Question Paper for Quiz 2 Examination
Course CS310

1 November, 2023

- The question paper carries 60 marks in total and consists of 4 questions.
- Justification is required for all questions unless explicitely stated as not required. Partial answers carry partial marks. Hence it is advantageous to show working of your answers.
- Illegible or un-understandable answers will get no marks. Pl. write clearly and accurately.
- This is a paper and pen examination. Answers must be written in an answersheet which must be submitted **with role number clearly marked**.
- The examination must be completed in 60 minutes. Do not spend too much time on a single questions.
- Students may keep 3 printed or handwritten A4 size sheets with them for reference. Use of books, notebooks, laptops, mobile phones etc. is not allowed.
- Good Luck!

Q1 (20 marks) For each of the following statements please state whether it is true or false. Please provide a short justification in no more than 3-5 lines for your answer. Justification will be given marks only if the true/false answer is correct.

1. There is a 10-state PDA which can accept a CFL which is not accepted by any 1-state PDA.
   False.
   If a CFL is accepted by n-state PDA then there exists a 1-state PDA with larger stack alphabet which accepts the same language.
2. Let $A = \{a^n b^n c^m a^m b^k c^k \mid n, m, k \geq 0\}$. Then, A is a context-free language.
   True.
   A is catenation of $a^n b^n$ and $c^m a^m$ and $b^k c^k$ each of which are CFL. Also CFL are closed under catenation.

3. 4-stack automata are more expressive than 2-stack automata.
False.
Every 4-stack automaton can be simulated by 5 tape TM. Morever, 5 Tape TM can be simulated by 1 tape TM. And 1 tape TM can be simulated by 2-stack automata.

4. 4-counter automata are more expressive than 2-counter automata.
False.
k-counter automata can be simulated using 2-counter automata. We can encode 4 counters using 1 counter using Godel numbering and second scratch counter is needed for doing inc/dec operations.

5. It is decidable whether a given one-tape turing machine M on given input y ever writes a symbol \$ on the tape. Here \$ is a symbol in the tape alphabet.
False.
We can reduce Halting problem to this problem. Given M, x we translate this to N, x where N has same transitions as M, but after any transition writing \$ the machine N transits to halt state (accept/reject state).

6. It is decidable that given an input $\langle M \rangle \# 0^n$, whether the turing machine M halts in n or less steps on the input word ϵ.
True
We can simulate M on ϵ for m steps. If machine does not halt within n steps we reject. A separate work tape may be used to count steps.

7. Recursive languages are closed under complementation.
True.
Any recursive language is accpeted by some TTM M. We transform M to N by exchanging accept and reject states. This TTM N accepts the complement language.

8. Intersection of two R.E. languages is always R.E.
True
We can run TMs for the two languages on two separate work tapes. We accept when both have accepted.

9. It is decidable whether the intersection of two R.E. languages is RECURSIVE. False.
Intersecion of two R.E. languages will be R.E.but not necessarily RECURSIVE in general. We can construct TM for this intersection language. But whether a TM accepts RECUrSIVE language is undecidable as shown in class.

10. Language $\{\langle M \rangle \mid M \text{ takes less than } 2^{481} \text{ steps on all inputs}\}$ is R.E.
True.
Actually this language is RECURSIVE and hence also R.E. The language is recursive as in 2^{481} steps the TM can examine only the first 2^{481} input

symbos of the input. We can systematically enumerate all such inputs and simulate the TM on these to check the property.

Rubric: For each item, 1 Mark for correct choice of True/False and additional 1 Mark for correct short justification. Do not check justification for wrong True/False choices.

Q2 (15 marks) Let the alphabet be $\{a, b, c\}$. Let $L_1 = \{a^n b^n c^n \mid n \geq 0\}$. Let $L_2 = a^*b^*c^* - L_1$ where $A - B$ denotes the set of all elements in A which are not in B.

