CIS 721 - Real-Time Systems **Name: ______________________**

Quiz #2 Solution

Spring 2013

Fri., Apr. 26, 2013 (100 points)

1. Consider a preemptive, priority-based system consisting of three periodic tasks, A, B, and C, with the following properties:

Task	**Run Time (C_i)**	**Period (T_i)**	**Deadline (D_i)**
A	1	4	4
B	5	12	12
C	9	36	36

Suppose that tasks are assigned priorities using a rate monotonic priority assignment.

(a) Is the task set feasible based on a Utilization Based Test? _Yes__. Explain briefly. Hint: $3(2^{1/3} - 1) \approx 28/36$ and 1.0 = 36/36.

The task set is simply periodic, 4 | 36 and 12 | 36, so the schedulable utilization is 1.0. U = 1/4 + 5/12 + 9/36 = 33/36 <= 1.0, so the task set is feasible.

(b) Compute the worst-case response time for task B using Response Time Analysis: ______ Show work below.

$W_B^0 = 5$, $W_B^1 = 5 + \text{ceil}(5/4) * 1 = 7$, $W_B^2 = 5 + \text{ceil}(7/4) * 1 = 7$, so WCRT of B = 7.

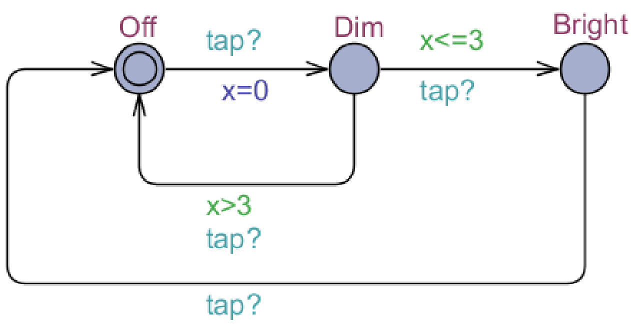
(c) Suppose that **only** task B needs to access a resource X for **k** time units. What is the maximum value of **k** that can be used and still ensure that the task set remains feasible if access is controlled using NonPreemptable Critical Sections (NPCS)? _______. Note that **k** is in the range 0 to 5 because the run-time of task B is 5. Explain briefly.

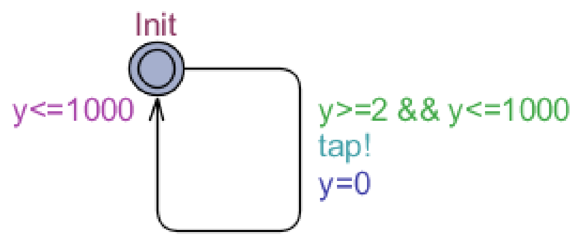
Using NPCS, task B will block all higher priority tasks when it holds a lock on X for k time units. Since task A can only tolerate up to 3 units of blocking time (it needs one unit to complete its own job), so the maximum value of k is 3.

(d) Suppose that **only** task B needs to access a resource X for **k** time units. What is the maximum value of **k** that can be used and still ensure that the task set remains feasible if access is controlled using the Original Priority Ceiling Protocol (OPCP)? ________. Explain briefly.

With OPCP, task B will only block tasks that also use the shared resource, so it will have no impact on the other tasks. Thus, task A will be able to preempt task B at any time, and k can be as large as B's run-time which is 5.

2. Consider the following timed automata, Light and User. They communicate via a global channel called tap. Light has a local clock x, User has a local clock y and a location invariant of y<=1000.





