12. [20th Feb]

    (i) In the proof of Thereom 5.9 on pg 223 of [Sip], assuming input TM M is an NTM, give details of Step (1) to simulate M on w while counting the number of IDs encountered along each branch of computation. You may detail the automata as well.

11. [19th Feb]

    (i) Devise an algorithm to implement Step (2) in the proof of Theorem 4.7 on pg 198 of [Sip]. (a solution could be provided in tomorrow's class)

    (ii) Given an undirected unweighted (finite) graph $G(V, E)$ in adjacency list form and a vertex $s \in V$, provide details of a TM M that does a breadth-first traversal of G, starting at s. The output tape of M should consists of all the vertices of G that belong to the same component as s.

    (iii) Given a directed unweighted (finite) graph $G(V, E)$ in adjacency matrix form and two vertices $s, t \in V$, provide details of a TM M that traverses G in depth-first order to find whether there is a path from s to t in G.

10. [18th Feb]

    (i) Construct a DTM that takes an encoding $< N >$ of an NFA N as input and outputs an encoding $< D >$ of DFA D corresponding to N such that $L(D) = L(N)$.

    (ii) Construct a DTM that takes an encoding $< R >$ of a regular expression R as input and outputs an NFA N corresponding to R such that $L(N) = L(R)$.

    (iii) Give the details of an UTM M_1 that takes an encoding $< P >$ of a DPDA P and $w \in \Sigma^*$ as input and simulates P on w such that M_1 always halts and $< P, w > \in L(M_1)$ iff P accepts w.

9. [13th Feb]

    (i) Following Theorem 9.4 on pg 222 of [HU], give an unrestricted grammar G so that $L(G) = L(M_2)$, where the DTM M_2 is as given on page 172 of [Sip]. And, show that $0000 \in L(G)$.

    (ii) For the grammar G in Example 9.4 on pg 220 of [HU], following the proof of Theorem 9.3 on pg 221 of [HU], give a two-tape NTM N with all the details (including the automata) so that $L(N) = L(G)$. Also, by giving a sequence of IDs, show that $aaaa \in L(N)$.

    (iii) Give a detailed description of a TM, including the automata, to non-deterministically print two positive unary integers, separated by a $\#$. Each branch of its computation needs to print a unique tuple of integers, and printing the tuple $200, 300$ causes graceful exit.

8. [12th Feb]

    (i) While following the construction on pg 162 of [HU] Fig. 7.9, give detailed descriptions' of four-track Turing machines which would simulate the steps of Turing machines that you gave for Problem (i) of 11th Feb.

(ii) Devise a DTM that prints positive integers in sorted order on one of its tapes, with every two successive numbers separated by a #.

(iii) Consider the NFA N in [HU] pg 20 Fig. 2.5. Give an NTM T such that $L(N) = L(T)$. Based on T's transition function, draw the computation tree of T. Identify at least one node in T that accepts 01001. List the IDs at nodes of this tree that are encountered in simultating T's computation with a DTM.

(iv) Supposing there is a node of maximum depth d in the computation tree T of an NTM N which accepts the input $w \in \Sigma^* \cap L(N)$, upper bound the total number of transitions executed in the DTM D that simulates N, wherein D is constructed as described in the proof of Theorem 7.3 on pg 164 of [HU].

7. [11th Feb]

(i) Give two-tape Turing machines to accept the following languages: (i) $\{ww \,|\, w \in \{0,1\}^*\}$, and (ii) $\{ww^R \,|\, w \in \{0,1\}^*\}$.

(ii) Design a Turing machine to add two floating point numbers. You may assume any standard representation for the floating point numbers.

(iii) Provide the corresponding transitions for a standard DTM (semi-infinite tape with L and R options for the tape head) so that it can simulate each step of a single-tape two-way infinite two-track DTM that has the stay put option as well.

6. [6th Feb]

(i) Considering the PDA P' constructed in class by dovetailing a DFA D with a PDA P to achieve $L(D) \cap L(P) = L(P')$, given any $w \in \Sigma^*$, prove by induction on i, $(\delta_D(d_0, w) = d$ and $(p_0, w, \gamma_1) \vdash_P^i (p, \epsilon, \gamma))$ implies $([d_0, p_0], w, \gamma_1) \vdash_{P'}^i ([d, p], \epsilon, \gamma)$.

5. [5th Feb]

(i) Using pumping lemma for context-free languages, show that $\{a^i b^j a^i b^j \,|\, i \geq 1, j \geq 1\}$ is not context-free.

(ii) Prove $L = \{a^i b^j c^k d^\ell \,|$ either $i = 0$ or $j = k = \ell\}$ is not context-free.

Considering $b^j c^k d^\ell \in L$, show the converse of pumping lemma for context-free languages not necessarily holds.

(iii) For $L = \{w \,|\, w \in \{a,b\}^*$ is not a palindrome$\}$, considering $a^p b^p \in L$, show the converse of pumping lemma for regular languages not necessarily holds. [Hint: As argued in a video lecture, since $\overline{L}$ is shown to be non-regular, L cannot be regular.]

4. [4th Feb]

(i) [Sip] pg 157 exer 2.30 $(a), (b)$, and (c).

3. [30th Jan]

(i) [Sip] pg 88 exer 1.29 $(a), (b)$.

(ii) With the algorithm in the proof of Theorem 2.4 of [HU], give a regular expressions corresponding to DFAs in [Sip] pg 86 exer 1.21.

(iii) With the algorithm in the proof of Theorem 2.3 of [HU], construct NFAs to accept languages of regular expression on [Sip] pg 86 exer 1.20 $(e), (h)$.

2. [29th Jan]

   (i) [Sip] pg 88 exer 1.31.

   (ii) Let N_1 and N_2 be as defined in the below problem. Using the construction given in showing the intersection operation is closed for regular languages, give a DFA for $L(N_1) \cap L(N_2)$.

   (iii) Let N_1 (resp. N_2) be the NFA in [Sip]: 86 exer $1.16(a)$ (resp. $1.16(b)$). Using the construction from the below problem, construct DFAs D_1 and D_2 such that $L(D_1) = L(N_1)$ and $L(D_2) = L(N_2)$.

   (iv) Given an NFA $N(Q, \Sigma, \delta, q_0, F)$, by combining proofs of Theorems 2.1 and 2.2 in [HU], provide a direct proof showing there is a DFA $D(Q'(\subseteq 2^Q), \Sigma, \delta', \epsilon\text{-}closure(q_0), F')$ with $L(D) = L(N)$. Here, $\forall_{S \in Q'} \delta'(S, \alpha) = \{q \in Q \,|\, q \in \epsilon\text{-}closure(\delta(q_s, \alpha))$ for some $q_s \in S$ and $\alpha \in \Sigma\}$, and $F' = \{S \in Q' \,|\, S$ contains an accept state of $N\}$.

1. [28th Jan]

   (i) Given an NFA N, determine whether the language of NFA obtained by swapping the accepts states of N to reject states and vice versa is $\overline{L(N)}$.

   (ii) Let N_1 (resp. N_2) be the NFA in [Sip]: 86 exer $1.16(a)$ (resp. $1.16(b)$). Using the construction given in today's class (proofs of Theorems 2.1 and 2.2 in [HU]), construct DFAs D_1 and D_2 such that $L(D_1) = L(N_1)$ and $L(D_2) = L(N_2)$. [Hint: For N_2, in the first phase, construct an NFA with no ϵ-transitions.]