COMP330 Assignment 6

Ryan Sowa, ID: 260886668

1. Rigorously establish the decidability or undecidability of the following languages:

   (a) (20 points)

   $$L = \{0^n \mid \text{the decimal expansion of } \pi = 3.14... \text{ contains } n \text{ (or more) consecutive 0's}\}.$$

   Proof. Suppose the number of consecutive 0's in π is bounded by some positive integer, m. Then, if $n > m$, reject (we cannot have any more consecutive zeros than m). Else, accept.

   Next, consider if the number of consecutive 0's in π is not bounded by some number (they go on infinitely). Then, the decimal expansion of π contains n consecutive $0's$ no matter what n is. Therefore, accept.

   Thus, L is decidable.

   □

   (b) (20 points)

   $$L = \{\langle D\rangle \mid D \text{ is a DFA and } L(D) \text{ is not recognized by a DFA with fewer states}\}.$$

   Proof. On input $\langle D\rangle$ to L, we count the number of states in D (finite) and call this number n. For $i = 0, 1, ..., n-1$, we build all possible Turing Machines with i states (finite set). Since we know (from class) that $L = \{\langle A, B\rangle \mid A \text{ and } B \text{ are DFA's and } L(A) = L(B)\}$ is decidable, we can compare the language of each of these Turing Machines with $L(D)$. If one (or more) of these Turing Machines has the same language as D, then accept. Else, reject.

   Thus, L must be decidable.

   □

2. (30 points) Prove that the following language is undecidable by giving a reduction from the Post Correspondence Problem.

   $$L = \{\langle G\rangle \mid G \text{ is a CFG and } L(G) \text{ contains at least one palindrome}\}.$$

Proof. Suppose by contradiction that L is decidable. Then, there exists a Turing Machine R which decides L.

On input $\langle P \rangle$ to PCP, we first count the number of dominoes, n. For every domino, we make an individual rule in our CFG as follows:

$$S \to N_{ir} S D_i$$

, where N_{ir} is the reversed numerator of the i'th domino and D is the denominator of the i'th domino, for $1 \leq i \leq n$. We also make the rule

$$S \to \varepsilon$$

so that our grammar will terminate.

Run the CFG outlined above on R. If R accepts, then accept (numerator could match denominator). If R rejects, then reject (numerator could not match denominator).

Since if L is decidable, we can decide PCP, L must be undecidable. □

Example (from class):

On input $\langle \frac{b}{ca}, \frac{a}{ab}, \frac{ca}{a}, \frac{abc}{c} \rangle$ to PCP, construct the following CFG:

$$S \to bSca \mid aSab \mid acSa \mid cbaSc$$

We can then build a palindrome as follows:

$$S \to cbaSc \to cbaaSabc \to \ cbaaacSaabc \to cbaaacbScaaabc$$

$$\to cbaaacbaSabcaaabc \to cbaaacbaabcaaabc$$

Since we could build a palindrome, we can match the numerator and denominators of our dominoes in a corresponding manner (accept in PCP).

3. (30 points) Let A be any language in P over the alphabet $\{0, 1\}$. Prove that

$$L_A = \{1^n \mid n \in \mathbb{N}, A \cap \{0,1\}^n \neq \emptyset\}$$

is in NP.

Proof. We use a verifier on $\langle A, y \rangle$, where y is the set of strings we need to verify. For $i = 0, 1, ..., m$ ($O(m)$ time), where m is the length of the longest string in A, we check to see if A contains a string of length i. Since $A \in P$, we know that checking if a string is in A will take polynomial time. For each string of length $i \in A$, if 1^i is in y ($O(1)$ time), then accept. Else, reject.

Since this computation takes $(O(m)$ * polynomial time * $O(1)) =$ polynomial time , we can conclude that this is an efficient verifier for L_A and therefore, $L_A \in NP$.

$\square$