# CS21004 Formal Languages and Automata Theory, Spring 2010–11

## End-Semester Test

Maximum marks: 50 | Date: April 2011 | Duration: 3 hours

---

**Roll no:** ____________________ **Name:** ____________________________________________

[ *Write your answers in the question paper itself. Be brief and precise. Answer all questions.* ]

**1.** Let $L$ be the language

$$L = \{w \in \{a,b\}^* \mid w \text{ contains an equal number of occurrences of } ab \text{ and } ba\}.$$

For example, $ababa \in L$ (two occurrences of $ab$, and two of $ba$), whereas $bbaba \notin L$ (one occurrence of $ab$, and two of $ba$).

**(a)** Give a regular expression whose language is $L$. **(5)**

**(b)** Design a DFA/NFA/$\epsilon$-NFA to accept $L$. **(5)**

**2.** Consider the following context-free grammar to generate arithmetic expressions in one variable $a$, involving addition and multiplication operations only. Here, $S$ is the start symbol.

$$S \to a \mid S + S \mid S \times S$$

**(a)** Draw *all* the parse trees for the string $a + a \times a + a$ following this grammar. **(5)**

**(b)** Design an unambiguous grammar to generate the same language. Force the operations to be evaluated from left to right (like in some digital calculators). This means that $+$ and $\times$ are given the same precedence and left-to-right associativity. For example, $a + a \times a + a$ is to be interpreted as $((a + a) \times a) + a$. **(5)**

**3.** Consider the following unrestricted (Type 0) grammar with the start symbol $S$, with non-terminal symbols $S, A, B, C$, and with terminal symbols $a, b, c$.

$$
\begin{array}{llllll}
S \to ABCS \ \mid \ \epsilon, & & & & & \\
AB \to BA, & BA \to AB, & AC \to CA, & CA \to AC, & BC \to CB, & CB \to BC, \\
A \to a, & B \to b, & C \to c. & & &
\end{array}
$$

**(a)** Show a derivation of the string $babacc$ using this grammar. Show each individual step in the derivation, and mention the rule used in each step. **(5)**

**(b)** What language over $\{a, b, c\}$ is generated by this grammar? Justify. **(5)**

**4.** Exactly one of the following languages is recursive, exactly one is not recursive but r.e., and exactly one is not r.e. Identify which one is what, and supply a corroborating proof in each case.

$$\begin{aligned} L_1 &= \{M_1\#M_2 \mid \text{NSTEPS}(M_1,\epsilon) < \text{NSTEPS}(M_2,\epsilon)\}, \\ L_2 &= \{M_1\#M_2 \mid \text{NSTEPS}(M_1,\epsilon) \leqslant \text{NSTEPS}(M_2,\epsilon)\}, \\ L_3 &= \{M_1\#M_2 \mid \mathcal{L}(M_1) \cap \mathcal{L}(M_2) \text{ is r.e.}\}. \end{aligned}$$

Here, $M_1$ and $M_2$ are (encoding of) Turing machines. For a Turing machine $M$ and for an input $w$ of $M$, the symbol NSTEPS$(M,w)$ stands for the number of steps that $M$ takes before halting, upon input $w$. If $M$ loops (that is, does not halt) on $w$, we take NSTEPS$(M,w) = \infty$. If $M'$ also loops on $w'$, we make the convention that NSTEPS$(M,w) =$ NSTEPS$(M',w')$ (that is, two infinities in this context are equal).

**(a)** $L_1$ is ________________________________________ .

*Proof* **(5)**

**(b)** $L_2$ is ________________________________________ .

*Proof* **(5)**

**(c)** $L_3$ is ______________________________________________ .

*Proof* **(5)**

**5.** Which of the following are properties of r.e. sets? First, supply an answer Yes/No. If the answer is *Yes*, mention whether the property is trivial and/or monotone. If the answer is *No*, supply a one-line justification. In each case, the property is specified by a Turing machine $M$. **(5)**

**(a)** $M$ accepts $\epsilon$.

---

**(b)** $M$ (explicitly) rejects $\epsilon$.

---

**(c)** $M$ halts on $\epsilon$.

---

**(d)** $\mathcal{L}(M)$ is a context-free language.

---

**(e)** $\mathcal{L}(M)$ contains a context-free language.

---