CPSC 536F Jan 11, 2022

- Goals:

1. Introduction to formula and circuit complexity

2. Intro to some tools all CS theoreticians should learn
   - probabalistic methods
   - eigenvalues in graph theory, expanders and mixing
   algorithms
   - uses of symmetry groups
   - etc.

3. Connections to more mathematical ideas: "relativistic approach", sheaf theory, etc. illustrated in graph thy

Circuit & Formula Complexity:

- For decades: 1960's – 1990's
  generated huge amount of the
  CS Theory work
- Still very exciting
- Still inspires research, but the
  fundamental open problems are
  quite difficult.

Probably: 1/3 – 1/2 course,
will include diversions/intros to
"probabilistic method", eigenvalues of
Laplacians, symmetry arguments, etc.

Next part of course:

eigenvalues/vectors

"spectral analysis" of finite

matrices in graph Laplacians

and adjacency matrices

Applications: to study certain

qualitative properties of graphs

useful in algorithms.

- expansion in graphs

- quick mixing in Markov chains

weighted directed graph

with some important properties

Logistics:

- We'll develop all concepts from scratch
- Typically focus on examples, theorem statements (not proofs)

Grading:

2-3 homework sets — homework problems assigned during class, compiled into some document

Grades!

≥ 95 strong encouragement in CS Theory

≥ 90 encourages research in CS Theory

≥ 85 students in other fields who are just starting in CS Theory, doesn't encourage research CS Thy without significantly more background & study

≥ 80 fulfilled expectations of a grad student

≤ 79 very unlikely

No exams...

Also, depending on demand, have some students give presentations, this can be in lieu of last problem set.

After break, we'll begin on the problem of minimum formula size in $\begin{cases} \text{Boolean formulas} \\ \text{algebraic formulas} \end{cases}$

Break at 10:09 ~ 10:14

Rough motivation for Boolean formula complexity:

(P vs NP) $\Leftrightarrow$ a question about building circuits for certain Boolean functions

A formula is a very restricted class of circuits. Even studying formulas

- appears very difficult
- inspires a lot of research questions
- at present: the best lower bound for formula size is roughly $\geq$ order n^3.
- we'll study <u>monotone</u> formulas

Definitions:

Let's consider Boolean formulas

over $\neg$ (negation)

$\wedge$ (AND)

$\vee$ (OR)

Formula: example

$$\neg\big((x_1 \vee x_2) \wedge (x_3 \vee x_4)\big) \wedge (\neg x_1)$$

$$\Big(\neg\big((x_1 \text{ or } x_2) \text{ AND } (x_3 \text{ or } x_4)\big)\Big)$$

AND

$(\neg x_1)$

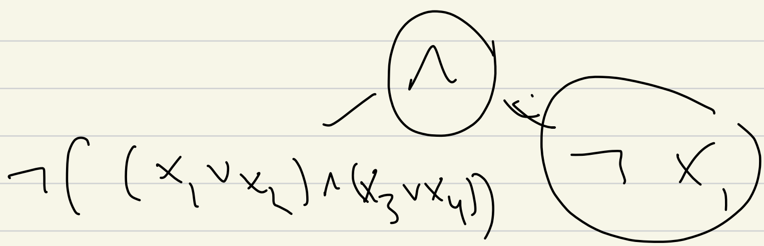
View: $\{F, T\} = \{\text{false}, \text{true}\}$

