Dependence of complexity on # tapes

Df: $L_{PAL} = \{\alpha\alpha^R \mid \alpha \in \{0,1\}^*\}$ "even palindromes"
(α^R — reversed)

- trivial 2-tape TM deciding L_{PAL} in $\Theta(n)$ time.
- but all machines we found with 1 tape run in $\Theta(n^2)$ time!

Thm: Every 1-tape machine deciding L_{PAL} runs in time $\Omega(n^2)$.

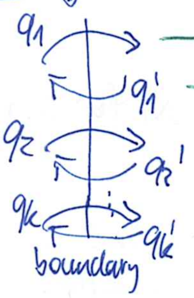
Proof: Assume there is M deciding L_{PAL} s.t. $T_M(n) \in o(n^2)$. ← that is: $\forall \varepsilon > 0\ \exists^\infty n : T_M(n) < \varepsilon n^2$.

- Consider inputs of type: for $6 \mid n$

part L	part Z	part R
α	$00\ldots0$	α^R
$n/3$	$n/3$	$n/3$

← answer is YES for every such string

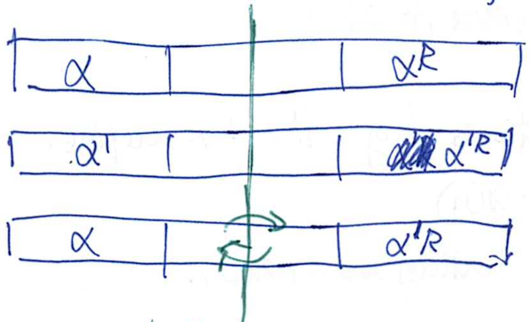
- Consider boundary between 2 zeros in part Z & how computation of M crosses it:



— first, we cross from left to right in some state q_1
— then we return in state q_1'
— WLOG computation ends in L
(we can modify M to "rewind" without affecting $o(n^2)$)

↳ crossing sequence $(q_1, q_1'), \ldots, (q_k, q_k')$

- If two inputs with α, α' have the same C.S. for the same boundary:



answers YES
answers YES
"mixed" input — also answers YES, although the answer should be NO ↯

both parts communicate only via crossings

- Similarly if they have same C.S. for different boundaries.
(part Z can have odd length, but that implies NO anyway)

- Let's use P.H.P. (Pigeon-hole principle) to show that such α, α' exist:

$\#$ c.s. of length k $= |Q|^{2k}$

$\#$ c.s. of length at most k $\leq c \cdot |Q|^{2k}$ for some constant c
(via sum of geom. series)

If $\#$c.s. $<$ $\#$ inputs, then $\exists$ two inputs with the same C.S.
↳ so we want $c \cdot |Q|^{2k} < 2^{n/3}$ … $2^{\underbrace{\log c + 2k \log|Q|}_{\leq 3k \log|Q|}} < 2^{\frac{n}{3}}$ … $k < \frac{n}{9 \log|Q|}$

- P.H.P. once again (we can find boundary with a small # crossings):
  - we have $n/3$ boundaries
  - Σ of lengths of their C.S. $\leq T_M(n) < \varepsilon n^2$
  - $\Rightarrow \exists$ boundary with at most $\frac{\varepsilon n^2}{n/3} = 3\varepsilon n$ crossings
- now set ε such that we have:

$$\text{min. \# crossings} \leq 3\varepsilon n < \frac{n}{9 \log|Q|}$$

such ε exists
& inequality satisfied for n large enough

- so there are 2 inputs with the same C.S. $\Rightarrow$ mixing produces contradiction.

Thm: For every multi-tape TM M there is 1-tape TM M' s.t. $L(M) = L(M')$ & $T_{M'}(n) \in O(T_M(n)^2)$.

Proof: Analyze reduction from previous lectures. [hint: at most $T_M(n)$ tape cells used on each tape]

Thm: For every multi-tape TM M there is 2-tape TM M' s.t. $L(M) = L(M')$ & $T_{M'}(n) \in O(T_M(n) \cdot \log T_M(n))$.

(proof omitted)

Time complexity classes

- $\text{DTIME}(f) := \{ L \subseteq \{0,1\}^* \mid \exists M \text{ multi-tape TM deciding } L \text{ in time } \underbrace{O(f)}_{T_M \in O(f)} \}$

$f: \mathbb{N} \to \mathbb{R}$

sometimes just $\text{TIME}(f)$
["D" is for "deterministic"]

every finite Σ s.t. $|\Sigma| \geq 2$ would work

this will be the default

makes things easier, but for TMs it's not necessary because of time compression thm.

- We will require f to have these properties:
  1. non-decreasing
  2. $\forall n\ f(n) \geq n$
  3. time-constructible $\equiv \exists$ TM M_f which for input 1^n produces output $1^{f(n)}$ in time $O(f(n))$

"proper time complexity function"

- $P := \bigcup_{i \geq 1} \text{DTIME}(n^i)$ ← polynomial-time decidable languages

Why we like P:
- Corresponds (roughly) to "efficiently solvable"
- independent of model of computation (RAM gives the same P as TM)
- ~~...~~ polynomials are the smallest set of functions containing constants & identity and closed under addition, multiplication and composition.

Examples:
- reachability in graphs
- evaluation of Boolean formulas

Classes of functions

- $\text{DTIMEF}(f) := \{g : \Sigma_1^* \to \Sigma_1^* \mid \exists \text{TM } M \text{ computing } g \text{ in time } O(f)\}$ — $\{0,1\}^*$
- $\text{PF} := \bigcup_{i\geq 1} \text{DTIME}(n^i)$

if $|f(n)| \in \text{poly}(n)$ ($O(n^k)$ for some fixed k): $L_g := \{\langle \alpha, i \rangle \mid i\text{-th bit of } g(\alpha) \text{ is } 1\}$ (i encoded in binary)

$L_g \in P \iff g \in \text{PF}$.

