CPTH 2

Saswata Mukherjee (BMC201945)
Somnath Bhattacharjee (BMC201954)

March 12, 2021

Question 1.

say, $L \in \Sigma_p^2$
$\implies x \in L \iff \exists u_1 \, \forall u_2 \, M(x, u_1, u_2) = 1$ where M is a poly-time DTM.
take $L' = \{(x, y) : \forall u_2 \, M(x, y, u_2) = 1\} \in$ co-NP.
$\implies x \in L \iff \exists u_1$ so that $(x, u_1) \in L'$.
so, $L \in \text{NP}^{L'}$ and $L' \in$ co-NP $\implies L' \in P^{SAT}$.
Now, on input x non-deterministically guess u_1 and reduce (x, u_1) to corresponding SAT instance. Then by oracle call SAT check whether (x, u_1) is in L'.
therefore $L \in \text{NP}^{SAT} = \text{NP}^{\text{NP}}$ as, SAT is NP-complete.
so, $\Sigma_p^2 \subseteq \text{NP}^{\text{NP}}$.

again say, $L \in \text{NP}^{\text{NP}} = \text{NP}^{SAT}$.
then, $\exists$ NDTM N so that it uses oracle query of SAT.
Now, $x \in L \iff \exists$ non-deterministic choices for x in N say, $\{c_1, c_2, \ldots, c_n\}$ where $c_i \in \{0, 1\}$. And, it makes oracle call k times and the sequence of the answers is $\{a_1, \ldots, a_m\}$ so that $a_i \in \{0, 1\}$.
Now, $a_i = 1 \implies \exists$ CNF ϕ_i so that $\exists$ assignment $s_i \in \{0, 1\}^{|\phi_i|}$ so that $\phi_i(s_i) = 1$
and $a_j = 0 \implies \exists$ CNF ϕ_j so that $\forall d_j \in \{0, 1\}^{|\phi_j|}$, $\phi_j(d_j) = 0$
so, $x \in L \iff \exists c_1, \ldots, c_n \, , \, a_1, \ldots, a_m \, , \, s_1, \ldots, s_m \;\; \forall d_1, \ldots, d_m$ so that,
N makes choice $c_1, \ldots, c_n$, & $a_1, \ldots, a_m$ are answers to oracle call
& $\forall i \in [k]$ if $a_i = 1$ then $\phi_i(s_i) = 1$
& $\forall i \in [k]$ if $a_i = 0$ then $\phi_i(d_i) = 0$
Hence, by definition of Σ_2^p , $L \in \Sigma_2^p \implies \Sigma_2^p = \text{NP}^{\text{NP}}$

Question 2.

If a reduction takes $O(\log n)$ space then clearly it takes $O(2^{\log n}) = O(n)$ time
Ans we know that NP is closed under polytime reduction
Hence NP is closed under log-space reduction

Now we will show that SPACE(n) is not closed under logspace reduction
Let $L \in \text{SPACE}(n^2)$, $L \notin \text{SPACE}(n)$
(We know such language exists from space hierarchy theorem)

Now define

$$\gamma(L) = \{\gamma(w) = w\#0^{n^2-n} \big| w \in L, |w| = n\}$$

Clearly γ is a log-space reduction since to write the number $n^2 - n$ takes $O(\log(n^2 - n)) = O(\log n^2) = O(\log n)$ space

Claim. $\gamma(L) \in \text{SPACE}(n)$

Note we have a DTM M which takes $O(|x|^2)$ space to conclude the membership for x in L
Hence we can have a DTM M' which converts $x \in \Sigma^*$ to $\gamma^{-1}(x)$ (By removing the zeros after #, provided they exist with proper cardinality, other wise clearly $x \notin \gamma(L)$)
then uses M to find whether $\gamma^{-1}(x) \in L$ or not
Clearly this takes logarithmic space for the conversion of x to $\gamma^{-1}(x)$ And $|\gamma^{-1}(x)| = \sqrt{|x|}$
Hence the entire thing will be done in $O(|\gamma^{-1}(x)|^2 + \log|x|) = O(\sqrt{|x|}^2 + \log|x|) = O(|x|)$ space
Hence our claim is true

Now $L \notin$ SPACE(n) but $\gamma(L) \in$ SPACE(n)
Hence SPACE(n) is not closed under log-space reduction

