CSE396, Spring 2019 **Final Exam** **May 14, 2019**

One notes binder allowed, no Internet, closed neighbors, 170 minutes. The exam has **seven** problems and totals 240 pts., subdivided as shown. You must do EXACTLY ONE of 7(A) or 7(B) and indicate clearly which one you are attempting. *Show your work*—this may help for partial credit. *Please write in the exam books only*—if you need more paper you may ask.

Notation and terms are as in the text, except that $\#c(x)$ stands for the number of times the character c occurs in x, the complement of a language A is denoted by $\tilde{A}$ or $\sim A$, difference of sets A and B by $A \setminus B$, and except for the following alternative terminology:

- *Turing-recognizable* language and *c.e. or r.e.* language are synonyms. The class of such languages is denoted by RE.
- *Decidable* language and *recursive* language are synonyms. The class of such languages is denoted by REC.
- Angle brackets $\langle M \rangle$ denote the encoding of a Turing machine M as a string over the alphabet ASCII, which can be further coded over $\Sigma = \{0, 1\}$. Fine details of such encodings are not important. A language using such an encoding *may be assumed not to be a* CFL or co-CFL—*unless it equals* $\emptyset$ or Σ^*. Similar remarks apply to the use of (e.g.) $\langle P, w \rangle$ to encode a program P and an input w as a single string.
- The notation $P(x)$ means "the computation of P on input x."

You may cite theorems and facts that were covered in lectures and/or text without further proof, so long as the cited item is clearly stated and your use of it is clear. This includes that the algorithms for E_{DFA}, ALL_{DFA}, EQ_{DFA}, E_{NFA}, A_{NFA}, E_{CFG}, and A_{CFG} run in polynomial time, as do conversions between CFGs and nondeterministic PDAs, complementing a deterministic PDA, and Cartesian product of a (nondeterministic) PDA N with a DFA M to yield $L(N) \cup L(M)$, $L(N) \cap L(M)$, or $L(N) \setminus L(M)$. You may refer to the following (not exhaustive!) list of languages and their classifications:

$\{ a^n b^n : n \geq 0 \}$	DCFL, but not regular
$\{ x \in \{ a, b \}^* : x = x^R \}$	CFL and co-CFL, but not a DCFL
$\{ a^n b^n c^n : n \geq 0 \}$	Co-CFL and in P, but not a CFL
$\{ a^m b^n a^m b^n : m, n \geq 0 \}$	Co-CFL and in P, but not a CFL
$\{ ww : w \in \{ a, b \}^* \}$	Co-CFL and in P, but not a CFL
$\{ \langle G, w \rangle : G \text{ is a CFG and } w \in L(G) \}$	A_{CFG}: In P but not a CFL (or co-CFL).
$\{ x\#y\#z : x = y \wedge y \neq z \}$	In P, but not CFL or co-CFL
$\{ \phi : (\exists a \in \{ 0, 1 \}^n)\ \phi(a) = 1 \}$	SAT: In NP, believed not in P
$\{ \phi : (\forall a \in \{ 0, 1 \}^n)\ \phi(a) = 1 \}$	TAUT: In $\mathsf{co\text{-}NP}$, not in NP unless $\mathsf{NP} = \mathsf{co\text{-}NP}$
$\{ \text{poly-time NTMs } N : \langle N \rangle \notin L(N) \}$	D_{NP}: Decidable but not in NP
$\{ \langle M, w \rangle : M \text{ is a TM and accepts } w \}$	A_{TM}: C.e. but not co-c.e.
$\{ \langle M \rangle : M \text{ is a TM and does not accept } \langle M \rangle \}$	D_{TM}: Co-c.e. but not c.e.
$\{ \langle M \rangle : \text{for all inputs } x,\ M(x) \text{ halts} \}$	TOT: Neither c.e. nor co-c.e.
$\{ \langle M \rangle : L(M) = \Sigma^* \}$	ALL_{TM}: Neither c.e. nor co-c.e.

(1) (50 pts.)

Classify each of the following languages $L_1, \ldots, L_{10}$ over $\Sigma = \{0, 1\}$ according to whether it is

(a) regular;

(b) a DCFL but not regular;

(c) a CFL but not a DCFL;

(d) A co-CFL but not a CFL;

(e) in P but not a CFL or co-CFL;

(f) decidable but either known not to be in P, or not in P unless $\mathsf{NP} = \mathsf{P}$.

(g) c.e. but not decidable;

(h) not c.e. (this includes neither c.e. nor co-c.e.)