Templates: Light **User**

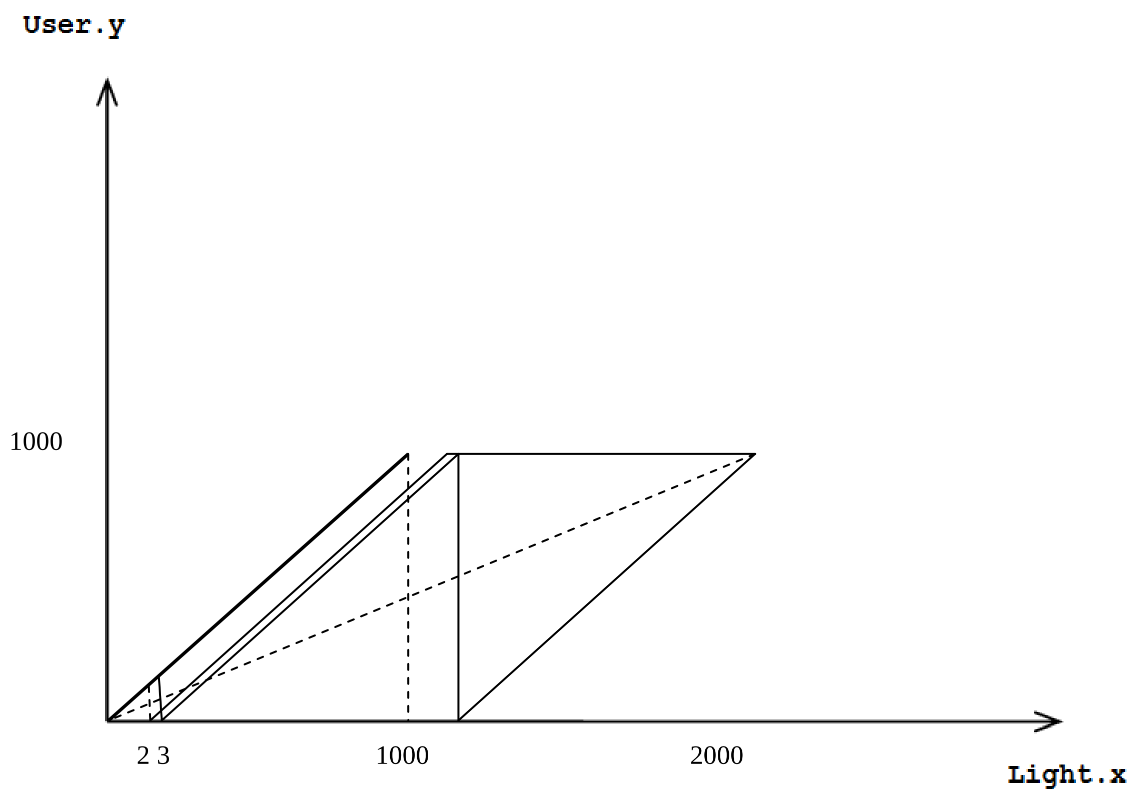
```
// Global Declarations:
chan tap;

// System Declarations
System Light, User;
```

(a) Determine which of the following properties are satisfied for the automata shown above; circle all properties that are satisfied, explain **briefly** why the properties that are not circled are not satisfied: SATISFIED IN RED

- `A[] not deadlock`
- `Light.Off --> Light.Dim`
- `Light.Dim --> Light.Bright` – if the user never clicks within 3 time units, then the light will just go from Off to Dim to Off, etc.
- `Light.Bright --> Light.Dim`
- `E<> (Light.x==1000 and User.y==1000)`
- `A[](Light.x==2003) imply Light.Off`
- `E<> (Light.x==2003 and User.y==1000)`
- `E<> (Light.x==2004)` – the max for Light.x is 2003, in Dim for 3, Bright for 1000, and Off for 1000

- `A[] (User.y <= Light.x)`
- `A[] (Light.x <= User.y)` `- y is reset on each tap, but x is not`

(b) As these automata have two clocks, Light.`x` and User.`y`, the reachable state space with respect to the clocks can be viewed as a point in a two-dimensional Cartesian plane, one axis for clock Light.`x` and one axis for clock User.`y`. A point (a, b), with non-negative a and b, can be used to denote that clock Light.`x` equals a and clock User.`y` equals b. Determine the reachable state space for this automaton; e.g., draw a 2-d graph.

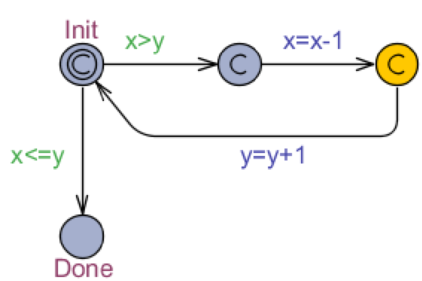


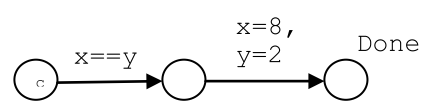
Fill in the trapezoidal region…

3. Consider the following Promela model:

```
int x,y;

proctype p() {
  do
    :: (x > y) -> x=x-1; y=y+1;
    :: else -> break;
  od;
}

proctype q() {
  (x == y);
  atomic{ x=8; y=2; }
}

init
{
  atomic { x=0; y=0; run p(); run q(); }
}
```

(a) Proctype **`p()`** can be modeled in UPAAL using the following template, draw an equivalent UPAAL template for proctype **`q()`**, mark the initial state **`Init`**, and the final state **`Done`**.





(b) Will the processes terminate; e.g., is the property **`A<>(p.Done and q.Done)`** satisfied? Explain briefly. Yes

(c) What are the possible final values for **`x`** and **`y`**? x=5,y=5, or x=8,y=2

(d) What if the atomic statement is removed around **`{ x=8; y=2; }`** ? Does your model for proctype **`q()`** need to be changed? If so, draw the new model below. Also, in the following list, circle all of the possible final values for **`x`** and **`y`**: Yes, add another state and transition, the existing transition for x=8, and the new next transition for y=2.

- `x=8, y=2`
- `x=8, y=0`
- `x=4, y=5`
- `x=5, y=5`
- `x=4, y=2`
- `x=3, y=3`