A Note on proving lower bounds on the delay and cost of decoders and encoders

1. An encoder is not specified for every input; it is only specified for inputs of weight 1. This means that the definition of a cone is useless for proving lower bound in encoders: flipping a bit renders the input illegal. However, all we need to show in order to apply the proofs of the lower bound theorems for cost and delay is that there exists a sink $y[j]$ in $DG(C)$ that is reachable from many sources.

   We say that a set of inputs A *influences* the output $y[j]$ if there exists two input strings X, X' that are identical outside A (and are different in A).

   If we can prove that A influences $y[j]$, then it follows that at least one of the inputs of A reaches the output $y[j]$.

   If we can present k disjoint sets $A_1, \ldots, A_k$ of inputs that influence $y[j]$, then it follows that at least k inputs (one from each set A_i) reaches $y[j]$. Then we have a $\Omega(\log k)$ lower bound on the delay and an $\Omega(k)$ lower bound on the cost.

   We apply this technique to an encoder. We show that every two consecutive inputs $\{x[i], x[i+1]\}$ influence the output $y[0]$.

   There are $2^n/2$ such sets, and hence we obtain the desired bounds.

   We remark that a similar technique can be applied if one can show many subsets that influence different outputs. In that case, if we can show that outputs are not directly fed by inputs, then at least one input from each influencing set is fed to a non-trivial gate. This leads to a lower bound on the cost.

2. A decoder has n inputs and 2^n outputs. If we apply "cone-arguments", then the strongest bound we can hope for is $\Omega(\log n)$ delay and $\Omega(n)$ cost. We strengthen the $\Omega(n)$ lower bound on cost to $\Omega(2^n)$ as follows.

   We show that every output terminal y_j is connected to a distinct non-trivial gate. To prove this we need to show several things:

   (a) The output terminal is not constant (i.e. it depends on the inputs). For that, we need to show that there exist two assignments of input variables, $X_1 = [x_{1_1} \ldots x_{n_1}]$ and $X_2 = [x_{1_2} \ldots x_{n_2}]$ such that $y_j(X_1) \neq y_j(X_2)$. It is easy to find such inputs in the case of decoders.

   (b) The output terminal y_j is not directly hard-wired to any input terminal. For that, we need to show that for all $i = 1 \ldots n$, there exists an assignment of variables $x_1 \ldots x_i \ldots x_n$ such that $y_j \neq x_i$. Formally stated:

   $$\forall i \in [1, n] \; \exists x_1 \ldots x_i \ldots x_n : y_j \neq x_i$$

   (c) Finally, in order to be able to count the gates by counting the output terminals, we need to show that no two output terminals can be fed by the same gate. Otherwise, maybe the number of gates is actually much smaller than the number of output terminals. To show that two output terminals y_j and y_k cannot be fed by the same gate, it suffices to show that there exists an input assignment $X = [x_1 \ldots x_n]$ such that $y_j(X) \neq y_k(X)$.

3. **Incorrect usage cone-arguments.** When considering the cone of a function for the purpose of calculating the lower bound on its delay and cost, note that you should always take the maximum size of the individual cones of each of the output bits, i.e $\max_{j\in[1,n]}\{|cone(y_j)|\}$. Considering the union of the cones of different output bits is useless in this respect.

   For instance, consider a bitwise-NOT(n). Evidently, the size of the union of the cones of the output bits, $|cone([n-1:0])|$ equals n. However, the cone of each individual output terminal is exactly 1. It is wrong to combine the cones and argue a $\Omega(\log n)$ lower bound on the delay.