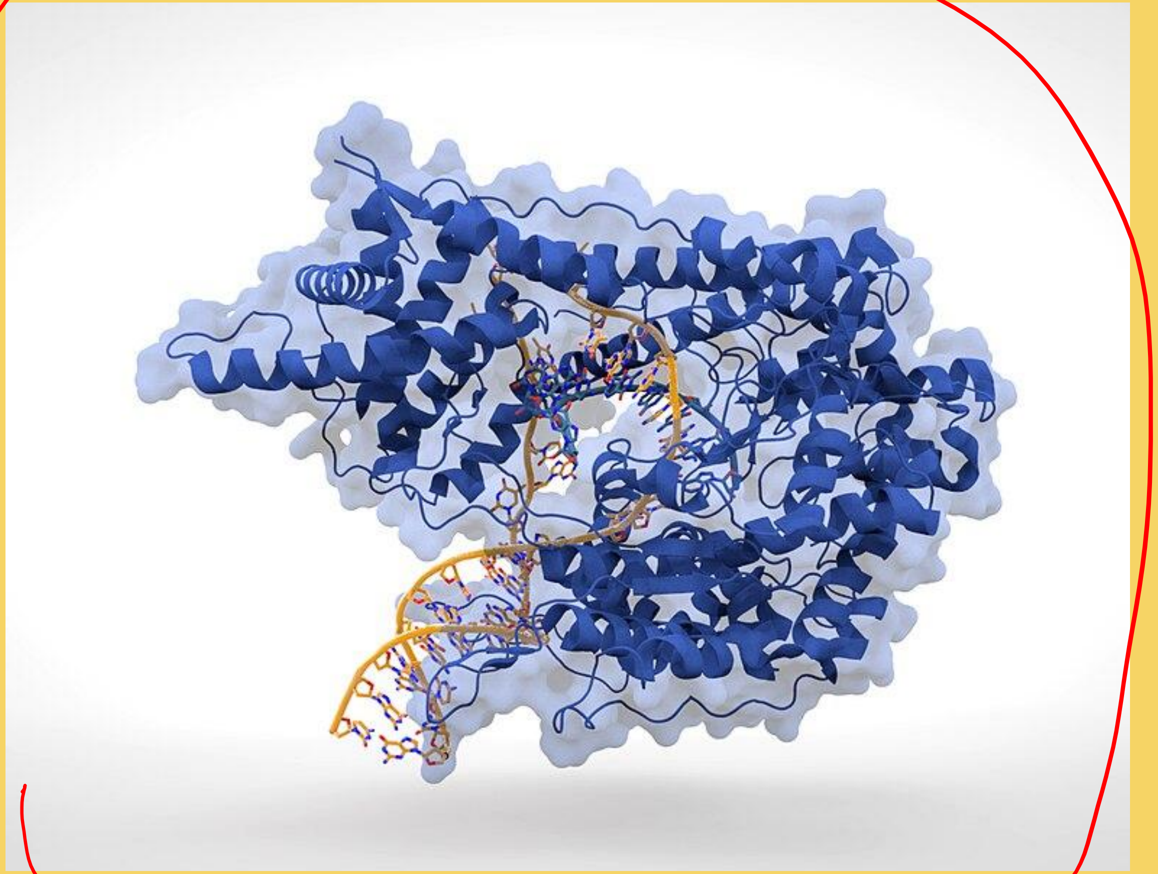


HW 5 due after Spring break

Quiz 6 due after Spring break

Class 12: Program as Data

University of Virginia
CS3120: DMT2
https://weikailin.github.io/cs3120-toc
Wei-Kai Lin

