On the complexity of various Lights Out variants

Daan van Berkel
Wieb Bosma

February 2, 2024

Abstract

We explore the complexity of various Lights Out variants.

1 Introduction

We will report on the complexity of solving various variants of Lights Out.

In general there are two problems

- *Existance* does a solution for the specific instance of a Lights Out problem exist.
- *Solution* what is a solution for the specific instance of a Lights Out problem, if it exists.

Each section will focus on a different class of problems.

2 Standard Lights out

Theorem 1 *The existance and the solution problem for standar lights out are in P.* ◇

Proof (TODO refer to standard result?) both problems use solving a linear set of equations over a suitable ring, which is polynomial. □

3 Nice Graphs

We call a graph $G := (V, E)$ *nice* if and only if it has the following properties

1. *reflexive* there is an self-edge for each vertex of G
2. *symmetric* if $(u, v) \in E$ then also $(v, u) \in E$
3. *equal weight* Each edge has the same weight.

Without the loss of generality, for nice graphs we can set the weights to 1.

Proof (TODO talk about equivalent problems) There is an *equivalent* problem with weights 1 for each nice graph. □

Lemma 2 *Given a nice graph G, and an instance L to the standard problem on G with a solution S.*

For every lit button $b \in L$ there is a button $s \in S$ within $d(b, s) \leq 1$. ◇

Proof Assume the contrary. I.e. All buttons in the solution have a distance greater then 1 to all lit buttons of L. Since the effect of pressing a button has reach 1, no lit button in L will change when pressing any $s \in S$. Contrary to S being a solution. □

Remark Something more general is true. G does not have to be a nice graph. ◁

Lemma 3 *Given a nice graph G*

A solution to the restricted problem on G exists if and only if a solution to the standard problem exists. ◇

Proof $\Longleftarrow$ A solution to the restricted problem is also a solution to the standard problem.

$\Longrightarrow$ We will use induction on the length of a solution to show that a solution S to the standard problem on G can be transformed to a solution S' to the restricted problem on G.

Assume all solution S to the standard problem on G with $|S| < k$ can be transformed to solution to the restricted problem. We will show that a solution S with $|S| = k$ can also be transformed. Let S with $|S| = k$ be given. There are two situations to consider.

1. S contains a lit button.
2. S does not contain a lit button.

Without loss of generality, if we are in situation 1, we assume that the first button in S we press is lit, since we can freely reorder the sequence of button presses in a solution for the standard problem. Since this button is lit we can press it. The remaining solution for the standard problem can be transformed to a solution to the restricted problem by the induction hypotheses.

In situation 2 we can apply lemma 2. Pick a lit button b and choose button $s \in S$ with $d(b, s) = 1$. Without loss of generality we can assume that s is the first button in S. The following alternate press sequence allows us to press s without altering the resulting light pattern. Assume $\upsilon(b) = k$.

1. press button b exactly $q - k$ times. Since G is reflexive b is now unlit. Since $d(b, s) = 1$ we find s in state $\upsilon(s) = q - k$ and therefor lit.
2. press button s which is possible because s is lit. Since G is symmetric s effects b, so b becomes lit.
3. press button b exactly k times.

Notice that this is a valid press sequence in the restricted problem. Furthermore button b is pressed q times, so it has not effected the resulting light pattern. More importantly the above sequence presses s.

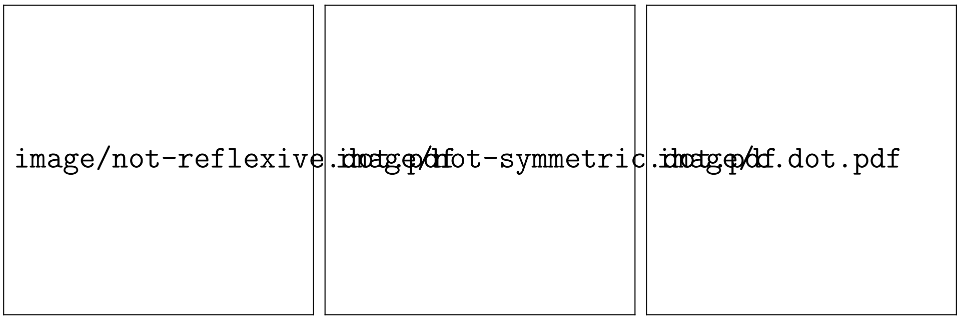
The remaining solution for the standard problem can again be transformed to a solution to the restricted problem by the induction hypotheses.

So by induction we have shown that any solution to the standard problem on G can be transformed to a solution to the restricted probem on G. □

Corollary 4 *Let S be a solution to the standard problem and S' the corresponding solution to the retricted problem.*

Then $|S'| \leq (q + 1)|S|$. ◇

Proof In the worst-case scenario we need to transform every $s \in S$ by our alternate press sequence. In each such transform we press $q+1$ buttons. □



(a) Not reflexive (b) Not Symmetric (c) Not same Weight

Figure 1: Counter examples

Theorem 5 *For a nice graph G we have*

> *The existance and solution problems for the restricted Lights Out variant are in P.* ◇

Proof By lemma 3 there exist a solution to the restricted problem if and only if there exists a solution to the standard problem. By theorem 1 the existance of a solution for the standard problem is in P.

