Computing Theory
COSC 1107/1105
Sample Exercise

1 Assessment details

1. Consider the grammar derivations below.

   $S \Rightarrow aSbb \Rightarrow aaSbbbb \Rightarrow aa\#bbbb$
   $S \Rightarrow ccBd \Rightarrow ccccBdd \Rightarrow cccc\#dd$
   $S \Rightarrow ccBd \Rightarrow ccAd \Rightarrow cc1A1d \Rightarrow cc1\#1d$
   $S \Rightarrow A \Rightarrow 2A2 \Rightarrow 2C2 \Rightarrow 2cCc2 \Rightarrow 2c\#c2$

   (a) From the above derivations, construct rules that must exist in any context-free grammar G for which these derivations are correct.

   (b) Assuming that these are all the rules in G, give $L(G)$ in set notation.

   (c) Is the grammar G regular? Explain your answer.

   (d) Is there a deterministic finite state automaton for $L(G)$? Explain your answer.

   (e) Construct a context-free grammar for the language L_2 below.

   $L_2 = \{w_1 \; a^{2i} \; c^j \; a^l \; d^k \; b^i \; w_2 \mid w_1, w_2 \in \{1,2\}^*, |w_1| = |w_2|, i \geq 1, j, k, l \geq 0, j < k\}$

   (f) Explain why there is not necessarily a context-free grammar for $\overline{L_1}$.

2. Legolas Brownbean, a friend of a more famous Legolas, has written the following discussion. There are 6 incorrect statements in the paragraph below. Identify all 6 incorrect statements and justify each of your answers.

   Note: A single sentence counts as one statement.

   "Algorithms are an expression of knowledge that makes computers work. In order to solve a particular computational problem, such as finding the factors of a given number, or sorting a list of students by their student number, an algorithm must be developed and implemented. It comes as a great surprise to many people that there are some problems for which there is no algorithm possible. These are known as undecidable problems and include the Halting problem for Turing machines, the busy beaver problem, the tile problem and the vertex cover problem. Alan Turing was the first to show there was an undecidable problem, and since that time many other problems have been shown also to be undecidable. This is done by reducing a problem (say A) with unknown status to a problem known to be undecidable (say B). From this reduction it is simple to show that problem A must be undecidable using the Pumping Lemma. The existence of undecidable problems means that certain properties of programs cannot be guaranteed. So anyone faced with an undecidable problem should refuse to work on it, as no instance of any such problem can ever be solved. The Church-Turing thesis, which states that anything computable can be performed by a Turing machine, indicates that we cannot avoid undecidable problems by changing technologies. Undecidable problems are only an issue for nondeterministic Turing machines, as anything that

can be executed on a deterministic Turing machine is decidable. Unrestricted grammars also do not have any issues with undecidable problems. Note though that NP-complete and other intractable problems are all decidable. The limits on computing with such problems is practical rather than fundamental."

3. (a) Katie the Koala loves playing Platypus tournaments. She is particularly fond of the 4-player variety, for which a tournament of n machines will require $n(n+1)(n+2)(n+3)/24$ matches. She ran her own tournament on her own laptop for 200 machines, which took 97.65 seconds. Encouraged by this, she resolves to run the largest tournament that she can in 10 hours. Calculate the largest number of machines for which Katie can run a tournament in this amount of time.

   (b) Katie is disappointed by the previous result and so turns to playing the same 4-player game but in a different tournament format suggested by her friend the Wombat. She works out that using this format the tournament takes $(1.1)^n/(1.05)^n$ matches for n machines where $n \geq 200$, using the Platypus game as in the previous question. Calculate the largest number of machines for which Katie can run a Wombat-style tournament in 10 hours.

   (c) Never one to quit, Katie asks some friends for help. They work out that they have 10 identical computers between them, each of which can run a 4-player game in exactly the same time as Katie, and they agree to divide up the matches equally between them. Katie decides that they will attempt to collectively run a tournament for 2,000 machines in the original 4-player format. How long will each machine have to run to complete each person's share (ie 10%) of the matches? We will call this time t_1.

