LECTURE 16: CIRCUIT CPLX AND WRAP-UP

CIRCUIT

Directed acyclic graph

inputs — source nodes — labelled by variables x_i

gates —
- internal nodes — indegree 1 labelled by $\neg$ (NOT)
- indegree 2 labelled by $\wedge$ (AND) or $\vee$ (OR)
- unique sink node — output

Actually can assume w/og that negations come right after variable source node, so that source nodes labelled x_i or $\neg x_i$ / $\overline{x}_i$

MONOTONE CIRCUIT No $\neg$ gates or $\overline{x}_i$

FORMULA Each node has outdegree 1 (except sink), i.e., DAG is a tree

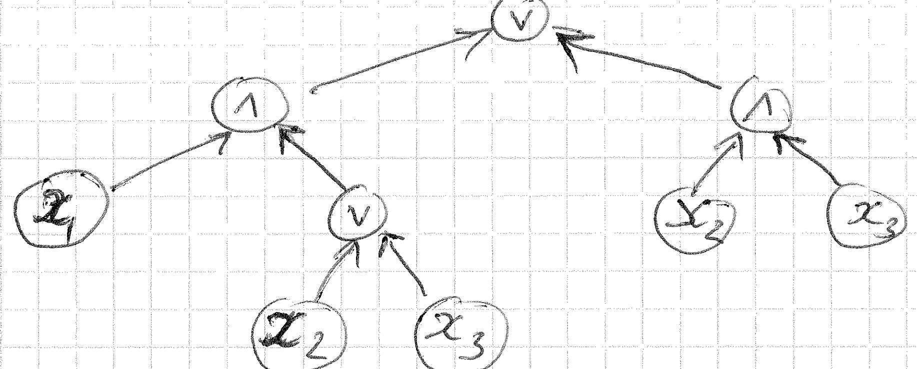
FUNCTION COMPUTED BY A CIRCUIT

inputs — compute x_i / $\overline{x}_i$ i.e., the literal they are labelled with

gates — compute the logical and/or of the functions computed by the nodes feeding into them

The circuit computes the function computed by output node

Example:



Monotone circuit that computes majority of three bits

For $x, y \in \{0,1\}^n$ say $x \leq y$ if $x_i \leq y_i \ \forall i \in [n]$

$f : \{0,1\}^n \to \{0,1\}$ is MONOTONE if $x \leq y \Rightarrow f(x) \leq f(y)$

<u>Prop</u> (a) Every function computed by a monotone circuit is monotone.

(b) For every monotone circuit, there is a monotone circuit computing it

<u>Proof</u> Exercise

COMPLEXITY MEASURES

Depth of circuit C $d(C)$ = longest path from source to sink

Size of circuit C $S(C)$ = # nodes

$d(f)$ = min depth of any circuit computing f

$S(f)$ = min size of any circuit computing f

For monotone circuits & functions, can define $d_m(C)$, $S_m(C)$, $d_m(f)$, $S_m(f)$ similarly.

RELATION

Subset R of $\mathcal{X} \times \mathcal{Y} \times \mathcal{I}$ (for some sets $\mathcal{X}, \mathcal{Y}, \mathcal{I}$) where for each $(x,y) \in \mathcal{X} \times \mathcal{Y}$ there is some $i \in \mathcal{I}$ (one or many) such that $(x,y,i) \in R$.

COMMUNICATION COMPLEXITY OF RELATION R

Alice given $x \in \mathcal{X}$
Bob given $y \in \mathcal{Y}$
They should find $i \in \mathcal{I}$ s.t. $(x,y,i) \in R$

INVERSE IMAGE

$$f^{-1}(b) = \{ z \mid f(z) = b \}$$

KARCHMER - WIGDERSON RELATION

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

$$KW_f \subseteq f^{-1}(1) \times f^{-1}(0) \times [n]$$

defined by $\{ (x,y,i) \mid x_i \neq y_i \}$

$D(KW_f)$ is the deterministic commcplx of the Karchmer-Wigderson relation of f.

THEOREM 1 For any (non-constant) Boolean function $f : \{0,1\}^n \to \{0,1\}$ it holds that

$$D(KW_f) = d(f)$$

(switch to variables $z_1, \dots, z_n$ to avoid confusion)

LEMMA 2 If circuit C computes f, then $\exists$ protocol $\mathcal{P}$ for KW_f with communication $\leq d(C)$

Proof Alice and Bob traverse circuit from output to some source, maintaining that current node computes 1 for Alice and 0 for Bob

Induction base True for single/output.