The result for the restricted solution problem follows from theorem 1 a solutions to the standard problem can be found in polynomial time. By lemma 3 and corollary 4 it can be transformed in polynomial time to a solution for the restricted problem. □

In general, when our graph is not nice we loose the property that the solutions to the lit problem are equivalent to the standard problem.

Remark For the not reflexive counter example we have an $\upsilon(a) = 0$ and $\upsilon(b) = 1$. A standard solution is to press a, which can be achieved in lit problem. Pressing the only lit button results in $\upsilon(a) = 1$, pressing any button in this state transforms back to the original or similar state.

For the not symmetric case with only $\upsilon(b) = 1$ and the rest is off, we have a solution by pressing c, d, e, a. Since neither of c, d, or e is lit, and will never be lit this solution can not be transfered from the standard problem to the lit problem. ◁

4 restricted

Theorem 6 *The existance and solution problems for restricted lights out is NP-hard.* ◇

image/a.dot.pdf

(a) clock

image/b.dot.pdf

(b) register

image/c.dot.pdf

(c) carry

Figure 2: twilight constructs

Proof (TODO expand proof) reduce SAT to restricted lights out. □

5 twilight

Theorem 7 *the solution problem is not in NP.* ◇

Take a look at figure 5. It displays various constructs used in the proof of theorem 7.

Construct 5 is called a clock. It has two vertices, which are both in the *active* state. The top vertex can be used to kill the clock. Pressing it will change the state of the clock vertices to 0, which is an *inactive* state. The bottom vertex of the clock construct will send a signal through the output edge. This signal can be used to increment the vertex that will be connected to it.

Construct 5 of figure **??** is called a register. It consist of a sole vertex connected to the input and output. Initially this vertex is in the state 0. It takes at least $q - 1$ input signals before the *register* construct can pass a signal to its output.

The last construct, 5 is called a carry. It is a sole vertex connected to the input. Initially the carry vertex is in state 1. Since it is only influenced by the input, it needs $q - 1$ input signals before it is turned off.

With these constructs we will define a family of problems $(P_n)_{n=0}^{\infty}$. To define our family we introduce a little notation. a will denote a clock construct, b denotes a register and c denotes a carry. Concatenation uv of verbs u and v means to connect the output of u to the input of v. Exponentiation will be interpreted as iterated concatenation.

With these conventions it is easy to describe our family: $P_n := ab^n c$. In figure 3 you see a depiction of P_3.

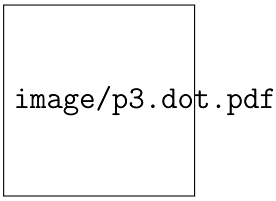


Figure 3: Problem P_3

The crux of the proof of theorem 7 can be found in the following lemma.

Lemma 8 *For construct b^n there need to be at least*

$$\frac{(q-1)^{n+1}-1}{q-2}$$

presses before an output signal occurs. ◇

Proof We will prove this with induction on the size of the construct. Notice that for $n = 0$ the input is directly coupled to the output so each input signal corresponds to one output signal, in accordance with lemma.

Take a look at construct b. Each input signal changes the state of the vertex from $i \mapsto i + 1$. In order for the vertex to reach state $q - 1$ we need at least $q - 1$ input signals. Before an output signal is send the vertex of b should be pressed. This sets the least amount of presses to send an output signal to $(q - 1) + 1$ which equals $\frac{(q-1)^2-1}{q-2}$.

Now assume that construct b^k needs at least $\frac{(q-1)^{k+1}-1}{q-2}$ input signals before an output signal occurs. We will show that the construct b^{k+1} will need at least $\frac{(q-1)^{k+1}-1}{q-2}$ input signal.

Since $b^{k+1} = b^k b$ and we know that b needs $q - 1$ input signals to reach state $q - 1$ before we can press its vertex to produce an output signal, we have for the least amount of presses for b^{k+1}

$$\left(\frac{(q-1)^{k+1}-1}{q-2}\right)(q-1)+1 = \frac{(q-1)^{k+2}-(q-1)}{q-2}+\frac{q-2}{q-2} = \frac{(q-1)^{k+2}-1}{q-2}$$

Proving our lemma. □

The crux can be used in the following theorem

Theorem 9 *The problem P_n needs $\frac{(q-1)^{n+2}-1}{q-2}$ presses to solve.* ◇

Proof Note that $P_n = ab^n c$. c needs $(q-1)$ input signals to transistion from 1 to 0. These need to be delivered by b^n, which by the preceding lemma, needs at least $\frac{(q-1)^{n+1}-1}{q-2}$ presses. Afterwards we need to kill the clock which can be achieved with one press of the kill switch.

So the least number of presses to solve P_n is $\frac{(q-1)^{n+1}-1}{q-2}(q-1)+1 = \frac{(q-1)^{n+2}-1}{q-2}$. □

The proof of theorem 7 is a consequence

Proof $|P_n| = |ab^n c| = |a| + |b^n| + |c| = 2 + n + 1 = n + 3$. By the preceding family the solution length is not bounded by a polynomial in the number of vertices. □