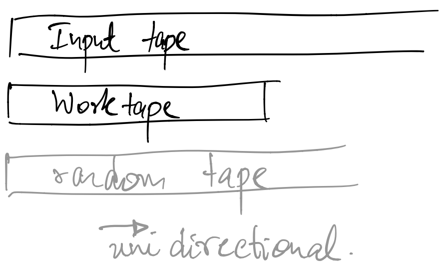
Computational Complexity: Lecture 16

Agenda:
- Randomized space complexity.
- Barrington's theorem.

Recap:
- RP, coRP, BPP, ZPP
- Error reduction, Chernoff bounds.
- BPP $\subseteq$ P/poly $\quad$ BPP $\subseteq \Sigma_2 \cap \Pi_2$.

Randomised Space classes.



$L \in \text{RSPACE}(s(n))$ if there is an machine s.t

$x \in L$: $\Pr_{r \in \{0,1\}^*}[M \text{ accepts}] \geq 1/2$

$x \notin L$: $\Pr_r[M \text{ accepts}] = 0$

Cool fact: Undirected s-t connectivity $\in$ RL.

Algo for deg-d graphs:

$u = s$.

For $i = 1, \ldots, n^{10}$:

  If $u = t$: Return "Yes".

  $u \leftarrow$ random neighbour of u.

Return "No".

If s & t are connected, the algo succeeds with prob $\gg 1/2$.

(Those attending the Toolkit course will see a proof of this).

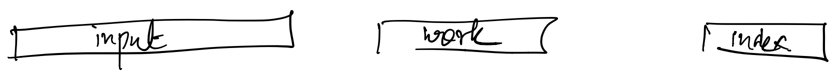
Revisiting low-space computation.

```
out = 0
for i=1...n:
    out = out ⊕ x_i
return out.
```

How much space are we using?

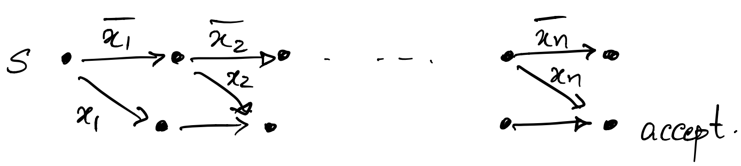
What if we allow the TM to access any input location instantly?

RAM model: Allow access to arbitrary tape location.



PARITY - $O(1)$ space

Branching programs:



OR

Accept x if there is a path from $s \rightsquigarrow t$ using only true literals.

Note: Vars need not be read in order and can be revisited.

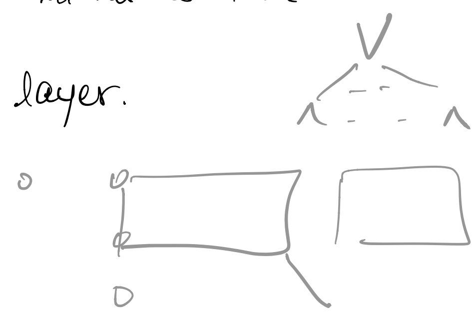
$\mathrm{SPACE}(s(n)) \subseteq \mathrm{BP}(?, ?)$

More formally:

- layered graph (edges only from layer $i \to i+1$).
- first layer with a unique vertex s.
- last layer has a special vertex t.
- Edges can either be constant, or labelled by a literal which signifies it is present iff literal is true.

Width = max # vertices in any layer.

Length = # layers.



width

length.

PARITY $\in$ BP$(2, n)$.

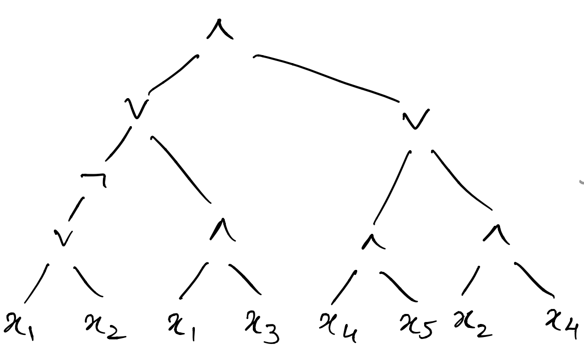
$\mathrm{Mod}_{100} \in$ BP$(100, n)$

Thm [Barrington]: Any f that is computable by a size s formula is in BP$(5, \mathrm{poly}(s))$

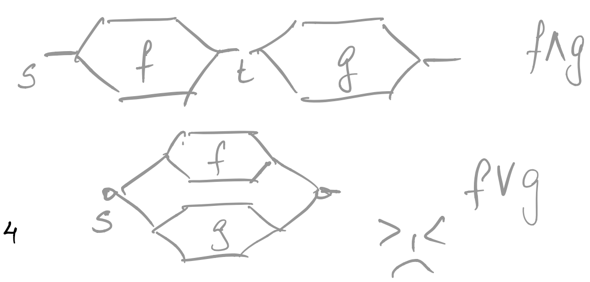
WHAT?!