Inductive step

OR-gate $g(z_1, \dots, z_n) = g_1(z_1, \dots, z_n) \vee g_2(z_1, \dots, z_n)$

Bob $g_1(\vec{y}) = g_2(\vec{y}) = 0$
Alice one of $g_1(\vec{x})$, $g_2(\vec{x}) = 1$

Alice sends 0 to go left to g_1 and
1 to go right to g_2

AND-gate $g(\vec{z}) = g_1(\vec{z}) \wedge g_2(\vec{z})$

Alice $g_1(\vec{x}) = g_2(\vec{x}) = 1$
Bob one of $g_1(\vec{y})$, $g_2(\vec{y}) = 0$
Bob decides whether to go left or right.

Finally A & B reach input node labelled by z_i / $\overline{z_i}$. Then A & B can answer i, since by induction $x_i \neq y_i$.

clearly # bits $\leq$ depth of C.

Illustration for circuit above

$$X = f^{-1}(1) = \{101, 111, 110, 011\}$$

$$Y = f^{-1}(0) = \{100, 000, 001, 010\}$$

	100	000	001	010
101	3	1	1	1
111	2	1	1	1
110	2	1	1	1
011	2	2	2	3

Alice goes left if possible
If so Bob goes left if possible
otherwise Alice again goes left if possible...

Yields protocol with partition
as above

LEMMA 3 If $\mathcal{P}$ is a (deterministic) protocol for KW_f, then $\exists$ circuit C for f with $d(C) \leq$ comm cplx of $\mathcal{P}$.

<u>Proof</u> Look at protocol tree and convert into circuit

Alice speaks (in internal node) — label $\vee$

Bob speaks — label $\wedge$

Each leaf is a monochromatic rectangle $A \times B$ labelled by $i \in \boxed{n}$.

For all $x \in A$ & $y \in B$ $x_i = \sigma$ and $y_i = 1 - \sigma$ for $\sigma = 0$ or 1

Suppose for some concrete $x^* \in A$ $x^*_i = \sigma$.

For all $y \in B$ i is correct answer for x^*, y

$\Rightarrow$ for all $y \in B$ $y_i = 1 - \sigma$

Same reasoning again wrt $x \in A$

for all $x \in A$ $x_i = \sigma$.

If $x_i = 1$, $y_i = 0$ label z_i

If $x_i = 0$, $y_i = 1$ label $\overline{z}_i$

Depth of circuit = depth of protocol tree = comm cost.

Need to prove circuit computes f

Prove by induction:

Let $A \times B$ be inputs that reach node v computing g_v. Then $\forall x \in A$ $g(x) = 1$

$\forall y \in B$ $g(y) = 0$

Sufficient, since for output the function computed is 1 on all $x \in f^{-1}(1)$
0 on all $y \in f^{-1}(0)$
i.e. function computed is f.

Bottom-up induction:

__Base case:__ True for input nodes by construction.

__Inductive step__

$$\begin{tikzcd} & v & \\ v_1 \arrow[ur] & & v_2 \arrow[ul] \end{tikzcd}$$

$g_v = g_{v_1} \circ g_{v_2}$

$\circ \in \{\wedge, \vee\}$

Suppose $A \times B$ reaches v

Suppose Alice speaks $\quad g_v = g_{v_1} \vee g_{v_2}$

Alice's ~~bits~~ partitions A into A_1 & A_2

By IH $\quad g_{v_1}(y) = g_{v_2}(y) = 0 \quad \forall y \in B$

$g_{v_1}(x) = 1 \quad \forall x \in A_1$

$g_{v_2}(x) = 1 \quad \forall x \in A_2$

Hence $\quad g(y) = 0 \vee 0 = 0$

$g(x) = 1 \vee * = 1 \quad x \in A_1$

$* \vee 1 = 1 \quad x \in A_2$

$g(x) = 1 \quad \forall x \in A = A_1 \cup A_2$

The case for Bob is symmetric. $\boxed{}$

For f monotone, define

$$KW_f^m = \{ (x,y,i) \mid x \in f^{-1}(1), y \in f^{-1}(0), x_i = 1, y_i = 0 \}$$

THEOREM 4

For every monotone Boolean function

$$D(KW_f^m) = d_m(f)$$

Proof The same as above.

Want to use this to prove lower bounds on circuit depth. Hard.