   (d) Flushed with their success, the group of friends that each one of them will dedicate the same time as calculated in the previous question for running as large a Wombat-style tournament as they can. Calculate the number of machines for which they can complete a Wombat-style tournament this way. In other words, what is the largest number of machines for which they can run a Wombat-style tournament in time t_1?

4. (a) Student numbers at the University of Minas Tirith are strings of length 7 over the alphabet $0, 1, 2, 3, 4, 5, 6, 7, 8, 9$. The first digit of each number must be 7 and the final digit must be 1. Construct a Turing machine which recognises such student numbers. Any string of length 6 or less should be rejected, as should any string of length 8 or more.

   (b) Which of the following machines could also be used to recognise such numbers? Briefly justify each of your answers.
   - A non-deterministic PDA
   - A deterministic PDA
   - A non-deterministic finite state automaton (NFA)
   - A deterministic finite state automaton (DFA)
   - A linear-bounded automaton

   (c) Given a Minas Tirith student number (as in the previous question), the internal string is the 5 digits of the number excluding the 7 at the start and the 1 at the end. For example, given the student number 7654321, the internal string is 65432. The Steward of the university, Denethor the Demented, announces that he is especially interested in student numbers whose internal string is either a palindrome or contains at least three 9s, which he calls Palantir numbers. Note that the string has to be a valid student number as well in order to be a Palantir number. For example, 7643461 and 7299991 are both Palantir numbers, and

7666441 and 7992281 are valid student numbers but not Palantir numbers, and 199991 and 63399332 are neither.

Which of the following machines can be used to recognise valid student numbers that are also Palantir numbers? Briefly justify each of your answers. You do not have to explicitly construct any machines in your answer.

- A deterministic Turing machine
- A non-deterministic PDA
- A non-deterministic finite state automaton (NFA)
- A deterministic finite state automaton (DFA)

(d) Some time later, Denethor changes his mind about both Minas Tirith student numbers and Palantir numbers. He decrees that from now on student numbers can be of any odd length, provided the first digit is 7 and the final digit is 1. He also decrees that Palantir numbers must be palindromes only. For example, 711 and 798654321 are now both valid student numbers, but 7992281 is no longer a Palantir number, whereas 71234543211 is a Palantir number.

Given these changes, which of the following machines can be used to recognise valid student numbers that are also Palantir numbers? Briefly justify each of your answers. You do not have to explicitly construct any machines in your answer.

- A deterministic Turing machine
- A non-deterministic PDA
- A non-deterministic finite state automaton (NFA)
- A deterministic finite state automaton (DFA)

5. (a) Construct a Turing machine M_1 which given as input a string of length 7 over $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ performs as follows.

- The fourth digit, whatever it is, is replaced with a blank.
- Any of the digits 5,6,7,8 or 9 is replaced with 0,1,2,3 or 4 respectively.
- The machine must terminate on all strings over $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}^*$ (including the empty string).
- The machine only halts in an accepting state if the input string has exactly 7 digits. Strings of any other length are rejected.

(b) Construct a Turing machine M_2 which given as input two string of digits over $\{0, 1, 2, 3, 4\}$ separated by a blank performs as follows.

- If either string has length less than three, the machine halts with rejection.
- If the final digit of both strings is 0, 2 or 4, then machine halts with acceptance.
- If the final digit of either string is 1 or 3, the machine loops forever (i.e. it does not terminate).

(c) Consider the machine formed by combining M_1 and M_2 as above in sequence into a new machine M_3, so that M_3 first performs as M_1 does, rewinds the tape head to the appropriate point on the tape and then performs as M_2 does. Clearly the behaviour of M_3 will depend on the behaviour of M_1 and M_2.

- For which inputs does M_3 halt with rejection?
- For which inputs does M_3 halt with acceptance?

- For which inputs does M_3 not halt?

In each case, justify your answer.