Question 3.

say S be a sparse language which is NP-complete.
$\implies \exists$ poly-time computable function f s.t. $x \in SAT \iff f(x) \in S$.
hence $\exists\, c$ s.t. $|f(x)| = O(|x|^c)$

here we define an algorithm for SAT: (on input ϕ)
(i) define $S_n = \{(f(\phi), \phi)\}$
(ii) say we have $S_n, S_{n-1}, \ldots, S_{k+1}$
then, $S_k = \bigcup_{\psi \in S_{k+1}} \{(f(\psi|_{x_k=1}), \psi|_{x_k=1}), (f(\psi|_{x_k=0}), \psi|_{x_k=0})\}$
and if $\exists (f(\psi_i), \psi_i), (f(\psi_j), \psi_j) \in S_k$ s.t. $f(\psi_i) = f(\psi_j)$ then remove one of them. (we do not allow repetition on $f(\psi)$)
(iii) construct $S_n, \ldots, S_0$, if $(f(1), 1) \in S_0$ then accept else reject.

we know if $f(\psi_i) = f(\psi_j) \implies \psi_i \in SAT \iff \psi_j \in SAT$
so, our algorithm is correct as, $\phi \in SAT \iff \forall\, k\, \exists\, \psi \in S_k$ s.t. $\psi \in SAT$

Claim. $\forall k\ |S_k|$ is polynomially bounded.

as, $\forall \psi \in S_k$, $|\psi| = O(n - k)$ as, $\psi \in S_k \implies \psi$ has $n - k$ variables.
$\implies |f(\psi)| = O((n-k)^c)$
again, $\exists\, d$ s.t. $|S^{=n}| = O(n^d)$ for all n
hence, $|S^{=O((n-k)^c)}| = O((n-k)^{cd})$

also, S_k has no repetition on $f(\psi) \implies |S_k| = O((n-k)^{cd}) = O(n^{cd})$
our claim is true.

so, this algorithm runs for n steps and each step take time $O(n^{cd})$
$\implies SAT \in P \implies$ P = NP.

Question 4.

a.
For intersection in BPP
We will take the constant $\frac{3}{4}$ instead of $\frac{2}{3}$
Let $L_1, L_2 \in$ BPP which can be recognised by the PTM M_1, M_2 respectvely
let us construct a PTM M which on input x simulates x on M_1 and M_2 both and returns $M_1(x) \wedge M_2(x)$
The probability constant for M will be $\frac{9}{16}$ (which is clearly $> \frac{1}{2}$)

Now we know that $P(A \cup B) \geq P(A), P(A \cap B) = P(A) \times P(B)$ and if $A \implies B$ then $P(A) \leq P(B)$
Now if $x \in L_1 \cap L_2$ then $M_1(x) = 1$ and $M_2(x) = 1 \implies M(x) = 1$

$$\begin{aligned} & x \in L_1 \cap L_2 \\ \implies & P(M_1(x) = 1) \geq \frac{3}{4} \textbf{ and } P(M_2(x) = 1) \geq \frac{3}{4} \\ \implies & P(M(x) = 1) \geq P(M_1(x) = 1 \text{ and } M_2(x) = 1) \\ & = P(M_1(x) = 1) \times P(M_2(x) = 1) \\ & \geq \frac{3}{4} \times \frac{3}{4} = \frac{9}{16} \\ \implies & M \text{ accpets } x \end{aligned}$$

Now suppose $x \notin L_1 \cap L_2$

if $x \notin L_1$ then $P(M(x) = 0) = P(M_1(x) = 0 \cup M_2(x) = 0) \geq P(M_1(x) = 0) = \frac{3}{4} > \frac{9}{16}$
Hence M rejects x

If $x \in L_1 - L_2$ then $M_1(x) = 1$ and $M_2(x) = 0 \implies M(x) = 0$
Hence

$$\begin{aligned} P(M(x) = 0) &\geq P(M_1(x) = 1 \text{ and } M_2(x) = 0) \\ &= P(M_1(x) = 1)P(M_2(x) = 0) \geq \frac{3}{4} \times \frac{3}{4} = \frac{9}{16} \end{aligned}$$