Cor: $\mathrm{Mod}_{71345} \in$ BP$(5, s^2)$ & MAJ $\in$ BP$(5, s^2)$.

How do we prove something like this?



Induction:



Formulas(s) $\subseteq$ BP(s, s).

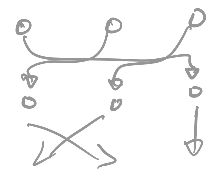
Rough roadmap: Induction

Given PBPs for f & g, build a PBP for $f \vee g$, $f \wedge g$, $\neg f$ by maintaining width.

Detour: Permutations

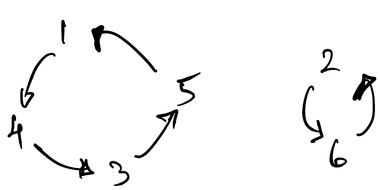
S_n = { set of all permutations of n-elements }.

$$\begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{pmatrix}\begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{pmatrix}$$



Forms a group - has identity, inverses, associative.

Cycle decomposition: $\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 4 & 6 & 5 & 3 & 1 & 2 \end{pmatrix}$



$(1\,4\,3\,5)\ (2\,6)$ α

$(1\,3\,2\,5)\ (4\,6)$ β

α, β are similar in cycle structure.

Obs: If $\sigma = \begin{pmatrix} 1 & 2 & \cdots & n \\ \sigma(1) & \sigma(2) & \cdots & \sigma(n) \end{pmatrix}$, then for any α, if $\beta = \sigma^{-1}\alpha\sigma$, then $\alpha(i) = j \iff \beta(\sigma(i)) = \sigma(j)$

Pf: $\sigma(i) \xrightarrow{\sigma^{-1}} i \xrightarrow{\alpha} j \xrightarrow{\sigma} \sigma(j)$ $\square$.

Cor: If two permutations α, β have the similar cycle type, then they are conjugates of each other.

Permutation branching program:

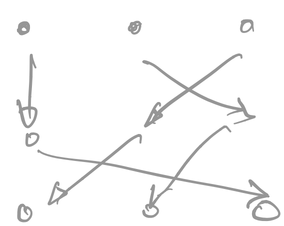
Sequence of instructions of this form:

(i, α, β) : If $x_i = 1$, apply α
else apply β.

Eg: Input $x = (1,1,0)$.

$$\big(1, \ (2\ 3), \ (1\ 2)\big)$$
$$\big(3, \ (1\ 2\ 3), \ (1\ 3\ 2)\big)$$
$$\big(1, \ (2\ 3), \ id\big)$$
$$\big(2, \ (1\ 2\ 3), \ (1\ 3)\big)$$

id



Eg: $(1, (1\ 2), id)$
$\vdots$
$(n, (1\ 2), id)$

"computing" $(1,2)$ PARITY.

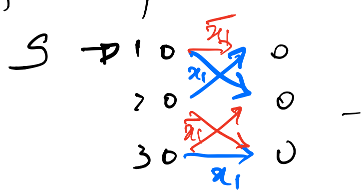
Defn: A PBP (σ, τ)-computes f if $\tau \to id$

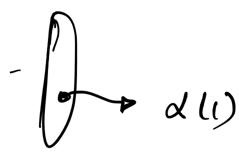
$\forall x$: $\quad f(x)=1 \Rightarrow PBP(x) = \sigma$

$\qquad\quad f(x)=0 \Rightarrow PBP(x) = \tau.$

Lemma: For any $\alpha \neq \beta \in S_m$, if f is (α,β)-computed by a PBP of length l, then $f \in BP(m, l)$.

Pf: $\alpha \neq \beta$ WLOG $\alpha(1) \neq \beta(1)$ $\qquad (1, (12), (23))$





$\square$.

$\therefore$ Suffices to construct a PBP for f.

Plan: ▸ Given PBP for f, get one for $\bar{f}$.

