CSE 105 A-B CSE105Sp15FinalExamV1

Antriksh Rajendra Prasad Yadav

TOTAL POINTS

100.5 / 110

QUESTION 1

1 True/False (6 / 10)

- 0 Correct
- 1 a
- 1 b
- 1 c
- 1 d
- 1 e
- **1 f**
- **1 g**
- **1 h**
- **1 i**
- 1 j

QUESTION 2

2 Yes/No (8 / 8)

- **0 Correct**
- 2 Incorrect 2a
- 2 Incorrect 2b
- 2 Incorrect 2c.a
- 2 Incorrect 2c.b

QUESTION 3

3 DFA (7 / 10)

- 0 Correct
- 1 Not a DFA (missing/extra transitions)
- 1 Did not label initial state
- **1 DFA does not accept (aa)* (including empty string) / accepts a(aa)***
- 1 DFA does not accept (bb)* / accepts b(bb)*
- **2 DFA does not accept even # of a's AND even # of b's (ex. aabb, abab) / accepts odd # of a's or odd # of b's (ex. aaab)**
- 1 Incorrect string (part b)
- 2 Incorrect sequence of states (part b)
- 2 Incorrect sequence (part c)
- 1 Incorrect string (part c)
- 10 Not attempted
- 1 Minor error in the DFA
- 4 Incorrect DFA

QUESTION 4

4 Pumping Lemma (10 / 10)

- **0 Correct**
- 2 String s not in the language
- 2 Incorrect value of i
- 4 Incorrect explanation (picked a string that doesn't work)
- 2 Cannot choose an xyz split
- 6 Incorrect explanation:(1)cannot choose an xyz split (2) picked a string that doesn't work/ picked an i that doesn't work
- 6 No explanation
- 6 Incorrect explanation: Cannot pick a value for p

QUESTION 5

Language Questions 10 pts

5.1 NFA Language (5 / 5)

- **0 Correct**
- 2 Strings (top half) that end in 10 are preceded by a positive but even number of 0's, but your description allows odd number or no of 0's
- 2 Strings that start with 1 must end in zero or more 0's
- 2 Your description allows strings that begin with more than one 1, or which are not in the language
- 2 Your description allows strings that contain exactly a positive even number of 0's, or which are not in the language
- 2 Incorrect use of operation instead of concatentation.

- **2.5** Missing or major error in half the description
- **5** Incorrect description of both parts of language
- **0** Click here to replace this description.

5.2 CFG Language (5 / 5)

- **0 Correct**
- **5** Circled Yes
- **1** Need to justify counterexample (i.e. why is this string not in the described langauge?)
- **1.5** Need to justify how you derived the counterexample
- **1** Incorrect Derivation
- **3.5** Incorrect counterexample

QUESTION 6

6 Closure Regular (10 / 10)

- **0 Correct**
- **2** Languages used to intersect with A not sufficiently demonstrated to be regular.
- **6** Drawing a DFA is not a viable proof strategy for closure properties. You need to ensure the string is in A.
- **3** Good overall strategy with some errors.
- **5** The language demonstrated to be regular does not require that elements be in A.
- **10** Incorrect.
- **1** Only very minor error(s).
- **9** The regular languages are not closed under "subset". Just because B is a subset of A, and A is regular, it does not follow that B is regular.
- **6** Solution shows some understanding of DFAs and the general proof by construction strategy for proving closure. However the constructed DFA does not recognize B.
- **2** Invalid/Nonsensical mathematical notation.

QUESTION 7

Closure Recognizable 14 pts

7.1 High-Level Description (8 / 8)

- **0 Correct**
- **4** Did not check all ways to split w.
- **4** Did not run M in parallel or nondeterministically.
- **4** Since M may not halt, your loop will not check all possible ways to split the input.
- **4** Checking only one case (a fixed number n or the longest string of a's -- or b's)
- **1** Did not consider case n = 0 (minor error)
- **2** Incorrect to check a^n (or b^n) for acceptance (with TM M or other machines)
- **1** Did not use M as subroutine to check if input in A. (minor error)
- **8** Did not attempt or Incorrect description
- **1** Using language A instead of TM M. (minor error)
- **4** Misunderstanding language A or Va (major error)
- **1** Incorrect input to MV or M
- **1** Misunderstanding Language A or Va (minor error).
- **1** Minor Notation error.
- **1** Minor correctness error

7.2 Correctness (6 / 6)

- **0 Correct**
- **2** Did not consider that M may not halt .
- **3** Only showed correctness in 1 direction.
- **2** Incomplete or insufficient argument in 1 of 2 directions.
- **4** Incomplete or insufficient argument in both directions.
- **6** Did not attempt or Incorrect proof
- **1** Minor notation/terminology error.
- **3** Correctness proof did not match your TM construction (major error).