Hence M rejects x

For intersection in RP
We will take the constant $\frac{3}{4}$ instead of $\frac{2}{3}$
Let $L_1, L_2 \in$ RP which can be recognised by the PTM M_1, M_2 respectvely
let us construct a PTM M which on input x simulates x on M_1 and M_2 both and returns $M_1(x) \wedge M_2(x)$

The probability constant for M will be $\frac{9}{16}$ (which is clearly $> \frac{1}{2}$)

Now we know that $P(A \cup B) \geq P(A), P(A \cap B) = P(A) \times P(B)$ and if $A \implies B$ then $P(A) \leq P(B)$
Now if $x \in L_1 \cap L_2$ then $M_1(x) = 1$ and $M_2(x) = 1 \implies M(x) = 1$

$$\begin{aligned}
& x \in L_1 \cap L_2 \\
\implies & P(M_1(x) = 1) \geq \frac{3}{4} \textbf{ and } P(M_2(x) = 1) \geq \frac{3}{4} \\
\implies & P(M(x) = 1) \geq P(M_1(x) = 1 \text{ and } M_2(x) = 1) \\
& = P(M_1(x) = 1) \times P(M_2(x) = 1) \\
& \geq \frac{3}{4} \times \frac{3}{4} = \frac{9}{16} \\
\implies & M \text{ accpets } x
\end{aligned}$$

Now if $x \notin L_1 \cap L_2$ then

$$\begin{aligned}
P(M(x) = 0) &= P(M_1(x) = 0 \text{ or } M_2(x) = 0) \\
&\geq \max\{P(M_1(x) = 0), P(M_2(x) = 0)\} = 1
\end{aligned}$$

Hence $P(M(x) = 0) = 1$

For union in RP
Let $L_1, L_2 \in$ RP which can be recognised by the PTM M_1, M_2 respectvely
let us construct a PTM M which on input x simulates x on M_1 or on M_2 randomly and returns the output accordingly
Now if $x \in L_1 \cup L_2$

$$\begin{aligned}
P(M(x) = 1) =& P(M_1(x) = 1 \cap M_1 \text{ has been choose }) + P(M_2(x) = 1 \cap M_2 \text{ has been choose}) \\
=& P(M_1(x) = 1 \big| M_1 \text{ has been choose }) P(M_1 \text{ has been choose}) \\
& + P(M_2(x) = 1 \big| M_2 \text{ has been choose }) P(M_2 \text{ has been choose}) \\
\geq & \frac{2}{3} \times \frac{1}{2} + \frac{2}{3} \times \frac{1}{2} = \frac{2}{3}
\end{aligned}$$

Hence M will accept x

Now if $x \notin L_1 \cup L_2$

$$\begin{aligned}
P(M(x) = 0) =& P(M_1(x) = 0 \cap M_1 \text{ has been choose }) + P(M_2(x) = 0 \cap M_2 \text{ has been choose}) \\
=& P(M_1(x) = 0 \big| M_1 \text{ has been choose }) P(M_1 \text{ has been choose}) \\
& + P(M_2(x) = 0 \big| M_2 \text{ has been choose }) P(M_2 \text{ has been choose}) \\
=& 1 \times \frac{1}{2} + 1 \times \frac{1}{2} = 1
\end{aligned}$$

Hence M will reject x

For union of BPP we will prove that BPP is closed under complementation in the next problem
Now using de morgan law $L_1 \cup L_2 = (L_1 \cap L_2)^c$ we can say BPP is closed under

union

b.
Let L is in BPP with PTM M
Let us define M' which on input x returns $\not M(x)$
Now

$$P(M(x) = L(x)) \geq \frac{2}{3}$$
$$\implies P(M'(x) = L^c(x)) \geq \frac{2}{3}$$

Hence M' recognises L^c

Now we can not say anything about complementation of RP. We know that RP=co-RP is a open problem. If RP is closed under complementation then clearly ZPP=RP

c.
We know that there exists a language L s.t. $L \in \text{DTIME}(2^{2^n})$ but $L \notin \text{EXP}$
Let 1^L be the unary representation of L
Then note for $w \in L$, $|1^w| = 2^{|w|}$

Claim. $1^L \notin \text{P}$

Suppose $1^L \in \text{P}$ then we have a polytime DTM M for 1^L
Now we can construct a TM M' which on input w writes 1^w which takes $O(2^{|w|})$ time
Now M' will use M to check whether 1^w is in 1^L or not which will take $O((2^{|w|})^c)$ time.
Clearly M' will stop the execution in exponential time and it can recognise L completely
which contradicts $L \notin \text{EXP}$
Hence our assumption was wrong, our claim is true

Now clearly $1^L \in P/poly$

d.

