CS121 Lecture 4: Defining Computation September 12th, 2019

Friday noon - Monday 4pm weekly quiz

Part 1
Finite computation
(boolean circuits)

Part 2
Uniform computation
(Turing machines)

Part 3
Efficient computation

Part 4
Randomized computation

Part 5
Advanced topics

How: Formula / Algorithm / Circuit

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

Compute f: map x to $f(x)$ using a sequence of basic operations

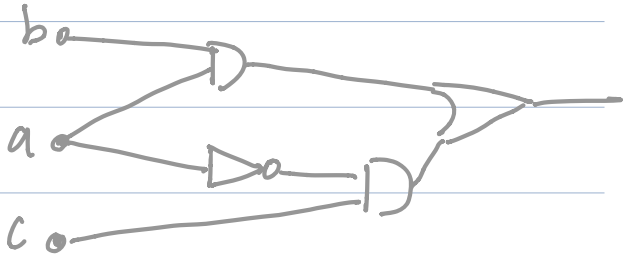
AND / OR: $\{0,1\}^2 \to \{0,1\}$ — and: $a \wedge b$ \wedge — or: $a \vee b$ \vee

NOT: $\{0,1\} \to \{0,1\}$

Ex: compute $f: \{0,1\}^3 \to \{0,1\}$ s.t.

$$f(a,b,c) = \begin{cases} b & a=1 \\ c & a=0 \end{cases}$$

$$\boxed{(a \wedge b) \vee (\bar{a} \wedge c)}$$



Boolean circuit: DAG w/ vertices as inputs or gates (AND, OR, or NOT)

AON-CIRC program: P has lines of form foo = AND(bar, blah)

Inputs are X[0] .. X[n-1]

Outputs are Y[0] ... Y[m-1]

Thm 3 (3.8): For every $f: \{0,1\}^n \to \{0,1\}^m$ and $s \in \mathbb{N}$, f is computable w/ s gates iff f is computable by an AON-CIRC w/ s lines

Ex: Write an AON-CIRC program that computes

XOR: $\{0,1\}^2 \to \{0,1\}$ where $XOR(a,b) = a+b \bmod 2$

$(a \vee b) \wedge (\overline{a \wedge b})$

foo1 = OR(X[0], X[1])

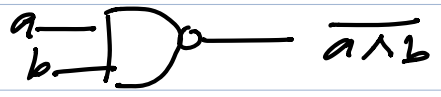
foo2 = AND(X[0], X[1])

foo3 = NOT(foo2)

Y[0] = AND(foo1, foo3)

NAND Operation

$NAND(a,b) = \overline{a \wedge b} = \begin{cases} 0 & a=b=1 \\ 1 & \text{o/w} \end{cases}$



Ex: Show how to compute NOT, AND, OR using NAND

NOT: NAND(a,1)

AND: NAND(NAND(a,b), NAND(a,b))

OR: NAND(NAND(a,1), NAND(b,1))

$\equiv$ NAND(NOT(a), NOT(b))

f computable by $\leq s$ NANDs $\to$ f computable by $\leq 2s$ AND/OR/NOT $\to$ f computable by $\leq 6s$

Boolean circuits $\rightleftharpoons$ AON-CIRC straightline programs

$\uparrow\downarrow$ $\uparrow\downarrow$

NAND circuits $\rightleftharpoons$ NAND-CIRC straightline programs

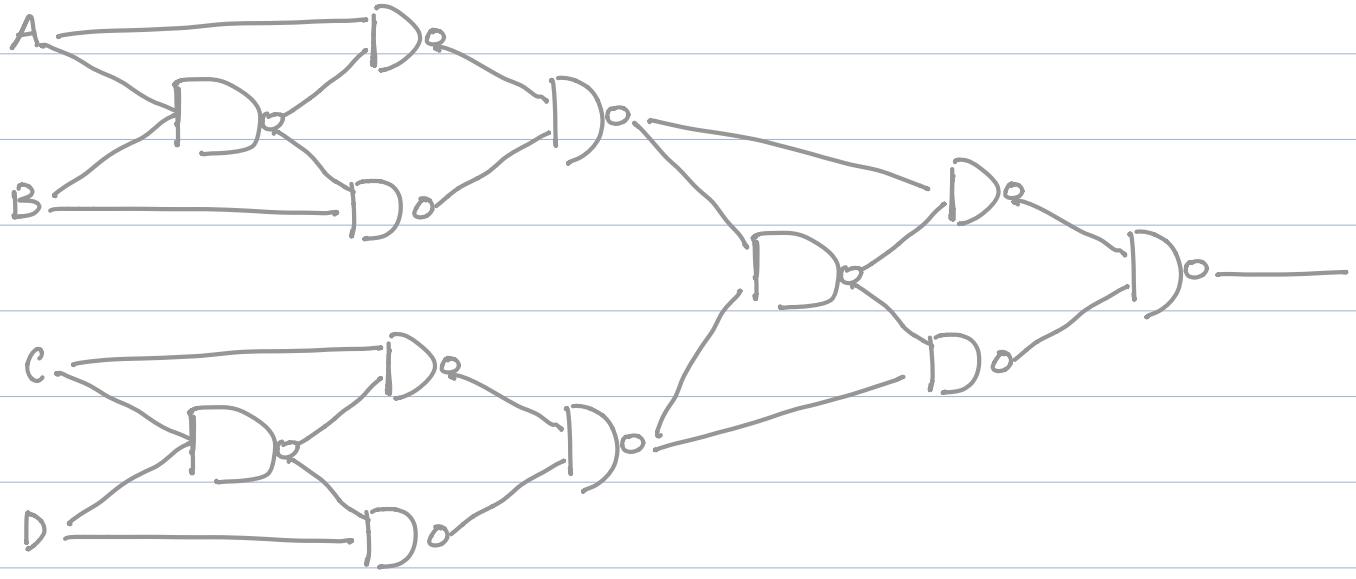
Ex: with NAND-CIRC code for XOR

```
t1 = NAND(NAND(X[0], X[1]), NAND(X[0], X[1]))
nott1 = NAND(t1, t1)
t2 = NAND(NAND(X[0], 1), NAND(X[1], 1))
Y[0] = NAND(NAND(nott1, t2), NAND(nott1, t2))
```

Ex: NAND circuit for XOR$_4$: $\{0,1\}^4 \to \{0,1\}$

XOR$_4$(a,b,c,d) = a+b+c+d mod 2

XOR(XOR(a,b), XOR(c,d))



Next: Theorem: For every function $f: \{0,1\}^n \to \{0,1\}^m$ there exists a Boolean circuit to compute f.

Corollary: For every such f, there is an AON-CIRC program / NAND-CIRC program to compute f.