Computational Complexity **(Due: Flexible)**

Assignment #2 (Arora Barak Ch - 2)

Instructor: Shreesh Maharaj *Name:* Jimut Bahan Pal

Q. 1. Prove that $3 - COL$ is **NP**-complete.

Soln. We will show that [1, 2] $3 - SAT \leq_P 3 - COL$, to prove that $3 - COL$ is NP-hard. Given an instance ϕ of $3 - SAT$, we will construct an instance of $3 - COL$, i.e., a graph $G(V, E)$ is 3-colorable iff ϕ is satisfiable.

> **Defn.:** Given a graph $G(V, E)$, return 1 if and only if there is a proper coloring of G using at most 3 colors.

Our verifier takes a graph $G(V, E)$ and a coloring c, and checks in $O(n^2)$ time whether c is a proper coloring by checking if the end points of every edge $e \in E$ have different colors.

We consider ϕ as a $3 - SAT$ instance and $C_1, C_2, ..., C_m$ be the clauses of ϕ defined over the variables $x_1, x_2, ..., x_n$. The graph needs to capture two things, mainly, establish the truth assignment for $x_1, x_2, ..., x_n$ via the colors of its vertices and somehow capture the satisfiability of every clause C_i in ϕ.

We create a triangle in G with three vertices T, F, B where T is True, F is False and B is Base. These can be thought of colors we will use to color the vertices of G. Since the triangle is a part of G, we need 3 colors to color G. We add two vertices v_i, $\bar{v}_i$ for every literal x_i and create a triangle B, (v_i, $\bar{v}_i$) pair as shown in Figure 1.

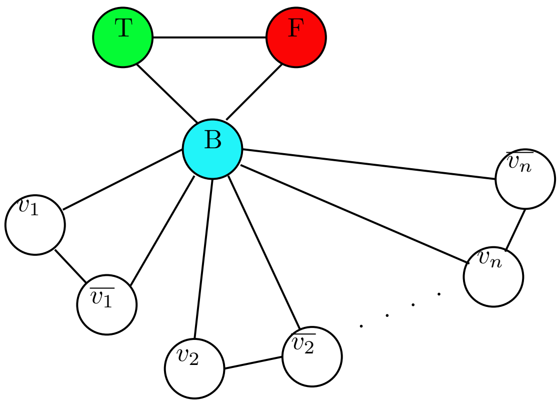

Figure 1: The coloring of the Graph G.

Since, G is 3-colorable, either v_i or $\bar{v}_i$ gets the color T, we just interpret this as the truth assignment to v_i. We now need to add constraints to G to capture the satisfiability of the clauses of ϕ. To perform this we introduce the **Clause Satisfiability Gadget**, a.k.a the OR-gadget.

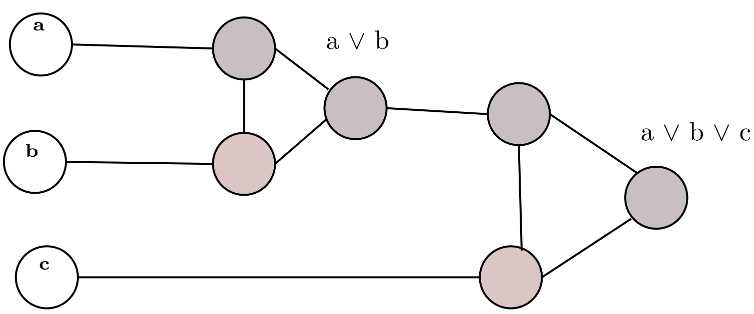

Figure 2: The OR-gadget construction where **(a ∨ b ∨ c) is the output node**.

For a clause $C_i = (a \vee b \vee c)$, we need to express the OR of its literals using our colors T, F, B. The OR-gadget for the gadget graph is constructed as shown in Figure 2. The output of this gadget graph's node shall be colored T if C_i is *satisfied* and F otherwise. The node labelled (a ∨ b) captures the output of (a ∨ b) and we repeat the operation for ((a ∨ b) ∨ c). The gadget satisfies the following three properties:

1. If a, b, c are all colored F in a 3-coloring, then the output node of the OR-gadget **has** to be colored F. Thus capturing the unsatisfiability of the clause $C_i = (a \vee b \vee c)$.
2. If one of the a, b, c is colored T, then there exists a **valid** $3 - coloring$ of the OR-gadget where the output node is colored T. Thus again capturing the satisfiability of the clause.

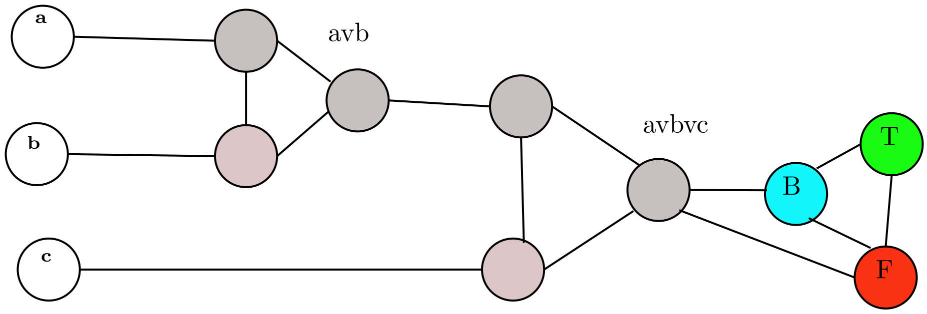

Figure 3: The modified OR-gadget by adding the base vertex and false vertex to the output node.

Once we add the OR-gadget of every C_i in ϕ, we connect the output node of every gadget to the base vertex and to the false vertex of the initial triangle as shown in Figure 3. We can prove that our initial 3-SAT instance is ϕ is satisfiable if and only if the graph as shown in Figure 3 is 3-colorable. Suppose ϕ is satisfiable and let $(x_1^*, x_2^*, ..., x_n^*)$ be the satisfying assignment. If x_i^* is assigned True, we color v_i with T and $\bar{v_i}$ with F, which is a valid coloring. Since ϕ is satisfiable, every clause $C_i = (a \vee b \vee c)$ must be satisfiable, hence output node is colored T. Conversely, if G is 3-colorable. We construct an assignment of the literals ϕ by setting x_i to True if v_i is colored T and vice versa. Now suppose this assignment is not a satisfying assignment to ϕ, then this means there exists at least one clause $C_i = (a \vee b \vee c)$ that was not satisfiable. That is, all of a, b, c were set to False. But if this is the case, then the output node of corresponding OR-gadget of C_i must be colored F. But this output node is adjacent to the False vertex colored F ; thus contradicting the 3-colorability of G.

References

[1] Lalla Mouatadid. Introduction to complexity theory: 3-colouring is np-complete, 2014. `http://cs.bme.hu/thalg/3sat-to-3col.pdf` last accessed June 7, 2020.

[2] Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach.* Cambridge University Press, USA, 1st edition, 2009.