EECS 144 & 244, Fall 2010 **Lee, Roychowdhury, Seshia**

Midterm 2 Solutions

November 22, 2010

YOUR NAME: Oski Bear

Instructions:

This exam is open-book, open-notes. Please turn off and put away electronic devices such as cell phones, laptops, etc.

You have a total of 80 minutes. There are **4** questions worth a total of **160** points. The questions vary in difficulty, so if you get stuck on any question, it might help to leave it and try another one.

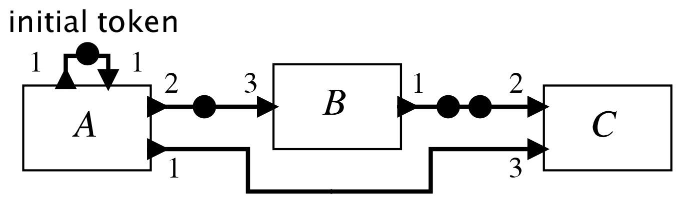
Answer each question in the space provided below the question. If you need more space, you can use the reverse side of that page. *You can use without proof any result proved in class but clearly state the result you are using.*

Do not turn this page until the instructor tells you to do so!

Problem 1	40
Problem 2	40
Problem 3	40
Problem 4	40
Total	160

Problem 1: (40 points)

Consider the dataflow graph shown below:



The production and consumption parameters are shown next to each port, and initial tokens on each connection are shown with dots.

1. Write down the balance equations in matrix form.

Solution: The balance equations are:

$$\begin{bmatrix} 0 & 0 & 0 \\ 2 & -3 & 0 \\ 0 & 1 & -2 \\ 1 & 0 & -3 \end{bmatrix} \begin{bmatrix} q_A \\ q_B \\ q_C \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Note that any permutation the rows of the matrix is valid; the rows may appear in any order. There is one row per connection between actors, and above, the connections are in top-to-bottom, left-to-right order, with the feedback connection being represented by the top row. Notice that the top row, which represents the self loop on actor A, is not really necessary; it represents a connection in which A produces one token and consumes one token, for a net production or consumption of zero tokens.

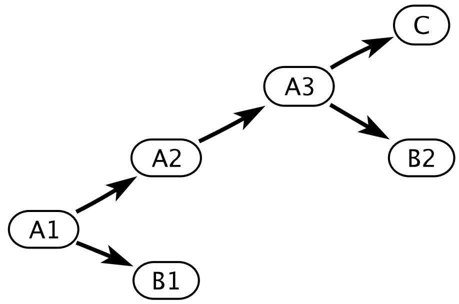
2. Is the graph consistent? If so, give the least positive integer solution.

Solution: The graph is consistent. The least positive solution is

$$q_A = 3, \quad q_B = 2, \quad q_C = 1.$$

3. Use symbolic execution to construct an acyclic precedence graph.

Solution: The APG is below:



4. Assume that the execution times are all one time unit. Construct the minimum makespan schedule for one iteration of the minimal balanced schedule. What is the makespan? How many processors are required to achieve this makespan?

Solution: A two-processor schedule with the first processor executing $A1, A2, A3, C$ and the second executing $B1, B2$ (starting at time 2, 4) achieves the minimum makespan of five.

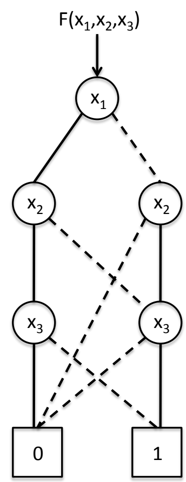
Problem 2: (40 points)

Consider the Boolean function F of three variables x_1, x_2, x_3 that evaluates to 1 if exactly two of the variables are 1 and evaluates to 0 otherwise.

1. Draw the reduced, ordered binary decision diagram (BDD) representing this function. State the number of non-terminal nodes in the BDD.

   Does the size of the BDD depend on the variable order you select? Why or why not?

Solution: The BDD has 5 nodes and is shown here.



No, the size of the BDD does not depend on the variable order. The structure of the BDD will be exactly the same for any variable ordering, since this is a symmetric function.

2. Write down the Boolean function in conjunctive normal form (CNF) using only variables x_1, x_2, x_3. Justify the correctness of your answer.