1. Answer whether L_1 is (A) Regular, (B) Context-free but not regular, (C) Recursive but not context-free, by choosing the correct option. **Formally, prove your answer**.

2. Answer whether L_2 is (A) Regular, (B) Context-free but not regular, (C) Recursive but not context-free, by choosing the correct option. Justify your answer.

Solution: (1) (C) L_1 is not CFL. We prove this using pumping lemma for CFL.
Let Demon choose k, We choose the word $z = a^k b^k c^k \in L_1$. Demon splits $z = uvwxy$ with $|vx| > 0$ and $|vwx| \leq k$.
Case 1. v or x has two different letters in it (e.g. $v = a^p b^q$). Then pumping with $i = 2$ gives word $z' = uv^2wx^2y$ which is not even of the form $a^*b^*c^*$, Hence $z' \notin L_1$.
Case 2. If vx contain a single letter, pumping up by $i = 2$ increases only the count of that single letter violating the condition that count of a,b,c must be equal.
Case 3. $v = a^p$. Then $x = b^q$ since all c are more than k distance from any a. Then $u = a^l$ and $w = a^m b^n$ and $y = b^r c^k$ for some $l, m, n \geq 0$ with $l+p+m = k$ and $n + q + r = k$. Then pumping up with $i = 2$ gives word $z' = a^{k+p} b^{k+q} c^k$ with $p + q > 0$. Hence, $z' \notin L_1$.
Case 4 $v = b^p$ and $x = c^q$. Argument is similar to case 3.

(2) (B) L_2 is CFL but not regular.
We first show that L_2 is CFL. Note that $L_2 = L2A \cup L2B \cup L2C \cup L2D$ where $L2A = \{x = a^i b^j c^k \mid i < j\}$, and $L2B = \{x = a^i b^j c^k \mid i > j\}$ and $L2C = \{x = a^i b^j c^k \mid j < k\}$, and $L2D = \{x = a^i b^j c^k \mid j > k\}$. We show that each of these is CFL by giving a grammar for it. (Alternatively, PDA can also be given). Hence, their union L_2 is also CFL.

Let $G2A$ have productions,
$S \to DC, \quad D \to aDb|bB, \quad B \to bB|\epsilon, \quad C \to cC|\epsilon$. Similarly, we can give grammars for other three languages.

We show that L_2 is not regular using the Myhill-Nerode theorem. Consider Nerode equivalence $\equiv_{L_2}$. We claim that for each $k \in N$ word $a^k b^k$ belongs to a separate equivalence class, i.e. $m \neq k$ implies $a^m b^m \not\equiv_{L_2} a^k b^k$. This is because, we can extend both words with c^k giving $a^m b^m c^k \in L_2$ and $a^k b^k c^k \notin L_2$. Hence, $\equiv_{L_2}$ has infinitely many equivalence classes. By Myhill Nerode theorem L_2 is not regular.
Remark: It seems difficult to prove this result using Pumping Lemma for regular languages.
Rubric:

- For Part (1) Give 1 mark if option chosen is correct.
  Additional 5 marks for correct use of pumping lemma with all cases men-

tioned. Do not cut marks if cases are similar and hence not elaborated.
Additional 1 mark for showing language RECURSIVE by a total turing machine.

- For Part (2) give 1 mark for correct option.
  Additional 4 marks if correct CFG or PDA is given for L_2.
  Additional 4 marks for showing that L_2 is not regular.

Q3 (10 marks) Let $MP = \{\langle M\#x\rangle \mid M \text{ accepts } x\}$. Show that the laguage MP is not recursive. Give a direct proof without using reduction from other undecidable problems.

Solution: We prove that MP is not Recursive.

Assume to contrary that there is TTM K s.t. $L(K) = MP$.

