CSci 423 Homework 8

Due: 12:30 pm, Thursday, 11/14/2019
Daniel Quiroga

Collaborators: Ethan Young, Will Elliot, Yang Zhang, Kevin Li

1. $E_{CFG} = \{< G > \mid G \text{ is a CFG and } L(G) = \emptyset\}$.
   We first convert G_L to a CNF of the language E_{CFG} with input $< G >$, and w represents the string of nonterminals $\in G_L$, (A, B, C...).
   Let the terminals in G_L be a TM that takes in $< M_T >$.
   Then G_M will be the TM that takes in w:
   (a) It first runs $< M_{Ta} >$ on $A \in G_L$.
   (b) If it accepts, then it will move onto the next terminal
   (c) If $< M_{Ta} >$ rejects, then G_M rejects.
   (d) It will loop until all terminals in G_L are accepted and return and empty string.
   (e) Then G_M accepts. Essentially if the language is just empty then we have a Turing machine for the specified language.

   Since we have constructed a Turing machine that can decide E_{CFG}, so L is a decidable language.

2. $L = \{< M > \mid \text{TM } M \text{ accepts at least one string in no more than 9 steps}\}$
   M* decides languages that follow the input $< M >$. Gets the length of $| < M > |$ (at most 9) and stores the value.
   Next, run the inputs length up to 9 / up to 9 steps as well. $\rightarrow$ accepts if M accepts at least one of the strings within 9 steps.
   Since the inputs are a finite length, the machine will eventually halt and thus is decidable.

(Hint for (b): What is the maximum number of tape squares can a TM scan in no more than 9 steps?)

(5, 5 points) Prove the following closure properties of TRLs.

1. If L_1 and L_2 are Turing-recognizable, so is L_1L_2.
   For decidable languages L_1 and L_2, let M_1 and M_2 recognize them, respectively. Design M that recognizes L_1L_2
   TM M = on input w
   For every way to split $w = w_1w_2$
   $\rightarrow$ Run M_1 on w_1 M_2 on w_2
   $\rightarrow$ If both accpet, accept
   $\rightarrow$ else: continue with the next iteration of the w_1w_2 combo
   If none are accepted after loop, reject

   Since there is a turing machine that recognizes L_1L_2 , then L_1L_2 is turing-recognizable

2. If L_1 and L_2 are Turing-recognizable, so is $L_1 \cap L_2$
   For recognizable languages L_1 and L_2, let M_1 and M_2 recognize them, respectively. Design M that recognizes $L_1 \cap L_2$
   TM M = on input w

   run M_1 on w
   If M_1 accepts:
   $\rightarrow$ Run M_2 on w
   $\rightarrow$ If M_2 accepts, accept w
   $\rightarrow$ else: reject w
   else: reject w

   Since there is a turing machine that recognizes $L_1 \cap L_2$, then $L_1 \cap L_2$ is turing-recognizable

(6 points) Let B be the set of all **infinite** binary strings over $\{0,1\}$. Show that B is uncountable.

If B only contains binary strings, we know that each position will have either a 1 or 0. Therefore we write out all of the strings in B via Table format.

B_0	*111111...1...
B_1	1*01111...1...
B_2	10*0111...1...
....	....
B_x	000000...*0...
....	....

By combining all of the * in the string, using diagonalization we can create a string composed of all the stars in their corresponding position and change the 0's to 1's and 1's to 0's. This would give a string that is not in B, therefore we can say that B is uncountable!

(6 points) Let $L = \{w_{2i}, \forall i = 1,2,3,\dots \mid w_{2i} \notin L(M_i)\}$. Prove by contradiction that L is non-TR.
Proof: Assume A_D is TR.
$\exists$ TM M that accepts A_D i.e. $L(M) = A_D = L(M_{2i})$
M = M_{2i} for some i
$w_{2i} \notin A_D$ iff $w_{2i} \in L(M_{2i})$ (by def of A_D)
$w_{2i} \notin A_D$ iff $w_{2i} \notin L(M_{2i})$ (by $L(M_{2i}) = A_D$)