So studying only languages in P is WLOG.

Consider these problems: (as usually: suitably encoded)

1. HAMILTON PATH — Input: undirected graph G, vertices u,v
   Question: $\exists$ path in G with endpoints u,v containing all vertices (exactly) once.
   (path vs. walk ← can repeat; path: doesn't repeat vertices)

2. 3-COLORING — Input: undirected graph G
   Q: $\exists f: V(G) \to \{1,2,3\}$ s.t. $\forall \{u,v\} \in E(G): f(u) \neq f(v)$
   (f: coloring of G with 3 colors)

3. INDEPENDENT SET — Input: undirected graph G, $k \in \mathbb{N}$
   Q: $\exists A \subseteq V(G): |A| \geq k \;\&\; \forall u,v \in A: \{u,v\} \notin E(G)$

4. CLIQUE — Input: G, $k \in \mathbb{N}$
   Q: $\exists A \subseteq V(G): |A| \geq k \;\&\; \forall u,v \in A: (u = v \lor \{u,v\} \in E(G))$

5. 0,1-Equations — Input: matrix A, vector b with 0/1 entries
   Q: $\exists x \in \{0,1\}^n: Ax = b$ (evaluated in integers, not $\mathbb{Z}_2$)

6. SAT (Boolean satisfiability): Input: Boolean formula $\varphi(x_1 - x_m)$ in CNF
   Q: $\exists x_1 - x_m \in \{0,1\}$ s.t. $\varphi(\vec{x})$ is true.

For all these: we are looking for something we can easily recognize (poly-time), but we don't know how to find it in poly time.

$\varphi \equiv (x_1 \lor x_2 \lor x_3) \land (\ldots) \land (\ldots)$ — clause

literals: either x_i or $\neg x_i$

(restriction to CNF is WLOG, see later)

(we cannot use thm. from Logic about equivalent formulas in CNF, because it blows up size exponentially)

Reductions will again prove themselves useful:

Df: For languages K, L: $K \leq_m^P L \equiv \exists f \in \text{PF}$ s.t. $\forall \alpha \in \Sigma_1^*$ $\alpha \in K \Leftrightarrow f(\alpha) \in L$.

($\leq_m^P$: polynomial-time many-to-one reduction)

Properties of $\leq_m^P$:

- reflexive & transitive (quasi-order)
- $\exists$ incomparable languages (exercise)
- $K \leq_m^P L$ and $L \in P \Rightarrow K \in P$ [composition of 2 algorithms running in poly. time is again poly-time]

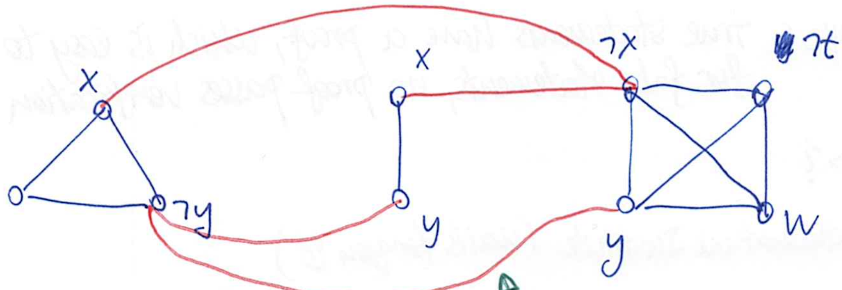
size of input for ALG2 is at most time complexity of ALG1

Example: SAT $\leq_m^P$ INDEP SET

$$(x \vee \neg y \vee z) \wedge (x \vee y) \wedge (\neg x \vee y \vee \neg t \vee w)$$

given a formula → produce a graph, $k := \#\text{clauses}$ ⇒ from each subgraph we must select exactly 1 vertex

each clause gets complete subgraph labelled with literals of the clause



add edges between conflicting literals

- $\exists$ satisfying assignment: for each clause, pick 1 satisfied literal, put its vertex to the indep. set → got I.S. of size k
- $\exists$ indep. set of size k: each vertex ~~selected~~ in I.S. selects a variable which will be set to 0/1 to satisfy the corresponding clause, red edges guarantee that we won't set var to both 0 and 1
  - remaining variables set arbitrarily

  ↳ got satisfying assignment
- the reduction runs in poly. time

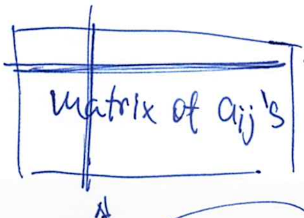
Example: INDEP SET $\leq_m^P$ SAT ... given G, k, construct formula φ s.t. φ is satisfiable $\Leftrightarrow$ G has ind. set of size k

Variables: x_1 — x_n : vertex v_i selected to ind. set

a_{ij} for $1 \leq i \leq k$, $1 \leq j \leq n$: vertex j is i-th in the ind. set ← order on vertices of the set

Clauses: $\forall \{v_i, v_j\} \in E(G)$: $\neg x_i \vee \neg x_j$

$\forall i,j \quad a_{ij} \Rightarrow x_j$ ← order describes the set (we allow unordered elements of set, which doesn't break anything)



$a_{ij} \Rightarrow \neg a_{ij'}$ no number used multiple times

$a_{ij} \Rightarrow \neg a_{i'j}$ no vertex used twice or more

$a_{i1} \vee \dots \vee a_{in}$ each number used at least once

implication $x \Rightarrow y$ is a clause $\neg x \vee y$ → so we get CNF