UNIVERSITI SAINS MALAYSIA

Second Semester Examination
Academic Session 2004/2005

March 2005

CPT314 - Automata Theory & Formal Languages

Duration : 3 hours

INSTRUCTIONS TO CANDIDATES:

- Please ensure that this examination paper contains **FIVE** questions in **FIVE** printed pages before you start the examination.
- Answer **ALL** questions.
- This is an 'Open book' examination.

ENGLISH VERSION OF THE QUESTION PAPER

1. Answer both 1(a) and 1(b) below.

   (a) Let $V_1 = \{a, b, c, d\}$, $V_2 = \{a, d, e, f\}$, $V_3 = \{0, 1\}$, and $V_4 = \{b, d, e, g\}$. Solve the following expressions based on the given sets:

   (i) $V_1 \cap V_3$

   (ii) $2^{V_1} \cap 2^{V_2}$

   (iii) $2^{V_1 \cap V_2}$

   (iv) $|V_1 \cup V_4|$

   (v) $|\{V_3, \{V_3\}\}|$

   (50/100)

   (b) Let $\Sigma = \{0, 1\}$. Give regular expressions generating the following languages:

   (i) P = {w | w has exactly a single 1}

   (ii) Q = {w | w has at least one 1}

   (iii) R = {w | w is a string of even length}

   (iv) S = {w | w starts and ends with the same symbol}

   (v) T = {w | the length of w is at most 5}

   (50/100)

1. Let *L* be the language consisting of all words over the symbols 0 and 1 that have a '0' in the third-to-last place.

   (a) Give **two** strings that belong to *L* and two strings that do not belong to *L*.

   (10/100)

   (b) Construct a non-deterministic finite state automaton *N* that accepts the strings belonging to *L*.

   (20/100)

   (c) Convert NFA *N* into DFA *M*.

   (30/100)

(d) Give the formal description of DFA *M* and draw its state diagram.

(20/100)

(e) Try to minimize DFA *M* into a smaller DFA *M'* (if any). What can you say about *M* and *M'*?

(20/100)

3. Consider the grammar G = ({S,A,B}, {a, b, c}, *P*, S), where *P* is given by

$S \rightarrow aABb$

$A \rightarrow bAb \mid c$

$B \rightarrow aB \mid b$

(a) Give a leftmost and rightmost derivation of the string *abcbabb* in G.

(10/100)

(b) Give two strings that are not in G.

(10/100)

(c) What is the shortest string in G? Give its length.

(10/100)

(d) Give the equivalent PDA *M* and draw its transition diagram.

(50/100)

(e) Simulate *M* for the string *abcbabb*.

(20/100)

4. Let Σ = {0, …, 9, +, -, *, /, ^,), (}. The language of all strings over the alphabet Σ represents syntactically correct arithmetic expressions on integers involving the following operators.

Operators	Precedence	Associativity	
–	3		Negation
^	2	Right	Exponentiation
*	1	Left	Multiplication
/	1	Left	Division
+	0	Left	Addition
–	0	Left	Subtraction

In the table above, the precedence of operators is listed from highest (3) to lowest (0) precedence. We say that operator **p** has precedence over operator **q** if an expression is of the form:

expr1 **p** *expr2* **q** *expr3* is interpreted only as *(expr1* **p** *expr2)* **q** *expr3*.

In arithmetic expressions, (well balanced) parentheses are used to modify the order of precedence and associativity of operators.

In arithmetic expression, except for 0 (zero), integers cannot start with zero. For example, 002 is not an integer.

Samples of well formed arithmetic expressions:

69*2+5^2   2*5+4/(4+5)

–(346–3)*–2   —0+10+687^(2 + 1)/5+7

Samples of not well formed arithmetic expressions:

45**23 + 2   234+((45*67)+2

00234+34^(2+1)   823++/4

(a) Add parentheses to the following expressions to make clear the precedence of the operators.

e.g: 2*4+22+452+ −1 → ((2*4)+22)+452)+(−1)

(i) 20+—20+68^2/23−23 →

(ii) 2+−2*3^3^4*8+2^4^3+2*5 →

(20/100)

(b) Find a grammar G that can generate all arithmetic expressions described above.

(40/100)

(c) Convert the grammar G into a PDA M that can recognise all well formed arithmetic expressions.

(40/100)

5. Let $L = \{0^{2^n} \mid n \geq 0\}$.

(a) Design a Turing machine T for the language L.

(50/100)

(b) Draw the state diagram of T.

(30/100)

(c) Show the ID's of the Turing machine T if the input tape contains the string "0".

(20/100)

-oooOooo-