Recent advances in Complexity CIS 6930/CIS 4930 September 03, 2002

Lecture 3

Lecturer: Dr. Meera Sitharam Scribe: Andrew Lomonosov

Today we are going to look at some basic facts about Boolean Circuits (BC) to build up some intuition (first two items below).

Overall plan:

1. Depth 2 boolean circuits
2. Arbitrary depth
3. Best known size lower bounds,
   - with no restrictions on depth,
   - Hastad's exponential size lower bounds for bounded depth circuits for parity.

Which boolean functions can we compute using boolean circuits?

We will need to build some basic circuit intuition before we go into showing complicated lower bounds. First we will note that the answer to question above is that any function

$$f = \{0,1\}^n \to \{0,1\}$$

can be computed by depth 2 boolean circuit.

Wlog we can assume a depth 2 unbounded fan-in circuit is either an OR-of-ANDs (DNF formula) or AND-of-ORs (CNF formula), see Figure 1.

All circuits we will be talking about are assumed to be of unbounded fan-in and have Boolean gates $\{\vee, \wedge, \neg\}s$, unless indicated otherwise.

Take a boolean function and corresponding truth table, which has n variables and 2^n rows.

x_1	x_2	x_3	
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

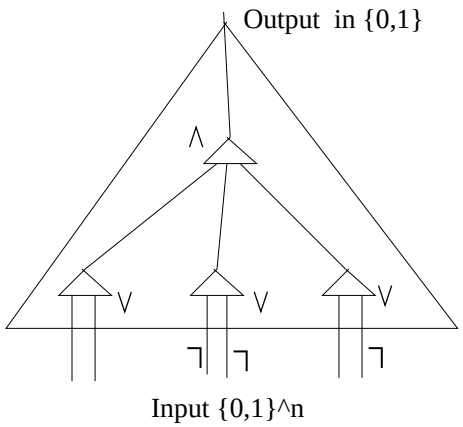

Figure 1: Depth 2 tree of alternating ANDs and ORs

How to write this function as a circuit?

Take all ones and construct *disjunctive normal form (DNF)*

$$f(x) = (\overline{x_1} \wedge \overline{x_2} \wedge \overline{x_3}) \vee (\overline{x_1} \wedge x_2 \wedge \overline{x_3}) \vee (\overline{x_1} \wedge x_2 \wedge x_3)$$

(further optimizations are possible even in depth 2 circuits - use Karnaugh maps).

Therefore all boolean functions can be computed by depth 2 boolean circuits. But the size of such circuits is potentially exponential.

Fact 1 *In [Muller, late 60's] it was shown that there exist a boolean circuit of size $O(2^n/n)$ for any boolean function of n variables.*

Electrical engineers use various heuristics (Karnaugh maps, Quine McCluskey method etc) to try to reduce number of gates used.

Aside - the problem of minimization is NP-hard. Input to this problem is a boolean function f and number K, output is "yes" iff f has a circuit of size no more than K. This problem is also in NP, hence it is NP-complete.

Now lets get back to question of "what can we do with depth 2 circuits of reasonable size"?

Parity function

$$f : \{0,1\}^n \to \{0,1\}$$

$$f(x_1, \ldots, x_n) = (\Sigma_{i=1}^n x_i) mod 2$$

Theorem 2 *The best depth 2 circuit for this function must have size at least $\Omega(2^n)$ (Lupanov 62).*

So what is it about the parity function that makes it difficult to implement it using few gates?

First thing one should try when proving lower bound is to construct the best possible circuit. Easiest circuit is DNF.

How do we optimize circuits using Karnaugh?

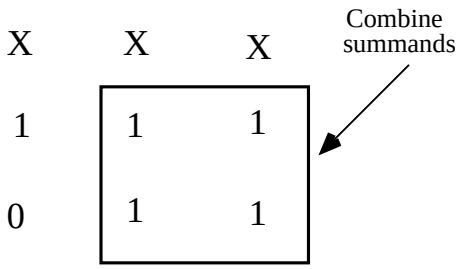

Figure 2: Karnaugh map

Slight detour - a variable i has *non-zero influence* on function f if there is some assignment a to $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n$ for which $f(a_1, \ldots, a_{i-1}, 0, a_{i+1}, \ldots, a_n) \neq f(a_1, \ldots, a_{i-1}, 1, a_{i+1}, \ldots, a_n)$.