Solution: There are at least two solutions:

4 clauses:

$$(x_1 + x_2)(x_2 + x_3)(x_1 + x_3)(\bar{x_1} + \bar{x_2} + \bar{x_3})$$

The first three clauses encode the fact that at least one out of any two variables must be 1. The last clause encodes the constraint that at least one variable must be 0. Together these ensure the desired constraint.

5 clauses:

$$(x_1 + x_2 + x_3)(\bar{x_1} + x_2 + x_3)(\bar{x_2} + x_1 + x_3)(\bar{x_3} + x_2 + x_1)(\bar{x_1} + \bar{x_2} + \bar{x_3})$$

This is a more straightforward encoding — each clause is the negation of a minterm that does not have exactly two variables 1.

3. Suppose F was a function of *four* variables x_1, x_2, x_3, x_4 rather than three. Its defintion stays the same: it evaluates to 1 if exactly two of the variables are 1 and evaluates to 0 otherwise.

   How many CNF clauses (at least) would you need to write F only using variables x_1, x_2, x_3, x_4? Write the CNF representation down.

Solution: At least 8 CNF clauses are needed to encode $F(x_1, x_2, x_3, x_4)$:

$$(x_1 + x_2 + x_3)(x_1 + x_3 + x_4)(x_1 + x_2 + x_4)(\bar{x_2} + \bar{x_3} + \bar{x_4})$$
$$(\bar{x_1} + \bar{x_2} + \bar{x_3})(\bar{x_1} + \bar{x_3} + \bar{x_4})(\bar{x_1} + \bar{x_2} + \bar{x_4})(\bar{x_1} + x_2 + x_3 + x_4)$$

Problem 3: (40 points)

1. Let p and q be atomic propositions. Suppose that system M satisfies the LTL properties $\mathbf{G}\,\mathbf{F}\,p$ and $\mathbf{F}\,\mathbf{G}\,q$. Then, does M satisfy $\mathbf{G}\,\mathbf{F}\,(p \cdot q)$? Explain your answer.

   Yes, M satisfies $\mathbf{G}\,\mathbf{F}\,(p \cdot q)$.

   Since M satisfies $\mathbf{F}\,\mathbf{G}\,q$, for every execution $\pi = s_0, s_1, s_2, \ldots$ of M, there exists an i such that q holds for each s_j for $j \geq i$.

   Consider the suffix of π starting at s_i. Since π satisfies $\mathbf{G}\,\mathbf{F}\,p$, p occurs infinitely often in this suffix. Hence this suffix satisfies $\mathbf{G}\,\mathbf{F}\,(p \cdot q)$, which implies that so does π.

2. Say you work for a CAD company Cadopsys, where your manager Pointy Hair has come up with an idea for performing sequential equivalence checking of two circuits C_1 and C_2, and wants you to implement it. For each circuit, the output is the entire state of that circuit. The idea goes as follows:

   - First, check that C_1 and C_2 generate the same output in the initial state.
   - Separately compute the set of reachable states R_1 of C_1 and R_2 of C_2. Store R_1 and R_2 as BDDs.
   - If $R_1 = R_2$, conclude that C_1 and C_2 are equivalent. Otherwise, report that they are not equivalent.

   Will Pointy Hair's idea work? Justify your answer.

   No, it will not work. Simply having the same set of reachable states does not mean that the two circuits generate identical outputs at each cycle.

   A simple example is if C_1 is a counter that counts up from 0000 to 1111 and C_2 counts down from 0000 (with wraparound). Both will enumerate all 4-bit unsigned integers, hence have the same set of reachable states. But they do not generate the same states (outputs) at each cycle. For example, at cycle 1, C_1 generates 0001 as output, but C_2 generates 1111.

Problem 4: (40 points)

Figure 1 shows an RC tree with N segments. $u(t)$ is an input voltage source; $R = C = 1$.

