CS109B Final Examination

Monday, June 12, 1995, 12:15–3:15PM

Directions

The exam is *open book*; any written materials may be used. Answer all 12 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.

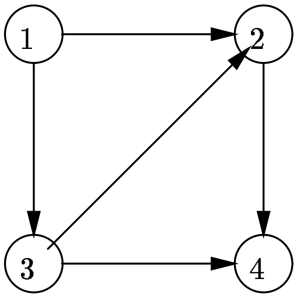
I acknowledge and accept the honor code.

____

Print your name here: ____

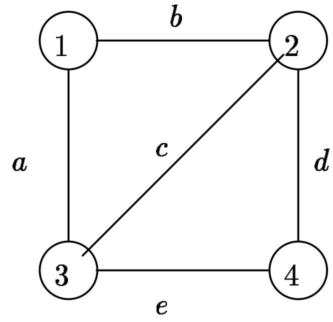
1	2	3	4
5	6	7	8
9	10	11	12

Problem 1: (15 points) Suppose we perform a depth-first search of the graph below:



with node 1 as the root. Show all the depth-first search trees that may result. You do not need to show non-tree arcs.

Problem 2: (40 points) Consider the following graph, where the letters a, b, c, d, e may be thought of both as the names of the edges and as weights (labels) on the edges. When they are weights, you may assume they are distinct.



a) If the weights a, b, c, d, e on the edges could be assigned arbitrary distinct values, how may different minimal-weight spanning trees (MWST's) could result? ______

(Optional) For partial credit, list the sets of edges forming the MWST's here.

---

b) Of the 120 possible orders of the weights a, b, c, d, e, how many will yield the MWST $\{a, c, e\}$? ______ *Hint*: This may be trickier than it looks. Note that c, e, d, a, b is one such order, for example.

(Optional) For partial credit, describe these orders here. The following notation may be useful: Let $\{u, v\}\{x, y, z\}$ represent all orders where u and v precede x, y, and z, but u and v may appear in any relative order; likewise x, y, and z. This notation extends to groups of different size and more than two groups.

---

c) How many orders of a, b, c, d, e will yield the MWST $\{a, b, d\}$? ______

(Optional) For partial credit, describe these orders here.

---

d) Suppose we run Dijkstra's algorithm with source node 1 on the above graph (assume edges are arcs in both directions with the same weight). What inequalities must hold among a, b, c, d, e so that the order in which nodes are settled is $1, 2, 3, 4$?

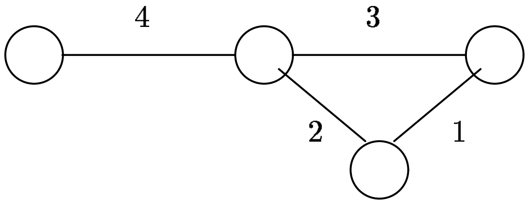
---

e) The above graph is not bipartite. Which edges, if removed (and the other 4 edges remain) will make the graph bipartite?

Problem 3: (30 points) Given an undirected graph G, we can construct its *dual* graph D as follows.

i. D has a node for each edge of G.

ii. There is an edge $\{e, f\}$ in D if and only if the edges e and f of G have a node in common. For example, if G is this graph:



then D is the graph of Problem 2 (ignoring the edge labels a, b, c, d, e on D).

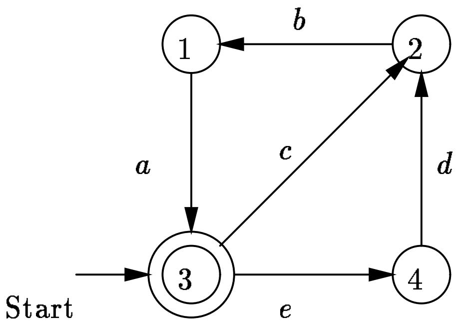
a) Below, draw the dual of the graph of Problem 2 (do *not* draw the dual of the graph above).

b) Sometimes, the dual of a graph G is G itself (renaming nodes and edges, if necessary). K_3 is an example of such a graph. Below, draw another graph G that is its own dual. *Hint*: Notice that a self-dual graph must have exactly as many nodes as edges.

c) For any $n \geq 1$, the dual of the complete graph K_n has $n(n-1)(n-2)/2$ edges. Briefly explain why.

d) If G has a node of degree d, what can you say about the clique number (maximum clique size) of its dual D?

