

CA313 Algorithms and Complexity

Spring 2007

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

Q 1.

Assume the following Type 0 grammar:

$$\langle \begin{array}{l} V_t = \{a, \textit{computer}, \textit{Peter}, \textit{sold}, \textit{is}, \textit{was}, \textit{by}, \textit{faulty}\} \\ V_n = \{S, NP, VP, V, D, N\} \\ P = \{S \rightarrow NP, VP \\ NP \rightarrow D, \textit{computer} \\ NP \rightarrow \textit{Peter} \\ VP \rightarrow V \\ VP \rightarrow V, NP \\ D \rightarrow a \\ V \rightarrow \textit{sold} \\ \textit{Peter sold a computer} \rightarrow \textit{the computer was sold by Peter} \\ \textit{a computer sold Peter} \rightarrow \textit{Peter was faulty}\} \end{array} \rangle$$

$S = S$)

(i) Write down the strings in the language permitted by this grammar. Show their derivations using trees.

(ii) Show which of the rules in the grammar could be rules in a CSG, a CFG, or an FSG. Explain why.

(iii) Which rules in the set P make this grammar a Type 0 grammar? Explain why.

Q 2.

(i) Define the constraints on α and β in a context-sensitive grammar.

Assuming the following ruleset:

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

(i) what are the first three strings in this language?

(ii) Show the derivation of the 2nd and 3rd shortest strings using trees;

(iii) Show the derivation of the 2nd and 3rd shortest strings using string manipulation.

(iv) Does it make a difference in which order you apply the rules? Explain your answer.

Q 3.

Describe in your own words:

(i) What a Turing machine consists of?

(ii) What it is useful for?

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

(iv) Assuming unary input, describe the transition function of the Turing machine **UN+1**. Comment on the approach you have taken compared to other possible solutions.

(v) For the initial configuration $(q_0, \underline{1}11\#)$, show how the Turing machine you provided in (iv) copes with that input tape.

Q 4.

(i) Give a formal definition of the notion(s) of complexity.

(ii) Order the following complexities:

$O(1000n^2)$, $O(\ln(n) + 5n)$, $O(\exp(n))$, $O(6\ln(n))$, $O(2n^3 + 500)$, $2\ln(n^2)$,
 $O(3n^3)$, $O(2n^3 \times \ln(n))$, $O(2^n)$, $O(4n^3)$.

(iii) Compute the space and time complexities of the following programs (in the worst-case scenario). Justify your answer.

```
// note: a is a sorted array (in increasing order)
int doSomething(int[] a, int x) {
    int s = 0;
    int b = a.length-1;
    do {
        int m = (s+b)/2;
        if (a[m] == x) {
            return m;
        } else if (a[m] > x) {
            b = m-1;
        } else {
            s = m+1;
        }
    } while (s <= b);
    return -1;
}
```

```
// note: a is an unsorted array
int doSomethingElse(int[] x) {
    int k = 0;
    int n = x.length;
    for (int i = 1; i < n; i++) {
        if (x[i] < x[k]) {
            k = i;
        }
    }
    return k;
}
```

(iv) What are these programs doing? Explain.

Q 5.

(i) Give an example of **one** problem that belongs:

- to the P class.
- to the NP class (other than the Traveling Salesman Problem).

(ii) Can you prove that the NP problem you mentioned (or the Traveling Salesman Problem if you have not answered the first question) is actually NP?

(iii) What would be an exact algorithm to solve it? What is its complexity?

(iv) Describe one algorithm that would lead to an approximate solution in reasonable time. How would you represent your problem in order to solve it using this algorithm?