Claim. $\overline{SAT} \in P^{SAT}$

For a input φ, a oracle machine M will check whether φ in SAT or not as an oracle call and returns the exact opposite
Clearly M runs in polytime
Hence our claim is true

Now from the last assignment we know that $\overline{SAT}$ is co-NP complete
Hence for any $L \in$ co-NP we have a polytime reduction γ to $\overline{SAT}$, we have a polytime oracle machine M which converts x to $\gamma(x)$ and then checks for $\gamma(x)$ in $\overline{SAT}$

as an oracle call
Hence $L \in P^{SAT}$
or, co-NP $\subseteq P^{SAT} =$ NP
Now in the last assignment we have proved that if co-NP $\subseteq$ NP then coNP=NP

e.
if NP $\subseteq$ BPP then, clearly, $SAT \in$ BPP.
$\implies$ $\exists$ randomized poly-time algorithm A for SAT
so that $\Pr[A(\phi) \neq SAT(\phi)] \leq 2^{-|\phi|}$

here we define an algorithm A' for : On input ϕ:
(i) otherwise, simulate A on $\phi|_{x_1=0}$ if accepts then fix $x_1 = 0$ and go to x_2.
if rejects then, fix $x_1 = 1$ and go to x_2.
(ii) in general simulate A on $\phi|_{x_i=0}$ (after setting values of $x_1, \ldots, x_{i-1}$) if accepts then fix $x_i = 0$ and go to x_{i+1}.
if rejects then, fix $x_i = 1$ and go to x_{i+1}.
(iii) say, b_i is the assignment for x_i from this above procedure. then check
if $\phi(b_1, b_2, \ldots, b_n) = 0$ then reject, else accept.

Here by the construction our algorithm if $\phi \notin SAT$ then, either $A(\phi) = 0$ at some stage of the above algorithm or, at the end A' rejects it.
$\implies$ if $\phi \notin SAT$ then, $A'(\phi) = 0$.
if $\phi \in SAT$ then, error probability of A at each iteration is $< 2^{-n}$ where $n = |\phi|$.
so, total error probability of A' is $< n2^{-n}$
(as, error at each iteration is independent.)
Now, $\frac{n}{2^n} \leq \frac{1}{2} \implies \Pr[A'(\phi) = 1] \geq \frac{1}{2}$ if $\phi \in SAT$
and if $\phi \notin SAT$ then, $\Pr[A'(\phi) = 1] = 0$
so, $SAT \in$ RP $\implies$ NP $\subseteq$ RP $\implies$ NP = RP.

f.
say, $L \in$ BPP $\implies$ $\exists$ PTM M so that
if $x \in \{0,1\}^n$ then, $\Pr_r[M(x) \neq L(x)] \leq 2^{-(n+2)}$
where $r \in \{0,1\}^m$ are the random bits.
Say, $x \in \{0,1\}^n$ and $Bad_x = \{r \in \{0,1\}^m : M(x,r) \neq L(x)\}$
so, $\left|\bigcup_{x\in\{0,1\}^n} Bad_x\right| \leq \sum_{x\in\{0,1\}^n} |Bad_x| \leq 2^n \frac{2^m}{2^{n+2}} < 2^m$.
so, $\exists r_n \in \{0,1\}^m$ so that $\forall x \in \{0,1\}^n$, $M(x, r_n) = L(x)$
as we know if we fix the random bits in M then it will behave like a poly time DTM.
so, if $x \in \{0,1\}^*$ and $|x| = n$ then, design the circuit C_n by fixing that good random bit r_n in M then, $C_n(x) = L(x)$ for $x \in \{0,1\}^n$.
so, $L \in P/poly \implies$ BPP $\subseteq P/poly$.

