CS109B Final Examination

Monday, March. 18, 8:30–11:30 am

Directions

The exam is *open book*; any written materials may be used. Answer all 16 questions on the exam paper itself. The total number of points is 300. You may make additional comments at the bottoms of pages or at the end if you feel the need to clarify or justify your answers. Do not forget to **sign the pledge** below.

I acknowledge and accept the honor code.

____

Print your name here: ____

1		5		9		13	
2		6		10		14	
3		7		11		15	
4		8		12		16	

Problem 1: (20 points) Suppose that a graph G is as drawn below.

2

1

a b

1 1 2

c d

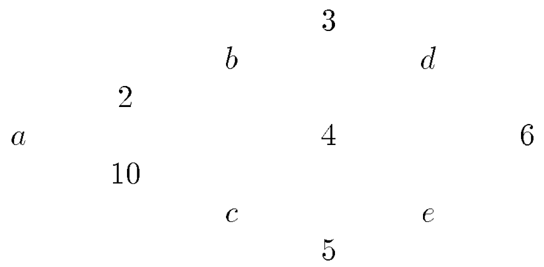
2

a) What is the weight of a minimum-weight spanning tree for G; that is, the sum of the labels of the selected edges?______________

b) How many different minimum-weight spanning trees are there for G?______________

c) Draw two different minimum-weight spanning trees below.

Problem 2: (15 points) Consider the graph H below.



a) If we apply Dijkstra's algorithm to H with source a, what is the order in which the five nodes are added to the "settled" region?

____

b) Give for each node its minimum distance from the source: a:________ b:________ c:________ d:________ e:________

Problem 3: (15 points) In the matrix below, give the initial distance matrix when Floyd's algorithm is applied to graph H from Question (2). This matrix is the one that holds before any pivoting is done.

	a	b	c	d	e
a					
b					
c					
d					
e					

Now, show the distance matrix after pivoting on node c (and no other nodes).

	a	b	c	d	e
a					
b					
c					
d					
e					

Problem 4: (15 points) Draw a depth-first spanning tree for the graph H of Question (2), starting at node a. Indicate the postorder numbers of your nodes.

Now, find another depth-first spanning forest for H that has the maximum number of trees. Again, indicate the postorder numbers of the nodes.

Problem 5: (30 points) Write regular expressions for the following languages.

a) The set of strings of a's, b's, and c's that begin with a, end with b, and have at least one c.

b) The language consisting of nonempty strings of alternating `a`'s and `b`'s. Examples of such strings are `a`, `b`, `ab`, `ba`, `aba`, `bab`, and so on.

c) The set of strings of a's, b's, and c's that have at least one a and at least one b.

Problem 6: (20 points) Consider the following finite automaton.

$$\begin{array}{ccccc} & b & & a & \\ \rightarrow\ 1 & & 2 & & 3 \\ & a & & b & \end{array}$$

a) We can begin to find a regular expression for its language by removing state 2 by "pivoting." Draw the resulting automaton below, showing the regular expression labeling each arc.

b) From your answer to (a), give a regular expression for the set of strings that label paths from state 1 to state 3 in the original automaton.

Problem 7: (20 points) Convert the following nondeterministic finite automaton

b b

2

a

$\rightarrow$ 1 a

b

3

a

into a deterministic finite automaton, below, by using the subset construction.

Problem 8: (20 points) In the following problem you *must* use the regular-expression-to-automaton conversion algorithm described in Section 10.8 of the notes. We shall convert the regular expression $\mathbf{a}^* \mid \mathbf{b}^*$.

a) Show the expression tree for this regular expression:

b) Convert the expression to an automaton with ϵ-transitions:

c) Eliminate ϵ-transitions from your answer to (b):

Problem 9: (20 points) The following grammar generates palindromes (strings that read the same forward as backward) with characters `a` and `b`, but with the center marked with either a `c` or a `d`. Examples of strings in the language of the syntactic category $< Pal >$ are `abaacaaba` and `bbdbb`.

(1) $< Pal > \rightarrow \mathbf{a} < Pal > \mathbf{a}$
(2) $< Pal > \rightarrow \mathbf{b} < Pal > \mathbf{b}$
(3) $< Pal > \rightarrow < Center >$
(4) $< Center > \rightarrow \mathbf{c}$
(5) $< Center > \rightarrow \mathbf{d}$

a) Put a Y after those strings that are in the language of $< Pal >$, and a N after those that are not (score = 2 for each right minus 2 for each wrong). `baab`________ `abccba`________ `d`________ `ababa`________

b) Describe the language of the syntactic category $< Center >$

c) In the space below, draw a parse tree for the string `abbcbba`.

