24/30

Sandipan Dey HW-5

Proof
- The machine takes its input 1^n. (use a separate read only tape)
- Scan the input tape to count #1s in binary and write on work tape (e.g. start with 0 on work tape, with every single 1 read by moving head on the input tape, add 1 to the binary number stored on work tape). this takes $O(n)$ space for scanning the input and $O(\log_2 n)$ space to write it in binary on the work tape.
- count the # bits in $(n)_2$ from the work tape, i.e. compute $\lfloor \log_2 n \rfloor$ and write on the work tape (in binary). Again $O(\log_2 n)$ scaning of the work tape and $O(\log_2 \log_2 n)$ writing on the work tape which now contains $n \# \lfloor \log_2 n \rfloor$. Hence this step is computed using $O(\log_2 n) + O(\log_2 \log_2 n) = O(\log_2 n)$ space.
  (n in binary; $\lfloor \log_2 n \rfloor$ in binary)
- Finally read the binary representations of the two numbers n and $\lfloor \log_2 n \rfloor$ from the work tape and use standard multiplication algorithm to write the result on the work tape. Since the result can be at most $O(\log_2 n + \log_2 \log_2 n)$ bits long and the work tape now contains $n \# \lfloor \log_2 n \rfloor \# n * \lfloor \log_2 n \rfloor$, total space needed for this step $= O(\log_2 n) + O(\log_2 \log_2 n) + O(\log_2 (n) * \log_2 \log_2 n) = O(\log_2 n)$ space.
  (n in binary; $\lfloor \log_2 n \rfloor$ in binary; $n * \lfloor \log_2 n \rfloor$ in binary)

Copy $n * \lfloor \log_2 n \rfloor$ onto the beginning of the work tape, erase other symbols written and halt.

Hence work tape now contains $f(n) = n \log_2 n$ and it required $O(n) + O(\log_2 n) + O(\log_2 n)$ space $= O(n) = O(n \log_2 n)$ spa

2.

Since $\lim_{n\to\infty} \frac{n^3}{n^5} = 0$, $n^3 = o(n^5)$.

By space hierarchy theorem, we know that $\exists$ a language A which is decidable in ~~$\in$~~ $SPACE[O(f(n))]$ but not in $SPACE[o(f(n))]$, by any space constructible function $f(n)$.

Let $f(n) = n^5$, then $f(n)$ is space constructible ($n^5 \geq \log n$, we need to multiply n 5 times, which is $O(n)$ hence $O(n^5)$),

$O(f(n)) = O(n^5) = n^5$ and $o(f(n)) = o(n^5) = n^3$.

For space constructible function n^5, $\exists$ a language A ~~which is decidable~~ $A \in SPACE[n^5]$ ~~but~~ but $A \notin SPACE[n^3]$.

But we need to construct an infinite A_{inf} such that all infinite subsets $\forall B_{inf} \subseteq A_{inf}$, $A_{inf} \in SPACE[n^5] \Rightarrow B_{inf} \notin SPACE[n^3]$ (to prove)

The following $SPACE[n^5]$ algorithm ALGO decides a language (A_{inf}) that is immune to $SPACE[n^3]$

M ALGO = "On input w:

1. Let n be the length of w.
2. Compute n^5 using space constructibility and mark off this much tape. If later stages ever attempt to use more, reject.
3. If w is not of the form $\langle M \rangle 10^*$ for some TM M, reject.
4. ~~Simulate M on w while counting # steps used in simulation. If count ever exceed 2^{n^3}, reject.~~

/* ALGO is a decider, must decide when M runs forever. M deciding any subset $B \subseteq A_{inf}$ can run at most 2^{n^3} if not forever. Also, if M decides $B_{inf} \subseteq A_{inf}$ in SPACE, ALGO must ~~do the opposite as~~ behave differently */

5. ~~If M accepts, reject. If M rejects,~~

/* L(M) can be $\subseteq A_{inf}$ */

Simulate M on w count # steps used in simulation.

If count ever exceeds 2^{n^3}, reject. ← bigger M might have a big alphabet.

