Name:

CS 341
Practice Final Exam

1 a	20	
b	20	
c	20	
d	20	
e	20	
f	20	
g	20	
2	10	
3	30	
4	12	
5	15	
Total	207	

1. Please write neatly. You will lose points if we cannot figure out what you are saying.

2. Whenever you answer a question with a machine or a grammar, add comments to help us figure out what you are trying to do.

3. DO NOT WRITE JUNK and hope for partial credit. If you are trying to prove something and you know that you haven't got it right, say so. That may get you partial credit. But if you give the impression that you think that junk is nonjunk, you won't get points.

(1) For each of the following languages:

(i.) **(4 points)** Determine the appropriate category, as listed below.

(ii.) **(8 points)** Prove that the language is in the category you have specified, except as noted below, in which case these points will be assigned to part iii. You can do this part by writing an appropriate grammar (or regular expression), exhibiting an appropriate recognizing machine, using closure properties, or by showing that the language is finite. What I mean by "exhibit a recognizing machine" is:

- For an FSM: Draw it.
- For a PDA: Draw it, but if it is very complicated, also provide a description of how it works. We will give partial credit for the description even if the details of the machine are missing or wrong.
- For a Turing machine: Describe an algorithm in clear English. It need not be specifically a Turing Machine program. Any clear algorithm is acceptable.

(iii.) **(8 points)** Prove that the language is not in the next most restricted category, except as noted below, in which case these points will be assigned to part ii.

Assume the following language categories:

A. *L* is regular. (In this case, you can skip part iii.)
B. *L* is not regular but is context free
C. *L* is not context free but is decidable (In this case, if *L* is a set of strings that include descriptions of machines (FSMs, PDAs or TMs), you do not need to prove that it is not context free.)
D. *L* is not in D but is in SD.
E. *L* is not in SD. (In this case, you can of course skip part ii.)

You may take as theorems that the following languages are not decidable:

H = {<*M*, *w*> : TM *M* halts on input string *w*}
H_ε = {<*M*> : TM *M* halts on the empty tape}
H_{ANY} = {<*M*> : there is any string on which TM *M* halts}
The corresponding languages A, A_ε, and A_{ANY}.

You may take as theorems that the following languages are not semidecidable:

¬H = {<*M*, *w*> : TM *M* does not halt on input string *w*}
$H_{\neg ANY}$ = {<*M*> : there does not exist any string on which *M* halts}
H_{ALL} = {<*M*> : TM *M* halts on all inputs}
EqTMs = {<M_a, M_b> : $L(M_a) = L(M_b)$}
The corresponding languages ¬A, $A_{\neg ANY}$ and A_{ALL}.

You may not use Rice's Theorem for these problems.

To get full credit for any reduction proof, you must justify, in detail, why the reduction is correct. So purely copying and pasting a standard reduction will not get full credit even if it happens to work.

Remember that:

- If *S* is a set, |*S*| is the cardinality of *S*.
- If *w* is a string, |*w*| is the length of *w*.
- $\#_a(w)$ is the number of a's in the string *w*.

(a) $L = \{<M, w> : M$ moves right exactly twice while operating on $w\}$

(b) $L = \{wx : |w| = 2\cdot|x| \wedge w \in \mathtt{a}^+\mathtt{b}^+ \wedge x \in \mathtt{a}^+\mathtt{b}^+\}$

(c) $L = \{<M_a, M_b> :$ if M_a does not accept ε then M_b rejects $\varepsilon\}$

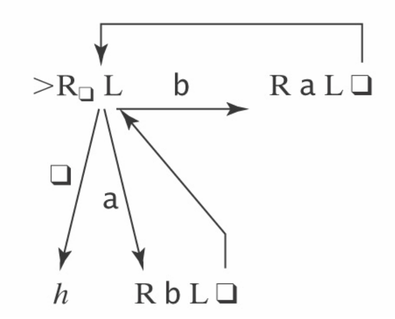
(d) $L = \{<M> : M$ rejects at least two even length strings$\}$

(e) $L = \{w = xyx^R : x \in \{\mathtt{0}, \mathtt{1}\}^+, y \in \{\mathtt{0}, \mathtt{1}\}^*\}$

(f) $L = \{<M> : L(M)$ is not regular$\}$

(g) $L = \{wxw : |w| = 2\cdot|x|, w \in \{\mathtt{a}, \mathtt{b}\}^*, x \in \{\mathtt{c}\}^*\}$

(2) Consider the following Turing Machine *M*, described in our macro language:



Give a short English description of what *M* does. Don't describe how it operates. Describe its result.

(3) Give an example that proves each of the following:

(a) It is possible, given two languages L_1 and L_2, both in SD/D, that $L_1 \cap L_2$ is in D.

(b) It is possible that, if L_1 and L_2 are context free, $L_1 \cap L_2$ is not context free.

(c) The SD languages are not closed under complement.

(d) Let L_A be the set of all languages defined over some alphabet A. Then let $md: L_A \rightarrow L_A$ be a function that maps from one language to another, defined as follows:

$$md(L) = \{x : \exists w \in L\ (w = sxt, s \in \Sigma^*, x \in \Sigma^*, t \in \Sigma^*)\}$$

If L is context free but not regular, $md(L)$ may be regular.

(e) It is possible that, if $L_1 \leq L_2$ and $L_2 \in$ SD then $L_1 \in$ D.

(f) There exists a language L that is Turing enumerable but not lexicographically Turing enumerable.

(4) Let M be an arbitrary Turing machine. Suppose that $timereq(M) = 3n^3(n+5)(n-4)$. Circle all of the following statements that are true:

a) $timereq(M) \in \mathcal{O}(n)$.

b) $timereq(M) \in \mathcal{O}(n^6)$.

c) $timereq(M) \in \mathcal{O}(n^5/50)$.

d) $timereq(M) \in \Theta(n^6)$.

(5) Show that SUBSET-SUM = {<*S*, *k*> : *S* is a multiset (i.e., duplicates are allowed) of integers, *k* is an integer, and there exists some subset of *S* whose elements sum to *k*} is in NP.