Plan

Circuit-Size Class

Program as Data

Universal Circuits

Textbook [TCS] Section 3 and 4

https://introtcs.org/public/lec_04_code_and_data.html

Code as data, data as code

N

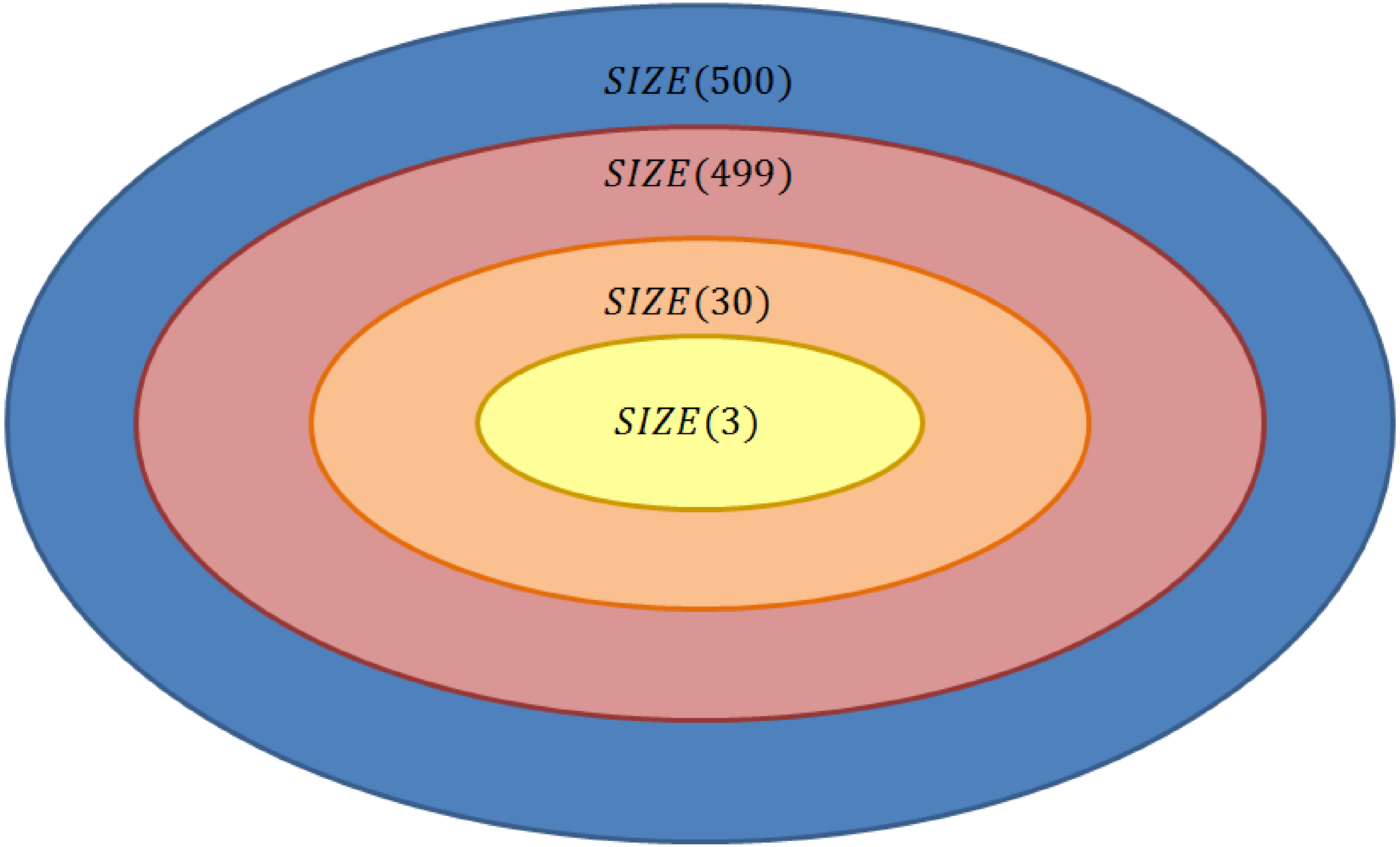
Recap: class SIZE(s)

$SIZE(s)$ The set of all **functions** that can be implemented by a circuit of at most s NAND gates

