CA313 Algorithms and Complexity
Spring 2008

Attempt **three** questions. All questions carry equal marks.

Q 1.
(i) Explain each term in the quadruple $\langle V_t, V_n, P, S \rangle$ by which a grammar for any language is defined.

(ii) Define the constraints on α and β in a rewrite rule $\alpha \longrightarrow \beta$ for the four classes of grammar in the Chomsky Hierarchy.

(iii) Assume the following ruleset:

S → NP, VP	PP → by NP
VP → V, NP, PP	NP → John
VP → V, NP	N → girl
NP → the N	N → park
NP → the N PP	V → saw

(a) State which type of grammar this is, and why;

(b) State which rules in the grammar could be rules in a less powerful grammar, and why;

(c) Give **both** trees (i.e. the grammar is ambiguous) that represent the structure of the sentence *John saw the girl by the park*. Note that in one reading of this sentence, *John* is 'by the park', while in the other, *the girl* is.

Q 2.

(i) Assume the following ruleset:

- $W \rightarrow xyz$
- $W \rightarrow xWYz$
- $zY \rightarrow Yz$
- $yY \rightarrow yy$

(a) State which type of grammar this is, and why;

(b) State which rules in the grammar could be rules in a less powerful grammar, and why;

(c) Show the derivation of the 2nd-shortest string in the specified language using trees;

(d) Show the derivation of the 3rd-shortest string in the specified language using string manipulation.

(ii) Give a formal definition of a Turing machine in terms of the 5-tuple $M = (Q, \Sigma, \Gamma, q_0, \delta)$.

(iii) How do deterministic Turing machines and non-deterministic Turing machines differ?

(iv) How are Turing machines useful in the definition of different classes of complexity problems?

Q 3.
(i) Define *time complexity*.

(ii) Polynomial solutions are usually considered efficient, while exponential solutions are considered inefficient. Give examples which show that this is not completely true.

(iii) Name the following complexities (e.g. $O(n^2)$ is 'quadratic'), and for each of them, say if they are usually considered efficient (e.g. quadratic complexities *are* considered efficient).

- $O(3)$:
- $O(n)$:
- $O(exp(n))$:
- $O(n^k)$ (k is a natural number):
- $O(ln(n))$:
- $O(n^3)$:
- $O(2^n)$:

(iv) Write the following functions using the O notation (e.g. $2n = O(n)$).

- $1000n^2$:
- $5ln(n) + 10$:
- $exp(n) + 3n^2$:
- $2n^3 + 500n^2$:
- $3n^2\ x\ ln(n)$:

Q 4.
(i) Define the following classes of complexities:

- P,
- NP,
- NP-hard,
- NP-complete.

(ii) What is a binary decision problem? Give **one** example of a binary decision problem.

(iii) Explain the Traveling Salesman Problem (TSP). Give four different applications of the TSP.

(iv) Explain in your own words why the TSP is actually in the complexity class NP.

(v) Describe **one** of the following algorithms that lead to approximate solutions for NP in reasonable time: Nearest neighbour **or** Genetic algorithms **or** Simulated Annealing.

(vi) Describe the problem of *local maxima* in hill climbing algorithms. How might these be overcome?

Q 5.
(i) What is an NP-intermediate problem? Give **one** example of an NP-intermediate problem.

(ii) Define the subset-sum problem (SSP). Explain the complementary problem of the SSP.

(iii) Define *space complexity.*

(iv) What is the difference between the two classes $DSPACE$ and $NSPACE$?

(v) Provide a diagram which makes clear the relationship between time and space, for the time complexity classes P, NP and *co-NP*, and the space complexity class $PSPACE$.