QUESTION 8

8 Undecideability (8.5 / 10)

- **0** Correct
- **2** Part (a) entirely incorrect.
- **1** Half credit for Part a
- **8** Part (b) entirely incorrect.
- **0.5 M is overloaded, that is M is being used as a symbol twice for different entities.**
- **2** Wrong input to X
- **2** Wrong input to M.
- **5** Reduction is backwards.
- **7** Construction doesn't have anything to do with M'

accepting w.

- 1 Point adjustment

What happens if x is not 01?

QUESTION 9

9 Yes/No (11 / 12)

- 0 Correct
- **- 1 Point adjustment**

QUESTION 10

10 Satisfiability (10 / 10)

- **- 0 Correct**
- - 5 Circled True
- - 5 Incorrect justification
- - 2.5 Partially correct justification
- - 1 Correct, but needs a little more specificity
- - 5 Incorrect justification - O(nm)
- - 2.5 Partially incorrect justification - O(n)
- - 2.5 Partially incorrect justification - O(n^n)
- - 1 Silly mistake in justification

QUESTION 11

11 Name/PID/Section (6 / 6)

- **- 0 Correct**
- - 1.5 No Name
- - 1.5 No PID
- - 1.5 No Lecture
- - 1.5 No Seat Number

ID Check? ✓

Name: ANTRIKSH YADAV

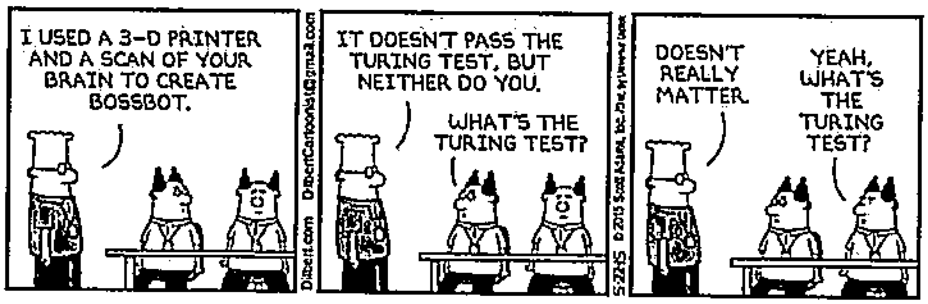
Seat Number: D-8

PID: A98002799

Lecture A00 / Lecture B00

CSE 105
Final Exam
June 6, 2015
Version 1

Version 1



dilbert.com

INSTRUCTIONS — READ THIS NOW

- Write your name, seat number, and PID, and circle your lecture number above; also write your name on every odd-numbered page right now. 6 points will be awarded for doing so.
- This exam is double-sided, and has 10 problems on 12 numbered pages worth a total of 110 points.
- Show your work. To receive full credit, your answers must be neatly written and logically organized. Use the boxes for answers if provided. If you need more space, write on extra page 12; label your work clearly.
- Unless otherwise indicated in the question, you may use theorems proved in class and HW but make sure to mention that you are doing so. You needn't cite theorems by number, but do give a statement of what was proved.
- Turn off and put away all cellphones, calculators, and other electronic devices.
- The exam is closed book and notes.
- This exam is three hours long.
- Academic integrity is expected of all students at all times, whether in the presence or absence of members of the faculty. Understanding this, I declare I shall not give, use, or receive unauthorized aid in this examination.

Signature of Student ____________

OFFICIAL USE ONLY

Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Total
(6)	(10)	(8)	(10)	(10)	(10)	(10)	(14)	(10)	(12)	(10)	(110)

(1) (10 points) Indicate whether each statement is True or False.
You do not need to justify your answers to this question.

(a) Every language over $\{0, 1\}$ is countable, and there are uncountably many languages over $\{0, 1\}$.

