Midterm 6892

Theodore S. Norvell

2012 Nov 13

Your name please:

The following definitions may help for question Q0.

procedure *expect*($a : A$)
precondition s is not empty
    if $\neg f$ then return end if
    if $s(0) = a$ then consume()
    else $f :=$ false end if
end *expect*

procedure *consume*()
precondition s is not empty
    $s := s[1, ..s.\text{length}]$
end *consume*

procedure *nextIsAtom*() : *Boolean*
precondition s is not empty
    return $s[0] \in \mathcal{I}$
end *nextIsAtom*

Q0. Consider the following context free grammar for an alphabet consisting of $\{\text{`('}, \text{`)'}, \text{`.'}, \$\} \cup \mathcal{I}$ where $\mathcal{I}$ is an infinite set of symbols disjoint from $\{\text{`('}, \text{`)'}, \text{`.'}, \$\}$.

$$
\begin{array}{rcl}
S' & \rightarrow & S\ \$ \\
S & \rightarrow & \text{`('}\ SS\ \text{`)'} \\
S & \rightarrow & a, \text{ for each } a \text{ in } \mathcal{I} \\
SS & \rightarrow & \epsilon \\
SS & \rightarrow & S\ MS \\
MS & \rightarrow & \text{`.'}\ S \\
MS & \rightarrow & S\ MS \\
MS & \rightarrow & \epsilon
\end{array}
$$

(a) [5] Write a recursive recognition procedure for nonterminal MS.

(b) [5] Remove the recursion from the procedure.

Q1.[10] Here are three sets of functions $O(n \log_2 n), \Omega(n \log_2 n), \Theta(n \log_2 n)$.
(a)[6] Which sets are the following functions in?
(i) $f(n) = 10 \times n \times \ln(n/2)$

(ii) $g(n) = n^2 + 2n + 1$

(iii) $h(n) = 100 \ln n$

(b)[4] Factoring is the problem of finding all prime factors of a number. The input size is generally the number of bits required to represent the number.
(i) If you develop an algorithm that can find all prime factors of any n-bit number in time $\Theta(n^7)$ in the worst-case, what does this imply about the complexity of the factoring problem?

(ii) If I were to show that no algorithm for can take $\Theta(n^5)$ worst-case time. What would this imply about the complexity of the factoring problem?

Q2. [5] Show (by giving an example graph and showing the actions of the algorithm on it) that if you modify Dijkstra's algorithm by replacing each $<$ with a $>$, it does not always find the longest path from the source s to every other node.

Q3. [10] Four Vikings need to cross a rickety bridge that can hold only two people at a time, in the dark. They have one torch that must be carried by someone in each party that crosses the bridge. Two people can cross the bridge only at the speed of the slower person. The times it would take the Vikings to cross individually are as follows: Frigga 1 minute, Eric 2 minutes, Bjork 5 minutes, Thormund Lameleg 10 minutes. They have 17 minutes to cross the bridge before their torch burns out.

(i) Explain how this problem can be modelled as a graph problem.

(ii) What algorithm can be used to efficiently find the quickest way for them to cross the bridge.

Bonus. [1] How can the Viking's cross the bridge?