You need not justify your answers, but *brief* justifications may help for partial credit–especially with some "close" answers. The languages are:

1. $L_1 = \{ \langle G \rangle : G$ is a context-free grammar and $\epsilon \notin L(G) \}$.
2. $L_2 = \{ \langle G \rangle : G$ is a context-free grammar and $L(G) \neq \Sigma^* \}$.
3. $L_3 = \{ \langle N \rangle : N$ is an NFA and $L(N) \neq \Sigma^* \}$.
4. $L_4 = \{ \langle N \rangle : N$ is an NFA and $L(N) \neq \emptyset \}$.
5. $L_5 = \{ x \in \{ a, b \}^* : \#a(x) - \#b(x)$ is a non-negative multiple of $3 \}$.
6. $L_6 = \{ x \in \{ a, b \}^* : \#a(x) - \#b(x)$ is a (possibly negative) multiple of $3 \}$.
7. $L_7 = \{ x \in \{ a, b \}^* : \#a(x) > \#b(x)$ and $x = x^R \}$.
8. $L_8 = \{ x \in \{ a, b \}^* : \#a(x) = \#b(x)$ or $x = x^R \}$.
9. $L_9 = \{$ Java programs P: on some input x, $P(x)$ outputs $777 \}$.
10. $L_{10} = \{$ Java programs P: for all inputs x, $P(x)$ outputs $777 \}$.

Please write your answers in this form: if L_{11} were the language of the Halting Problem, you could write "11. g" or "11. (g)" or (safest) "L_{11} is (g) c.e. but undecidable." Not every category need be used.

(2) $10 \times 5 = 50$ pts. *Multiple Choice*: Please write the one best answer clearly in the exam book. *No justifications are needed*, though they could help for partial credit.

1. In a Myhill-Nerode proof that the language $L = \{\, a^{3n}b^n : n \geq 0 \,\}$ is non-regular, the proof can begin with:
   (a) Take $S = a^*b^*$;
   (b) Take $S = (aaa)^*$;
   (c) Take $S = b^*$;
   (d) Take $S = (aaa \cup b)^*$.

2. Continuing the proof, when considering any pair $x, y \in S$, you can distinguish them by:
   (a) Take $z = b^{|x|/3}$;
   (b) Take $z = b^{|x|}$;
   (c) Take $z = b^{3|x|}$;
   (d) Take $z = a^{3|y|}$.

3. To prove a language B is NP-complete, after first showing $B \in \mathsf{NP}$, we could:
   (a) Show FACTORING $\leq_m^p B$.
   (b) Show SAT $\leq_m^p B$.
   (c) Show $A_{\mathsf{TM}} \leq_m^p B$;
   (d) Show $B \leq_m^p$ SAT.

4. The union of two non-regular DCFLs can possibly be:
   (a) Regular;
   (b) A non-regular DCFL;
   (c) A CFL that is not a DCFL;
   (d) Any of the above.

5. For every language A, the concatenation A^*A^* equals:
   (a) $(AA)^*$;
   (b) Σ^*;
   (c) A^*;
   (d) None of the above.

6. If M_1, M_2, and M_3 are DFAs with 100 states each and the same alphabet Σ, then $L(M_1) \cup L(M_2) \cup L(M_3)$ is:
   (a) Always equal to Σ^*;
   (b) Always recognized by a DFA with 300 states;
   (c) Always recognized by an NFA with 301 states;
   (d) Possibly non-regular.

7. In the CFG $G = S \longrightarrow 0S \mid 1S \mid \epsilon$:
   (a) The string 011 is ambiguous;
   (b) The string 010 is ambiguous;
   (c) The empty string ϵ is ambiguous;
   (d) No string in $L(G)$ is ambiguous.

8. The language $\{ \langle M, G \rangle : M$ is a DFA and G is a CFG and $L(M) = L(G) \}$ is (known to be):
   (a) In P;
   (b) NP-complete;
   (c) Empty, because $L(G)$ must be nonregular;
   (d) Undecidable.

9. In a proof that $\{a^i b^j c^k : i < j < k\}$ is not a CFL, upon being given a "pumping length" $N \geq 1$, you should start by taking:
   (a) $s = a^N b^N c^N$;
   (b) $s = a^N b^{N+1} c^{N+2}$;
   (c) $s = a^N b^{2N} c^{3N}$;
   (d) Any of the above—they all work.

