Problem Set 4 is due
This Friday, Feb 14 (10pm)

Class 9: *Lower Bound of Circuit Sizes*

University of Virginia
cs3120: DMT2
Wei-Kai Lin

Maybe related: Complexity Zoo,
https://complexityzoo.net/Complexity_Zoo

Recap: Representing Circuits as Bits

Strings

- Equivalent: NAND straightline program
- n-bit input, ℓ lines, m-bit output.
- Circuit size $s = \ell + m$ (# gates)

$s = n + \ell + m$

- Represented by a sequence of:
  - $2(s+1)$ natural numbers (at most)
  - $O(s \log s)$ bits

Example

```
def CIRCUIT(X[0],X[1]):
    temp2 = NAND(X[0],X[1])
    temp3 = NAND(X[0],temp2)
    temp4 = NAND(X[1],temp2)
    temp5 = NAND(temp3,temp4)
    return temp5
```

```
2, 1      // 2-bit in, 1-bit out
0, 1      // 1st line. 0 and 1 are input
0, 2      // 2nd line. 2, 3, … are temp
1, 2      // 3rd line (and so on
3, 4
5         // return, one line per bit
```

4

temp5

1st

1 line:
input & output lengths

0:	n,	m
n:	$v_{n,1}$,	$v_{n,2}$
n+1:	$v_{n+1,1}$,	$v_{n+1,2}$
…		
last:	$r_1, r_2, \ldots r_m$	

ℓ lines:
one NAND per line

return

m lines:
one output bit per line

r_2

r_3

Representing a sequence in bits

- Chars (e.g. ASCII): represents English letters, digits, punctuation in 8 bits

  26×2 10 <20

  total $< 100 < 2^8$

  - Represent any (finite length) sequence in chars

    $2, 11, 31, 20 \longmapsto$ '2' ',' '1' '1' ',' . . . .

    8-bit $8\times 10 = 80$ bits

- Other encodings. E.g. a sequence of natural numbers

  - '0' is 00
  - '1' is 01
  - Separator ',' is 11

  "10,1011,11111,10100"

  0100 11 0100 . . . .

Subtle: How many variables?

n-bit input, ℓ lines, m-bit output. Circuit size $s = \ell + m$ (# gates)

Num variables: $n + \ell$ (if # > s, need more than $\log s$ bits)

Without loss of generality, $n \le s$, $n + \ell \le 2s$

1 line: input & output lengths	0:	n,	m
ℓ lines: one NAND per line	n:	$v_{n,1}$,	$v_{n,2}$
	n+1:	$v_{n+1,1}$,	$v_{n+1,2}$
	…		
m lines: one output bit per line	last:	$r_1, r_2, \ldots r_m$	

s+1 lines

≤ 2 nat nums

each num ≤ 2s

⇒ log s + 1 bits

$(s+1) \cdot 2(\log s + 1) = O(s \log s)$

Proved:

Theorem. There is a constant c such that for any s, any circuit of size s can be represented in $c \cdot s \log s$ bits.

$O(s \cdot \log s)$

Theorem 5.1 (Representing programs as strings)

There is a constant c such that for $f \in SIZE(s)$, there exists a program P computing f whose string representation has length at most $cs \log s$.

Recap: SIZE(s)

