Homework 8

Huimin He , section 1

March 6, 2015

1. 8.1 (a)

   For each x_i, there is 10 possibilities, so $|C| = 10^n$.

   (c1)

   Denote x to be the number that satisfies M. $x(i)$ to be the i^{th} digit of x.

   Input: n, the number of digits Output: a set C_m with all numbers satisfying M with n digits.

   Pseudocode:

   Construction of configuration space.

```
$x = 0, C_m = \emptyset$ let x have all the preceding zeros
while ($x < 10^n$)
    $i$ = search()
    if $i \neq 0$
        $x(i-1) += 1$ changes the carrying step
        for $j$ from $i$ to $n$
            $x(j) = x(i-1) + 1$
    add $x$ to $C_m$
return $C_m$
```

   search (search for the first digit that is 9)

```
for $i$ from 1 to $n$
    if $x(i) == 9$
        return $i$
    else
        return 0
```

   Justification of this algorithm: We want the number to be monotone. The only problem is when incrementing a number x, the carry reduces the digit that reached 9. We want the addition to avoid the carry problem. So the algorithm above changes the carry step to avoid decrease on any digit. Because no digit is decreasing, the result is monotone.

   To find a configuration that satisfy predicate T in C_m, we have to loop through the configuration space which has size $\binom{n+9}{n}$.

   c2

   The number of element in C_m is a. To construct C_m .

2. 8.2 transitivity of karp reduction
   It is easy to showthat if $L_1 \preccurlyeq L_2 \preccurlyeq L_3$ then $L_1 \preccurlyeq L_3$.
   proof: Knowing $(\forall x \in \Sigma_1^*)(x \in L_1 \iff f(x) \in L_2)$
   $(\forall x \in \Sigma_2^*)(x \in L_2 \iff g(x) \in L_3)$

   NTW $(\forall x \in \Sigma_1^*)(x \in L_1 \iff f(x) \in L_3)$

   we have
   $$g(f(x)) : \Sigma_1^* \to \Sigma_3^*$$
   for all $x \in \Sigma_1^*$, we have
   $$x \in L_1 \iff f(x) \in L_2 \iff g(f(x)) \in L_3$$
   The above two statements define the karp reduction from L_1 to L_3.

   The exponent of the reduction of L_1 to L_3 is cd. In other words, the Karp reduction f can be computed in time $O(n^{cd})$.

3. 8.4
   3-colorability of graphs and RSA decryption. Breaking RSA is not a decision problem as 3-colorability is. However, the question that does integer n have a factor smaller than k is a decision problem. We can use binary search to find the factor of n in polynomial time by asking the decision question mentioned

4. 8.7 Clay mathematics institue carries the prize the correctio solution for the P versus NP problem. If there is a proof that RSA can not be broken in polynomial time, then the proof that integer factorization can not be solved also must be provided. Since the 3-colorability problem is more or equally hard than the integer factorization problem (decision version)from HW8.6. It can be concluded that 3-colorability can not be solved in polynomial time. Since 3-colorability problem is NP-complete, then NP $\neq$ P is proved.

5. 8.8 First run the bellmanford through the graph G with an origin s. Detect any negative cycle that is accessible from s. Second, delete all vertices that are accessible from s. Denote this as a new graph G'. Third, run G' using bellmanford again and detect any negative cycle in this new graph G'. Then repeat step 2 and 3 until all vertices are deleted.
   Running time should be $O(|V||E|)$ because the number of vertices visited is equal to the number of vertices deleted at termination.

6. 8.1 (b) extra credit

   This problem can be thought of unordered sampling with replacement since the monotonicity removes the ordering from part (a). After sampling with replacement from 0 9, we remove the sampling order and fill each x_i using the monoticity rule. The total number of ways to do this is $\binom{n+k-1}{k}$ (from probability thoery knowledge). In this problem $k = n$ and $n = 10$. So the unordered sampling has $\binom{n+9}{n}$ ways to do the unordered sampling.