10. The undecidability of the Halting Problem, noting also the Church-Turing thesis, means that:
    (a) Extraterrestrial civilizations may be able to build computers that can solve it, even though human beings cannot;
    (b) There is no program that solves every given instance of the Halting Problem with a yes/no answer;
    (c) Human beings should not even bother to try to solve any instances of the Halting Problem;
    (d) Turing machines are too weak a model of computation to solve it; random-access machines or quantum computers are needed to solve it.

(3) 5 × 3 = 15 pts. *True/false*. You must write out the word `true` *or* `false` *in full*, and justifications are not needed.

(a) *True/false?:* The intersection of a CFL and a DCFL is always a CFL.

(b) *True/false?:* The generally quickest way to tell if a given string x matches a given regular expression r is to convert r into an equivalent DFA M_r and then run $M_r(x)$.

(c) *True/false?:* The union of two regular languages is always a DCFL.

(d) *True/false?:* The problem of whether a Boolean formula $\phi(x_1, \ldots, x_n)$ has a truth assignment $a \in \{0,1\}^n$ that makes $\phi(a) = \texttt{false}$ is NP-complete.

(e) *True/false?:* For every nondeterministic PDA N there is a deterministic PDA M such that $L(M) = L(N)$.

(4) (6 + 18 + 6 + 6 = 36 pts.)

Let $\Sigma = \{a, b\}$. Let $E = \{x \in \Sigma^* : |x| \text{ is even and } |x| \geq 2\}$. Consider the context-free grammar $G = (\{S, Y\}, \Sigma, \mathcal{R}, S)$ with rules $\mathcal{R}$ given by

$$\begin{array}{lcl} S & \longrightarrow & aSa \mid bY \mid Yb \\ Y & \longrightarrow & YS \mid a \mid b \end{array}$$

(a) Is G ambiguous? If you say yes, find an ambiguous string and give two different parse trees for it. If you say no, prove it. (6 pts.)

(b) Prove by structural induction that $L(G) \subseteq E$. (Reasonable proof shortcuts are OK. 18 pts.)

(c) The grammar's designer intended $L(G) = E$, but there's a "bug." Find three different strings in $E \setminus L(G)$. (Or optionally, write a regular expression r that matches infinitely many strings in $E \setminus L(G)$. 6 pts.)

(d) Add one rule to create a context-free grammar G' such that $L(G') = E$. *You need not prove your answer*, but should give a short explanation. (6 pts., for 36 total)

(5) (18 + 6 + 6 + 5 = 35 pts.)

Design a DFA M for the language of the regular expression $r = (a+bb)^*(b+aa)^*$. You must either give comments that explain why your M is logically correct (comments that merely say what it does are insufficient), or first write an NFA N with 4 states such that $L(N) = L(r)$ and convert N into a DFA. Also answer the following questions:

(a) Find the shortest string x such that $x \notin L(r)$. Then find a string y such that $xy \in L(r)$.

(b) Is there a string x such that for all strings $y \in \Sigma^*$, $xy \notin L(r)$? If so, find a shortest such string. Else, explain why not using your M.

(c) Is there a string x' such that for all strings $y \in \Sigma^*$, $x'y \in L(r)$? If so, find a shortest such string. Else, explain why not using your M.

(6) (18 + 12 = 30 pts.)

Let $\Sigma = \{ a, b, c, d \}$, and let $A = \{ x \in \Sigma^* : \#a(x) = \#b(x) \wedge \#c(x) = \#d(x) \}$. Prove via the "CFL Pumping Lemma" that A is not a CFL.

Then describe in prose how you would create a 2-tape deterministic TM M such that $L(M) = A$ *and* M runs in $O(n)$ time. You may suppose the tapes (in particular, the second tape) are infinite in both directions and you may move the input-head tape left (including a full "rewind" to the first cell if you wish).

(7) (18 + 6 = 24 pts.)

Choose ONE of the following two alternatives. You must indicate clearly which one you are attempting. You are welcome to use and cite facts from lecture and assignments, including those on the exam front page.

(A) Consider the following decision problem:

INSTANCE: A deterministic Turing machine M with input alphabet $\Sigma = \{0, 1\}$.

QUESTION: Does every string accepted by M (if any) have a substring 00 somewhere in it?

Show that the language of this problem is undecidable (18 pts.), and for the full 24 pts., show that the language is not c.e. either. (You may combine both steps into one.)

XOR

(B) Consider the following decision problem:

INSTANCE: A context-free grammar G with terminal alphabet $\Sigma = \{0, 1\}$.

QUESTION: Does every string generated by G (if any) have a substring 00 somewhere in it?

Show that the language of this problem is decidable (18 pts.), and for the full 24 pts., show that it belongs to P. (You may combine both steps into one.)

END OF EXAM.