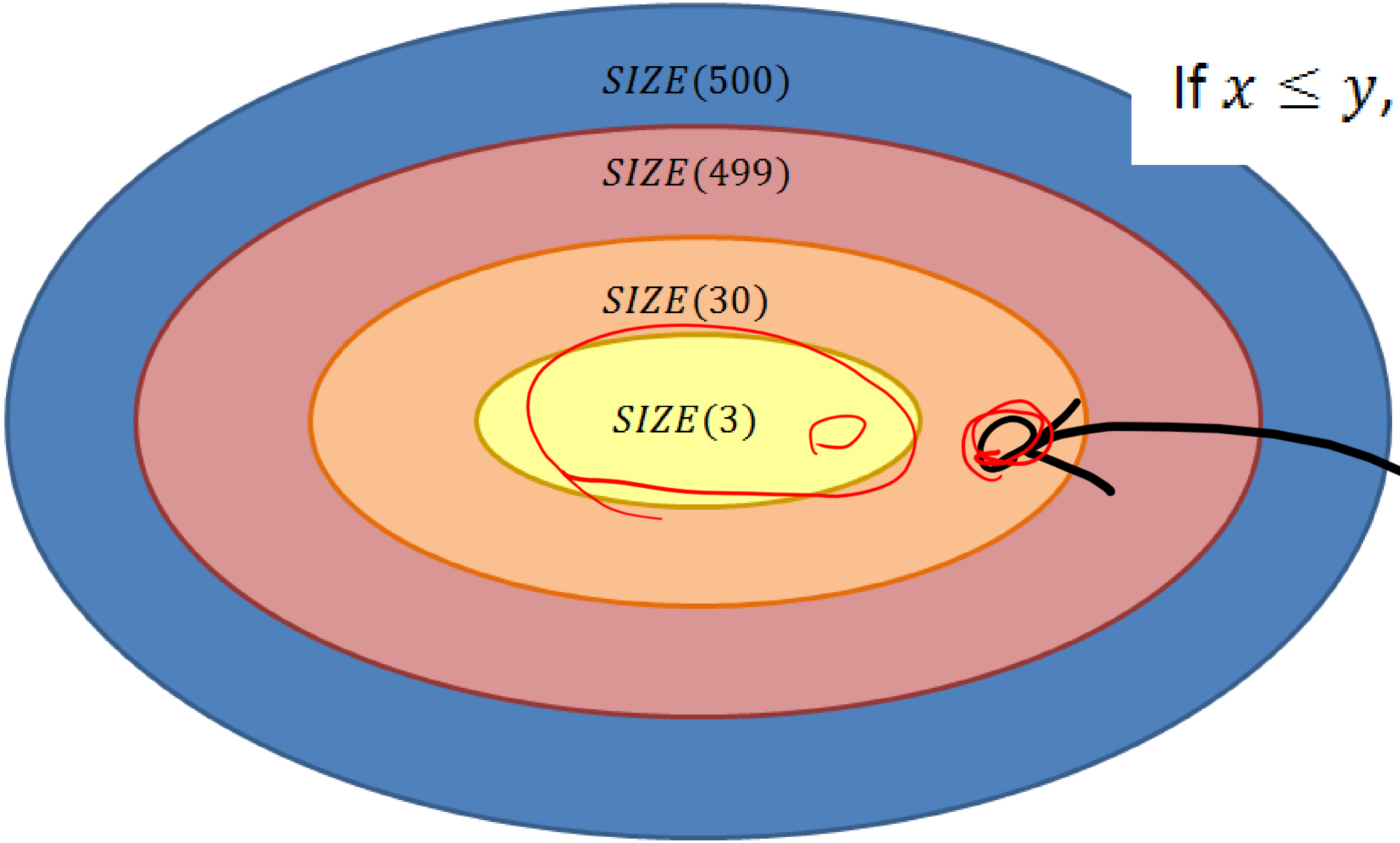
- SIZE(s): set of all *functions* f such that are "computable" by NAND circuits of s gates.
  - f is "computable" by something: there exists something computing f

Definition 4.18 (Size class of functions)

For all natural numbers n, m, s, let $SIZE_{n,m}(s)$ denote the set of all functions $f : \{0,1\}^n \to \{0,1\}^m$ such that there exists a NAND circuit of at most s gates computing f. We denote by $SIZE_n(s)$ the set $SIZE_{n,1}(s)$. For every integer $s \geq 1$, we let $SIZE(s) = \cup_{n,m} SIZE_{n,m}(s)$ be the set of all functions f for which there exists a NAND circuit of at most s gates that compute f.

Circuit Complexity

If $x \leq y$, then $SIZE(x) \subseteq SIZE(y)$



But is the inclusion **strict**? Is there a function *here*?

Goal today:
Prove that the inclusion is **strict.**

Proof idea:
1. Show that $SIZE(s)$ is small (in cardinality)
2. Show that there are many more (finite) functions that are computable in 10x circuit size

Spoiler: counting

Theorem 5.1 (Representing programs as strings)

There is a constant c such that for $f \in SIZE(s)$, there exists a program P computing f whose string representation has length at most $cs \log s$. (bits)

Theorem: Every circuit of size s can be written using $O(s \log s)$ bits.

"" "0" "1" "$b_1 b_2 \dots b_x$"

How many strings are at most x bits?

(for any natural number x)

$2^0 + 2^1 + 2^2 + \cdots + 2^{100} = 2^{100+1} - 1$

$2^0 + 2^1 + \cdots + 2^x = 2^{x+1} - 1$

Consequence of Programs as Data

Theorem: Every circuit of size s can be written using $O(s \log s)$ bits.

$x = c \cdot s \log s$

How many different circuits of size s can exist?

