VICTORIA UNIVERSITY OF WELLINGTON

Te Whare Wananga o te Upoko o te Ika a Maui

EXAMINATIONS — 2003

END-YEAR

COMP 202

Formal Methods of Computer Science

Time Allowed: 3 Hours

Instructions: There are seven (7) questions, worth fourteen (14) marks each, making ninety-eight (98) marks in total.
Answer all the questions.

You may use printed foreign language dictionaries.
You may not use calculators or electronic dictionaries.

Question 1.

The following program takes as input two strings, s and t, which are assumed to be the same length (i.e., $|s| = |t|$). Strings are indexed starting from 0. It is possible to determine from the final value of i whether s and t are in fact the same string.

begin
  $i := |s| - 1$;
  while $i \geq 0$ **and** $s[i] = t[i]$ **do**
    $i := i - 1$
  od
end

(a) Give a specification (signature, precondition, and postcondition) for the problem that this program satisfies. [4 marks]

(b) Explain how a **loop invariant** may be used in the verification of a while-program. [5 marks]

(c) State a loop invariant that may be used to verify the above program. **You do not need to complete the proof.** [5 marks]

Question 2.

$M_1 = (Q, \Sigma, \delta, q_0, F)$, where:

- $Q = \{S_0, S_1, S_2, S_3\}$
- $\Sigma = \{a, b\}$
- $q_0 = S_0$
- $F = \{S_3\}$

and δ is given by the table:

δ	a	b
S_0	$\{S_1, S_2\}$	$\{\}$
S_1	$\{S_1\}$	$\{S_0, S_3\}$
S_2	$\{S_2\}$	$\{S_0\}$
S_3	$\{S_3\}$	$\{S_1\}$

(a) Draw M_1. [4 marks]

(b) Find an FA which accepts the same language as M_1. [10 marks]

Question 3.

(a) Let r and s be regular expressions, and let $L_{\mathbf{r}} = \text{Language}(\mathbf{r})$ and $L_{\mathbf{s}} = \text{Language}(\mathbf{s})$.

Give regular expressions which describe the following languages:

1. $L_{\mathbf{r}} \cup L_{\mathbf{s}}$
2. $L_{\mathbf{r}}^*$
3. $L_{\mathbf{r}} \cap \overline{L_{\mathbf{r}}}$ [3 marks]

(b) Let: $\Sigma = \{a, b\}$
$L_1 = \text{Language}(\mathbf{a}(\mathbf{a} + \mathbf{b})^*)$
$L_2 = \text{Language}((\mathbf{ba})^*)$
$L_3 = \text{Language}(\mathbf{a}^* + \mathbf{b}^*)$

Give a string which:

1. is in L_1 but not in L_2 or L_3
2. is in L_2 but not in L_1 or L_3
3. is in L_3 but not in L_1 or L_2 [3 marks]

(c) State Kleene's theorem. [4 marks]

(d) *Either* give an example of a finite language which is not regular *or* show that every finite language is regular. [4 marks]

Question 4.

Consider the context free grammar

$$\begin{array}{ll} (1,2) & S \to aS \mid T \\ (3,4,5) & T \to aSbS \mid U \mid \Lambda \\ (6) & U \to b \end{array}$$

(a) List the **nullable nonterminals**. [1 mark]

(b) List the **unit productions**. [1 mark]

(c) Find an equivalent grammar with no unit productions. [4 marks]

(d) Give two different **leftmost derivations** of the string $aabb$. [2 marks]

(e) Find an equivalent **unambiguous grammar**. [4 marks]

(f) Using your grammar from part **(e)**, draw a parse tree for the string $aabb$. [2 marks]

Question 5.

The following is a context-free grammar for a fragment of **HTML**: the markup language used for web documents. Nonterminal symbols are written in *Italics*; terminal symbols are enclosed in "quotation marks".

$$
\begin{array}{rlcl}
(1,2) & \mathit{Doc} & \rightarrow & \mathit{Element}\ \mathit{Doc} \mid \mathit{Element} \\
(3) & \mathit{Element} & \rightarrow & \text{“<OL>”}\ \mathit{List}\ \text{“</OL>”} \\
(4) & \mathit{Element} & \rightarrow & \text{“<UL>”}\ \mathit{List}\ \text{“</UL>”} \\
(5,6,7) & \mathit{Element} & \rightarrow & \text{“a”} \mid \text{“b”} \mid \text{“c”} \\
(8,9) & \mathit{List} & \rightarrow & \text{“<LI>”}\ \mathit{Element}\ \mathit{List} \mid \Lambda
\end{array}
$$

Thus, the nonterminals of the grammar are $\{Doc, Element, List\}$, and the terminals are {"<OL>", "</OL>", "<UL>", "</UL>", "<LI>", "a", "b", "c"}.

(a) Explain what it means for a grammar to be in LL(1) form. [3 marks]

(b) Show that the above grammar is **not** in LL(1) form. [2 marks]

(c) Rewrite the grammar so that it is in LL(1) form. [4 marks]

(d) Complete the *ParseList* procedure whose heading appears below, for a recursive-descent parser to recognize the *List* nonterminal. You may assume that procedures *ParseDoc* and *ParseElement* for recognizing the other nonterminals are already written. The parameter *ss*, providing the input for the parser, is a sequence of terminal symbols. You do not need to return a parse tree.

procedure *ParseList* (**in out** *ss*)
begin
  . . .
end

[5 marks]

Question 6.

(a) Define a pushdown automaton that accepts each of the following languages:

(i) The language described by the regular expression $\mathbf{a}^*\mathbf{b}^+$ [2 marks]

(ii) The language of even-length palindromes, $\{ww^R \mid w \in \{a, b\}^*\}$ [2 marks]

(iii) The language consisting of strings over the alphabet $\{a, b\}$ with the same number of as and bs, but occurring in any order [3 marks]

(iv) The language $\{a^m b^n c^{m+n} \mid m, n \geq 1\}$ [3 marks]

(b) Define a pushdown automaton that accepts the language generated by the following grammar, using a **bottom-up** (shift-reduce) strategy.

$$\begin{aligned} S &\rightarrow TU \mid \Lambda \\ T &\rightarrow aS \mid Sb \\ U &\rightarrow aU \mid a \end{aligned}$$

[4 marks]

Question 7.

(a) Let L_1 and L_2 be regular languages. State whether it is possible to write a program which decides whether L_1 and L_2 are the same language. Justify your answer.
[4 marks]

(b) Give an example of a language which is:

(i) context-free but not regular [1 mark]

(ii) computable but not context-free [2 marks]

(iii) computably enumerable but not computable [3 marks]

(iv) not computably enumerable [4 marks]

****