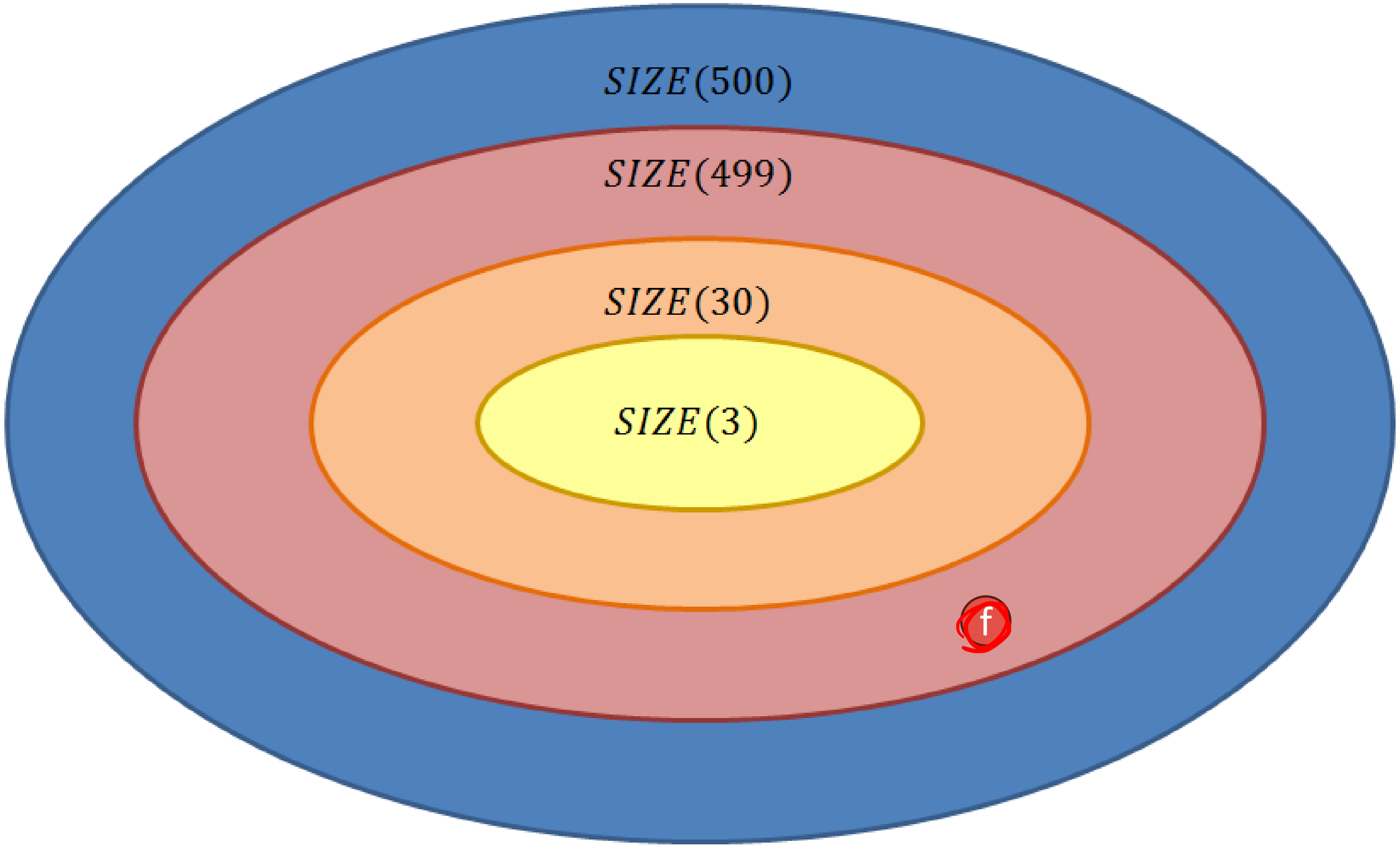
Is circuit $C \in SIZE(s)$?

$\in EV$

Category Error!



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



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

But is the inclusion **strict**?

