Polytime reductions

Turing vs Many-one reductions.

$A \to B$. A uses B as an oracle. if $B \in NP \Rightarrow A \in NP$? ← equivalent to $NP = co\text{-}NP$? (we don't know)

If $B \in NP$ and $A = \overline{B} \to B$ then $\overline{B} \in NP$ i.e. $NP = co\text{-}NP$

$SAT \to \overline{SAT}$

oracle access, flip ans.

Primality testing

Primes $\in co\text{-}NP$

↳ Give composite numbers as proof.

Primes $\in P$ {AKS test}

↳ Primes $\in NP$ $\{1, \ldots, N-1\}$ is cyclic iff N is prime. (& order $N-1$)

1. Guess a s.t. $a^n \neq 1$ for any $n < N-1$

$a^{N-1} \equiv 1$. So, we need $a^{(N-1)/p_i} \neq 1$ for every prime divisor of $N-1$

2. Guess $p_1, \ldots, p_k$ $(N-1$.

Certificate for p_i s being prime.

1. Largest prime factor of $N-1$ has size $\leq \sqrt{N}$
2. Atmost $\log(N-1)$ primes

$S(\log N) = \sum_i S(\log p_i) + \underbrace{\log^2(N)}_{\text{verif } a^{(N-1)/p_i} \neq 1}$

$\leq \log^2 N + \sum (\log p_i)^c$

$\leq \log^2 N + \left(\sum \log p_i\right)^c$ i.e. on $\log N$ numbers.

$\leq \log^2 N + (\log(N-1))^c$ exp. takes time $\log N$

$= \vartheta((\log N)^c)$ ∎

ZPP = RP ∩ co-RP

① $L \in RP \cap co\text{-}RP$ (M_1) (M_2)

Monte-carlo TMs for L and $\bar{L}$

1. run w on M_1, if M_1 accepts, accept
2. run w on M_2, if M_2 accepts, reject

→ if $w \in L$, has 50% chance of working } 50% in each round.
$w \notin L$, 50% chance

Time = $2p(n)$

Next, repeating, p(reaches round i) = $\left(\frac{1}{2}\right)^i$

So $\mathbb{E}$[running time] = $2p(n)\sum\left(\frac{1}{2}\right)^i = 4p(n)$.

Q: Almost sure termination ? → Yes, $\mathbb{E}$[running time] = stronger.

② $L \in ZPP$.

M_1 accepts or rejects in expected time $p(n)$

M_2: simulates M_1 for $2p(n)$ steps.

Let p(acceptance within $2p(n)$ time) = $c < 1/2$

$\mathbb{E}_{M_1}$(time) $\geq (1-c)\,2p(n) > p(n)$ i.e. contradiction.

$\{2(1-c) \geq 1\}$

Hence, $L \in RP$.

$\bar{L} \in ZPP$ so, similarly $L \in co\text{-}RP$.

RP ⊆ NP

$L \in RP$

$\exists$ TM M s.t. if $w \notin L$, M rejects w } all runs in time poly(n)
$w \in L$, 50% accept

Instead of having p's introduce non-deterministic transitions.

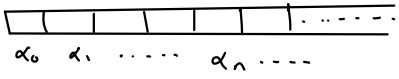
So, $w \in L$, then accepted.

Cook's Thm (SAT is NP-complete)

1. SAT is in NP.
   - Guess an assignment x. If encoded E is of length n, $O(n)$ time suffices
   - Evaluate assignment T, if $E(T)=1$, accept

2. If L is in NP, it reduces to SAT.

Let $\exists$ single tape NTM s.t. $p(n)$ steps almost for $w \in L$.



M accepts input w then,

1. α_0 = initial ID

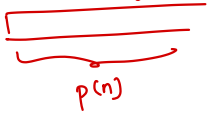
$\alpha_0 \to \alpha_1 \to \cdots \to \alpha_k$, where $k \le p(n)$

α_k = accepting

2. α_i = sequence of symbols $X_{i0} X_{i1} \cdots X_{ip(n)}$. X_{ij} = state, we need only $p(n)$ tape symbols

(state of turing machine tape ↳ head + symbols

$y_{ij,A} : X_{ij} = A$



ID	0	1	...				...	p(n)
α_0	X_{00}	X_{01}						$X_{0,p(n)}$
α_1	X_{10}	X_{11}						$X_{1,p(n)}$
α_i				$X_{i,j-1}$	$X_{i,j}$	$X_{i,j+1}$		
α_{i+1}				$X_{i+1,j-1}$	$X_{i+1,j}$	$X_{i+1,j+1}$		
$\alpha_{p(n)}$	$X_{p(n),0}$	$X_{p(n),1}$						$X_{p(n),p(n)}$

