Homework 11

Joe Baker, Brett Schreiber, Brian Knotten

February 6, 2018

16

Initial Approach: Construct a TM U which takes an input x, runs $T(x)$, and then $S(T(x))$ and returns the result. U correctly accepts x if and only if $x \in A$, however the space needed to store the temporary result $T(x)$ as input to S could be more than $log(|x|)$ in space.

New Approach: Instead modify T to produce output for S on-demand. That is, have U use a modified T' which takes a string x and int i and only writes the i^{th} symbol of x to the output tape. Adding a counter to T to track this space only requires an additional $log(|x|)$ space. Next, have U use a modified S' which runs $T'(x, i)$ when it needs the i^{th} symbol.
The new approach will have a TM U which runs T' and S' which are both $log(|x|)$ in space, so U is $log(|x|)$ in space while also accepting A.

17

a

Let $C = \left\{ \left(M, I, 2^{|I|^k} \right) \mid \text{ TM } M \text{ accepts } I \text{ in at most } 2^{|I|^k} \text{ space } \right\}$.
First we will show that $C \in EXPSPACE$:
Construct a TM T that takes in the tuple $(M, I, 2^{I^k})$, reads I, and runs M on I with a tape of size $2^{|I|^k}$.
T accepts if M accepts using less than $2^{|I|^k}$ space and rejects otherwise.
Because the TM M uses exponential space, T must also have exponential space available and therefore $C \in EXPSPACE$.

Next, we will show that $\forall C' \in EXPSPACE$, $C' \leq_{poly} C$.
Let $C' \in EXPSPACE$.
Since $C' \in EXPSPACE$, $\exists$ TM M and $k \in \mathbb{Z}$ such that $M(x)$ accepts in $2^{|I|^k}$ space iff $x \in C'$

b

The problem with the previous proof was that the space requirements for an arbitrary < M > could grow faster than 2^n. Consider a TM T such that $L(T) = C$. Since C passes in an arbitrary base c, T needs to account for that difference in growth rate. But this c is negligible, because c^{n^k} can be expressed as $2^{n^{k+\epsilon}}$ for some ϵ. Since the number of steps can be written as $2^{n^{x+\epsilon}}$, and since T can handle any 2^{n^k}, the base does not matter.