ECE 606, Fall 2019, Assignment 9

Zhijie Wang, Student ID number: 20856733

zhijie.wang@uwaterloo.ca

November 12, 2019

1. *Proof.* Suppose each state remember the past 4 bits in order to make decision on where the fourth from last bit is '0'. If there is a DFA which has states fewer than 16, then, there must be some state, said state q, where both $a_1a_2a_3a_4$ and $b_1b_2b_3b_4$ lead to this state. $(a_i, b_i \in \{0, 1\})$

   Since $a_1a_2a_3a_4 \neq b_1b_2b_3b_4$, then they must differ in at least one bit, say $a_i \neq b_i$, here we assume $a_i = 1, b_i = 0$.

   If $i = 1$, then, since $1a_2a_3a_4$ has the fourth from last bit equals to 0, then state q must be acceptable, which makes contradiction because $0b_2b_3b_4$ is unacceptable.

   If $i = 2$, then, we have $a_11a_3a_4$ and $b_10b_2b_3b_4$ at state q. Suppose we have input 0 to pass state q to state p, then, $1a_3a_40$ is acceptable, which makes contradiction again because $0b_3b_40$ is unacceptable.

   If $i = 3, 4$, suppose we input 00 and 000, then, it will make the same contradiction as before.

   Therefore, no DFA of fewer than 16 states exists for the following language: bit-strings of length at least 4, whose fourth from last bit is '0'.

2. Suppose we have a solution consists of an $n^2 \times n^2$ matrix, therefore, for each row, column and square, it takes $\Theta(n^2)$ times to check if every element is distinct. We have n^2 rows, n^2 columns and n^2 squares, and also k slots which have been filled before, so, it takes $3n^2\Theta(n^2) + \Theta(1) = \Theta(n^4)$ times. If n is binary-encoding, suppose $n = 2^s$, then it takes $\Theta(2^{4s})$ times to check a solution.

   Therefore, this problem is $\notin$ **NP**, but $\in$ **NEXP**.

3. Suppose $G =< V', E' >$, $H =< V, E >$.

   ```
   S ← {}
   for each i from 1 to |V'| do
       Non-deterministically pick a vertex v_i in V\S
       f(v'_i) = v_i
       S ∪ {v_i}
   for each i from 1 to |E'| do
       e'_i = (a', b')
       if e_i(f(a'), f(b')) is not exist then
           return False
   return True
   ```

   Here we have a brief discussion. For the first for-loop, we build a 1-1 f from vertexes in G to some vertexes in H, which is done with non-deterministically pick vertexes from H. Therefore, we check for each edge $e' = (a', b')$ in G, whether H also has a edge $e = (f(a'), f(b'))$. If all the edges exist, then there is a subgraph of H which is isomorphic to G.

   The first for-loop runs $\Theta(|V'|)$ times, and computing $V \backslash S$ takes $\Theta(|V|)$ times. The second for-loop runs $\Theta(|E'|)$ times for the worst case. Therefore, the algorithm runs $\Theta(|V'||V| + |E'|)$ for the worst case.

4. a9p4.py