is always the root).

In general function `Next` is undefined for some arguments (e.g. v is already first vertex of meta-path and have no children). However, since we assumed that the tree can not be l-colored without l-repetitions of type (2') and function `Next` always chooses a subtree for which current partial coloring C can not be extended, some l-repetition must occur before hitting a vertex for which `Next` is undefined.

Algorithm 3: Function Next

```
input: C partial l-coloring of the tree
input: v vertex the tree
if d(C, v) is not defined then
    let (v_1, ..., v_k) be the enumeration of children of v
    let j be the greatest number for which d(C, {v_1, ..., v_j}) is defined
    v' ← v_{j+1}
    C' ← d(C, v_1, ..., v_j)
else
    C' ← d(C, v)
    if v is the central point of meta-path ℙ then
        let B be the set of vertices on the left of v on ℙ together with v
        if d(C, B) exists then
            v' ← right(v)
            C' ← d(C, B)
        else
            v' ← left(v)
        endif
    endif
    else if v is on the left of the central point of meta-path ℙ then
        v' ← left(v)
    endif
    else if v is on the right of the central point of meta-path ℙ then
        v' ← left(v)
    endif
endif
return (C', v')
```

Let us run the algorithm for some input S for M steps. We record information about evaluation of the algorithm in a tuple (P, F, R, G) where F and R are defined as in the proof of Lemma 2.4 P is a sequence of p's where p_i is a number of **explicilty** coloured vertices after i steps of an evaluation of the algorithm and G is the sequence such that g_i is the number of vertices of a gap that belongs to the root-path. By the definition every g_i is not larger than the length of corresponding repetition.

Note that vertices on the root-path can be coloured only explicitely and last colouring which created repetition had to be also exlicit, so when a repetition of length r occurs it can contain only explicitly coloured vertices since they are forming a path.

Compression We analyse the asymptotic number of tuples (P, F, R, G) that describes the first M steps of the evaluation of the algorithm on some inputs, with M tending to infinity. We can bound P, F and R just like in the proof of Lemma 2.4. Also number of possible G is bounded in similar way since parts of gaps are bounded by the repetition length just like whole gaps in the Lemma 2.4. Taking all estimation together we get that the number of possible descriptions is:

$$o(\left(2^5n\binom{N-1}{n-1}\right)^M) = o(\left(2^5n\frac{n}{N-n+1}\binom{N}{n}\right)^M) = o(\binom{N}{n}^M).$$

That shows that, for sufficiently large M, the number of possible descriptions is smaller than the number of possible initial segments of length M of an input sequence S. To obtain a contradiction we show that descriptions (P, F, R, G) are different for input sequences which differ on initial segment of length M.

Losslessness We prove that it is possible to reconstruct the first M elements of input sequence S from a tuple (P, F, R, G). Given (P, F, R, G) we are going to decode $S(M)$ and (P', F', R', G') - tuple constructed by algorithm running $M-1$ steps on our particular S. Then by simple iteration we will extract all $S(i)$ for $i = 1, \dots, M$.

If $p_M = p_{M-1} + 1$ then no repetition was retracted in the last step. From every partial colouring, in particular from F, we can deduce which values was assigned explicitly. The vertices with these values form a path $(v_1, \dots, v_m)$ with v_1 being the root. Then $S(M)$ is simply the index of sublist assigned to v_m in F. Then P' is P without the last element, $R' = R, G' = G$ and F' is a partial colouring determined by the sublists assigned (explicitly) to $(v_1, \dots, v_{m-1})$ in F.

If $p_M = p_{M-1} - k + 1$ where $k > 0$ then in the length l-repetition (of type (2')) was retracted. Let g, r be the last elements of G, R. Let $r = (\alpha_1, \dots, \alpha_k)$. We reconstruct a colouring that contained the repetition in k phases. Starting from F we determine the path of vertices $(v_1, \dots, v_m)$ with explicitly assigned sublists. That path with assigned sublists determines uniquely the next vertex v' to be explicitly coloured. It is also the last vertex whose assignment was retracted in the i-th step of the algorithm. We know also that the sublists previously assigned to v' had nonempty intersection with the list assigned to some vertex on the root-path. That vertex w is uniquely determined by value g and position of v' within the tree. Vertex w still have assigned some sublist A and we can read it form F. Finally we know that α_k is the index of the sublist previously assigned to v' from among n-sublists of $L(v')$ having nonempty intersection with A. That enables to reconstruct the sublist preciously assigned to v'. After k such phases we obtain a partial coloring C from just before the l-repetition was retracted. Let $(v_1, \dots, v_m)$ be the path starting from the root with explicitly assigned colors in C, then $S(M)$ is simply the index of sublist assigned to v_m in C. Then P', R', G' are the corresponding sequences without their last elements, and just like before F' is a partial colouring determined by the sublists assigned explicitly assigned (explicitly) to $(v_1, \dots, v_{m-1})$ in C.