Programs as Data

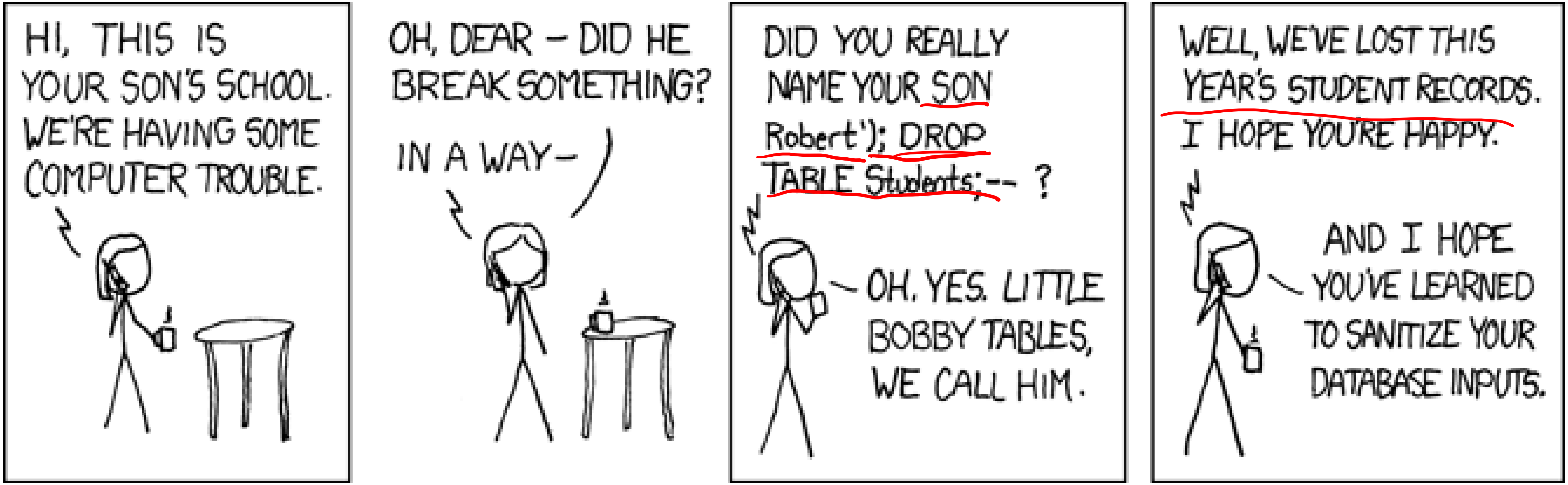
A Big Idea in Theory of Computing

Big Idea 6

A program is a piece of text, and so it can be fed as input to other programs.

Program: an instance (in a computation model) that performs computation (on some data)

Data: a sequence of symbols, such as bits (which can be computed, such as copy, truncate, concatenate….)



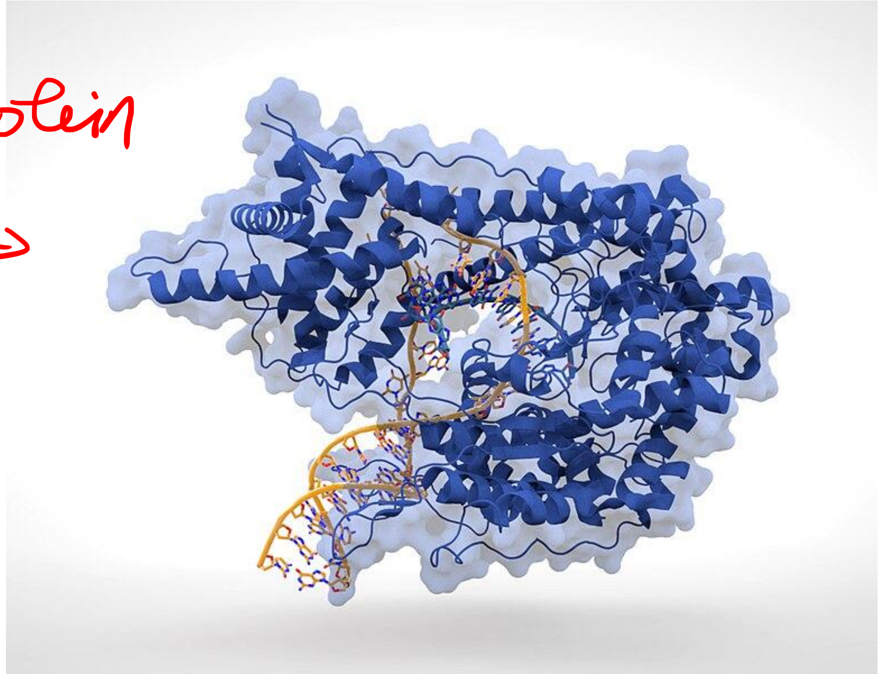
As illustrated in this xkcd cartoon, many exploits, including buffer overflow, SQL injections, and more, utilize the blurry line between “active programs” and “static strings”.

