CS 172 , Spring, 1999
Professor M. Blum

Problem #1

a) Define the number of steps taken by a NDTM on input x.
b) Define the number of steps taken by a NDTM on inputs of length of n.

a) # NDTMi [x] =
case1: min(y belongs to (summation *)) { # DTMi[y,x]} (this is the deterministic half of the NDTMi)

if there exists y such that DTMi[y,x] accepts (i.e. enters an accepting state)

case2: 1 otherwise

b) # NDTMi(n) = Max{# NDTMi[x]}, |x| = n

Problem #2

Define two (computational) problems p1, p2 to be poly-time equivalent iff it is possible to solve p1 in polynomial time given an algorithm to solve p2 in polynomial time (p1 <= p2), and vice-versa (p2 <= p1).

Are the following two problems poly-time equivalent?
If so, prove it.
If not, explain why not.

Decision:
Instance: NDTMi, x in {0,1}*, m in unary
........m
(ie 1.....1 = 1).
Question: Does NDTMi accept x in m steps? ie does there exist a y in {0,1}* s.t. DTMi accepts (y,x) in m steps, if any
(ie if such y exists);
"NONE" if there is no such y.

1) YES! <u>Decision <= Optimization:</u>

- If optimization program returns y, then the Decision program returns YES
- If optimization program returns "NONE", then the DEcision program returns NO.

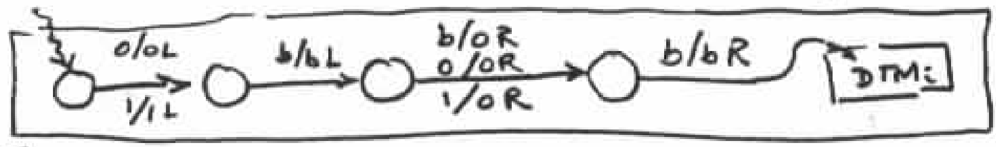
The running time of the Decision algorithm = running time of the Optimizaton algorithm + O(1).

2) <u>Optimization <= Decision:</u>

- If Decision algor returns No, then the Opt. algor returns NONE.
- If Dec. algor returns YES, then the Opt. algor must find y. It can do this by finding the bits of y = Yk Y(k-1) ... Yo one at a time starting say with Yo.

To see if there is a solution with $y_0 = 0$, replace

by



This DTM$|_{y_0=0}$ overwrites the guessed symbol y_0 by 0.

If the Dec. algor rejects [NDTMi|(Yo = 0), x,m], then we know that Yo = 1. Else we know that it's ok to let Yo = 0.
In general, having determined Y(i-1) = A(i-1),..., Yo = Ao, one can determine Yi by augmenting the NDTMi to NDTMi|oA(i-1)...Ao, which overwrites the guessed Yi Y(i-1) ... Yo with oA(i-1)...Ao.
If the NDTMi|oA(i-1)...Ao rejects, then we know that Ai = 1.
Else ok to let Ao = 0. The size of the augmented DTMi|Ai...Ao is just |DTMi| + O(m), which is poly in (|NDTMi| + |x| + m).

Problem #3

Explain what problems if any you encounter in doing the above reductions
In the case that m is given in binary instead of unary.

- Decision <= Optimization as before, but
- Optimization (NOT <=) Decision:
  The reason is that the required output y (y = 2^x) man be terribly long, length |y| = x, for inputs of length n = |NDTMi| + |x| + lgm (m = poly(|y|)
  = |NDTMi| + |x| + O(|x|)
  = poly(|x|).
  The decision algorithm can take just poly(n) steps, because it knows the existence of y without having to exhibit it. But the optimization algorithm must exhibit (print) y.