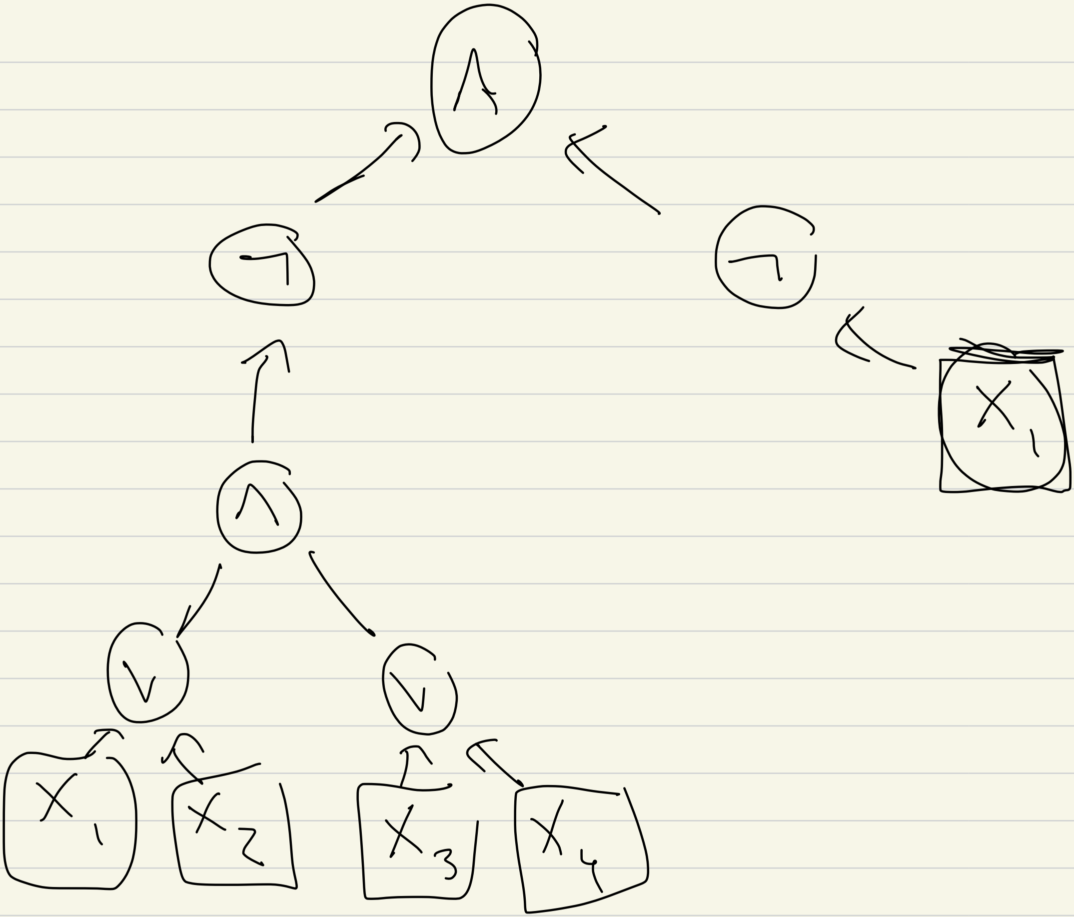
OR $\{0, 1\}$

formula:

$$S = \neg\big((x_1 \vee x_2) \wedge (x_3 \vee x_4)\big) \wedge (\neg x_1)$$

also equivalent tree





Size of a formula

$\stackrel{\text{def}}{=}$ # of occurrences of variables

$=$ # of leaves in the tree

Each formula S (or tree)

gives a Boolean function.

Here: formula size is 5

and $\left\{\begin{array}{l}\text{computes}\\ \text{describes}\end{array}\right\}$ a function

$f = f(x_1, x_2, x_3, x_4)$, formally

$f : \{F, T\}^4 \longrightarrow \{F, T\}$

OR $\{0, 1\}^4 \longrightarrow \{0, 1\}$

Note!