But can prove lower bounds for monotone circuits.

Depth interesting measure in its own right

Also gives bounds on formula size

If f is computed by a formula F of size S, then there is another formula F' computing f in depth $\approx \log S$

So depth lower bounds yield formula size lower bounds.

(Not too exciting, but good enough for today).

$\mathrm{Match}_n : \{0,1\}^{\binom{n}{2}} \to \{0,1\}$

Interpret input as characteristic vector of edge set of undirected graph on n vertices

$\mathrm{Match}_n(E(G)) = 1$ iff G has perfect matching

Match_n is monotone (why?)

Want to prove

THEOREM 5

$$d_m(\mathrm{Match}_n) = \Omega(n)$$

<u>Proof outline</u> Do randomized reduction from $DISJ_n$ to $\mathrm{Match}_{n'}$ for $n' = O(n)$.

Then conclude that

$$d_m(\mathrm{Match}_{n'}) = D(KW^m_{\mathrm{Match}_{n'}}) \geq R(DISJ_n) = \Omega(n) = \Omega(n')$$

Reduction in several steps

(i) DISJOINTNESS TO COVERING PROBLEM

$DISJ_n$ - instance

Alice gets $X \subseteq [n]$ Bob $Y \subseteq [n]$

Alice builds graph

$V = \{ a_i, b_i, c_i \mid i \in [n] \}$

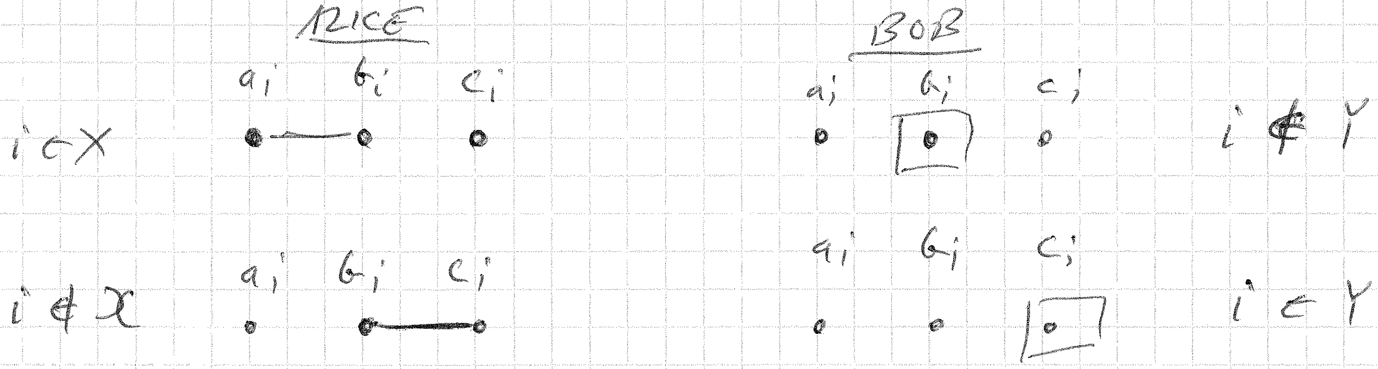
$i \in X$ (a_i, b_i) edge

$i \notin X$ (b_i, c_i) edge

Bob gets $T \subseteq V$

$i \in Y \Rightarrow c_i \in T$

$i \notin Y \Rightarrow b_i \in T$



Alice's graph matching of size n on $3n$ vertices

$X \cap Y = \emptyset$ <u>iff</u> T covers edges of Alice's graph.

(ii) COVERING PROBLEM TO Match$(n, 3n)$

Match$(n, 3n)$: $\{0,1\}^{\binom{3n}{2}} \to \{0,1\}$

1 iff graph has matching of size n

Alice gets graph G_A as above – 1+ instance.

Bob gets ~~set~~ vertex T. Need to turn into graph without matching of size n.

Pick random $v \in T$ and let $T' = T \setminus \{v\}$

Let graph G_B for Bob be complete bipartite graph between T' and $V \setminus T'$

$|T'| = n-1$ so no n-matching

A & B use shared randomness to permute vertices. Say random permutation σ and resulting graphs G_A^σ, G_B^σ

Run protocol for $KW^m_{\text{Match}(n, 3n)}$

returns $e = (x,y) \in E(G_A^\sigma) \setminus E(G_B^\sigma)$.