(d) An evil wizard now changes M_1 to another machine M_0. This is the same as M_1 except that rather than deterministically replacing 5, 6, 7, 8 or 9 with 0, 1, 2, 3 or 4, M_0 *non-deterministically* replaces each of these digits with one of 0, 1, 2, 3, or 4. For example, if there is a transition such as $\delta(q_i, 5) = (q_j, 0, R)$ in M_1, there are 5 transitions in M_0, i.e. $\delta(q_i, 5) = \{(q_j, 0, R), (q_j, 1, R), (q_j, 2, R), (q_j, 3, R), (q_j, 4, R),\}$ in M_0.

The same combination technique as above is used to combine M_0 and M_2 into M_4.

- For which inputs does M_4 halt with rejection?
- For which inputs does M_4 halt with acceptance?
- For which inputs does M_4 not halt?

In each case, justify your answer.

6. (a) Prove that the language $L_1 = \{a^i b^j c^j d^i | i, j \geq 0\}$ is not regular by using the Pumping Lemma. Use the string $a^n b^n c^n d^n$ at an appropriate point in the proof.

(b) Write out the proof of the same result, which uses the string $1^{2n} 2^{2n}$ and $i = 4$ instead. Which steps are different? Which steps remain the same?

(c) Give a PDA for the language $L_1 = \{a^i b^j c^j d^i | i, j \geq 0\}$. Is your machine deterministic or non-deterministic? Briefly explain your answer.

(d) Is there a context-free grammar for the language $L_2 = \{a^i b^j c^i d^j | i, j \geq 1\}$? Explain your answer.

(e) The language $L_3 = \{a^i b^j c^j d^i | i, j \geq 2\}$ is regular. Construct a finite state automaton which accepts L_3 (and only L_3). Your machine can be either deterministic or non-deterministic.

(f) The language $L_4 = \{a^i b^j c^i d^j | 1 \leq i, j \leq 6\}$ is also regular. If you were to extend your previous machine to accept this language, how many transitions would you need? In other words, if there are k transitions in your machine for L_3, give a formula for your estimated number of transitions required in the machine for L_4.

(g) Give a context-free grammar for $L_5 = \{a^i b^j c^j d^i \mid i, j \geq 1\}$ with at most 4 rules. The number of rules is the number of different right-hand sides in the grammar; for example, the grammar below has 6 rules.

$S \to aA | Aa | a$
$A \to aBaa | Baa$
$B \to abc$

7. Snivelling Sam the Shonky Snake Oil Salesman has heard about your work on Turing machines. This is of great interest to Sam, who wishes to set up a Gallery of Important Turing Machines which are Useful and Brilliant (GITMUB). This will showcase various Turing machines, which must be of the acceptor/rejector type (i.e. Turing machines which accept a given language by terminating in an accepting state). Sam is keen to ensure that he only has the "best and brightest" Turing machines on display from those submitted to GITMUB by the general public, and so insists on the following policy on adding machines. Let the machine to be added to GITMUB be M, and the machines already in GITMUB be $M_1, \ldots, M_n$. If $L(M) \neq L(M_i)$ for all $1 \leq i \leq n$, then M is added to GITMUB. Otherwise, (i.e. $L(M) = L(M_i)$ for some $1 \leq i \leq n$) then Sam makes a choice between M and M_i, and either rejects M as a member of GITMUB, or ejects M_i from

GITMUB and replaces it with M (according to what he perceives as the 'inherent artistic merit' of each machine). This means that Sam can boast that GITMUB's machines are all guaranteed to be unique representatives of machines accepting a particular language.

(a) Explain why, under the above policy, GITMUB can never be guaranteed to contain more than one machine.

(b) Suggest one way that Sam could relax his policy on adding machines to the gallery in order to have a larger collection of machines in GITMUB while remaining as true as possible to his original aim.

(c) Sam is also interested in the history of Turing machines, and runs a competition to find the Greatest Ingenious Turing Machine of All Time (the GITMOAT). Sam asks you for your choice of a Turing machine to enter into this competition. What machine would be your choice? Justify your answer.