1. Every infinite *r.e.* set contains an infinite regular subset.

2. Suppose $\{L_1, L_2, ...\}$ is an infinite collection of regular languages. Then, $\cup_{i=1}^{\infty} L_i$ is a recursively enumerable language.

3. If some NP-complete language is context-free, then $P = NP$.

4. $A \leq_m B$ implies that $\bar{A} \leq_m \bar{B}$ for every pair of languages A and B. ($\leq_m$ denotes *many-one* reduction.)

5. There does not exist any algorithm for deciding whether a given TM M accepts a given string w within $|w|^2$ time.

6. It is decidable whether the union of two given r.e. sets is r.e.

7. The following problem is in P: 'Given a graph G, does it contain a set of 5 nodes such that all of these nodes are pairwise connected in G?'

8. It is decidable to determine, given a TM M and input string w, whether there is some state which M enters no more than 3 times during its computation on w.

9. The intersection of a finite collection of recursive sets is recursive.

10. $\{a^n b^n c^n : n \geq 1\}$ is in P.

C. (15 pts) Given a directed graph and two nodes x and y, y is said to be *reachable* from x, denoted by $x \rightsquigarrow y$, iff there exists a path from x to y. Let $G = (V, E)$ be a directed graph with an initial node v. We define a *reachable knot* (see Figure 1) to be a subgraph $G' = (V', E')$, $E' \neq \emptyset$, of G such that

1. $\exists$ a node $x \in V'$ such that $v \rightsquigarrow x$,

2. $\forall w \in V', \forall u \in V, w \rightsquigarrow u$ implies $u \in V'$, and

3. G' is strongly connected.

Show that the following problem is NLOGSPACE-hard (NLOGSPACE=NSPACE(log n)):
Instance: Given a directed graph $G = (V, E)$ and an initial node v.
Question: Does there exist a reachable knot? (Yes or no?)

(Hint. Recall that the graph reachability problem (i.e., the problem of deciding, given a directed graph G and two nodes x and y, whether $x \rightsquigarrow y$) is NLOGSPACE-complete.) Explain (briefly) why your reduction can be carried out in DSPACE(log n).

16. There exists a language which can be accepted by a deterministic 2-stack machine (2PDA) but not by any nondeterministic 1-stack machine (NPDA).

17. There exists a language $L \subseteq 0^*$ which is not recursive.

18. Given an NFA M, it is decidable whether M accepts $(ab)^{481}$.

19. Given a PDA M, it is decidable whether M accepts $(ab)^{481}$.

20. Given a deterministic one-tape TM M, it is decidable whether M accepts $(ab)^{481}$.

21. Given a TM M, it is decidable whether M runs for at least 2^{481} steps on input $(ab)^{481}$.

22. Given a TM M, it is decidable whether M will ever move its head left more than ten times on input $(ab)^{481}$.

23. Given a TM M, it is decidable whether M ever reenters its start state on any input.

24. Given CFGs G and G', it is decidable whether $L(G) \subseteq L(G')$.

25. It is decidable whether two given C programs compute the same function.

26. It is decidbale whether a given C program will ever get into an infinite loop on some input.

27. Rice's Theorem can be used to prove that A = $\{\rho(M) : L(M)$ is finite$\}$ is not recursive.

28. Rice's Theorem can be used to prove that B = $\{\rho(M) : L(M)$ is accepted by some Turing machine with either fewer than 10 or more than 20 states$\}$ is not recursive.

29. Rice's Theorem can be used to prove that C = $\{\rho(M)$: there is some input string w, such that the head of M visits infinitely many tape squares during the computation of M on input $w\}$.

30. There exists a total function which is not Turing-computable.

31. There exists a primitive recursive function which cannot be computed by any TM which always halts.

32. If $NLOGSPACE = P$, then it must be the case that $P = NP$.

33. The halting problem is EXPSPACE-hard.

34. The complement of a language in NP is recursive.

35. If there is a problem Q which is NP-complete and in co-NP, then NP=co-NP.

B. (50 pts) For each statement below, decide whether it is TRUE or FALSE. To receive the full credit (i.e., 5 points for each problem (TRUE/FALSE: 2 pts; justification: 3 pts)) you must provide a brief (no more than 5 lines) justification of your answer. No penalty for guessing.

Theory of Computation

Final Exam. (Open Book)
Spring' 96 (YEN)

Name: .....

Student I.D.: .....

A. (70 pts) For each statement below, decide whether it is true or false. (No proofs needed.) (Score=max{0, right-$\frac{1}{2}$wrong}) Mark 'O' for 'true,' and 'X' for 'false.' (Notice that we use $\rho(M)$ to denote the encoding of TM M.)

1. Suppose that the language of a TM M is not recursive. Then, the number of input strings on which M does not halt must be infinite.
2. The class of languages in P is closed under complementation.
3. The class of NP languages is closed under intersection with regular languages.
4. Let L denote the language of encoded 3-SAT formulas which are *not* satisfiable. If $P \neq NP$, then $L \notin P$.
5. Every infinite regular set contains a non-r.e. subset.
6. Nondeterministic Turing machines can accept non-r.e. sets.
7. For Turing machines M, L(M) is recursive if and only if M halts on all inputs.
8. There is a TM with five tapes that accepts a set that cannot be accepted by any TM with four tapes.
9. The family of r.e. sets is closed under intersection.
10. The set of theorems of Peano Arithmetic is recursively enumerable.
11. $\{\rho(M) : L(M) \in NP\}$ is decidable.
12. There is an effective procedure (TM which always halts) which, for every Turing machine M, produces a context-free grammar which generates all strings that are halting computations of M on the empty string.
13. PATH = {(G, a, b) : G is a graph containing vertices a and b such that there is a path from a to b in G} is NP-complete.
14. The set of unsatisfiable Boolean formulas in 3-CNF is $\leq_P^m$-complete for co-NP. (the set of languages whose complements are in NP).
15. $CLIQUE \leq_P^m 3 - SAT$.