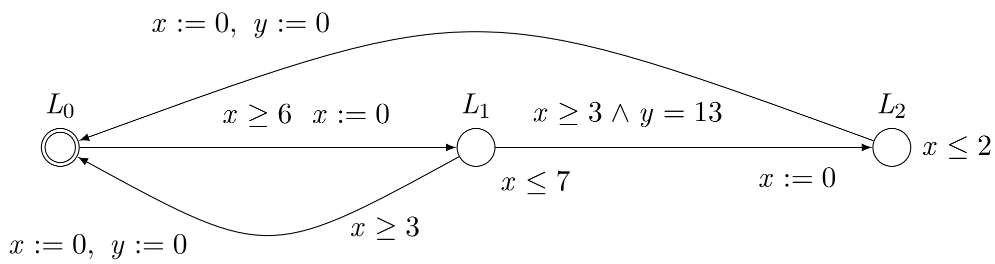
Suggested Solutions for

Written Exam, May 29, 2019

PROBLEM 1

Question 1.1

The constraints are satisfied by the following timed automaton:



Question 1.2

(a) **No.** L_0 does not have to be left, ever.

(b) **Yes.** May stay in L_0 for any amount of time.

(c) **No.** Cannot both be at L_1 and at L_0 or L_2 at the same instant. .

(d) **Yes.** All paths will start in L_0 for at least 6 time units.

(e) In any path, infinitely often F holds.

(f) **Yes.** Locations L_1 and L_2 must eventually be left, leading to L_0.

PROBLEM 2

Question 2.1

(a) `A[] A imply x <= 2` OR `true --> B`

(b) `E<> A and x > 2`

(c) `E<> x > 4`

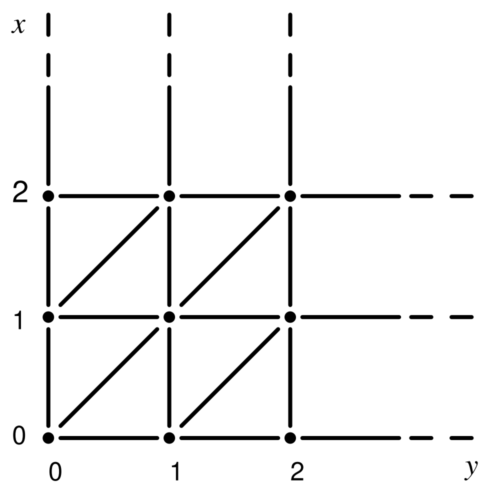
Question 2.2

(a) $(A, x = 0 \wedge y = 0)$ and $(B, x = 1.5 \wedge y = 0)$

(b) $(A, x = 1 \wedge y = 1) \xrightarrow{0.5} (A, x = 1.5 \wedge y = 1.5) \longrightarrow (B, x = 1.5 \wedge y = 0)$

Question 2.3

(a)



(b) There are 9 point regions, 6 half-lines, 16 line segments, 8 bounded areas and 5 unbounded areas. In total, 44 clock regions.

(c) With 2 locations and 44 clock regions, the maximum number of states of the region automaton would be $2 \times 44 =$ **88** states.

(d) $(A, [0 \leq x \leq 1, y = x]) \xrightarrow{time} (A, [x = y = 1]) \xrightarrow{discrete} (B, [x = 1, y = 0])$

PROBLEM 3

Question 3.1

(a) Individual and total loads (utilization):

	T	C	U
a	8	4	50 %
b	12	3	25 %
c	25	5	20 %
Total			95 %

(b) For $N = 3$ the Liu-Layland criterion is 78 %. Hence schedulability cannot be guaranteed.

[For this exam, the Bini criterion was not used. According to this, the product of $1 + U_i$ should not exceed 2. For the given tasks, the procduct becomes $(1.5 \times 1.25 \times 1.2) = 2.25$. Hence the task set cannot be guaranteed by this criterion either.

Finally, we note that none of the periods are harmonic and hence a the notion of *task families* is of no relevance.]