Can be GOOD

protein

DNA (produce)→ Creatures

Creatures (copy, modify) → ~~NDA~~ DNA



> *"The term code script is, of course, too narrow. The chromosomal structures are at the same time instrumental in bringing about the development they foreshadow. They are law-code and executive power - or, to use another simile, they are architect's plan and builder's craft - in one."* , Erwin Schrödinger, 1944.

Image credit: https://en.wikipedia.org/wiki/T7_RNA_polymerase
T7 RNA polymerase (blue) producing m-RNA (light blue) from DNA (orange)

How can we represent a straightline NAND program?

graph ?

2 0 1

~~temp0 = NAND~~(X[0],X[1])

3 0 2

~~temp1 = NAND(X[0]~~,temp0)

temp2 = NAND(X[1],temp0)

Y[0] = NAND(temp1,temp2)

99 characters. Can it be shorter?

How can we represent a straightline NAND program?

Recall: we have n input variables (gates), ℓ lines of code that each is also a “gate”, and m outputs. Total number of gates $s = n + \ell + m$

To encode: write (n, ℓ, m) followed by a list of ℓ “internal-gate descriptions” followed by m “output-gate descriptions”.

Each of the NAND gates, ~~NAND~~(input1, input2), is described by:

~~output~~ ~~int~~

$$(\underbrace{input1}_{int}, \underbrace{input2}_{int})$$

So, we have a list of $\ell + m \leq s$ triples, each of size $O(1) + 2\log s$ 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
```

in bits	out bits
2	1
0,	1
0	2
1	2
3	4
5	

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:	$r_1, r_2, \ldots r_m$	

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
```

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: $r_1, r_2, \ldots r_m$

Representing a sequence in bits

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

Represent any (finite length) sequence in chars

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

‘0’ is 00

‘1’ is 01

Separator ‘,’ is 11

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.

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 $cs \log s$.

Universal Circuits

Consequences of Programs as Data

later

1. We can count the number of programs of certain size.

2. We can also feed a circuit as input to other circuits.

Run Data as Program

Can define the following function, whose outputs is based on running a program given as input:

P computes on x

$$EVAL_{s,n,m}(px) = \begin{cases} P(x) & p \in \{0,1\}^{S(s)} \text{ represents a size-}s\text{ program } P \text{ with } n \text{ inputs and } m \text{ outputs} \\ 0^m & \text{otherwise} \end{cases}$$

Are the P's the same?

$p \neq P$

$p \neq p' \in \{0,1\}^n$ $\cancel{P' \; P'}$ $P(x) \equiv P'(x)$

But can we implement EVAL?

Theorem 5.9 (Bounded Universality of NAND-CIRC programs)

For every $s, n, m \in \mathbb{N}$ with $s \geq m$ there is a NAND-CIRC program $U_{s,n,m}$ that computes the function $EVAL_{s,n,m}$.

