Illinois Institute of Technology
Department of Computer Science

Third Examination

CS 330 Discrete Mathematics
Spring, 2015

11:25am–12:40pm, Wednesday, April 29, 2015
113 Stuart Building

Print your name and student ID, *neatly* in the space provided below; print your name at the upper right corner of *every* page. Please print legibly.

Name:	Rachael Affenit
Student ID:	A20303915

This is an *open book* exam. You are permitted to use the textbook (hardcopy only), hard copies of any class handouts, anything posted on the web page, any of your own assignments, and anything in your own handwriting. Foreign students may use a hardcopy dictionary. *Nothing else is permitted*: No calculators, laptops, cell phones, Ipads, Ipods, communicators, GPSes, etc.!

Do all five problems in this booklet. *All problems are equally weighted, so do not spend too much time on any one question.*

Show your work! You will not get partial credit if the grader cannot figure out how you arrived at your answer.

Question	Points	Score	Grader
1	20	8	
2	20	5	
3	20	20	
4	20	2	
5	20	2	
Total	100	37	

8

1. **Breadth First Search of Graphs**

In the lecture of April 1 (see bottom of page 3 of the April 1–6 notes) Professor Reingold stated the following property of the values $d[v]$ assigned by BFS:

> Suppose that during the execution of BFS on a graph $G = \langle V, E \rangle$, the queue Q contains the vertices $\langle v_1, v_2, \ldots, v_r \rangle$, where v_1 is the head of Q and v_r is the tail. Then, $d[v_r] \leq d[v_1] + 1$ and $d[v_i] \leq d[v_{i+1}]$ for $i = 1, 2, \ldots, r-1$.

shortest path from s to v

Use induction on the number of queue operations to prove this property.

base case: $i = 2$, $Q = \{v_1, v_2, v_3\}$

$d[v_3] \leq d[v_1] + 1$

Assuming that $s = v_3$, $d[v_3] = 0$.

v_1 is either adjacent to v_3 or v_2, so $d[v_1] = 1, 2$

In the worst case, $0 \leq 1 + 1 = 2$ ✓

$d[v_2] \leq d[v_3]$

$d[v_3] = 0$

$d[v_2] = 0, 1$ because it could be at same dist or below the previous vertex

$0 \leq 0$ ✓

inductive step: $i > 2$ $Q = \{v_1, v_2, \ldots v_i, \ldots v_r\}$ → Assume all true for i

$d[v_r] \leq d[v_1] + 1$

since $s = v_r$, $d[v_r] = 0$

$0 \leq d[v_1] + 1$ ✓ since all dist values are positive

$d[v_{i+1}] \leq d[v_{i+2}]$

$d[v_{i+1}] = d[v_{i+2}]$

$d[v_{i-1}] \leq d[v_i]$

=

$d[v_{i+1}] \geq d[v_i]$

v_{i+1} is either on the same level as v_{i+2} or just below it.

$\therefore d[v_{i+1}] = d[v_{i+2}]$, or $d[v_{i+2}] + 1$

take best case

2. **Depth First Search of Graphs**

Describe and analyze a DFS algorithm to determine if an undirected graph is 2-colorable, that is, if the vertices can be colored with 2 colors so that no two adjacent vertices have the same color. Your algorithm should either produce the 2-coloring or report the vertices of an odd-length cycle (the only thing that renders a graph not 2-colorable; why?). You will get full credit for an $O(|V| + |E|)$ algorithm and partial credit for slower algorithms.

5

Rather than marking each vertex **visited**, mark each edge visited in DFS. Let this new algorithm be DFS-EDGE. This algorithm also applies the opposite color to the second vertex in an edge if the first is already colored. The initial vertex is colored randomly.

ODD CYCLE SEARCH

1. choose any vertex in graph G, and run DFS-EDGE until we need to backtrack. When we do, a circuit is found. Store this circuit as CURRENT-CIRCUIT.

2. if CURRENT-CIRCUIT has an odd number of verticies, stop traversal and return those verticies.

3. pick the first vertex along the CURRENT-CIRCUIT that has an outgoing edge that is not visited. Run DFS-EDGE starting with that vertex until we need to backtrack. At this point, another circuit has been found.

4. replace the value of CURRENT-CIRCUIT with the value of this new circuit.