Problem 4: (20 points) There's that gosh darn graph again; now it's a finite automaton.



We can find a regular expression for this automaton by eliminating states until only its start/accepting state 3 remains (that is, use the reduction process of Section 10.9).

a) Below, draw the automata (with arcs labeled by regular expressions) that result from eliminating states 4, 1, and 2, in that order.

After eliminating 4 | After eliminating 1 | After eliminating 2

b) What is the regular expression for this automaton?

Problem 5: (20 points) Write regular expressions for the following languages whose strings consist of a's, b's, and c's.

a) The set of all strings that begin with a and end with either b or c.

---

b) The set of strings that begin and end with a.

---

c) The set of strings such that every a is followed immediately by a b, and every b is followed immediately by a c. Examples of such strings are `ccc` and `cabcbc`.

---

d) The set of strings without two consecutive a's.

Problem 6: (25 points) In each line below, there are two *or more* expressions that are equivalent, while the remaining expressions on that line are equivalent to nothing else on the line. Circle all the equivalent expressions on each line. For example, given the line

(sample) $p + q$ $\quad pq$ $\quad q + p$ $\quad \bar{p} + q$

we would circle the first and third, but not the other two.

(a) $p \to q$ $\quad q \to p$ $\quad \bar{q} \to p$ $\quad \bar{q} \to \bar{p}$

(b) $\bar{p} + \bar{q}$ $\quad p + q$ $\quad \bar{p} \to q$ $\quad \bar{q} \to p$

(c) p NAND q $\quad p$ NOR q $\quad$ NOT(p NAND q) $\quad$ NOT(p NOR q) $\quad pq$

(d) $p + q$ $\quad p + 1$ $\quad q + 1$ $\quad p + 0$ $\quad q + 0$

(e) $pq \to r$ $\quad p\bar{q} \to r$ $\quad pq \to \bar{r}$ $\quad q \to (\bar{r} \to p)$ $\quad p \to (q \to r)$

Problem 7: (25 points) Consider the grammar below, with start symbol $< A >$:

(1) $< A > \rightarrow < B > < A >$
(2) $< A > \rightarrow \epsilon$
(3) $< B > \rightarrow \texttt{0} < C >$
(4) $< C > \rightarrow \texttt{1}$
(5) $< C > \rightarrow \epsilon$

a) Each syntactic category happens to have a language that is a regular expression. Below, write regular expressions for each of these languages:

For $L(<C>)$: ____________________

For $L(<B>)$: ____________________

For $L(<A>)$: ____________________

b) In the box below, give the parsing table for this grammar.

	0	1	ENDM
$<A>$			
$<B>$			
$<C>$			

Problem 8: (15 points) Design a grammar such that the language of one of its syntactic categories (indicate which) is the set of strings of 0's and 1's having odd length. Examples of strings in the language are `0`, `100`, and `10110`.

Problem 9: (30 points) We would like to prove that from hypotheses $p \oplus q$ and $q \oplus r$ we may conclude $p \equiv r$, by resolution and proof by contradiction. Our first task is to convert the hypotheses and negation of conclusion into products of clauses.

a) $p \oplus q$ is equivalent to the product of clauses $(p+q)(\bar{p}+\bar{q})$. Below is a sequence of steps that turns the `AND`/`OR`/`NOT` definition of $p \oplus q$, which is $p\bar{q} + \bar{p}q$, into this product of clauses. At each step, one equivalence from the group of laws found in Sects. 12.8–9 of FCS is used (sometimes twice). For each step, give the name or number of the law used. For example, if we went from $0 + x$ to x, then for the latter step we would put "0 is the identity for `OR`" or "12.11." The fact that 12.11 is stated in a commutative variant, $p + 0 \equiv p$, is irrelevant. This law, like many in the book, is intended to represent several variants derived by commutativity of `OR`, `AND`, or $\equiv$.

$p\bar{q} + \bar{p}q$ Beginning

$(p\bar{q} + \bar{p})(p\bar{q} + q)$ ______________________________

$(p + \bar{p})(\bar{q} + \bar{p})(p + q)(\bar{q} + q)$ ______________________________

$(1)(\bar{q} + \bar{p})(p + q)(1)$ ______________________________

$(\bar{q} + \bar{p})(p + q)$ ______________________________

$(p + q)(\bar{q} + \bar{p})$ ______________________________

$(p + q)(\bar{p} + \bar{q})$ ______________________________

b) Give the result of converting $q \oplus r$ to a product of clauses.