3. i) $q_0 w$ is initial state
   ii) Next move is correct
   iii) Finishes right → there is some ID i.e. accepting
   (correct)

Starts Right

X_{00} must be the start state q_0 of M, X_{01} through X_{0n} must be w (where n is the length of w), and the remaining X_{0j}, must be the blank, B. That is, if $w = a_1 a_2 \cdots a_n$, then:

$$S = y_{00q_0} \wedge y_{01a_1} \wedge y_{02a_2} \wedge \cdots \wedge y_{0na_n} \wedge y_{0,n+1,B} \wedge y_{0,n+2,B} \wedge \cdots \wedge y_{0,p(n),B}$$

Since we assume that an accepting ID repeats forever, acceptance by M is the same as finding an accepting state in $\alpha_{p(n)}$. Remember that we assume M is an NTM that, if it accepts, does so within $p(n)$ steps. Thus, F is the OR of expressions F_j, for $j = 0, 1, \ldots, p(n)$, where F_j says that $X_{p(n),j}$ is an accepting state. That is, F_j is $y_{p(n),j,a_1} \vee y_{p(n),j,a_2} \vee \cdots \vee y_{p(n),j,a_k}$, where $a_1, a_2, \ldots, a_k$ are all the accepting states of M. Then,

$$F = F_0 \vee F_1 \vee \cdots \vee F_{p(n)}$$

Construction of N

$N = N_0 \wedge N_1 \wedge \cdots N_{p(n)-1}$

{ α_{i+1} is one of possibilities allowed by TM M using α_i } ← N_i:

$N_i = \bigwedge_{j=0}^{p(n)} (A_{ij} \vee B_{ij})$ ← head cannot influence state j

State is X_{ij} ← State of α_i is at position j (head) and ∃ move to transform $X_{i+1,j} = X_{i,j}$

B_{ij} is the easier to write. Let $q_1, q_2, \ldots, q_m$ be the states of M, and let $Z_1, Z_2, \ldots, Z_r$ be the tape symbols. Then:

$$B_{ij} = (y_{i,j-1,Z_1} \vee y_{i,j-1,Z_2} \vee \cdots \vee y_{i,j-1,Z_r}) \wedge \quad \leftarrow X_{i,j-1} \text{ is a tape symbol}$$
$$(y_{i,j,Z_1} \vee y_{i,j,Z_2} \vee \cdots \vee y_{i,j,Z_r}) \wedge \quad \leftarrow X_{i,j}$$
$$(y_{i,j+1,Z_1} \vee y_{i,j+1,Z_2} \vee \cdots \vee y_{i,j+1,Z_r}) \wedge \quad \leftarrow X_{i,j+1}$$
$$\to ((y_{i,j,Z_1} \wedge y_{i+1,j,Z_1}) \vee (y_{i,j,Z_2} \wedge y_{i+1,j,Z_2}) \vee \cdots \vee (y_{i,j,Z_r} \wedge y_{i+1,j,Z_r}))$$

$X_{i,j} = X_{i+1,j}$

$X_{i,j-1}$, X_{ij}, $X_{i,j+1}$, $X_{i+1,j-1}$, $X_{i+1,j}$, and $X_{i+1,j+1}$. An assignment of symbols to each of these six variables is *valid* if:

1. X_{ij} is a state, but $X_{i,j-1}$ and $X_{i,j+1}$ are tape symbols.
2. Exactly one of $X_{i+1,j-1}$, $X_{i+1,j}$, and $X_{i+1,j+1}$ is a state.
3. There is a move of M that explains how $X_{i,j-1}X_{ij}X_{i,j+1}$ becomes

$$X_{i+1,j-1}X_{i+1,j}X_{i+1,j+1}$$

Possibilities:

$$y_{i,j-1,D} \wedge y_{i,j,q} \wedge y_{i,j+1,A} \wedge y_{i+1,j-1,p} \wedge y_{i+1,j,D} \wedge y_{i+1,j+1,C}$$

$$y_{i,j-1,D} \wedge y_{i,j,q} \wedge y_{i,j+1,A} \wedge y_{i+1,j-1,D} \wedge y_{i+1,j,C} \wedge y_{i+1,j+1,p}$$

$D\,q\,A \to p\,D\,C$, $\delta(q,A) = (p, C, L)$

$\hookrightarrow D\,C\,p$, $\delta(q,A) = (p, C, R)$

A_{ij} = OR of all valid terms

$$E_{M,w} = S \wedge N \wedge F$$

as a function of both M and w, the fact is that only the "starts right" part S that depends on w, and it does so in a simple way (w is on the tape of the initial ID). The other parts, N and F, depend on M and on n, the length of w, only.

