CS 325: Analysis of Algorithms
Group Assignment 4

Busra Dedeoglu
Hyun Taek Oh
Yu-Hao, Shih

14 March 2024

Description of Algorithm

In this assignment, we are required to get answers such as “Yes” or “No” from the program that turns off all the lights in the OSU campus. To find the answers, we need to convert the lights and switches problem to 2-CNF problems since turning a switch twice means not turning it, and changing the order of toggling the switches does not change the results. The reduction from the problem to 2-SAT, which encapsulates the problem within black box that is used as 2-CNF problem and can be applied for solving 2-SAT problems. We will use the given 2-SAT solver, which uses Kosaraju’s algorithm and finds contradiction with strong connected components.

We are required to define Boolean variables, which represent the action of toggling the switches. If the switch is toggled, the variable can be True whereas if the switch is not toggled, the variable can be False.

To build constraints for the 2-SAT problem, the program checks each initial state of the lights one by one and its connection to two switches. For example, one light has two switches: S_1 and S_2. For each light, if the initial state of the light is “1”, which means “ON”, we need to turn off the light to change current state. The lights can be changed by other switch since each light is connected to two switches, which means we carefully choose which switch is toggled. So, we choose the clauses below that work together as a formula, ensuring that at least one switch is toggled.

$$(S_1 \vee S_2)$$

$$(\neg S_1 \vee \neg S_2)$$

$$(S_1 \vee S_2) \wedge (\neg S_1 \vee \neg S_2)$$

On the other hand, if the initial state of the lights is “0”, which means “OFF”, we need to maintain this state. If both switches are toggled, it can be achieved since one switch would turn on and another switch would turn off the light. The clauses below ensure that remaining

its state.

$$(S_1 \vee \neg S_2)$$

$$(\neg S_1 \vee S_2)$$

$$(S_1 \vee \neg S_2) \wedge (\neg S_1 \vee S_2)$$

Thus, these meet the condition of turning off each light or remaining its state. With these clauses, we use the given 2-SAT solver function, which provides the result of the problem that whether 2-CNF is satisfiable or not. Then, we can finally get the answer “Yes” or “No”.

Running Time Analysis

In the reduction process, there are two things we need to consider: the number of lights and switches, which are m and n respectively. For each light, the program generates two clauses checking the connected switches and the light’s initial state, which takes linear time $O(m)$. For each switch, the program changes them into Boolean variables, which are direct and take linear time $O(n)$. Therefore, the overall time complexity of the reduction is polynomial, which is $O(m+n)$, solving the problem efficiently and meeting the condition of polynomial time solvability with 2-SAT algorithms.

Proof of Correctness

To prove that the proposed reduction is correct, we need to show that the original problem of switches and bulbs is satisfiable if and only if the solution to 2-SAT problem using boolean constraints is satisfiable. To show this we need to prove both directions: 1. Prove the solution to 2-SAT problem using boolean constraints is satisfiable if the original problem of switches and bulbs is satisfiable; 2. Prove the original problem of switches and bulbs is satisfiable if the solution to 2-SAT problem using boolean constraints is satisfiable.

For the first assertion, we consider the following array of switches $K = [k_1, k_2, \dots, k_n]$, which makes the original problem satisfiable, e.g., all bulbs (L bulbs) are off. For every bulb in the circuitry, we add a constraint, which relates 2 of the elements in K. That means there are L constraint clauses are generated:

$$C_1 = b_1(k_{x_1}, k_{y_1}) = 0$$

$$C_2 = b_2(k_{x_2}, k_{y_2}) = 0$$

$$\dots$$

$$C_L = b_L(k_{x_L}, k_{y_L}) = 0$$

It is important to note that, if a bulb is connected to more than 2 switches, the logical operation can be represented using binary tree of logical gates, e.g.:

$$C_M = b_M\left(k_{x_M}, k_{y_M}, k_{z_M}\right) = b''_M(b'_M\left(k_{x_M}, k_{y_M}\right), k_{z_M})$$

We here note that this increases the number of constraints to $L' > L$. Once we have all constraints, these constraints are added to 2-SAT solver. Let us assume here that the feasibility solution to $\{C_1, C_2, \dots, C_{L'}\}$ is solvable and at least a light bulb is on. That implies that at least one constraint $C_r = 1$, because, by definition, the bulb r must be off if and only if $C_r = 0$. However, this is a contradiction, because that implies that 2-SAT solver did not satisfy a constraint. Hence, this proves that the solution to 2-SAT problem using boolean constraints is satisfiable if the original problem of switches and bulbs is satisfiable.

For the second assertion, we consider a feasible solution to the constraint set:

$$C_1 = b_1\left(k_{x_1}, k_{y_1}\right) = 0$$

$$C_2 = b_2\left(k_{x_2}, k_{y_2}\right) = 0$$

$$\dots$$

$$C_L = b_L\left(k_{x_L}, k_{y_L}\right) = 0$$

We here notice that a constraint $C_M = 0$ corresponds to making bulb M off. Let us assume this solution leaves at least one bulb on. Then, the constraints corresponding to on bulb must satisfy $C_r = 1$. However, this leads to a contradiction since $C_r = 0$ was the original constraint which led to a feasible solution to 2-SAT problem. Therefore, $C_r = 0$ must be true, and hence the original problem of switches and bulbs is satisfiable if the solution to 2-SAT problem using boolean constraints is satisfiable.

Since we now proved both assertions, we prove that the original problem of switches and bulbs is satisfiable if and only if the solution to 2-SAT problem using boolean constraints is satisfiable. This completes the proof.