Definition 3 *influence of variable i on function f*

$$I_i(f) = \frac{|\{a_1 \ldots a_{i-1} a_{i+1} \ldots a_n : \{0,1\}^{n-1}\}|}{2^n},$$

$$s.t f(a_1, \ldots, a_{i-1}, 0, a_{i+1}, \ldots, a_n) \neq f(a_1, \ldots, a_{i-1}, 1, a_{i+1}, \ldots, a_n)$$

Intuitively parity function has high influence for each variable and this makes it a hard to compute (up to a point).

Fact 4 *(second relationship between depth and size of circuits) if $d_C(f) \geq \Omega(log^2 n)$ then $L_C(f) \geq \Omega(n^{\log n}) = \Omega(2^{\log^2 n})$, where $d_C(f)$ is the depth of boolean circuit C of function f and $L_C(f)$ is the size of circuit C.*

Definition 5 *Minterm(f) is a prime implicant of f. Prime implicant of f is any minimal (i.e none can be removed) set of variables which when fixed, fix the function f to be 1 as well.*

For example

x_1	x_2	x_3	
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

One implicant is $\overline{x_1} x_3$, i.e $x_1 = 0, x_3 = 1$. The following implication is valid: if $\overline{x_1} x_3 = 1$ then $f(x) = 1$. (That is why it is called "prime implicant"). Another possible notation $x_1^0 x_3^1$. Also

$$x_1 x_2 x_3 = X$$

$$\overline{x_1} x_3 = X^{0*1}$$

$$p \in \{0, 1, *\}^n, X = x_1^0 x_2^* x_3^1$$

Fact 6 *1. Any minimal size OR-of-ANDs circuit for boolean function f is wlog an OR of prime implicants.*

2. $|f^{-1}(1)| \leq \Sigma_i |P_i^{-1}(1)|$ where $f^{-1}(1)$ is set of inputs X for which $f(x) = 1$. Also $P_i(1)$ are prime implicants (or the summands in the OR-of-ANDs circuit for f).

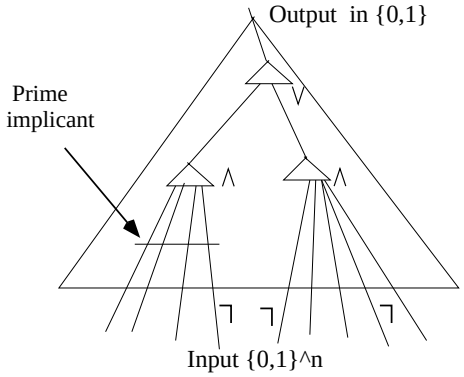

Figure 3: Prime implicant

Proof of Fact 2.2

$$f^{-1}(1) = \cup_i P_i^{-1}(1)$$

hence

$$|f^{-1}(1)| \leq \Sigma_i |P_i^{-1}(1)|$$

Going back to the proof of theorem 2.
Proof:

1. For parity function each prime implicant must have all variables (every variable has influence = 1/2)
2. $|P_i^{-1}(x)| = 1$
3. $|Parity^{-1}(1)| = 2^{n-1}$
4. Minimal OR-of-ANDs circuit for parity is an OR of P_is and

   $$|Parity^{-1}(1)| \leq \Sigma_i |P_i^{-1}(1)|$$

   Therefore minimal OR-of-ANDs for parity has at least 2^{n-1} ANDs.

■

Same size lower bounds and same arguments hold for the Majority function and many symmetric functions

Definition 7 *Majority function is*

$$M(x_1, \ldots, x_n) = 1 iff \Sigma_{i=1}^{n} x_i \geq n/2 = TH_{n/2,n}$$

Exercise 1 *Any circuit of depth d can be converted into one that looks like Figure 4 (a directed acyclic graph with alternating levels of ANDs and ORs. All NOT gates are at the bottom) by increasing size S to atmost square of original size $O(S^2)$ and depth remaining the same or increasing atmost by 1. Directions - any proof like this has to use induction.*

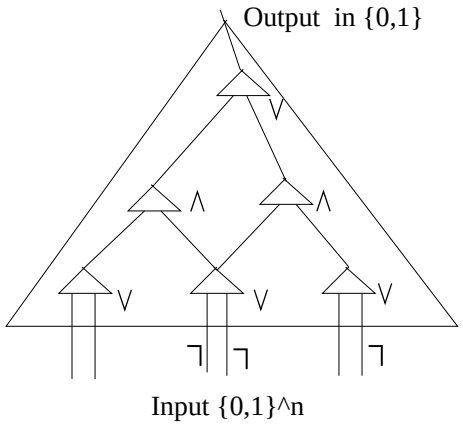

Figure 4: Tree of alternating ANDs and ORs

Positive results i.e upper bounds

Boolean addition, multiplication, matrix multiplication, matrix determinant all have $O(n)$ size, $O(\log n)$ depth circuits of fan-in 2. Proofs are in Wegener's book.

