CIT3130 Theory of Computation

Second mid-semester examination, September 2002

Time: 75 minutes
Perusal: 5 minutes
Marks: 40 marks

Please write neatly in ink and show all working.

Statements of the pumping lemmas for regular and context-free languages are appended in case they are required.

1. Transform the following deterministic finite automaton A into an equivalent regular expression E (*i.e.*, so that E defines the same language that A accepts).

	a	b
$\rightarrow p$	q	r
$* q$	r	s
$* r$	r	q
s	s	s

(Note that A has two final states.)

(6 marks)

2. Construct a context-free grammar that defines the following language:

$$L_a = \{\, w \in \{a, b\}^* \mid w \text{ contains strictly more } a\text{'s than } b\text{'s} \,\}$$

Clearly show the variables, terminals and start symbol of the grammar.

(6 marks)

3. Construct a pushdown automaton (PDA) that accepts the following language:

$$L_{mn} = \{\, a^m b^n c^m \mid m, n \geq 1 \,\}$$

Clearly show the states, start state, final state and stack alphabet of your PDA.

State whether your PDA is deterministic or nondeterministic.

(6 marks)

4. Prove that the following language is **not** context-free:

$$L_{abc} = \{\, a^p b^q c^r \mid 0 \leq p < q < r \,\}$$

(6 marks)

5. Suppose that L_1 and L_2 are context-free languages and that L_3 is a regular language. Which of the following are context-free languages?

   (a) $L_1 \cup L_2$
   (b) $L_1 \cup L_3$
   (c) $L_1 \cap L_2$
   (d) $L_1 \cap L_3$
   (e) $L_1 L_3$
   (f) $L_1^* L_3$

   (3 marks)

6. The "halting problem" is the problem of determining whether a given program P halts when executed on a given input I.

   Complete the following (incomplete) proof that the halting problem is undecidable, *i.e.*, no algorithm (or C program) can determine whether a given program P halts when executed on a given input I.

   Proof Suppose H is a C program thata solves the halting problem. That is, given a program P and an input I, $H(P, I)$ prints "yes" and halts if $P(I)$ halts, and prints "no" and halts if $P(I)$ loops forever.

   Modify H into H_1 by replacing every statement in H that prints "yes" by an infinite loop (*e.g.*, `while (true) {}`). Then $H_1(P, I)$ loops forever if $P(I)$ halts, and prints "no" and halts if $P(I)$ loops forever.

   To be completed ...

   (In this proof, the notation $P(I_1, I_2)$ denotes the execution of program P on inputs I_1 and I_2.)

   (5 marks)

7. Construct a Turing machine that accepts the following language:

   $$L_n = \{\, a^n b^n \mid n \geq 0 \,\}$$

   (Partial marks will be awarded for a "structured program" equivalent to the Turing machine.)

   (6 marks)

8. (a) What is a computably enumerable language?
   (b) What is a computable language?
   (c) How do you represent an instantaneous description (ID) $a_1 \ldots a_m q b_1 \ldots b_n$ of a Turing machine in a stack machine with two stacks?
   (d) Does a queue machine with two queues have more or less expressive power than a Turing machine?

   (2 marks)

Pumping lemma for regular languages

Let L be a regular language. Then there exists a constant $n \geq 1$ such that, for every string w in L with $|w| \geq n$, we can write $w = xyz$ in such a way that:

1. $|xy| \leq n$ (the initial section is not too long).
2. $y \neq \epsilon$ (the string to pump is not empty).
3. For all $k \geq 0$, the string $xy^k z$ is in L (the string y may be pumped any number of times, including 0, and the resulting string is still in L).

Pumping lemma for context-free languages

Let L be a context-free language. Then there exists a constant $n \geq 1$ such that, for every string z in L with $|z| \geq n$, we can write $z = uvwxy$ in such a way that:

1. $|vwx| \leq n$ (the middle section is not too long).
2. $vx \neq \epsilon$ (at least one of the strings to pump is not empty).
3. For all $k \geq 0$, the string $uv^k wx^k y$ is in L (the strings v and x may be pumped any number of times, including 0, and the resulting string is still in L).