EVAL(P , x)

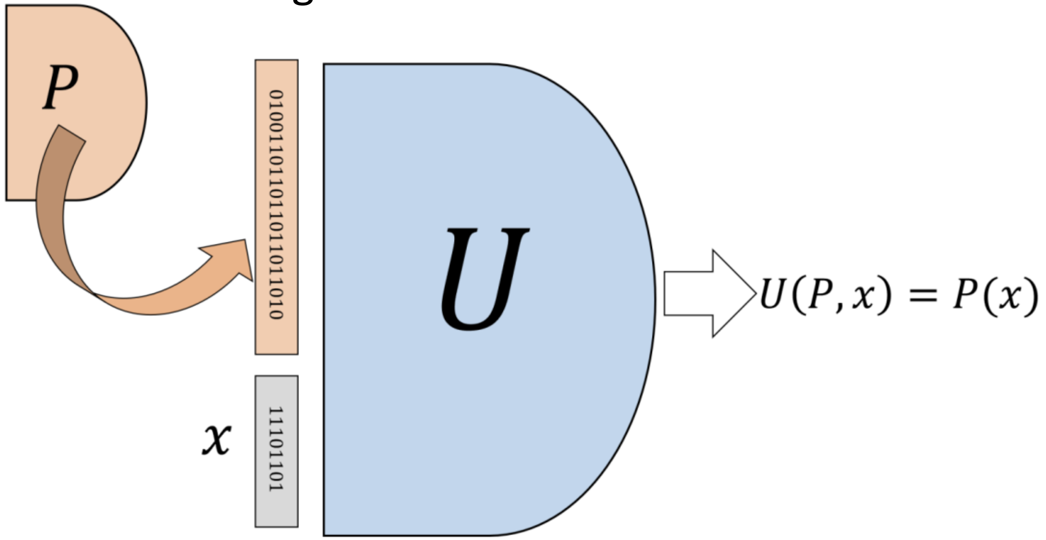
$|P| = c \cdot s \log s + m$ bits

$|x| = n$ bit

$U_{s,n,m}(P, x)$ using $\mathrm{LOOKUP}_{|P|+|x|}$

Universal Circuit/Program

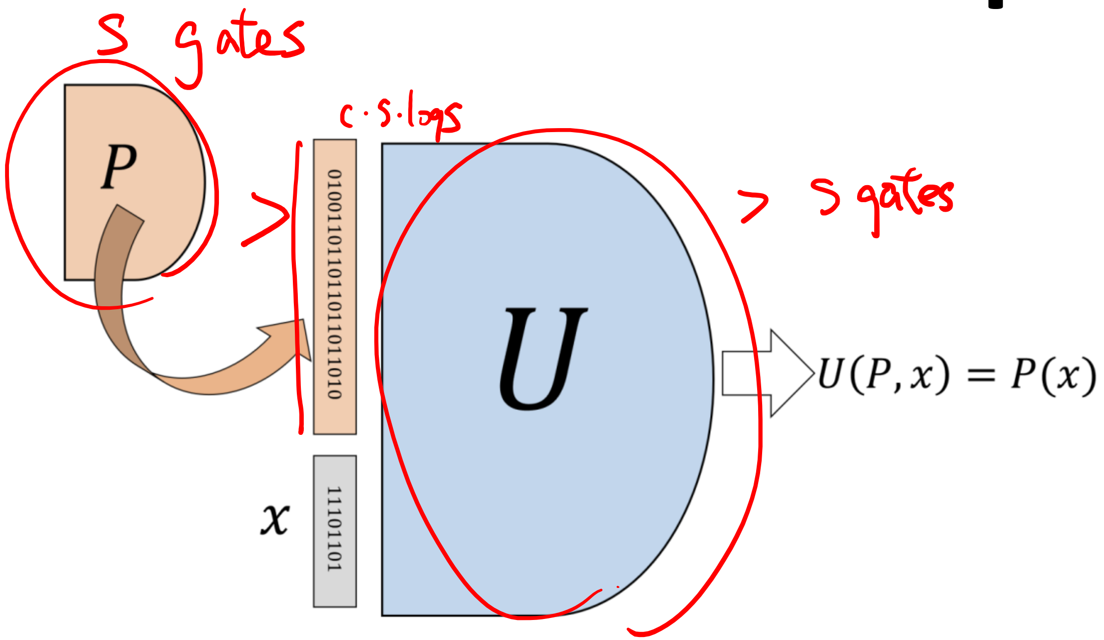
Figure 5.6 from TCS Book



program U takes a program description P and input x as its input, and "simulates" running P on x:

$$U(P, x) = P(x)$$

Points to pause and think



"running a program using another program" is already something to pause and appreciate