Exercise 2 (a) *Show that any symmetric boolean function can be done with $O(n)$ size, $O(\log n)$ depth circuits. You can use unbounded fan-in if you wish.* **(b)** *Muller and Preparata (JACM, 22). Say it in your own words.*

Exponential lower bound for circuit size of non-explicit function

Fact 8 *Almost all boolean functions require exponential size bounded fan-in circuits. I.e they are as hard as the hardest function in terms of circuit size complexity.*

Best known (linear) lower bound for circuit size of explicit function

Fact 9 *The best known lower bound for an explicit boolean function is linear (we knew this for depth 2 circuits, but now we have no depth restrictions, do have a bounded fan-in though). Example - $TH_{2,n}(x)$, defined as $TH_{K,n}(x)$ iff $\Sigma_{i=1}^{n} x_i \geq K$. See below.*

Proof: of fact 8.

Total number of boolean functions is 2^{2^n}.

Aside - Above result places no restrictions on depth of circuit. On Thursday we will give an exponential size lower bound for parity function, in case of bounded depth circuits.

Proofs of fact continues.

Number of boolean functions computable by bounded fan-in=2 circuits of a fixed size S is $(2(S+2n+2)^2)^S$, or equivalently number of distinct circuits of size S is less than or equal to $(2(S+2n+2)^2)^S$. Every gate can have up to two inputs. This 2 inputs can be chosen from: S outputs of other gates, $2n$ variables $x_i, \overline{x_i}$ and 2 constants 1/0. Thus total possible number of combinations of inputs for one gate is $\binom{S+2n+2}{2}$ and is $\leq (S+2n+2)^2$. This expression is doubled because there are two gate types possible AND and OR. Finally, everything is raised to S^{th} power, because there are S gates.

Taking S to be $(2^n/10n)$ we get $2^{2^n/5}$.

Therefore most boolean functions i.e all except $2^{2^n/5}$ functions out of 2^{2^n} need circuits of size at least $2^n/10n$. ■

Proof: of fact 9

It follows from this lemma

Lemma 10 *for $TH_{2,n}$ at least $2n-4$ gates are required.*

■

The proof of lemma relies on the following claim.

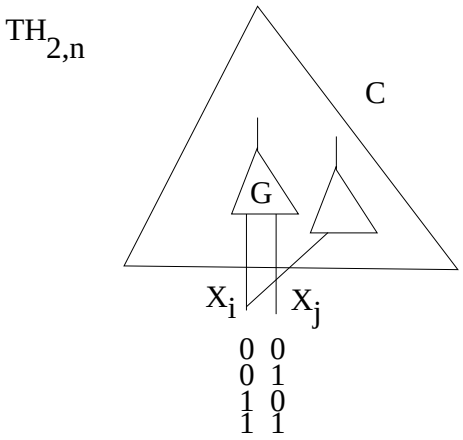

Figure 5: $TH_{2,n}$

Claim 11 *There is a variable x_i such that setting $x_i = 0$ reduces C by at least 2 gates.*

Proof: of the claim: take any 2 variables x_i and x_j that feed into the same bottom level G in C. The different functions that C computes over various settings of x_i, x_j are $TH_{0,n-2}(x_i = 1, x_j = 1), TH_{1,n-2}(x_i = 0, x_j = 1; x_i = 1, x_j = 0), TH_{2,n-2}(x_i = 0, x_j = 0)$. We want to say that x_i or x_j should feed into at least 1 other gate. If neither x_i nor x_j feed into any other gate, then C would only compute 2 different functions (over 4 different settings of x_i and x_j) since the G has only 2 possible output values, contradiction. ■

Proof: Proof of lemma 10 by induction: basis is true for $n = 2, n = 3$ (verify this). Induction step: let C be a circuit computing $TH_{2,n}$. Let x_i be the variable feeding into at least two gates, whose existence was proven in the claim

above. Clearly $C|_{x_i=0}$ computes $TH_{2,n-1}$ (by nature of TH) and by induction hypothesis circuit size of $TH_{2,n-1}$ is at least $2(n-1)-4$, therefore we have circuit size of $TH_{2,n}$ is at least $2n-4-2+2=2n-4$. ■