THC Homework #1

Let $\Sigma = \{0, 1\}$ for all exercises.

1. Give the state diagram of a DFA recognizing the following language:

   $L_1 = \{w \mid$ the number of 1s in w is divisible by 2 and the number of 0s is divisible by $3\}$.

2. Give the state diagram of a NFA (or DFA) recognizing the following language:

   $L_2 = \{w \mid w$ contains two 1s such that the number of 0s between them is divisible by $4\}$.

   (There may be some other 1s between the two 1s.)

3. For any word $w = w_1 w_2 \ldots w_n$, the reverse of w, written w^R, is the word in reverse order, $w_n w_{n-1} \ldots w_1$. For a language A, let $A^R = \{w^R \mid w \in A\}$. Show that if A is regular, so is A^R.

4. Prove that the following language is regular:

$$L_3 = \{w \mid \text{if } w \text{ is viewed as a binary number, then it is divisible by } 3\}.$$

THC Homework #2

1. Prove that any DFA that accepts the language $(0 \cup 1)^*0(0 \cup 1)^{n-1}$ has at least 2^n states. *(Hint: Show that if it has less states, then it cannot distinguish two words, where one of the words should be accepted and the other should be rejected.)*
2. Write a regular expression for the following language over the alphabet $\{0, 1\}$: The set of all strings with at most one pair of consecutive 0's and at most one pair of consecutive 1's.
3. Is the following language regular: The set of binary strings that are not palindromes. (w is a palindrome if $w = w^R$ where w^R denotes the reverse of w.)
4. Prove that every language with nonempty complement is contained in a regular language with nonempty complement.

THC Homework #3

1. Give a pushdown automaton recognizing the following language:

$$\{w \mid \text{the length of } w \text{ is odd and the middle symbol is a } 0\}$$

($\Sigma = \{0, 1\}$)

2. Prove the following language is context-free:

$$\{x\#y \mid x, y \in \{a, b\}^* \text{ and } x \neq y\}$$

($\Sigma = \{a, b, \#\}$)

3. Prove that the following language is not context-free:

$$\{x_1\#x_2\# \ldots \#x_k \mid k \geq 2, \text{ each } x_i \in \{a, b\}^*, \text{ and for some } i \text{ and } j \neq i, x_i = x_j\}$$

($\Sigma = \{a, b, \#\}$)

4. Give a language which is context-free, but its complement is not context-free.

THC Homework #4

1. Let M be the following 2-tape Turing Machine: $M = (Q, \Sigma, \Gamma, \delta, q_0, F)$, where $k = 2$, $Q = \{q_0, q_1, q_2, q_3, q_4, q_5\}$, $\Sigma = \{0, 1\}$, $\Gamma = \{X, 0, 1, *\}$, $F = \{q_5\}$,

δ :

	reading		writing				
state	1st tape	2nd tape	1st tape		2nd tape		new state
q_0	0	*	0	S	X	R	q_1
	1	*	1	S	X	R	q_1
	*	*	*	S	*	S	q_5
q_1	0	*	0	R	0	R	q_1
	1	*	1	R	1	R	q_1
	*	*	*	S	*	L	q_2
q_2	*	0	*	S	0	L	q_2
	*	1	*	S	1	L	q_2
	*	X	*	L	X	R	q_3
q_3	0	0	0	S	0	R	q_4
	1	1	1	S	1	R	q_4
q_4	0	0	0	L	0	S	q_3
	0	1	0	L	1	S	q_3
	1	0	1	L	0	S	q_3
	1	1	1	L	1	S	q_3
	0	*	0	S	*	S	q_5
	1	*	1	S	*	S	q_5

(If δ in not defined, and the current state is not in F, then we reject.) What is L_M? (Which 0–1 sequences are accepted by M?)

2. Give the state diagram of a 1-tape TM which recognizes $L = \{w\#w^R \mid w \in \{0,1\}^*\}$, where w^R is the reverse of w.