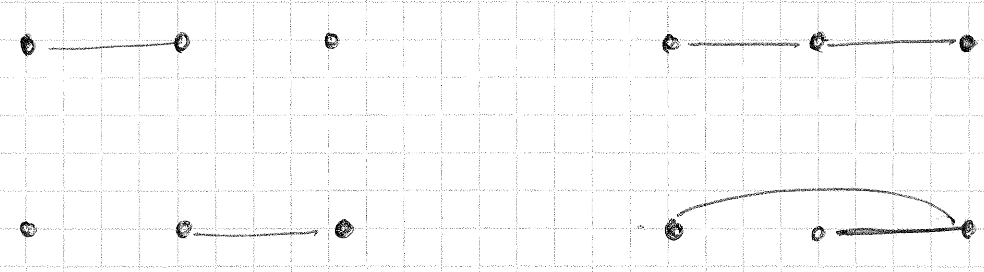
If e is incident on $\sigma(v)$ Bob says T cover

Otherwise says T not cover.

If T covers G_A then for all $v \in T'$ have

$i \in X$ $i \notin Y$

$i \notin X$ $i \in Y$



so Bob has all edges Alice does except for v removed to get T'. Missing edge involving T will be found $\Longrightarrow$ Bob declares cover.

If T does not cover G_A then for some u have

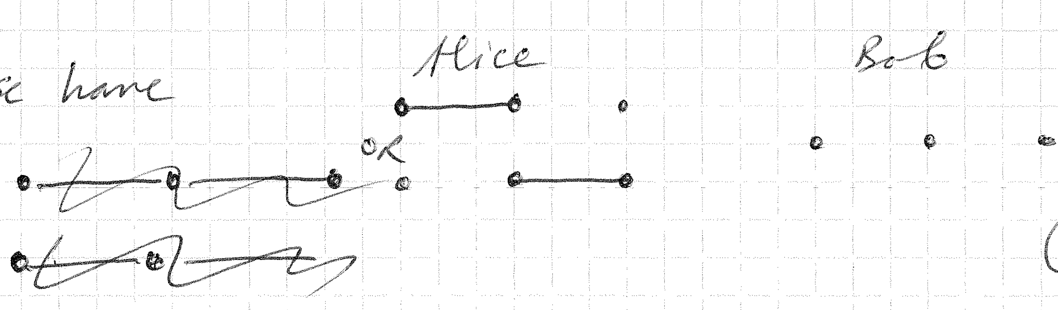
(a)

If ~~u~~ no $a_u\, b_u\, c_u$ was ~~not~~ removed, this edge of Alice is a valid answer and Bob will say <u>not cover</u>

If c_u was removed, have situation for v:

(b)

otherwise have



(c)

Hence, if $v = c_n$ was removed (the unique intersection, say) Bob always gets it right FIG 6

Otherwise Bob has at least two edges missing; all except one of these will make him answer correctly.
Namely Fig (a) + Fig (c) (with one of alternative ~~alternaedges~~ edges for Alice, both of which might be bad)
But Fig (a) - edge always good.

Hence, with probability $\geq 1/2$ over public randomness, Bob decides the cover problem correctly. Repeat to reduce error.

(iii) $KW^{m}_{Match(n, 3n)}$ to $KW^{m}_{Match_{4n}}$

Alice & Bob add n new identical vertices to G_A & G_B and connect them to all other vertices

Alice's graph now has perfect matching

For Bob, new vertices can match with $V \setminus T'$ but T' can still only give matching of size $n-1$ And matching new vertices with T' makes this worse, since then it will be harder to match $V \setminus T'$

Note that a protocol for $KW^{m}_{Match_{4n}}$ must discover an edge in $E(G^{\sigma}_{A}) \setminus E(G^{\sigma}_{B})$ since all other newly added edges are present in both graphs.

What did we cover in the course

Two-party comm cplx, deterministic
& randomized

Some multi-party CC

Set disjointness lower bound — fundamental problem

Applications:
- Streaming
- Property testing
- Proof cplx
- Circuit cplx

Some discussion of research frontier /
open problems.

What did we NOT cover?

Nondeterministic CC
Prover sends proof, A & B verifies.

Quantum CC

Multi-party CC (except a little)

Techniques

Linear programming
Linear algebra
Pattern matrix method Sherstov [except a little]
Information cost [much more here]

... And more ...

Applications

Data structures (static & dynamic) [Patrascu]
VLSI chip design
Combinatorial auctions
Circuit complexity [much more here]
Pseudorandomness

... And more ...