Question 5.

a.

say, we allow to visit the certificate tape back and forth.
and the machine allows $O(\log|x|)$ space $\implies$ M takes time at most $2^{O(\log|x|)} = O(|x|^c)$ time.
and for each step it can visit at most the whole certificate tape.
so, each of the steps take time $O(p(|x|))$ time.
as, length of the certificate is $O(p(|x|))$.
hence total time taken by the TM is $O(p(|x|)|x|^c)$ time.
hence, M takes poly-time and $x \in L \iff M(x,u) = 1$ where $u \in \{0,1\}^{p(|x|)}$.
so, it is the definition of NP.

b.
$A = \{\langle G\rangle | G \text{ is strongly connected}\}$.
given a graph G on n vertices, then convert G to G' in following procedure.
$\forall v \in V(G) \setminus \{s,t\}$ add edges $t \to v$ and $v \to s$.

Claim. G' is strongly connected $\iff \langle G,s,t\rangle \in PATH$

say, G' is strongly connected $\Rightarrow \exists$ a path $s \rightsquigarrow t$ is G'
and the path cannot use any edge $e \in E(G') \setminus E(G)$ by construction of G'.
so, $\langle G,s,t\rangle \in PATH$
again, say $\langle G,s,t\rangle \in PATH$ then take any $u,v \in V(G')$
then $u \to s, t \to v \in E(G')$ and $\exists$ a path $s \rightsquigarrow t \in G'$
hence $\exists$ a path $u \rightsquigarrow v$ in G'.
so our claim is true.

given $\langle G,s,t\rangle$ then convert G to G'.
and check if $\langle G'\rangle \in A$
hence it is a reduction from $PATH$ to A and by our construction of G' from G' it can be done by $O(\log|\langle G,s,t\rangle|)$ space $\implies A$ is NL-hard.

given G of n vertices and $V(G) = \{1,2,\ldots,n\}$
check whether $p_i = \langle G,i,i+1\rangle$ is in $PATH$
if true then go to p_{i+1} using the same space by removing p_i
if $\forall i \in [n], p_i \in PATH$ then, $\langle G\rangle \in A$ else reject. [take vertex $n+1$ = vertex 1]
as, $PATH \in NL \implies A \in NL \implies A$ is NL-complete.

Question 6.

We know there exists a binary linear size full-adder
Now using this we can find the value of $\sum x_i$
Now we will write it bit wise and then we will compare it with binary representation of $\frac{n}{2}$
That thing can be done bit-wise lexicographically
if number of bits for $\sum x_i >$ number of bits for $\frac{n}{2}$ we can simply say the output is

1
if number of bits for $\sum x_i <$ number of bits for $\frac{n}{2}$ we can simply say the output is 0
other wise we will compare bit by bit from the left
To compare each bit we can use binary comparator

Question 7.

1.
definition of $P/poly$:
$L \in P/poly$ if $\exists$ a circuit family $\{C_n\}_{n\in\mathbb{N}}$ so that
(i) $\exists$polynomial $p(n)$ s.t. for each n, $|C_n| = O(p(n))$
(ii) $\forall x \in \Sigma^*$ of length n $x \in L \implies C_n(x) = 1$ and $x \notin L \implies C_n(x) = 0$

Now, say $L \in P/poly$ then $\exists$ circuit family $\{C_n\}$ of polynomial size for L
say, $|C_n| = O(n^k)$ $\forall n$. Take, $\{\alpha_n\}_n = \{\langle C_n\rangle\}_n$ so clearly $|\alpha_n| = O(n^k)$
Define poly-time TM M so that:
On input $\langle x, \alpha_{|x|}\rangle$, simulate $C_{|x|}$ on x
if accepts then accept, if rejects then reject.
clearly M will run in time $O(n + n^k)$ (poly-time).
so, L agrees with the given advice string definition.

again say L agrees with the given advice string definition.
then, $\exists\{\alpha_n\}_n$ and poly-time DTM M so that $|\alpha_n| = O(n^k)$ $M(x, \alpha_n) = 1 \iff x \in L$
$M \in P \implies M \in P/poly$ and say $\{C_n\}_n$ be the poly-size circuit family for M.
C_n has $O(n + n^k)$ number of input gates (n for x and n^k for its advice)
construct $\{C'_n\}_n$ by fixing α_n in C_n for all n
so, in C'_n only n number input gates and it is poly-size circuit family as $\{C_n\}$ is.
$\{C'_n\}$ is a poly-size circuit family for $L \implies L \in P/poly$.

so, two definitions are equivalent.