3. Describe a TM in words (you don't need to give all the state diagram, just a precise description), which computes the following function: $x\#y \to x + y$, where x, y are considered as (multiple bit) binary numbers, and the output is the sum in binary form.

4. A Turing machine without a left move is like an ordinary TM except, that it can never move its head to the left (but the head can stay, or move to the right). Show that this variant is not equivalent to the original model. What class of languages do these machines recognize?

THC Homework #5

1. Prove that the language of all prime numbers (say in binary representation) is recursive (Turing-decidable).
2. Define the notion of a Turing machine with a two-dimensional tape. Show that a 2-tape ordinary Turing machine can simulate it. Estimate the efficiency of the above simulation. (Express the running time of the new machine with the running time function $T_M(n)$ of the original machine.)
3. Prove that if L_1, L_2 are recursive languages than $L_1 \oplus L_2 := (L_1 \setminus L_2) \cup (L_2 \setminus L_1)$ (the symmetric difference of the two languages) is also recursive.
4. Show that any infinite recursively enumerable language contains an infinite recursive language.

THC Homework #6

1. Let L be the language of all words $w \in \Sigma^*$ for which M_w (the TM described by w) exists and there exists a word $s \in \Sigma^*$ such that M_w stops in at most 100 steps started with input s. Is L recursive?

2. Show that the following problem is undecidable (the corresponding language is $\notin \mathcal{R}$): Given two Turing machines, M_1 and M_2, is the intersection of the languages accepted by M_1 and M_2 empty?

3. Show that the following language is not recursively enumerable:

$$L = \{w \mid M_w \text{ exists and it accepts a word } 0s \text{ iff it accepts } 1s\}$$

4. Show that L is Turing-decideable if and only if $L \leq_m 0^*1^*$.

THC Homework #7

1. In the Silly Post Correspondence Problem, SPCP, in each pair the top string has the same length as the bottom string. Is the SPCP decidable?
2. In In the Binary Post Correspondence Problem, BPCP, the size of the alphabet is 2 (i. e. $|\Sigma| = 2$). Is the BPCP decidable?
3. Define the notion of the 3 dimensional domino problem and prove that it is undecidable.
4. Show that the following modification of the domino problem is also undecidable. We use tiles marked on the corners instead of the sides and all tiles meeting in a corner must have the same mark.

THC Homework #8

1. Let

   $$\text{MODEXP} = \{(a, b, c, p) \mid a, b, c \text{ and } p \text{ are binary integers such that } a^b \equiv c \mod p\}.$$

   Show that MODEXP $\in \mathbf{P}$.

2. Show that $\mathbf{P}$ is closed under the star operation. (*Hint: Use dynamic programming*)

3. Prove that if $L \in$ SPACE$(3 \log n)$ then $L \in \mathbf{P}$.

4. Let TRIPLE $= \{xxx \mid x \in \Sigma^*\}$. Prove that TRIPLE $\in$ SPACE$(\log(n))$.

THC Homework #9

1. Let f be a length-preserving one-to-one function over binary strings, and suppose that f is computable in polynomial time. Let

   $$L = \{y \mid \exists x(f(x) = y \text{ and the first bit of } x \text{ is } 1)\}.$$

   Prove that $L \in \mathbf{NP} \cap \mathbf{coNP}$. *(Note that we cannot assume that the inverse of f is computable in polynomial time.)*

2. Suppose we have a polynomial algorithm to determine the size of the largest clique in a given graph. Describe a polynomial algorithm *to find* a maximum size clique.

3. Show that an algorithm that makes at most a constant number of calls to polynomial-time subroutines runs in polynomial time, but that a polynomial number of calls to polynomial-time subroutines may result in an exponential-time algorithm.

4. Show that, if $\mathbf{P} = \mathbf{NP}$ then every language $A \in \mathbf{P}$ except $A = \emptyset$ and $A = \Sigma^*$ is $\mathbf{NP}$-complete. *(Hint: The proof is not good enough if it doesn't show why do we have the two exceptional languages.)*

THC Sample Midterm

1. Give an NFA, that accepts the following language: The set of stings over the alphabet $\{0, 1, \ldots, 9\}$ such that the final digit has not appeared before.