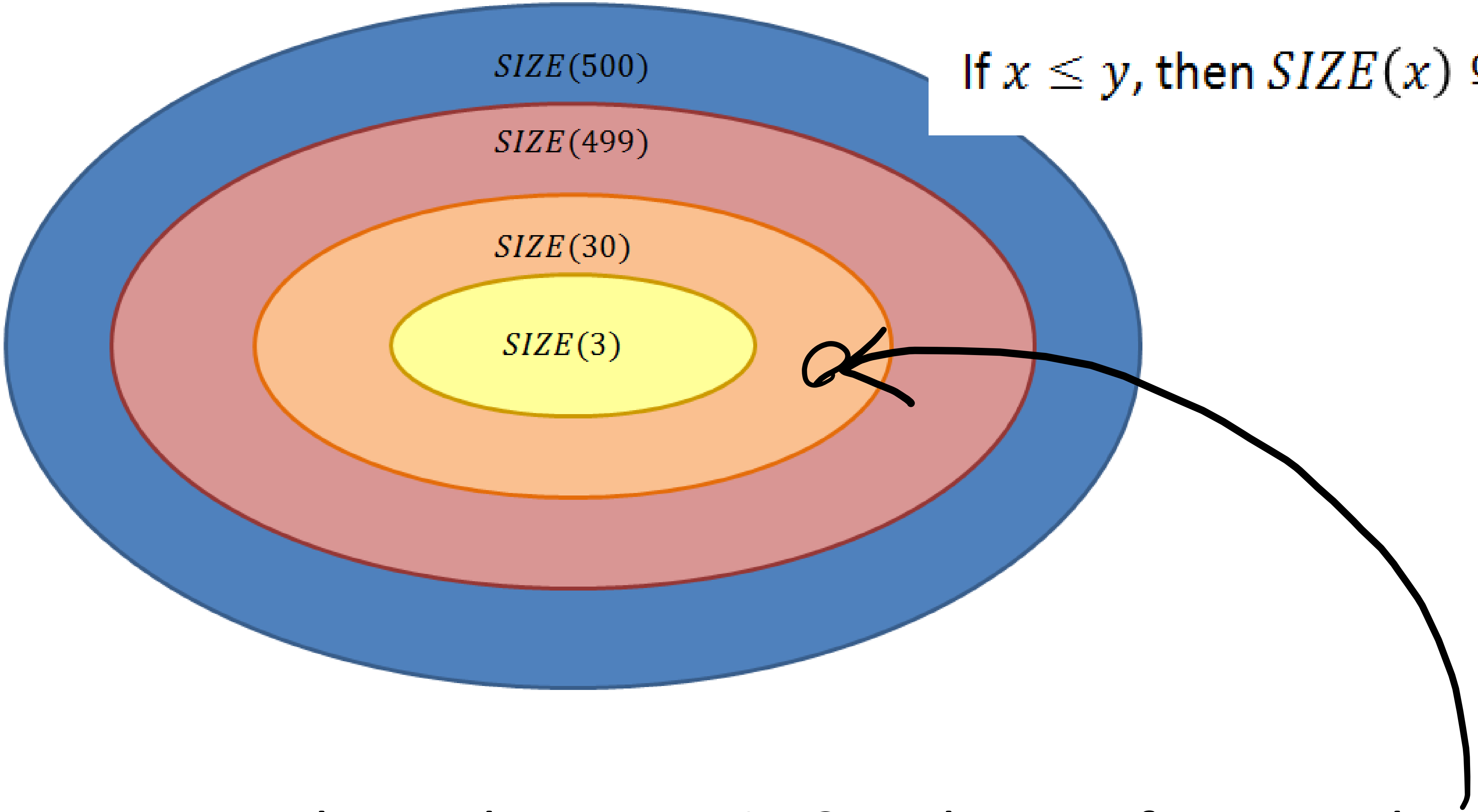
But do we really want to use such inefficient simulation?