$\neg$ (negation) : $\{F, T\} \to \{F, T\}$

$$\neg F = T$$

$$\neg T = F$$

(or $F \leftrightarrow 0$, $T \leftrightarrow 1$

$$\neg 0 = 1, \quad \neg 1 = 0$$

or

$$\neg x = (1 - x)$$

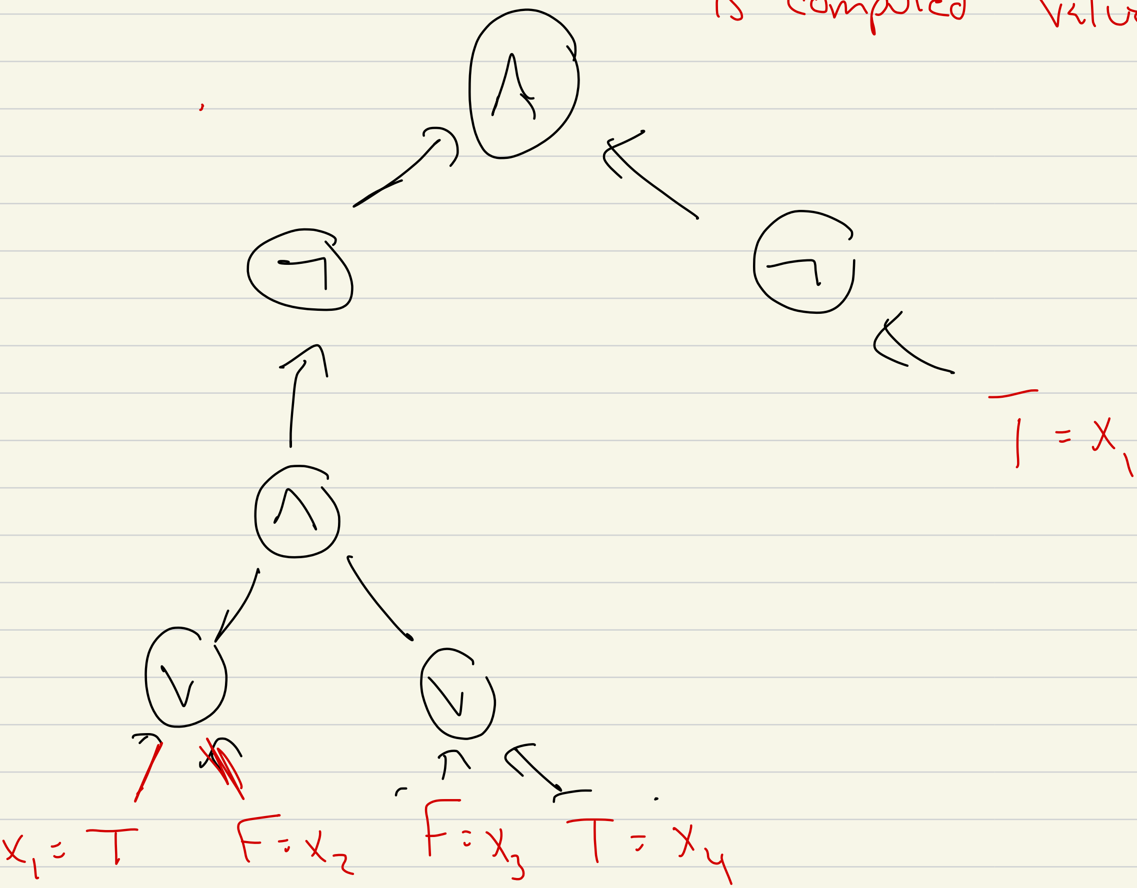
But $\wedge$ AND, $\{F, T\}^2 \to \{F, T\}$

$$F \wedge F = F \wedge T = T \wedge F = F$$

$$T \wedge T = T$$

Example!

root of tree is "computed" value



$x_1 = T, \quad x_2 = F, \quad x_3 = F, \quad x_4 = T$

Given a Boolean function on n variables, $f = f(x_1, \ldots, x_n)$,

i.e. $f : \{F, T\}^n \to \{F, T\}$

the $\left\{ \begin{array}{l} \text{formula size complexity} \\ \text{minimum formula size} \end{array} \right\}$

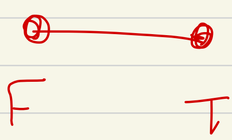
of f is the size of the smallest formula $\left\{ \begin{array}{l} \text{computing} \\ \text{describing} \end{array} \right\}$ f. Understand: $\neg$, $\wedge$, $\vee$ the gates we allow

Given n:

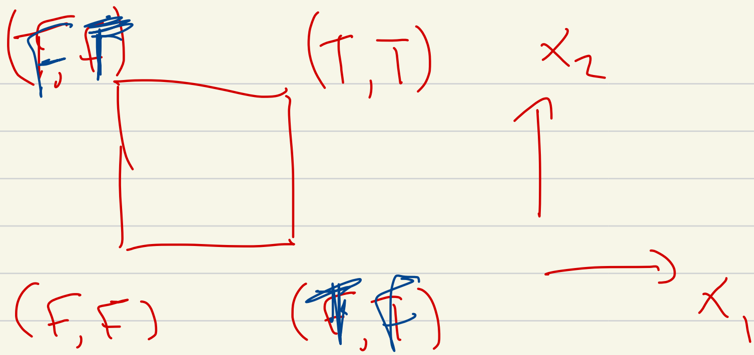
- the number of Boolean functions on $x_1, \ldots, x_n$ is