2. Let $\Sigma = \{0, 1, +, =\}$ and

   $$ADD = \{x = y + z \mid x, y, z \text{ are binary integers, and } x \text{ is the sum of } y \text{ and } z\}.$$

   Shaw that ADD is not regular.

3. Give a context-free grammar for the following language:

   $$\{a^n b^m c^k \mid k > n + m; n, m \geq 0\}.$$

   ($\Sigma = \{a, b, c\}$).

4. Consider the language $L = \{s \mid s \text{ contains twice as many 0s as 1s}\}$ over the alphabet $\{0, 1\}$. Design a TM (with full details) that recognises L.

5. A one-tape, 1-superhead Turing machine M_s is an obvious extension of a one-tape machine. The superhead can read and write the k cells right from its actual position in one step, then jump at most k positions left or right.

   Prove that such a machine can be simulated by an ordinary one-tape, one-head Turing machine and estimate the efficiency of the simulation. (i.e. If M_s makes $T_{M_s}(n)$ steps on an input of length n, how many steps the simulating machine needs to make?)

6. Show that any infinite recursively enumerable language contains an infinite recursive language.

THC Sample Final

Solve 1. and 2. with closed books, closed notes. When you hand the solutions for these in, you can open your book and notes to solve the other questions.

1. Define the set of recursive and recursively enumerable languages. What is the relation of the two sets? Give a language which is contained in one of the sets but not in the other.
2. Prove that if $L_1 \leq_P L_2$ and $L_2 \leq_P L_3$ then $L_1 \leq_P L_3$. ($\leq_P$ denotes the polynomial (Karp)-reduction. You can use previously proved theorems without proof here.)
3. Let $\Sigma = \{1, 2, 3, 4\}$ and $C = \{w \in \Sigma^* \mid$ in w, the number of 1s equals the number of 2s, and the number of 3s equal the number of 4s$\}$. Show that C is not a context free language.
4. Show that the class **NP** is closed under concatenation.
5. Let A be the language of properly nested parentheses. For example, $(())$ and $(()(()))()$ are in A, but $)($ is not. Show that $A \in DSPACE(log(n))$.
6. What is the complexity of the following language? *(Is it in* $\mathbf{P}$*? Is it* $\mathbf{NP}$*-complete?)*

   $$L_4 = \{(G, k) \mid G \text{ contains a spanning tree with maximum degree} \leq k \}$$

   (k is a number given in binary form.)

List of theorems that may appear in the final

Instructor: Gyula Y. Katona

1. NFA and DFA has equal computing power. (The idea, without the precise details.)
2. Regular languages are closed under regular operations.
3. If a language is decribed by a regular expression, then it is regular.
4. Pumping lemma for regular languages. (The idea, without the precise details.)
5. Pumping lemma for CFL. (The idea, without the precise details.)
6. One-way infinite 1-tape TM is equivalent to two-way infinite 1-tape TM.
7. A language is $\mathcal{RE}$ iff an enumerator enumerates all the words.
8. A language is $\mathcal{R}$ iff an enumerator enumerates all the words in inreasing order.
9. The diagonal language is not $\mathcal{RE}$.
10. $A_{\text{TM}} \in \mathcal{RE} \setminus \mathcal{R}$
11. $L \in \mathcal{R}$ iff $L \in \mathcal{RE}$ and $\overline{L} \in \mathcal{RE}$
12. $HALT_{\text{TM}} \in \mathcal{RE} \setminus \mathcal{R}$
13. Witness theorem.
14. Karp-reduction is transitive.
15. 3-SAT is NP-complete. (Assuming that SAT is NP-complete.)
16. MAXSTABLE is NP-complete. (Assuming that 3-COLOR is NP-complete.)
17. MAXCLIQUE is NP-complete. (Assuming that MAXSTABLE is NP-complete.)
18. HAMCYCLE is NP-complete. (Assuming that HAMPATH is NP-complete.)