Diff string $\Rightarrow$ diff ckt $\Rightarrow$ $2^{x+1} - 1 = 2^{O(s \cdot \log s)}$

Theorem: There are at most $2^{O(s \ \log s)}$ many circuits of size s

$f(n) = O(g(n))$

$f(n) - 1 = O(g(n))$

$10 \cdot 2^n = O(2^n)$

$10 \cdot 2^n \in 2^{O(n)}$

bcs $10 \cdot n = O(n)$,

$2^n \neq 2^{9n}$

$2^{10n} \in 2^{O(n)}$

How many (distinct) functions computable in circuit size s?

$$2^{x+1}-1 \le 2^{x+1} \le 2^{O(x)}, \quad x = s \log s$$

$$2^{x+1}-1 \in 2^{O(x)}$$

$$2^{c \cdot s \log s + 1} \le 2^{c' \cdot s \log s}$$

Consequence of Programs as Data

Theorem: Every circuit of size s can be written using $O(s \log s)$ bits.

Theorem: There are at most $2^{O(s \log s)}$ many circuits of size s

SIZE(s)

How many (distinct) functions computable in circuit size s?

How many (distinct) functions can be computed using y many (distinct) circuits? (for any natural number y)

Each function can be computed by more than 1 circuits

But

Two distinct functions must be computed by two distinct circuits

f_1

f_2

$C_1 \neq C_2$

Consequence of Programs as Data

Theorem: Every circuit of size s can be written using $O(s \log s)$ bits.

Theorem: There are at most $2^{O(s\ \log s)}$ many circuits of size s

Corollary: at most $2^{O(s\ \log s)}$ many functions are in $SIZE(s)$

Proof:

#funcs $\leq$ #ckts $\leq$ #strings

SIZE(s) $2^{O(s \log s)}$

1st

Corollary: at most $2^{O(s\ \log s)}$ many functions are in $SIZE(s)$

$|SIZE(s)| = 2^{O(s \log s)}$ for all s

set funcs

$|SIZE(3)| < X$

$|SIZE(30)| > Y > X$

$|SIZE(3)| \leq 2^{c \cdot 3 \log 3}$

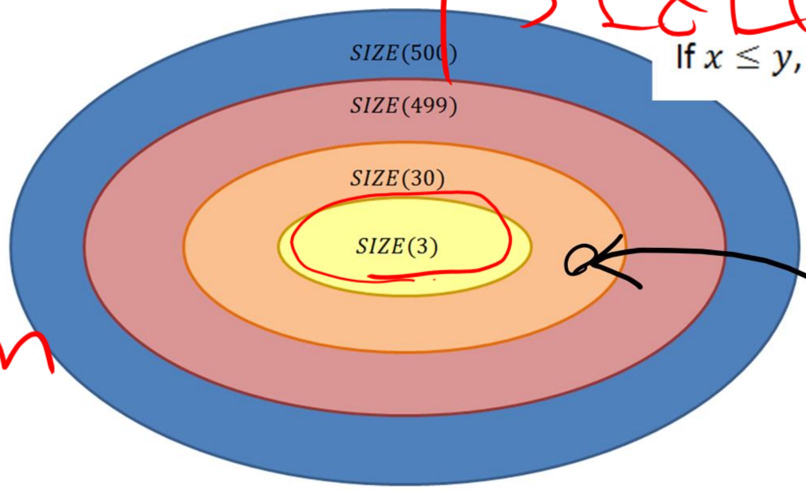
and

$|SIZE(30)| \leq 2^{c \cdot 30 \log 30}$

the inclusion

Is it **strict?**

If $x \leq y$, then $SIZE(x) \subseteq SIZE(y)$



But is the inclusion **strict**? Is there a function *here*?

How many functions of n-bit input are there?

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

$S_f = f(0)f(1)\ldots$

There are 2^{2^n} many Boolean function on n inputs

2^n bits

$2^{(2^n)}$ distinct strings of 2^n bit

Corollary: **Not** all functions can be computed by a circuit of size at most $\frac{2^n}{c \cdot n}$

$S = \frac{2^n}{c \cdot n}$

Proof: # func in $\{0,1\}^n \to \{0,1\}$ is 2^{2^n}

$$|SIZE(S)| = 2^{c \cdot s \log s} = 2^{c \cdot \frac{2^n}{c \cdot n} \cdot (n - \log(c \cdot n))} < 2^{2^n} \quad \square$$

2nd

Corollary:

There is a constant $\delta > 0$ such that for any n, there is a n-bit-input function such that requires more than $\frac{2^n}{\delta \cdot n}$

Charge

Circuit size lower bound

Counting number of circuits vs functions

PS4: due this Friday 10:00pm

Questions?
(maybe for Wei-Kai)

Prob 2 (PS_4)

$\underline{(XOR, 0, 1)}$ not universal.

$$f(x_1, x_2 \cdots x_n) = a_1 x_1 + a_2 \cdot x_2 + \cdots a_n x_n + b \mod 2$$

$\underline{x_1 + 1 \mod 2}$