Aside: we often think of functions in terms of the Boolean hypercube:

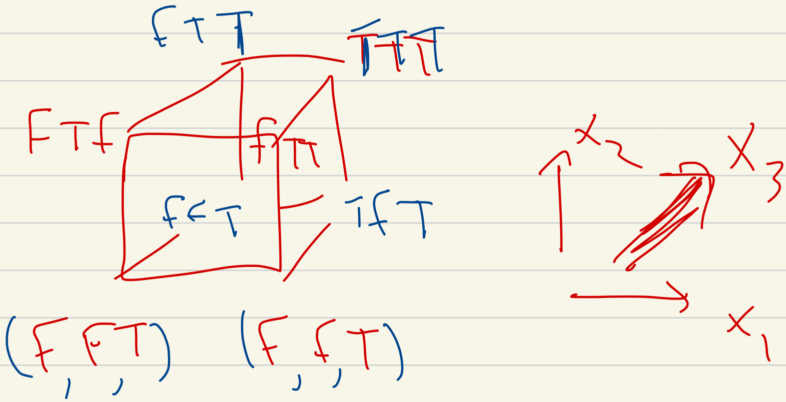
$\mathbb{B}^1$



$\mathbb{B}^2$:



$\mathbb{B}^3$



formally

$\mathbb{B}$ set $\{F,T\}$

$\mathbb{B}$ set $\{F,T\}^2$

$= \{F,T\} \times \{F,T\}$

$= \{(F,F),(F,T),(T,F),(T,T)\}$

We often view

$\mathbb{B}^n$ as a graph

vertex set is $\{F, T\}^n$

edges between vertices

of distance one

$\leftarrow$

Size of $\{F, T\}^n = 2^n$

Size of set of functions

$\{F, T\}^n \to \{F, T\}$

is $2^{(2^n)}$

$n = 1$ $\quad \cdot \quad \cdot \quad \mathbb{B}$
$\quad F \quad T$

2 points in $\mathbb{B}$

$\Rightarrow 2^2$ Boolean functions

① $f(F) = F$ $\quad f(T) = F$

"constant function" F

② $f(F) = F$, $f(T) = T$

identity

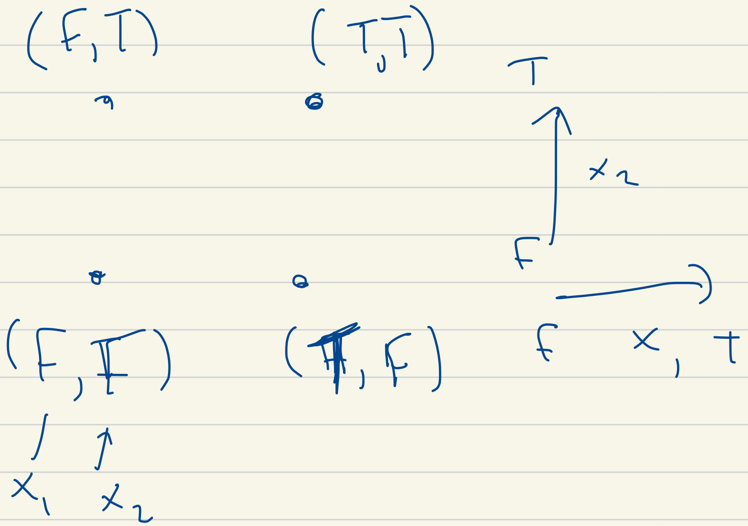
③ negation $f(F) = T$, $f(T) = F$

④ $f(F) = f(T) = T$

"constant function" T

$=$

$n = 2$



4 points in $\{F, T\}^2$

$\Rightarrow$ 2^4 Boolean function

on $f = f(x_1, x_2)$

Next time: function on n-Boolean vars

1. All functions have ~~size~~ formula complexity $\leq n \cdot 2^n$

2. (Shannon) Most functions have complexity

$$\geq 2^n / n^{1.001}$$

Class ends

Room ICSS Room 246