/* ALGO decides A_{inf}, hence must halt. M deciding any subset in $SPACE[n^3]$ ($B \subseteq A_{inf}$) must halt in 2^{n^3} steps, o/w it loops */

5. If M accepts, reject. If M rejects, accept."

The problem is that you will only try to spite M on inputs that look like $\langle M\rangle 10^n$. M might know its own index and reject all such inputs. I can construct M_1 & M_2 s.t. $L(M_1) = \overline{L(M_2)}$ and M_1 rejects all strings of the form $\langle M_1\rangle 10^n$ and accepts all other strings. Then $L(M_2)$ will be an infinite subset of your $L(A_{inf})$

$L(M_{ALGO}) = A_{inf}$ and no steps run forever ~~hence~~ also uses at most $O(n^5)$ space, hence M_{ALGO} decides A_{inf} in $SPAE[n^5]$. i.e.,

$A_{inf} \in SPACE[n^5]$.

Now, let's assume to the contrary that $\exists$ an TM M s.t. $L(M) = B_{inf} \subseteq A_{inf}$ ($w \in L(M) = B_{inf} \Rightarrow w \in A_{inf}$) and M decides B_{inf} (infinite subset of A_{inf}) in $SPACE[n^3]$, i.e., $B_{inf} \in SPACE[n^3]$.

By construction of M_{ALGO}, step 4 can simulate M in dn^3 space and ~~can complete $dn^3 \leq n^5$~~ $\exists n_0 > 0 \mid dn^3 \leq n^5 \; \forall n \geq n_0$, i.e., using input $w = \langle M\rangle 10^{n_0}$, step 4 must complete (since uses at most n^5 space)

So step 5 will be executed. But, ~~$w \in L(M)$~~ M accepts $w \Rightarrow$ $w \in L(M) \Rightarrow M_{ALGO}$ rejects $w \Rightarrow w \notin A_{inf}$ (since M_{ALGO} decides A_{inf} by assumption)

But, by assumption, $w \in L(M) = B_{inf} \Rightarrow w \in A_{inf}$, hence a contradiction

On the other hand, if M rejects $w \Rightarrow w \notin L(M) = B_{inf} \subseteq A_{inf}$ $\Rightarrow w \in B_{fin} \vee w \in \overline{A_{inf}}$. But, ~~then~~ M_{ALGO} accepts $w \Rightarrow$ $w \in A_{inf}$. ~~Hence only possibility is $w \in B_{fin} \subseteq A_{inf}$, i.e.,~~

~~Also by~~ Hence the only possibility is $w \in B_{fin} \subseteq A_{inf}$.

Hence, $B_{inf} \notin SPACE[n^3]$ when $B_{inf} \subseteq A_{inf}$ and $A_{inf} \in SPACE[n^5]$

Your proof does not consider what M does on input that starts with $\langle M'\rangle 10^n$ when $M \neq M'$

3. To prove: Any Turing recognizable set A can be written as

$A = C/D = \{x \in \Sigma^* \mid w = xy$ for some $w \in C \wedge y \in D\}$

for some C and D in SPACE[log n]

7

Proof: ~~Let~~ $L[M_C] = C$, $L[M_D] = D$, ~~where M_C & M_D are~~ TMs that decide C & D resp., in SPACE[log n]

Set A is T-R ⇒ ∃ TM M_A that recognizes A.

$M_A(x)$

~~if~~ writes string z on its tape and halts.

ow, loops forever.

Let $z_m = \{\arg\max |z| \mid M_A$ on input x writes z on the output tape$\}$

~~Construct TM M_D that counts the # 1's of input on its input tape and outputs~~

Now, TM M_D : starts with input $y = 1^d$ on its input tape and writes y in binary (using $\lfloor \log d \rfloor$ bits) on its output tape and halts.

$\overbrace{x\ 11\cdots 11}^{w}$ (the 1's labelled y)

M_C : Starts with input $w = xy$ on its input tape and writes w in binary using $\lfloor \log w \rfloor$ bits on its output tape and halts.

Clearly both C, D are Space[log n].

Now, by condition, $\lfloor \log w \rfloor \geq \log \lfloor z_m + \lfloor \log d \rfloor \rfloor$

$\lfloor \log(x+d) \rfloor \geq \log \lfloor z_m + \lfloor \log d \rfloor \rfloor$

$\Rightarrow \quad x + d \geq z_m + \log d$

$\Rightarrow \quad d \geq (z_m - x) + \log d \Rightarrow d - \log d \geq z_m - x$

Can always Choose such a d and use C, D to recognize A

What is z? and why can you assume that M_A behaves this way?

I don't understand your construction. You need to define C & D s.t. A = C/D and show that C & D are in SPACE[log n]. I don't see what C & D languages are.