Question 3.2

For the given tasks, rate monotonic priority assignment yields the order a. b, and c. The response times are then found using the iterative formula:

$$R_i^{k+1} = C_i + \sum_{j \in hp(i)} \left\lceil \frac{R_i^{k}}{T_j} \right\rceil C_j$$

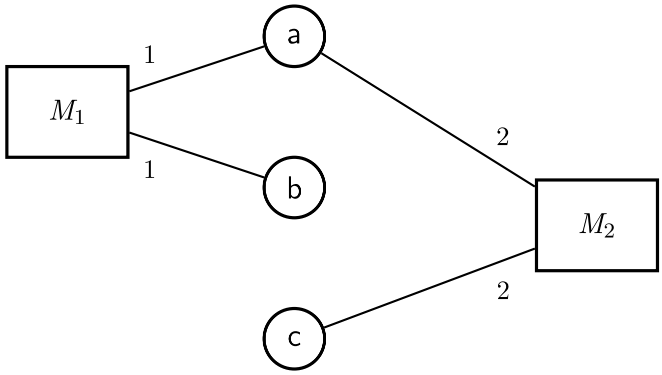
as presented in the following scheme

Task	T	C	R^0	R^1	R^2	R^3	R^4	...
a	8	4	4	$\underline{4}$				
b	12	3	3	7	$\underline{7}$			
c	25	5	5	12	16	19	23	$\underline{23}$

Hence, $R_{\mathsf{a}} = 4$, $R_{\mathsf{b}} = 7$, and $R_{\mathsf{c}} = 23$.

Question 3.3

(a) The system may be illustrated by:

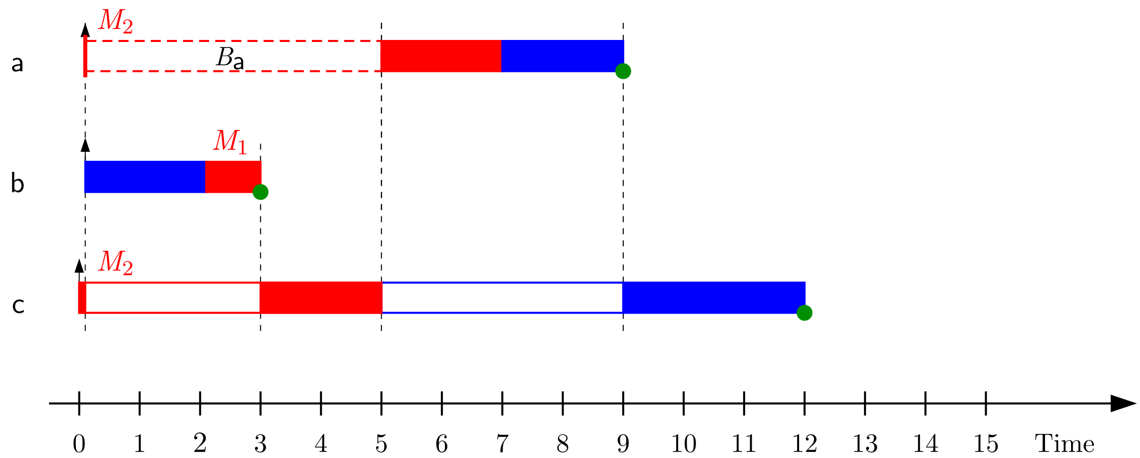


where the numbers on the connections indicate the maximum use of the given resource by the given task.

From this we conclude that $B_{\mathsf{c}} = 0$ as the lowest priority task can never be blocked by any lower priority task.

Also $B_{\mathsf{b}} = 0$ as task b does not share any resources with lower priority tasks (here task c).

(b) The worst case blocking time suffering from priority inversion can occur in the following scenario where tasks a and b are released just when task c has acquired the resource M_2:



(c) By applying standard *priority inheritance*, the computation time C_{b} cannot any longer prolonge the time task a is waiting for M_2 which is then only 2. However, now task a may addtionally be blocked on M_1 adding 1 to the blocking time so that $B_{\mathsf{a}} = 3$ with priority inheritance.

As the response time $R_{\mathsf{a}} = 4 + 3 = 7$ is now within the deadline and as the other response times are unaffected, the scheduling is now feasible.