□

Lemma 2.6. *For every k there exists such N_k that if T is a tree of path-width k with lists of size N_k in each vertex and without repetitions of type* (1) *it is possible to choose sublists of size n in such a way that there is no colouring from sublists with a repetition of type* (2)*.*

Proof. For the proof it is enough to apply procedure from 2.5 separately for every meta-path together with its downset. In this way we will forbid repetitions of type (2) with stable part on any of this paths. Note that every path has at most $k-1$ ancestors in the tree so its list will be shortened at most k times and hence there is bounded required size of lists. □

Lemma 2.7. *For every k there exists such N_k that if T is a tree of path-width k with lists of size N_k in each vertex and without repetitions of type* (1) *it is possible to choose sublists of size n in such a way that there is no colouring from sublists with a repetition of type* (3)*.*

Proof. It is enough to direct meta-paths of the decomposition into the other direction and apply Lemma 2.6. □

Lemma 2.8. *If T is a tree of path-width k with lists of size at least 2^{k+1} which contains no l-repetitions of types* (1), (2), (3) *then it can be nonrepetitively coloured from these list.*

Proof. Let us order the vertices of the tree wrt increasing distance from the root (breaking ties in an arbitrary way). We are going to color greedily vertices in its order, but each time a vertex v get color c, this color is erased from the lists of all vertices of $v\downarrow$. Since the depth of the contracted tree is smaller than the length of the lists no list will ever be empty. Suppose that there is a repetition in a tree coloured in this way. If it is a one-way repetition then the mid-edge must by on the same level as the lower end of the repetition (EXPLAIN). Then it it easy to check that the repetition is either of type (1) or (2) or (3) but those have been excluded.

Note that we had already ruled out all one-way repetitions so we only need to deal with two-way ones (DEFINE two-way). Just like in the previous case a two-way repetition must have its mid-edge on the stable part. Then the mid-edge is closer to one ends of the stable part, suppose wlog that to the right end. In this situation it forms also a repetition of type (3) which is impossible by assumption. □

Theorem 2.1 follows by application of Lemmas 2.4, 2.6, 2.7 and 2.8. The necessary bound on the lengths of lists are expressed only in terms of path-width of the tree.

3. General graphs

Example 3.1. Let $D_{n,k}$ be a path of length $2n$ with every odd vertex blown up to $\binom{kn}{k}$ vertices. Let the lists on even vertices be all disjoint and of length k and lists on all copies of every blown up vertex be all possible k-tuples of colours used on even vertices.

Claim 3.2. *Every colouring of $D_{\lceil e^k \rceil,k}$ have at least one repetition.*

Proof. In the proof by D_i we will mean generalised vertex on i-th position of $D_{\lceil e^k \rceil,k}$, which is regular vertex or vertex blown up to $\binom{kn}{k}$ vertices. A segment $[D_i, D_{i+2j}]$ is a repetition iff there is a path of length $2j$ starting in one of vertices of D_i and eding on one of vertices in position D_{i+j} which is a repetition. Let C_k be a set of colours used in colouring C on vertices on position k(consisting of one colour for even k). Note that $C_{2k} = C_{2l}$ only if $k = l$ and for every $2k+1$ there is at most $k-1$ colours not in C_{2k+1}.

Suppose now that there is a k-colouring C of $D_{\lceil e^k \rceil,k,k}$ such that no segment is a repetition. Lets keen now only on odd repetitions. Note that for every segment $[D_i, D_{i+4j+2}]$ there must be an even l such that $C_l \not\subset C_{l+2j+1}$ if $l \leqslant i+2j+1$ or $C_l \not\subset C_{l-2j-1}$ otherwise. Moreover if colour C_{2i} is forbidden in $C_{2i+2l+1}$ ($l \in (Z)$) then it forbids only repetitions of length $|2l+1|$ containing D_{2i} and $D_{2i+2l+1}$ so only $2l+1$ repetitions. Since for $D_{n,k}$ we need to forbid $\lceil \frac{n}{2l+1} \rceil$ colours in odd vertices to forbid all repetitions of length $2l+1$ so to forbid all repetitions we need to forbid $\Sigma_{i=1}^{n}\lceil\frac{n}{2i+1}\rceil \geqslant \frac{1}{2}nH_n$ colours in total, what makes an average of $H_{\lceil e^k\rceil}$ colours per odd vertex which is more than k.

References

[1] F. Fiorenzi, P. Ochem, P. Ossona de Mendez, and X. Zhu, *Thue choosability of trees*, Discrete Appl. Math., 159 (2011), pp. 2045–2049.

[2] J. Grytczuk, J. Kozik, and P. Micek, *A new approach to nonrepetitive sequences.* to appear in *Random Structures Algorithms.*

[3] J. Kozik and P. Micek, *Nonrepetitive colorings of trees.* submitted.

[4] A. Thue, *Über unendliche zeichenreichen*, Norske Vid. Selsk. Skr., I Mat. Nat. Kl., Christiania, (1906), pp. 1–22.

□