Then K on $\langle M\#x\rangle$

Halts and Accepts if M accepts x

Halts and Rejects if M rejects or Loops on x

- Given K Construct TM N.
  TM N any input $x \in \{0,1\}^*$ behaves as follows:
  - Writes $x\#x$ on Input tape.
  - Runs K on input tape.
  - if K accepts then N rejects (loops will also work) , and
    if K rejects then N accepts
- Note: Given 9-Tuple for K, we can write 9-Tuple for N.
- Does N accept $\langle N\rangle$?
- N accepts $\langle N\rangle$
  $\Leftrightarrow$ K rejects $\langle N\#N\rangle$
  $\Leftrightarrow$ N rejects $\langle N\rangle$

Contradiction. Hence the assumption that we have TTM K for MP is incorrect.

Rubric:

- Only 1 mark if Many-to-One Reduction is used for proving the result.
- 7 marks in total if correct construction of N with diagonalization is given but contradiction is not shown.
- Only 4 Marks if in above the idea of diagonaliation using K is attempted but N is formulated wrongly.
- 10 marks for correct construction of N and correctly showing contradiction for N accepting $\langle N\rangle$.

Q4 (15 marks) For $A, B \subseteq \Sigma^*$ define

$$A/B \;=\; \{x \in \Sigma^* \;\mid\; \exists y \in B.\; xy \in A\}$$

Show that if A and B are R.E. then A/B is also R.E.

Solution: Let M_1 and M_2 be Turing machines accepting A and B. We give a TM M accepting A/B.

TM M has read-only input tape taking input x. M uses additional 6 work tapes.

- M uses WT1 to enumerate $n \in N$ in unary.
- M uses WT2 to enumerate $y \in \Sigma^*$ such that $|y| \leq n$
- M alternates between Enumeration phase and Checking phase.
- In each enumeration phase it generates a new (y, n) pair such that all $(y, n) \in \Sigma^* \times N$ occur at least once in the systematic enumeration.
- In each checking phase, given (y, n) on WT2 and WT1,
  - M simulates M_1 on xy for n steps using WT3. Additional work tape WT5 may be used as step counter.
  - M simulates M_2 on y for n steps using WT4. Additional work tape WT6 may be used as step counter.
  - If M_1 on WT3 accepts within n steps AND M_2 on WT3 accepts within n steps then M accepts x and halts.
  - M_1 or M_2 reject or do not halt within n steps then M finishes the Checking phase and it goes to next Enumeration phase.
  - Note that a checking phase cannot loop forever.
- If $x \in A/B$ then there exists y s.t. $xy \in L(M_1)$ and $y \in L(M_2)$ and n is the max of steps taken by M_1, M_2 to accept. Then, in systematic enumeration, (y, n) will occur in some Enumeration phase and M will accept and halt in associated Checking phase.
- If $x \notin A/B$ then no such y exists and M will loop forever trying all (y, n) pairs.
- Hence M is a TM accepting A/B.

Alternative proof: We construct a Nondeterministic TM M' which

- takes input x is on input tape IT.
- non-deterministically guesses y on Work tape WT0
- Copies xy on WT1.
  Simulates $M1$ on xy using WT1.
  If M_1 halts and accepts, go to next step. Otherwise, M' rejects x if M_1 rejects xy.

- Copy y from WT0 to WT2.
  Simulate M_2 on y.
  M' accepts x if M_2 accepts y. It rejects/loops if M_2 rejects/loops on y.

Then it is clear that $L(M') = A/B$..
It is known that every such NTM M' can be simulated by a deterministic Turing machine M.

Rubric:

- 5 Marks if machines M_1 and M_2 are run on separate work tapes with inputs xy and y for some choice of y with correct acceptance.

- Additional 10 Marks for systematic enumeration of (y, n) pair, and alternating enumeration and checking phase.

- In above, additional 3 marks only if y is enumerated but no step bound n is used.

- For alternative proof, additional 10 marks if y is guessed non-deterministically in place of enumrating (y, n). and it is stated that NTM can be simulated by deterministic TM.
  If it is not stated that NTM can be simulated by deterministic TM, cut 2 marks.