Theorem 11.3: If M is a polynomial-space-bounded TM (deterministic or nondeterministic), and $p(n)$ is its polynomial space bound, then there is a constant c such that if M accepts its input w of length n, it does so within $c^{1+p(n)}$ moves.

Pf: Idea (ID must repeat in those many moves ⇒ smaller sequence)

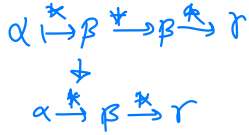
t = number of tape symbols

s = number of states in M

Number of different IDs used = $s\,p(n)\,t^{p(n)}$

position for head, state = $s\,p(n)$

tape possibilities = $t^{p(n)}$



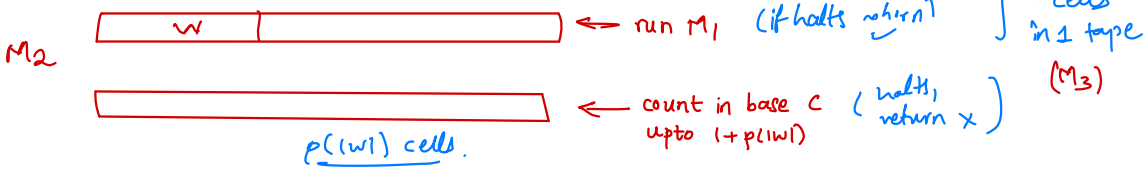
Pick $c = s + t$

$$(t+s)^{1+p(n)} = t^{1+p(n)} + s\,t^{p(n)}\,(1+p(n)) \geq \#\text{ids}.$$

∎

Theorem 11.4: If L is a language in $\mathcal{PS}$ (respectively $\mathcal{NPS}$), then L is accepted by a polynomial-space-bounded deterministic (respectively nondeterministic) TM that halts after making at most $c^{q(n)}$ moves, for some polynomial $q(n)$ and constant $c > 1$.

L is accepted by M_1, s.t. $p(n)$ symbols, accepts in $c^{1+p(|w|)}$ steps.



Savich's thm

Theorem 11.5: (*Savitch's Theorem*) $\mathcal{PS} = \mathcal{NPS}$.

DTM

can NTM I → J in m steps

$\mathcal{PS} \subseteq \mathcal{NPS}$ (trivial since DTM ⊆ NTM)

Let L be accepted by some NTM N with space bound $p(n)$

$\Rightarrow$ accepts with $c^{1+p(n)}$ steps

[I_0, J, m]

1. I_0 is the initial ID of N with input w.
2. J is any accepting ID that uses at most $p(n)$ tape cells; the different J's are enumerated systematically by D, using a scratch tape.
3. $m = c^{1+p(n)}$.

We argued above that there will never be more than $\log_2 m$ recursive calls that are active at the same time, i.e., one with third argument m, one with $m/2$, one with $m/4$, and so on, down to 1. Thus, there are no more than $\log_2 m$ stack frames on the stack, and $\log_2 m$ is $O(p(n))$.

$\log_2(c^{1+p(n)}) \Rightarrow$ space = $O(p(n)^2)$

[I, J, m] ← poly size each

QBF is PS – Complete

1. QBF is in PS

1. The expression is of the form (E). Then E is of length $n-2$ and can be evaluated to be either 0 or 1. The value of (E) is the same.
2. The expression is of the form $\neg E$. Then E is of length $n-1$ and can be evaluated. If $E = 1$, then $\neg E = 0$, and vice versa.
3. The expression is of the form EF. Then both E and F are shorter than n, and so can be evaluated. The value of EF is 1 if both E and F have the value 1, and $EF = 0$ if either is 0.
4. The expression is of the form $E + F$. Then both E and F are shorter than n, and so can be evaluated. The value of $E + F$ is 1 if either E or F has the value 1, and $E + F = 0$ if both are 0.
5. If the expression is of the form $(\forall x)(E)$, first replace all occurrences of x in E by 0 to get the expression E_0, and also replace each occurrence of x in E by 1, to get the expression E_1. Observe that E_0 and E_1 both:
   (a) Have no free variables, because any occurrence of a free variable in E_0 or E_1 could not be x, and therefore would be some variable that is also free in E.
   (b) Have length $n-6$, and thus are shorter than n.

   Evaluate E_0 and E_1. If both have value 1, then $(\forall x)(E)$ has value 1; otherwise it has the value 0. Note how this rule reflects the "for all x" interpretation of $(\forall x)$.
6. If the given expression is $(\exists x)(E)$, then proceed as in (5), constructing E_0 and E_1, and evaluating them. If either E_0 or E_1 has value 1, then $(\exists x)(E)$ has value 1; otherwise it has value 0. Note that this rule reflects the "exists x" interpretation of $(\exists x)$.

1. Recursively evaluate using stack as TM.

2. PePS → QBF. (Similar to pf of cook's thm) ∎