CFX-200 scientific calculator watch
https://en.wikipedia.org/wiki/Calculator_watch

Samsung Gear 2 smartwatch
https://en.wikipedia.org/wiki/Smartwatch

Counting the number of Circuits in SIZE(s)

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



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

Consequence of Programs as Data

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

How many different circuits of size s can exist?

$|\{0,1\}^n| = 2^n$

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

How many (distinct) functions computable in circuit size s? $\leq 2^{c\, s \log s}$

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

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:

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

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

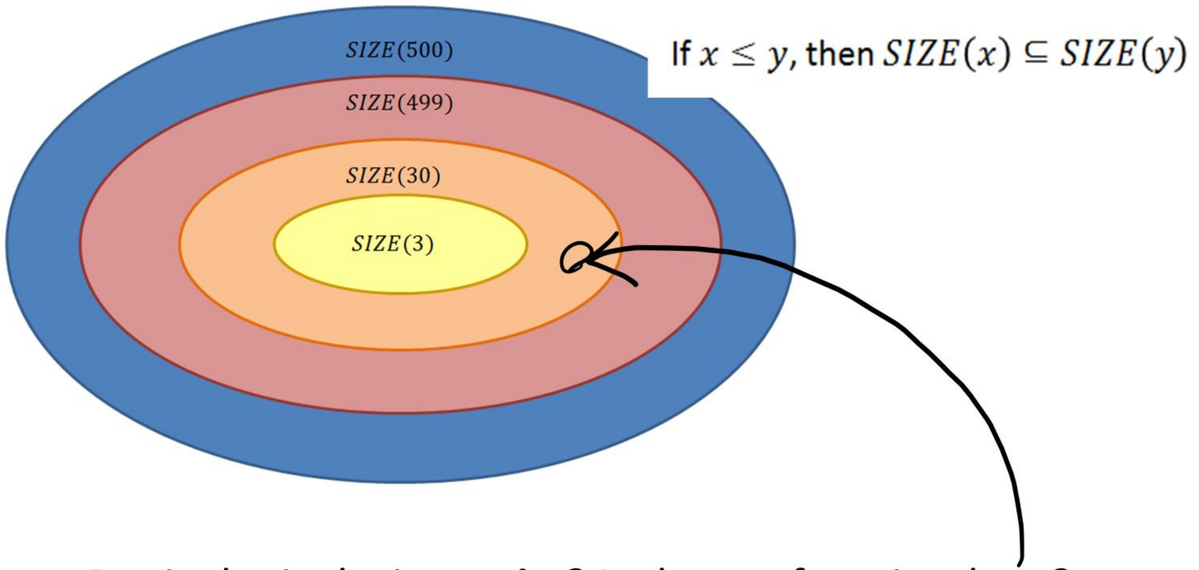
$$|SIZE(3)| \leq 2^{c\cdot 3\log 3}$$

and

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

$SIZE(30) \stackrel{?}{=} SIZE(3)$

Did we solve this?



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

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

How many functions of n-bit input are there?

$|f| = 2^n$ bits

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

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

Proof: $|SIZE(s)| \le 2^{c' s \log s} = 2^{c' \cdot \frac{2^n}{c \cdot n} \cdot \overbrace{\log\left(\frac{\cdots}{\cdots}\right)}^{n}}$

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

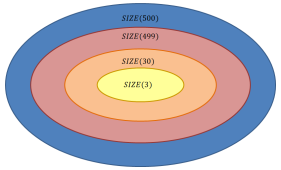
Corollary:
There is a constant $\delta > 0$ such that for any n, there is a n-bit-input function such that requires more than $\frac{2^n}{\delta \cdot n}$

All n-bit functions, $\{0,1\}^n \to \{0,1\}$

$SIZE_n(\frac{2^n}{100n})$, many f, but NOT ALL

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

Strict subsets of $SIZE_n(\frac{2^n}{100n})$?



$SIZE_n(1000n)$

ADD_n here

Does it answer the question?
Why or why not?

Plan

Circuit size hierarchy

Proof

[TCS] Textbook, Section 5

https://introtcs.org/public/lec_04_code_and_data.html

Quiz 6 coming soon, due after Spring break