Umans Complexity Theory Lectures

Circuit Lower Bounds:

-Shannon's Counting Argument

Lower bounds

- Recall: "**NP** does not have polynomial-size circuits" (**NP** $\not\subset$ **P/poly**) implies **P** ≠ **NP**
- **Goal**: prove lower bounds on (non-uniform) circuit size for problems in **NP**
  - believe exponential
  - super-polynomial enough for **P** ≠ **NP**
  - best bound known: 4.5n
  - don't even have super-polynomial bounds for problems in **NEXP**

Shannon's counting argument for Circuits

- Circuit size s= number of gates
- Shannon showed: *almost all* functions require huge circuits

Theorem (Shannon):

With probability at least 1 – o(1),

a random function

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

requires a circuit of size $s = \Omega(2^n/n)$.

(Definition of Ω notation: There exists constants c and n_0 where for all $n \geq n_0$, a circuit of size $s \geq c(2^n/n)$ is required.)

Shannon's counting argument for Circuits

Proof (counting):

Let B(n) = # binary functions $f:\{0,1\}^n \rightarrow \{0,1\}$.

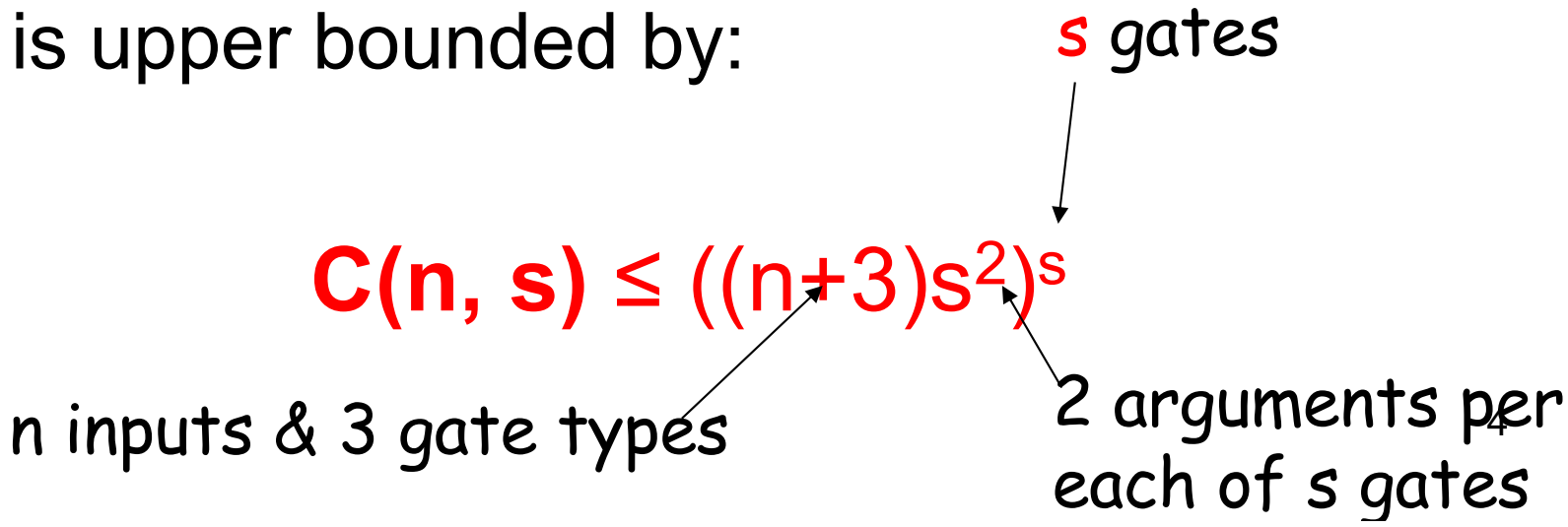
The number of inputs to each f is $N=2^n$.

So number of binary functions with N inputs is

$B(n) = 2^N = 2^{2^n}$.

Let **C(n, s) = # circuits with n inputs and size s.**

C(n,s) is upper bounded by:



Shannon's counting argument for Circuits

So far, we showed # circuits with n inputs and size s, is at most: $C(n, s) \le ((n+3)s^2)^s$

So $C(n, s) \le ((2n)s^2)^s$

Now we set the size $s = c2^n/n$.

$C(n, c2^n/n) < ((2n)(c^2 2^{2n}/n^2))^{(c2^n/n)}$

$< o(1)2^{2c2^n}$

$< o(1)2^{2^n}$ if $c < ½$

Recall $B(n) = 2^{2^n}$ = # functions $f:\{0,1\}^n \to \{0,1\}$.

Hence probability a random function has a circuit of size $s = (½)2^n/n$ is at most

$C(n, s)/B(n) < (o(1)2^{2^n})/(2^{2^n}) = o(1)$

Shannon's counting argument for Formulas

- *Almost all* functions require huge **formulas**
  A **formula** is a tree circuit with no shared subcomponents.

Theorem (Shannon): With probability at least 1 – o(1), a random function

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

requires a **formula** of size $\Omega(2^n/\log n)$.

(Definition of Ω notation: There exists constants c and n_0 where for all $n \geq n_0$, a formula of size $s \geq c(2^n/\log n)$ is required.)

Shannon's counting argument for Formulas

Proof (counting):

Let $B(n)$ = **# binary functions** $f:\{0,1\}^n \rightarrow \{0,1\}$.

Recall: $B(n) = 2^{2^n}$

Without loss of generality assume formula is a tree where:

- The formula leaves are n inputs or their n negations.
- The formula gates are are just 2 types: AND or OR.

Let $F(n, s)$= **# formulas with n inputs and size s.**

$F(n, s)$= is at most:

2n choices per leaf

$$F(n, s) \leq 4^s 2^s (2n)^s = (16n)^s$$

$4^s = 2^s * 2^s$ = number binary trees with s internal nodes

2 gate choices per each of s internal nodes

Shannon's counting argument for Formulas

So far, we showed: # formulas with n inputs and size s, is at most $F(n, s) \leq (16n)^s$

Now we set the size $s = c2^n/\log n$.

$F(n, c2^n/\log n) < (16n)^{(c2^n/\log n)}$

$< 16^{(c2^n/\log n)}\, n^{(c2^n/\log n)} = 16^{(c2^n/\log n)} 2^{(c2^n)}$

$= (1 + o(1))2^{(c2^n)} < o(1)2^{2^n}$ if $c < ½$

Recall $B(n) = 2^{2^n}$ = # functions $f:\{0,1\}^n \to \{0,1\}$.

Hence probability a random function has a **formula** of size $s = (½)2^n/\log n$ is at most

$F(n, s)/B(n) < (o(1)2^{2^n})/(2^{2^n}) = o(1)$.

Lower bounds

- Derived Shannon's known lower bounds for circuits and formulas:
  - **Shannon's** size $s = \Omega(2^n/n)$ **lower bound for circuits**
  - **Shannon's** size $s = \Omega(2^n/\log n)$ **lower bound for formulas**

  – Other Examples of known computational complexity lower bounds in restricted classes circuits:
    - Harnik and Raz gave a $\exp(\Omega(n^{(1/2)} / \log(n)))$ **lower bound for monotone circuits (with no logical negation operations)**