Problem 10: (20 points) The grammar of Question 9 can be parsed by a table-driven parser of the type described in Section 11.7.

a) How does one decide which production for $< Pal >$ to use?

---

b) How does one decide which production for $< Center >$ to use?

---

c) Draw the parsing table in the space below.

Problem 11: (15 points) In the lines on the next page, design a grammar for the set of strings that consist of zero or more `a`'s followed by at least as many `b`'s. Examples of strings in this language are ϵ, `ab`, `aabbbbb`, and `bbbb`. Strings not in the language include `aaabb` (more `a`'s than `b`'s) and `aba` (all strings in the language are also in the language of the regular expression $\mathbf{a}^*\mathbf{b}^*$. *Hint*: We covered in class a grammar for strings of the form $\mathtt{0}^n\mathtt{1}^n$, from which a simple modification should give you a grammar for strings (including ϵ) with an equal number of `a`'s and `b`'s. From this grammar, can you arrange to allow for more `b`'s than `a`'s?

Explain what the language of each of your syntactic categories is

Problem 12: (20 points) Transform the logical expression $ab + \bar{a}\bar{b}$ into the expression $(a + \bar{b})(b + \bar{a})$, using the algebraic laws from Section 12.8. Show each step on a line, and justify it, either by citing the number of an algebraic law from Section 12.8 or giving its name. If several changes are justified by the same rule, you may combine them on a single line. Also, you may use the associative and commutative laws for `AND` and `OR` without stating them explicitly and without using a line to show the transformations.

Problem 13: (20 points) Below is a Karnaugh map.

		rs			
		00	01	11	10
pq	00	0	1	1	1
	01	1	1	1	1
	11	1	1	0	1
	10	1	1	1	1

It represents the function that is `TRUE` unless all or none of p, q, r, and s are `TRUE`.

a) List the prime implicants (do not list any nonprime implicants), giving the AND-of-literals form for each.

---

b) Only a subset of the prime implicants are needed to cover the map, that is, to include all the points with 1's. Give a minimal such set below.

Problem 14: (15 points) Let E be the logical expression

$$\Big(\big(p \to (q + r)\big)\big(p \to (q + \bar{r})\big)\Big) \equiv (p \to q)$$

a) In the space below, draw the expression tree for E (called "syntax tree" in the notes).

b) Construct the truth table for all the subexpressions of E, including E itself. You may construct the various columns of the truth table in the space provided below. You should number the nodes of your tree from (a) and use these numbers to head the columns of the truth table.

p	q	r	
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Problem 15: (15 points) Below is a sequence of steps whereby we turn the expression $\big((p \to (q+r)\big)\big(p \to (q+\bar{r})\big)$ into the expression $p \to q$, thus showing their equivalence. For each step, give a justification by citing an algebraic law from Section 12.8 by number, or giving the name of the transformation. Uses of the commutative or associative laws for `AND` and `OR` may be omitted.

$\big((p \to (q+r)\big)\big(p \to (q+\bar{r})\big)$ Start

$(\bar{p}+q+r)\big(p \to (q+\bar{r})\big)$ ____________________

$(\bar{p}+q+r)(\bar{p}+q+\bar{r})$ ____________________

$\bar{p}+q+r\bar{r}$ ____________________

$\bar{p}+q$ ____________________

$p \to q$ ____________________

Problem 16: (20 points) Now, we shall prove the implication

$$\Big(\big((p \to (q+r)\big)\big(p \to (q+\bar{r})\big)\Big) \to (p \to q)$$

by resolution. Note that this implication is "half" of the equivalence with which we have been dealing in the two previous questions, but is not the same tautology.

a) Convert the hypotheses $\big((p \to (q+r)\big)\big(p \to (q+\bar{r})\big)$ into a product of clauses.

b) Convert the *negation* of the conclusion $p \rightarrow q$ into a product of clauses. Explain your reasoning by showing how NOT$(p \rightarrow q)$ is converted into a product of clauses by a series of algebraic laws from Section 12.8 (you need not name the laws).

c) Prove the conclusion from the hypotheses by starting with your clauses from (a) and (b) and deriving the empty clause by a sequence of resolutions. Show each new clause on a line below, and indicate the clauses that are resolved to get it.

New Clause	Derived from Clauses: