TECHNISCHE UNIVERSITEIT DELFT
Faculteit Elektrotechniek, Wiskunde en Informatica

IN4390 – Quantitative Evaluation of Embedded Systems

January 25th, 2021, from 09:00 to 12:00

Koen Langendoen, