The Gate Elimination Method

Uses induction to provide one of the strongest known lower bounds for complete bases.

Method:

a) Show that assigning values to a few variables result in function of the same type.

b) Count number of eliminated gates. After all variables eliminated, function is constant.

c) The circuit size is at least number of gates eliminated, giving lowerbound.

Definition $Q_{2,3}^{(n)}$ is the set of $f : \mathbb{B}^n \rightarrow \mathbb{B}$ such that for every pair of variables $x_j$ and $x_k$, $f$ has at least three distinct subfunctions as the two variables range over all four values. Also, there is an $x_j$ such that for some value $c_i$ the subfunction obtained is in $Q_{2,3}^{(n-1)}$.

Theorem Over the basis of all Boolean functions on 2 inputs, $f$ in $Q_{2,3}^{(n)}$ for $n \geq 3$ has $C(f) \geq 2n-3$.

Proof $f$ depends on each of its variables because if not, there is an $x_i$ such that $f$ doesn’t depend on it. If so, then picking any second variable, $f$ has at most two subfunctions, a contradiction.

We show that fan-out from some input is at least two. Consider gate $g$ such that the path to the output from it is longest. Both of its inputs are from variables. If each input to that gate has fan-out one, $f$ has at most two subfunctions on these input variables, a contradiction. Thus, for $n=3$, there are at least 4 edges from inputs from which the lower bound follows from the simple linear lower bound.

Assume that $C(f) \geq 2n-3$ for $n \leq k$. For $n = k+1$, fix an input with fan-out 2, thereby deleting two gates.

Lemma For $n \geq 3$, $Q_{2,3}^{(n)}$ contains $f_{mod\ 3,c}^{(n)}$ where $f_{mod\ 3,c}^{(n)}(x_1, x_2, ..., x_n) = ((y+c) \ mod\ 3) \ mod\ 2)$ for $c$ in $\{0,1,2\}$ where $y = x_1 + x_2 + ... + x_n$.

Proof The functions $f_{mod\ 3,c}^{(i)}$ for $c$ in $\{0,1,2\}$ are $x_1$, $\overline{x}_1$, and 0, respectively. For $n = 2$, $f_{mod\ 3,c}^{(2)}$ has value 1 exactly when $y = 1, 0, 2$ for $c$ in $\{0,1,2\}$, respectively. We now show that the property holds for $n \geq 3$.

In $f_{mod\ 3,c}^{(n)}(x_1, x_2, ..., x_n)$ fix any two variables and let $y^*$ be the sum of the remaining $n-2$ variables and $c^*$ be the sum of $c$ and the values of the two fixed variables. Then, $((y+c) \ mod\ 3) \ mod\ 2) = (((y^* \ mod\ 3 + c^* \ mod\ 3) \ mod\ 3) \ mod\ 2)$. Since the value of $y^* \ mod\ 3$ is in $\{0,1,2\}$ and $c^* \ mod\ 3$ has values in $\{0,1,2\}$, from above $(((y^* \ mod\ 3 + c^* \ mod\ 3) \ mod\ 3) \ mod\ 2)$ has one of 3 different functions. With $c_i = 0$ resulting subfunction is in $Q_{2,3}^{(n-1)}$.♥
The Gate Elimination Method

We derive a lower bound using this method for the storage access function \( f_{SA}(a_{k-1}, \ldots, a_0, x_{n-1}, \ldots, x_0) = x_{|a|} \) where \(|a|\) is the integer specified by \((a_{k-1}, \ldots, a_0)\).

This function can be implemented with inputs having fan-out one. Thus, we have to look closely at the functions used in the basis.

**Definition** \( f : B^{n+k} \rightarrow B \) belongs to \( F_{s}^{n,k}, 2^k \geq n \) if for some set \( S \) subset of \( \{0,1,\ldots,n-1\} \), \(|S| = s\) if the following holds for \(|a|\) in \( S \).

\[
f(a_{k-1}, \ldots, a_0, x_{n-1}, \ldots, x_0) = x_{|a|}
\]

Clearly \( f_{SA} \) is a member of \( F_{s}^{n,k} \). We show that every function in \( F_{s}^{n,k} \) has circuit size at least \( 2s-2 \).

Let \( B_2 \) be the basis containing all Boolean functions on two inputs.

The Gate Elimination Method

**Theorem** Let \( f : B^{n+k} \rightarrow B \) be in \( F_{s}^{n,k}, 2^k \geq n \). Over \( B_2 \), \( C(f) \geq 2s-2 \).

**Proof**

Proof is by induction on \( s \). Base case \( s=1 \) is obvious. Suppose hypothesis of the theorem holds for \( s = s'-1 \). We show that it holds for \( s = s' \). Cases:

a) For \( i \) in \( S \), \( x_i \) has fan-out 2. Replacing \( x_i \) by right constant gives \( f' \) in the class with \( s = s'-1 \) and eliminates two gates.

\[
C(f) \geq 2 + C(f') \geq 2 + 2s'-2 = 2s -2
\]

Note: all gates \( B_2 \) are either AND-type (computing AND or OR) or XOR-type (computing \( x_i \) XOR \( g \) XOR \( c \))

b) For \( i \) in \( S \), \( x_i \) has fan-out 1 to AND-type gate computing function \( (x_i^a \) AND \( g^b \) XOR \( c \)) . Setting \( x_i^a \) to 0 makes gate constant, eliminating it & \( \geq 1 \) successor.