CSCC63 TUT 0002

Tutorial 3

Definitions
Coming up with Complements
Using Enumerations
Sample Reduction Proofs

Recognizable: A language L is recognizable if there exists a recognizer R (TM) for which L(R) = L
Co-recognizable: A language L is corecognizable if there exists a recognizer R-c (TM) for which L(R-c) = L-c

Prove that L is recognizable:
1. Write a recognizer
2. A reduces B and B is recognizable, then A is recognizable
3. Certificate

= { M | M accepts some string that ends with 101 }

= { M | exists x in Sigma*, such that M accepts x + 101 }

- c = { M | forall x in Sigma*, such that M doesn't accept x + 101 }

X1 = { M | M accepts some string that ends with 101 }
X1 = { M | M(x) returns true for some x ending in 101 }

```
let w_i be enumerations over {0, 1}*, w_0 = e, w_1 = 0, w_2 = 1, w_3 = 00 ..
R on input <M>:
  for i = 0 to infinity:
     for k = 0 to i:
        run M on w_i for k steps
        accept if M accept if w_i ends in 101
```

if M is in X1 then eventually M will accept some w_i within k steps
if M is not in X1 then R will loop on M forever, never accepting

X1 - c = { M | M rejects/loops on all strings that end with 101 }
HALT = { <M, w> | M halts on w }
HALT-c = { <M, w> | M loops on w }

HALT reduces to X1, HALT is not co-recognizable => X1 is not co-recognizable
HALT-c reduces to X1-c, HALT-c is not recognizable => X1-c is not recognizable => X1 is not co-recognizable

Consider the following reduction from HALT-c to X-c
P on input <M, w>:
  define M' on input <x>:
    1. [pass]
    2. run M on w
    3. [accept if x ends with 101]
  return M'

WTP <M, w> in HALT-C iff <M'> in X1 - c
1: Show that if <M, w> in HALT-c then P(<M, w>) = M' rejects/loops on strings ending with 101
  Assume that <M, w> in HALT-c (this means that M will loop on w)
  If M loops on w, then line 3 will never run, M' will accept nothing => it rejects/loops on all strings that end in 101

2: Show that if <M, w> in HALT then P(<M, w>) = M' accepts some string that ends with 101
  Assume that <M, w> in HALT, line 2 will finish executing, line 3 will run and thus accept all strings that end with 101.
  if M' accepts all strings that end with 101 => it accepts some strings thus M' in X1

Suppose that L is recognizable, show that L reduces to HALT

HALT = { <M, w> | M halts on w }

If L is recognizable, then L has a recognizer R_L, L = { y | some property of y holds }
P on input <y>:
define M' on input x:
1. run R_L on x
2. if R_L accepts, accept otherwise loop

return <M', y>

WTP L is recognizable iff <M', y> is in halt
1. Show that L is recognizable, and y in L => <M', y> is in halt
Assume that L is recognizable, exists recognizer R_L
R_L will accept x if and only if x in L, M' halts if and only if y/x belongs to L

2. Assume that y not in L then R_L will reject/loop on y, then M' will loop on y as well.

Question 4 Clarification:
L4 = { <M, N> | |L(M)-C| < |L(N)| }
whether the size of the set of all M rejects/loops is less than the size of the set of all that N accepts.

Question 6 Clarification:
L6 = { <M> | M must accepts x if x = 1*, M can accept anything else }

Question 7 Clarification:
Definition: M^L TM is a TM that has access to L oracle
Show that for any fixed language A, no M^A can solve { <M^A, w> | M^A halts on w }
Consider what happens if such M^A exists, what if you used it to try and come up with a liar's paradox?