2.
given $S_1, S_2, \ldots, S_k$ are sparse sets.
and $\forall i \in [k]$, $|S_i^{=n}| = O(p_i(n))$ for some polynomial p_i.
Now, consider $S = \{\langle x, i\rangle | i \in [k], x \in S_i\}$

Claim. S is a sparse set.

$|\langle x, i\rangle| = |x| + \log i$
hence clearly, $|S^{=n}| \leq \sum_{i=1}^{k} p_i(n - \log i) \leq \sum_{i=1}^{k} p_i(n - c_i) = \sum_{i=1}^{k} q_i(n) = P(n)$
we know $\forall$polynomial $p_i(n - \log i)$, $\exists$ $c_i \in \mathbb{N}$ so that $p_i(n - \log i) \leq p_i(n - c_i)$
and $p_i(n - c_i) \geq 0$ $\forall n \geq 0$
so, $\exists P(n)$ so that $|S^{=n}| \leq P(n) \implies$ our claim is true.

Now, define M so that on input s:
(i) check if $s = \langle x, i\rangle$ for some $i \in [k]$ (if not then reject)

(ii) call oracle S and check if $s \in S$
(iii) if true then accept, else reject.

3. say $L \in \mathrm{P} \implies \exists$ poly-time DTM M for L.
here we define an algorithm A for α_n:
on input n, return y
if input is taken in unary(resp. binary) then, y is unary(resp. binary) form of n
clearly $|y| = O(|n|)$ and A runs in poly-time.

so, for $x \in \Sigma^*$, $\alpha_{|x|} = A(|x|)$

Now, define M' : on input $x, \alpha_{|x|}$:
(i) simulate M on x.
(ii) if accept then accept, else reject.
this can be done in poly-time.
So, $\exists$ poly-time algorithm A for finding the advice string α_n and $\exists$ poly-time DTM M so that $x \in L \iff M(x, \alpha_{|x|}) = 1 \implies P \subseteq P/poly_{det}$

again, say, $L \in P/poly_{det}$
here we describe an DTM M_0 for L:
On input x:
(i) find in poly-time $A(|x|)$ (advice string for x)
(ii) run $M(x, A(|x|))$ as, $\exists$ poly-time DTM M so that $x \in L \iff M(x, \alpha_{|x|}) = 1$
(iii) if accepts then accept otherwise reject.
clearly M_0 will run in poly-time.
$\implies L \in P \implies P = P/poly_{det}$

4. $L \in P/poly \implies \exists$ circuit family $\{C_n\}_{n \in \mathbb{N}}$ so that $|C_n| = O(n^d)$
and $x \in L \iff C_{|x|}(x) = 1$. Say, $b_{n1} \dots b_{n,cn^d} = \langle C_n \rangle$ here c is a constant.
take $S = \{\langle 1^n, i, b_{n1} b_{n2} \dots b_{ni} \rangle : i \in [cn^d],\ n \in \mathbb{N}\}$

Claim. S is a sparse set.

proof: say $n \in \mathbb{N}$ then, if $k > n$ then, clearly, $|\langle 1^k, i, b_{k1} b_{k2} \dots b_{ki} \rangle| > n$
moreover for each n, S contains cn^d number of strings.
so, $|S^{=n}| \leq c(1^d + 2^d + \dots + n^d) \leq cn^{d+1}$ for all $n \in \mathbb{N}$.
so, our claim is true.

Claim. $\exists$ poly-time DTM M for L with oracle access S.

define M in following way: On input x:
(i) say, $|x| = n$.
then we find C_n from S step by step.
(ii) check which one of $\langle 1^n, 1, 0 \rangle$ and $\langle 1^n, 1, 1 \rangle$ is in S.
by construction, both simultaneously cannot be in S.

say, $\langle 1^n, 1, 0\rangle \in S$ then find which one of $\langle 1^n, 2, 01\rangle$, $\langle 1^n, 2, 00\rangle$ is in S
(iii) in general, if we say, $\langle 1^n, i, b_{n1} \dots b_{ni}\rangle \in S$ then, find $\langle 1^n, i+1, b_{n1} \dots b_{n,i+1}\rangle$ from oracle S.
(iv) by this procedure in $O(n^d)$ time find $\langle C_n\rangle$.
(v) simulate C_n on x, if $C_n(x) = 1$ then accept else reject.

clearly M runs in $O(n^d)$ time.
and M is a DTM for L with oracle access S.