Replace this file with prentcsmacro.sty for your meeting, or with entcsmacro.sty for your meeting. Both can be found through links on the ENTCS Web Page.

An Example Paper

My Name[1,2]

My Department
My University
My City, My Country

My Co-author[3]

My Co-author's Department
My Co-author's University
My Co-author's City, My Co-author's Country

Abstract

This is a short example to show the basics of using the ENTCS style macro files. Ample examples of how files should look may be found among the published volumes of the series at the ENTCS Home Page http://www.elsevier.com/locate/entcs.

Keywords: Please list keywords from your paper here, separated by commas.

1 Introduction

2 Começou

Coloring(r, ℓ) can be easily solved in polynomial time when $(r, \ell) \in \{(0, 1), (1, 0), (1, 1), (2, 0)\}$. Therefore, in order to completely classify the complexity when $r + \ell \leq 2$ remains to analyse the $(0, 2)$-case.

Lemma 2.1 Coloring$(0, 2)$ *can be solved in polynomial time.*

Proof. $(0, 2)$-graphs, also known as co-bipartite, is a subclass of perfect graphs [**?**], consequently its chromatic number equals to its clique number, which can be found in polinomial time using a polynomial algorithm for Independent Set on bipartite graphs (see [**?**,**?**]). □

As $(3, 0)$-graphs (tripartite graphs) have chromatic number at most three, and to verify if the chromatic number of a graph G is one or two is trivial, we know that Coloring$(3, 0)$ can be solved in polynomial time. Next we consider $(4, 0)$-graphs.

Lemma 2.2 Coloring$(4, 0)$ *is NP-Complete.*

Proof. It is well-known that any planar graph is $(4, 0)$ (4-colorable) [**?**], and X and Y provide an $O(n^2)$ algorithm to find a 4-coloring of a planar graph. In addition, to determine whether a planar graph G is 3-colorable is NP-complete [**?**] (even when a $(4, 0)$-partition of G is also provided). Therefore to determine the chromatic number of $(4, 0)$-graphs is NP-complete. □

[1] Thanks to everyone who should be thanked

[2] Email: myuserid@mydept.myinst.myedu

[3] Email: couserid@codept.coinst.coedu

Is noticeable that any (r, ℓ)-graph is simultaneously a $(r+1, \ell)$-graph and a $(r, \ell+1)$-graph, since we can choose any vertex to act as both independent set and a clique. Therefore we can state that the problem is NP-Complete for any (r, ℓ)-graphs $\forall r \geq 4, \ell \geq 0$

The following theorem serves as base for the next demonstrations.

Theorem 2.3 List-Coloring(r, ℓ) *is equivalent to* Coloring$(r, \ell+1)$.

Proof. Let G be the graph of an instance of List-Coloring(r, ℓ), and $C = \{c_1, c_2, \ldots, c_n\}$ the color set formed by the lists of colors for each vertex.

For each color $c_i \in C$ create a vertex u_i and connects it to all others u_j $\forall j < i$. If a vertex $v \in V(G)$ does not have the color c_i in its list, then adds an edge from v to u_i. This steps generates a new graph H. It is simple to notice that if H can be proper colored, then we found a solution to the List-Coloring of G.

The counterpart of the prof can be done by finding the maximum clique Q of H and building the color list of the remaining vertices v as follow:

$\forall u_i \in Q$ if $(v, u_i) \notin E(H)$ adds c_i to the list of v. Note that $H \backslash Q = G$ and therefore a solution for List-Coloring of G can be easily extended to the Coloring of H. □

Based in the previous demonstration we can extract the following corollaries

Corollary 2.4 Coloring$(1, 2)$ *is NP-Complete.*

Proof. Derivates from the NP-Completeness of list coloring on split graphs [?]. □

Corollary 2.5 Coloring$(2, 1)$ is Np-Complete.

Proof. Derivates from the NP-Completeness of bipartite graphs demonstrated by Fellows at [?] □

Now remain to us to demonstrate the complexity of Coloring(0,2). In order to do so we will use the following theorem, which is based on a theorem proposed by Jensen et.al [?]. The proof is based on a reduction of the Restricted 3-SAT problem to the List-Coloring(0,2)

Restricted 3-SAT

Given: A set of unnegated variables $X = \{x_1, \ldots, x_n\}$ and negated variables $\overline{X} = \{\overline{x}_1, \ldots, \overline{x}_n\}$, a collection of clauses $c_1, \ldots, c_m$ such that

- each clause c_i contains two or three different literals $y_{i,k} \in X \cup \overline{X}$
- there are at most three clauses that contain x_j or $\overline{x}_j$ for each $1 \leq j \leq n$.

Question: Does there exists a truth assignment for the variables such that each clause is satisfied?

Theorem 2.6 List-Coloring(0,2) is Np-Complete

Proof. Let I be a instance of a restricted 3-SAT problem, we construct an instance I' of list-coloring(0,2) problem as follows:

For each variable j we create six vertices $a_j^{(1)}$, $a_j^{(2)}$, $a_j^{(3)}$; $b_j^{(1)}$, $b_j^{(2)}$, $b_j^{(3)}$. each vertex has a color set as follow:

- $a_j^{(k)} := \{x_j^{(k)}, \overline{x_j}^{(k)} \}$
- $b_j^{(k)} := \{\overline{x_j}^{(k)}, x_j^{((k \pmod 3))+1)} \}$

Let A be the set of all $a_j^{(k)}$ and B the set of all $b_j^{(k)}$ build a clique with $A \cup B$. Note that this clique can only be colored in two manners:

- (1) $f(a_j^{(k)}) = x_j^{(k)} => b_j^{(k)} = \overline{x_j}^{(k)}$
- (2) $f(a_j^{(k)}) = \overline{x_j}^{(k)} => b_j^{(k)} = x_j^{((k \pmod 3))+1)}$

Futhermore, for each clause create a vertex c_i and populate their lists as follows:

For each variable j or the negation of it $\overline{j}$ in the clause i, adds $x_j^{(k)}$ to the list of c_i where k is the current number of occurences of j. For example, let I be the following 3-SAT:

$(p \vee q \vee r) \wedge (\neg p \vee q \vee r) \wedge (\neg p \vee \neg r \vee s)$

their clauses would be transformed in these vertex:

- c_1 with list: $\{p^1, q^1, r^1 \}$
- c_2 with list: $\{\overline{p}^2, q^2, r^2 \}$

- c_3 with list: $\{\overline{p}^3,\ \overline{r}^3,\ s^1\ \}$

Let C be the set containing all c_i build a clique with $C \cup A$.

Using the variable p to demonstrate, if p is TRUE then $a_p^{(1)}, a_p^{(2)}, a_p^{(3)}$ shall be colored with p'^1, p'^2, p'^3, allowing the color p^x,and denying the color p'^x to be choosed to color a clause or another variable in the neighbourhood of $a_p^{(1)}, a_p^{(2)}, a_p^{(3)}$.

Therefore is easy to see that if I' can be list-colored then exists a truth assignment in I that satisfies each clause. And that by construction if a thruth assignment satisfies I then exists a list-coloring of I' in the following way:

$$color(a_j^{(k)}) = \begin{cases} x_j^{(k)}, \\ \overline{x}_j^{(k)}, \end{cases}$$

□

3 Bibliographical references

References