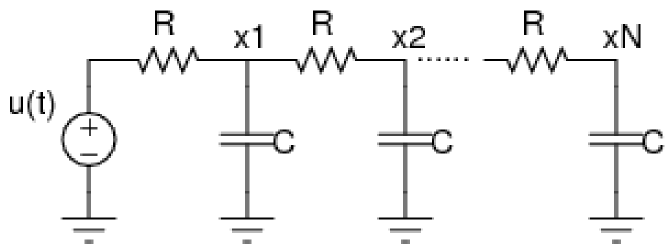


Figure 1: A simple RC circuit. (Problem 4)

1. Write out differential equations for the circuit in the matrix form $C\frac{d\vec{x}}{dt} + G\vec{x} + Bu = 0$, with the unknown variables being $x_1, x_2, \cdots, x_N$. (10 points)

   The equations are

$$
\begin{aligned}
C\frac{dx_1}{dt} + \frac{1}{R}(x_1 - u(t)) + \frac{1}{R}(x_1 - x_2) &= 0, \\
C\frac{dx_2}{dt} + \frac{1}{R}(x_2 - x_1) + \frac{1}{R}(x_2 - x_3) &= 0, \\
\vdots &= \vdots \\
C\frac{dx_{N-1}}{dt} + \frac{1}{R}(x_{N-1} - x_{N-2}) + \frac{1}{R}(x_{N-1} - x_N) &= 0, \\
C\frac{dx_N}{dt} + \frac{1}{R}(x_N - x_{N-1}) &= 0.
\end{aligned}
\tag{1}
$$

   In the matrix form, we have

$$
\frac{d}{dt}C\begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & \ddots & & \\ & & & 1 & \\ & & & & 1 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{N-1} \\ x_N \end{bmatrix} + \frac{1}{R}\begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 1 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{N-1} \\ x_N \end{bmatrix} + \frac{1}{R}\begin{bmatrix} -u(t) \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}.
\tag{2}
$$

2. Let $N = 2$, *i.e.*, let there be only two stages. Let the output be the node voltage at node 2. Derive the expression of the transfer function $H(j\omega) = \frac{X_2(j\omega)}{U(j\omega)}$.[1] (10 points)

[1] You may find the equality $\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{ad-bc}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$ useful.

When $N = 2$, the differential equation is

$$\frac{d}{dt}\begin{bmatrix} 1 & \\ & 1 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} -1 \\ 0 \end{bmatrix} u(t) = 0. \tag{3}$$

Since the output is x_2, we have the transfer function

$$H(j\omega) = c^T(j\omega C + G)^{-1}(-B), \tag{4}$$

where C, G, B are matrices defined in equation (3), and $c^T = [0, 1]$, selecting the second variable as the output.

Therefore,

$$\begin{aligned} H(j\omega) &= \frac{1}{(j\omega + 2)(j\omega + 1) - 1}\begin{bmatrix} 0 & 1 \end{bmatrix}\begin{bmatrix} j\omega + 1 & 1 \\ 1 & j\omega + 2 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \end{bmatrix} \\ &= \frac{1}{(j\omega + 2)(j\omega + 1) - 1} \end{aligned} \tag{5}$$

3. Suppose the input signal is $u(t) = 10 + \sin(t) + \sin(2t)$. Solve for $x_2(t)$ when the circuit settles to its periodic steady state. (In your answer, you may leave $\arctan(\cdot)$ or $\tan^{-1}(\cdot)$ expressions uncalculated.) (20 points)

   The solution $x_2(t)$ is a linear combination of responses to input 10, $\sin(t)$ and $\sin(2t)$, respectively, because the circuit is linear.

   To compute the steady state solution for these inputs, we compute $H(j\omega)$ for $\omega = 0, 1, 2$ respectively. We have

$$\begin{aligned} H(0) =& 1, \\ H(j1) =& \frac{1}{(j+2)(j+1) - 1} = \frac{1}{3j} = \frac{1}{3}e^{-\frac{\pi}{2}j}, \\ H(j2) =& \frac{1}{(2j+2)(2j+1) - 1} = \frac{1}{-4 + 6j + 2 - 1} = \frac{1}{-3 + 6j} = \frac{1}{3\sqrt{5}}e^{-(\arctan(-2)+\pi)j}. \end{aligned} \tag{6}$$

   Therefore, the steady state solution of x_2 is

$$x_2(t) = 10 + \frac{1}{3}\sin(t - \frac{\pi}{2}) + \frac{1}{3\sqrt{5}}\sin(2t - \arctan(-2) - \pi). \tag{7}$$