True / False

(b) The stack of a PDA can only contain symbols from the input alphabet.

True / False

(c) The Turing Test was formulated by Alan Turing to test whether any Turing Machine M accepts a string w.

True / False

(d) The Halting Problem is undecidable and Turing-recognizable.

True / False

(e) A language is recognizable by a DFA if and only if it is recognizable by an PDA.

True / False

(f) The set $\{\langle M \rangle \mid M$ is a Turing machine over $\{0, 1\}\}$ is uncountable.

True / False

(g) $P \subseteq NP$.

True / False / Unknown

I hope P = NP. Heh-heh!

(h) For all languages A over the alphabet $\{0, 1\}$, $A \leqslant_m \{0, 1\}^*$ if and only if $A = \{0, 1\}^*$.

True / False

(i) The language $\{w\ w\ w | w \in \{0, 1\}^*\}$ is not context-free.

True / False

(j) Turing recognizable languages are closed under union, intersection, and complement.

True / False

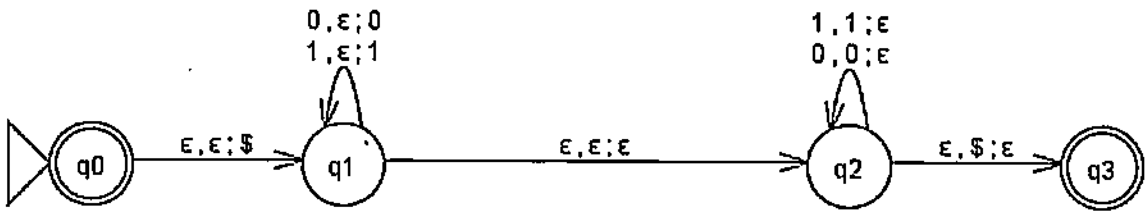
Name: ANTRIKSH Y

(2) (8 points) You do not need to justify your answers to this question.

(a) Alphabet: $\{0,1\}$. Consider the regular expression $R = 0((1 \cup 10)^*\underline{1 \cup 00})^*$.

Is $010100 \in L(R)$? Yes or No or Not enough information

(b) Alphabet: $\{0,1\}$. Consider the PDA D:



Is $0101 \in L(D)$? Yes or No or Not enough information

(c) Alphabet: $\{0,1\}$. Consider the TMs M_1 and M_2 (both over $\{0,1\}$) given as follows:

M_1 = " On input $\langle M \rangle$, where M is a TM,
1. Halt and accept."

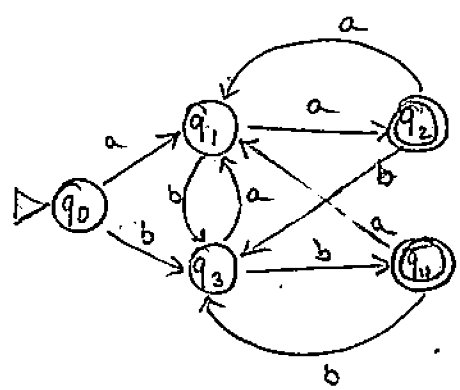
M_2 = " On input $\langle M \rangle$, where M is a TM,
1. Halt and reject."

Is $\langle M_1 \rangle \in L(M_2)$? Yes or No or Not enough information

Is $\langle M_2 \rangle \in L(M_1)$? Yes or No or Not enough information

(3) (10 points) Consider the following language
$T = \{ w \in \{a, b\}^* \mid w$ contains an even number of b's and $\mid w \mid$ is even $\}$

(a) Design the state diagram of a DFA which recognizes T. Use no more than 6 states. Label/name each state.



(b) Give an example of a string of length 4 not in T: $\boxed{a\,a\,b\,a}$

List the sequence of states visited by the DFA you designed when running on this string.

$$q_0 \xrightarrow{a} q_1 \xrightarrow{a} q_2 \xrightarrow{b} q_3 \xrightarrow{a} q_1$$

(c) Give an example of a string of length 4 in T: $\boxed{b\,b\,b\,b}$

List the sequence of states visited by the DFA you designed when running on this string.

$$q_0 \xrightarrow{b} q_3 \xrightarrow{b} q_4 \xrightarrow{b} q_3 \xrightarrow{b} q_4$$

(accept)

Name: ANTRIKSH Y

(4) (10 points) **Fill in** the blanks to complete the following proof.

Theorem: The language

$$E = \{a^j b^k c^\ell \mid j, k, \ell > 0 \text{ and } j + k > \ell\}$$

over the alphabet $\{a, b, c\}$ is not regular.

Proof: Assume (towards a contradiction), that E is regular. Therefore, the Pumping Lemma applies to E. Let p be a pumping length of E.

Choose s to be the string $s =$ $a^{\frac{p}{2}} b^{\frac{p}{2}+1} c^{p}$

The Pumping Lemma guarantees that, since $s \in E$ and $|s| \geqslant p$, s can be divided into parts x, y, z such that $s = xyz$ and for any $i \geqslant 0$, $xy^iz \in E$, and $|y| > 0$, and $|xy| \leqslant p$.

But, if we let $i = 0$, then xy^iz is not in E because

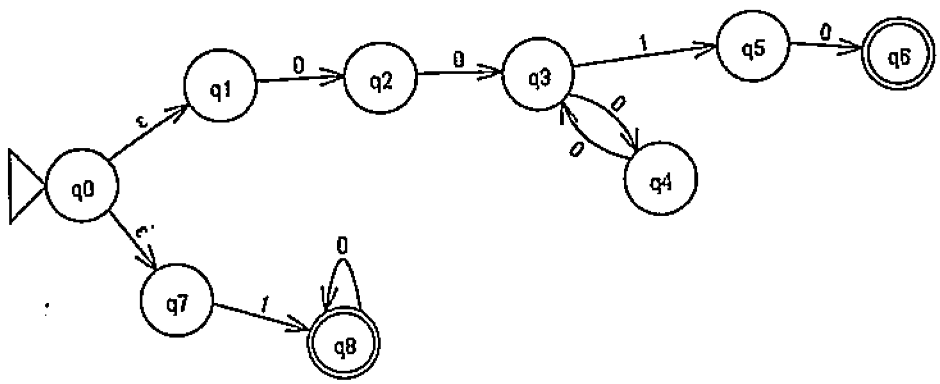
xy will be contained within the $a^{\frac{p}{2}} b^{\frac{p}{2}}$ part of the string at most. Minimum possible $|y|$ is 1. If y is removed, $|xy|$ will become less than p. We have chosen the string to contain exactly p instances of c, which will exceed $|xy|$, no matter what length of y or xy is chosen, thus breaking the rules of the language, E.

[Before the string is pumped, it has $j + k = \ell + 1$, so that it follows the rules of E]

This is a contradiction with the definition of p as a pumping length. Therefore, the assumption must have been false and E is not regular.

(5) (10 points)

(a) Consider the NFA N with alphabet $\{0,1\}$ given below.



Describe the language of N, $L(N)$, clearly and concisely, using a regular expression or an English description:

$(10^*) \cup (0\,0(00)^*1\,0)$

(b) Consider the CFG below over the alphabet (set of terminals) $\{0,1\}$ and with start variable S.
L is the language generated by the grammar

$$S \to S\,0\,S\,0\,S \mid 1\,S \mid \varepsilon$$

Does $L = \{w \in \{0,1\}^* \mid w \text{ contains an even number of 0's and an odd number of 1's}\}$.

Yes or No

If chose "No", give a counterexample and justify it.

$S \to S0S0S \to \varepsilon 01S01S \to 01\varepsilon 01\varepsilon \to 0101$

$S \to 1S \to 11S \to 11\varepsilon \to 11$

Above examples show that this CFG has nothing enforcing an odd number of 1s.
S can also be replaced with ε, creating an empty string with 0 (even) 1s.

Name: ANTRIKSH Y

(6) (10 points) Prove that if A is a regular language over the alphabet $\{a, b\}$, then the language

$B = \{w \in A \mid w$ has odd length and w has an even number of a's$\}$ is also regular.

Given: A is regular.

To prove: B is regular.

Proof:

Consider language:

$C = \{v \in \{a,b\}^* \mid v$ has odd length and an even number of a's$\}$

C is regular b/c there is an NFA for it:

Since $B = C \cup A$ and because regular languages are closed under union, B is a regular language.

(7) (14 points) Fill in the missing pieces in this closure proof:

Theorem: For every Turing-recognizable language A over $\{a, b\}$, the language V_A over $\{a, b\}$

$V_A = \{ a^n w \mid n \geqslant 0 \text{ and } w \in A\}$

is also Turing-recognizable.

Note: you may not *use any general (closure) theorems in this proof.*

Proof: Let A be a Turing-recognizable language over $\{a, b\}$. By definition, this means there is a TM M such that $L(M) = A$. Give a high-level description of and proof of correctness for a new TM M_V that will recognize V_A.

High-level description of M_V:

M_V takes string input x and checks if it starts with a. If it does, it splits the "a" part (incl. left and to right of all a's) of x nondeterministically (every possible split) and runs the second half through M. If it accepts, it accepts, otherwise rejects.
If x does not start with an a, it runs M with x as input and accepts if M accepts, otherwise rejects.
[some text outside box]

Proof that $L(M_V) = V_A$:

If string starts with an "a" or a series of "a"s V_A does not know where (a possible) substring w starts. This is why it runs x through M after removing 0 to even all "a"s. If it does not start with an "a", M_V just needs to make sure that x is accepted by M (because n is allowed to be 0). If any of the w strings above are not in A, they automatically won't be in V_A because M_V rejects when M rejects.

(8) (10 points)

(a) Define A_{TM} below:

A_{TM}= $\{\langle M, w\rangle \mid \langle M\rangle$ is TM encoding, w is string and M acc. $w\}$

(b) Let $L_{01} = \{\langle M\rangle \mid L(M)$ contains the string $01\}$, so L_{01} contains exactly all the encodings of Turing machines whose language contains the string 01.

Show that L_{01} is undecidable by showing that if there is a decider R for L_{01}, then there is a decider M for A_{TM}. If your decider M includes the construction of a TM X, give a full high level description of the TM X.

Note: You do not need to include a proof of correctness for your construction.

Construction of M from R:

M = "On input ⟨M, w⟩:
1. Construct a TM X:
"On input x:
1. If x is 01, run M on w.
2. If M acc., acc. otherwise reject."
2. Run ⟨X⟩ as input through R.
3. If acc., acc., otherwise reject.

(9) (12 points) Answer Yes or No for each of the properties with each set definition. You do not need to justify your answers for this question.

Recall: A set is co-Turing recognizable if its complement is Turing recognizable.

(a) $\{\langle N \rangle \mid N$ is an NFA that accepts input $\varepsilon\}$.

Decidable — Yes or No

Turing-recognizable — Yes or No

(b) $\{\langle M \rangle \mid M$ is a TM and M halts on input $\varepsilon\}$.

Turing-recognizable — Yes or No

Co-Turing-recognizable — Yes or No

(c) $\{\langle x, y, z \rangle \mid x, y, z$ are positive integers and z is the greatest common divisor of x and $y\}$.

Decidable — Yes or No

In P — Yes or No

(d) $\{a^n b^{2n} c^m \mid n, m > 0\}$.

Regular — Yes or No

Context free — Yes or No

(e) $\{a^n b^{2n} c^{3n} \mid n > 0\}$.

Context free — Yes or No

In P — Yes or No

(f) $\{\langle G, s, t \rangle \mid G$ is a directed finite graph, and there is a path in G from s to $t\}$

In P — Yes or No

In NP — Yes or No

(10) (10 points) Consider the set of finite Boolean expressions E over the operations $\wedge$ (and), $\vee$ (or), and $\sim$ (not). A Boolean expression is **satisfiable** if there is some assignment of 0's and 1's to its variables that make E evaluate to 1 (representing TRUE), using the laws of Boolean logic.

Consider the following high-level description of a Turing machine M.

M = "On input $\langle E \rangle$ where E is a Boolean expression:

1. For all possible assignments of 0 and 1 to each distinct variable in E
   a. If E evaluates to 1, accept.
   b. Otherwise, check next assignment.
2. If have not yet accepted, reject."

The worst-case running time of M has a polynomial bound: **True** or **False**

Give a justification of your answer:

For every variable, M has to assign a 0 and a 1, and each time, it has to try every possible combination of 0s and 1s for every other variable, through brute force. This is an exponential time algorithm in the worst case, and therefore does not have a polynomial bound.

Extra page