▸ Given a PBP for f, g, " " " $f \wedge g$.

Transforming PBPs.

Lemma: Suppose we have a PBP of length l that (α, id)-computes f. Then there is a PBP that (β, id)-computes f for any β of the same cycle type as α.

Pf: $\beta = \sigma^{-1} \alpha \sigma$

$$(1, \sigma^{-1}, \sigma^{-1}) \rightarrow \boxed{PBP \quad (\alpha, id)} \leftarrow (1, \sigma, \sigma)$$

$\square$.

Lemma: If we have a PBP that (α, id)-computes f, then we also have a PBP that (α, id) computes $\bar{f}$.

Pf: $\boxed{\text{PBP for } f} \leftarrow (1, \alpha^{-1}, \alpha^{-1})$

is a PBP that (α^{-1}, id) accepts $\bar{f}$.

Obs: α & α^{-1} have the same cycle structure

$\Rightarrow$ By prev lemma, we are done.

$\square$.

Computing $f \wedge g$?

P_1 : (α, id)- computing f.

P_2 : (β, id)- computing g.

$P_1 \; P_2$

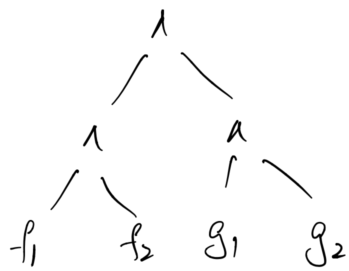
f	g	$P_1 P_2 P_1^{-1} P_2^{-1}$
1	1	$\alpha\beta\alpha^{-1}\beta^{-1}$
1	0	id
0	1	id
0	0	id

Cool idea: $\boxed{PBP_{f,\alpha}} \leftarrow \boxed{PBP_{g,\beta}} \leftarrow \boxed{PBP_{f,\alpha^{-1}}} \leftarrow \boxed{PBP_{g,\beta^{-1}}}$

$\alpha \quad id \quad \alpha^{-1} \quad id$

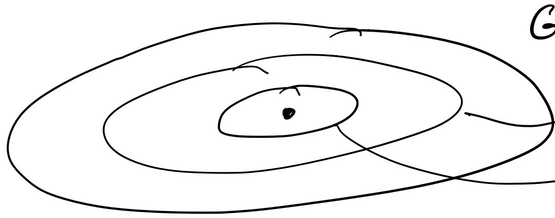
α (1 2) β: (2 3) $\alpha\beta\alpha^{-1}\beta^{-1}$ - 3cycle.

Lemma: Suppose P_1 (12)-computes f & P_2 (23)-computes g. Then the "program" $P_1 P_2 P_1^{-1} P_2^{-1}$ (132)-computes $f \wedge g$.



$(1\ 3\ 2)$- computation.

$(1,2)$- computations.



$G = G_0$

$G_1 = \{\alpha\beta\alpha^{-1}\beta^{-1} : \alpha, \beta \in G_0\}$

$G_2 = \{\alpha\beta\alpha^{-1}\beta^{-1} : \alpha, \beta \in G_1\}$

Fact: There are five cycles π_1, π_2, π_3 s.t $\pi_3 = \pi_1 \pi_2 \pi_1^{-1} \pi_2^{-1}$

Eg: $\pi_1 = (1\,2\,3\,4\,5)$ $\quad \pi_2 = (1\,3\,5\,4\,2)$

$\pi_3 = \pi_1 \pi_2 \pi_1^{-1} \pi_2^{-1} = (1\,3\,2\,5\,4)$

Lemma: Let $\alpha = (1\,2\,3\,4\,5)$. If P_1 & P_2 are PBPs of length $\leq l$ α-computing f, g respectively. Then, there is a PBP of length $\leq 4l$ that α-computes $f \wedge g$.

5-cycle

Pf: $\alpha = (1\,2\,3\,4\,5)$

$\square$.

Cor: If f is computable by a formula of depth $\leq d$, then we have a PBP of width 5 and length $\leq 4^d = 4^{O(\log s)} = \text{poly}(s)$ computing f.

Fact: Any formula of size s has an equivalent formula of size $\text{poly}(s)$ & depth $O(\log s)$

$$G \supseteq \{\pi_1, \pi_2, \pi_3\}.$$

$$G_1 := \{\alpha\beta\alpha^{-1}\beta^{-1} : \alpha, \beta \in G\}$$
$$\supseteq \{\pi_1, \pi_2\pi_3\}$$