____

c) Give the result of converting NOT$(p \equiv r)$ to a product of clauses.

____

d) Using the clauses $(p + q)$ and $(\bar{p} + \bar{q})$ from (a), plus your answers to (b) and (c), derive 0 (FALSE) by resolution. On each line below give a new clause and the two clauses from which the new clause is derived.

New clause ______________ from clauses ______________ and ______________

New clause ______________ from clauses ______________ and ______________

New clause ______________ from clauses ______________ and ______________

New clause ______________ from clauses ______________ and ______________

New clause ______________ from clauses ______________ and ______________

New clause ______________ from clauses ______________ and ______________

Problem 10: (25 points) We would like to design a "divide-and-conquer" circuit that takes n inputs and determines the number of 1's among these n inputs, modulo 3. That is, let m be the number of 1's among the inputs, and let r be the remainder when m is divided by 3. The circuit will have three outputs: a, b, and c; a will be 1 iff $r = 0$, b will be 1 iff $r = 1$, and c is 1 iff $r = 2$.

a) Design below (on the next page) a basis-case circuit that takes one input and produces the three outputs described above. *Hint*: an output can be attached to a constant, 0 or 1, as well as to a circuit input or the output of a gate. For example, an output attached to 1 will always have the value 1 (TRUE).

b) Design below the inductive-case circuit for $2n$ inputs. The outputs from the first n inputs are $x0$, $x1$, and $x2$, representing remainders of 0, 1, and 2, respectively, and $y0$, $y1$, and $y2$ are the outputs from the second n inputs. Your circuit must produce outputs $z0$, $z1$, and $z2$ from these 6 inputs. You may use 3-input gates if you like.

c) Write a recurrence relation for $D(n)$, the delay of the circuit when there are n inputs (n may be assumed a power of 2).

---

d) Solve your recurrence.

Problem 11: (20 points) We would like to evaluate the predicate logic expression $(\forall X)(\exists Y)p(X,Y)$ for the interpretation I, where the domain is the integers and predicate p is interpreted as the function that is `TRUE` iff $X + Y = 10$.

a) To evaluate this expression, we have to evaluate subexpression $(\exists Y)p(X,Y)$ for a collection of interpretations I_i, where i can be any integer. Explain the difference between interpretations I_i and I.

b) To evaluate $(\exists Y)p(X,Y)$ for interpretation I_i, we need to evaluate subexpression $p(X,Y)$ for a collection of interpretations I_{ij}, where j can be any integer. Explain the difference between interpretations I_{ij} and I_i.

c) For what values of i and j is the value of $p(X,Y)$ `TRUE` for interpretation I_{ij}?

d) For what values of i is the value of $(\exists Y)p(X,Y)$ `TRUE` for interpretation I_i? Explain your reasoning.

e) What is the value of the full expression $(\forall X)(\exists Y)p(X,Y)$ for interpretation I?

Problem 12: (35 points) Answer the following five questions about ML.

a) Here is a structure definition.

```
structure Foo = struct
    fun f(x) = x+1;
    val i = 3;
end;
```

Below, show the signature of this structure `Foo`.

b) Below, write a statement defining `BARSIG` to be a signature that could be used in the following definition of structure `Bar`:

```
structure Bar: BARSIG = Foo;
```

so that on opening `Bar` the function `f` is available but not the integer `i`.

c) Here is a (sketch of a) functor `Fun` that takes as argument a structure `Stru` consisting of an integer i and a real r.

```
functor Fun(Stru: sig val i: int; val r: real; end) =
    struct (* some structure definition *) end;
```

Write the definition of a structure `S1` that could be the value of parameter `Stru` in an application of functor `Fun`.

d) Now write a statement in which the functor `Fun` of part (c) is applied to structure `S1` to produce a new structure `S2`.

e) Write a recursive function `make3(A,n)` that sets the first n elements of array A to 3. Your function must be a simple, two-pattern recursion and must not use a while-loop or any similar attempt to replace recursion by iteration. You may assume the structure `Array` has already been opened.