5. Repeat 2, 3, and 4 until all edges are visited.

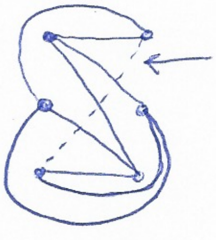
6. Return 2-colored graph with DFS-EDGE assigned colors.

3. **Graph Structure**

In the lecture on April 13, Professor Reingold proved that in a planar graph $|E| \leq 3|V| - 6$ (see the middle of page 2 of the notes, or Corollary 1 on page 722 of Rosen). Is the *converse* true? That is, can we conclude that the graph is planar just because $|E| \leq 3|V| - 6$? Prove your answer.

No, you cannot conclude planarity from this inequality.

Consider the case of $K_{3,3}$.



$K_{3,3}$ by definition is non-planar, as it contains at least 1 crossed edge by design.

However, it has 9 edges and 6 vertices.
If we apply our inequality, $9 \leq 3(6) - 6 \leq 18 - 6 \leq 12$, which is true.

If this inequality was able to prove the planarity of a graph, then $K_{3,3}$ would be planar.

We know this is not true, so by contradiction, you cannot conclude planarity from $|E| \leq 3|V| - 6$.

4. **Regular Languages**

State, with proof, whether each of the following languages is regular:

(a) Odd numbers, written in binary, read least significant bit first ~~(that is, from left to right)~~.

2

Set of numbers divisible by 2 when 1 is added
The set of #'s divisible by 2 is irregular, so our set is also irregular.

−4 X

(b) Numbers not divisible by 3, written in binary, read most significant bit first (that is, from right to left).

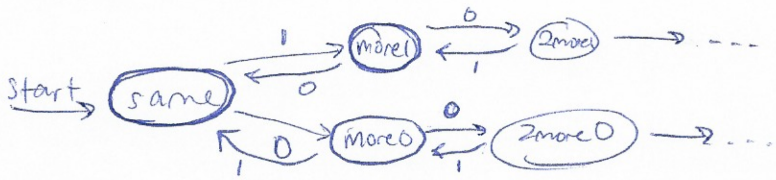
This is an infinite set, and cannot be represented as an FSM.
∴ it is irregular.

X −4

Also set divisible by 3 is irregular, so since regularity is closed under complementation, this set is irregular.

(c) The set of strings of zeros and ones containing at the same number of zeros and ones.

A Finite state machine cannot be constructed, so this language is irregular.



−3

(d) The set $\{0^n1^m | n \text{ and } m \text{ are congruent modulo } 3\}$.

$\{0^n1^m | n=m\}$ was shown to be irregular in class.
$\{0^n1^m | n=m, \%3\}$ is ~~a~~ an infinite subset of this, so it is also irregular.

X −4

(e) The set $\{0^n1^m | n > 2m\}$.

This set is a subset of $\{0^n1^m | n \neq m\}$.
$\underbrace{\{0^n1^m | n \neq m\}}_{A} \cup \underbrace{\{0^n1^m | n=m\}}_{B} = \underbrace{\{0^*1^*\}}_{U}$ $B = U - A$

−3

B was shown irregular in class, and U must be regular.
∴ A cannot be regular, so our infinite set cannot be regular.

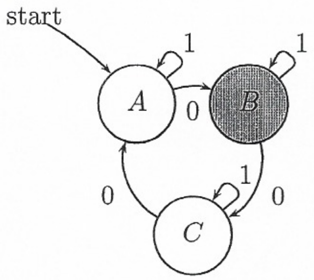
X

$\boxed{\{0^n1^m | n \geq 2m\} \text{ is irregular}}$

2

5. **Finite State Machines**

Consider the following finite state machine in which state A is the starting state and state B (shaded) is the only accepting state:



(a) Describe clearly and succinctly *in words* the language recognized.

+2 (b) Construct a regular expression for that language.

a) The language recognized is any string of 0's and 1's of any length in any order that are finite.

The accepted strings are strings of 0's & 1's ~~that are~~ finite of any length where the final state is the accepted state

b) $R_2^3 = R_2^2 + R_2^2 (R_9^2)^* R_6^2$ where $R_2^0 = \{x \in \{0,1\} \mid \delta(1, \{0,1\}) = 2\}$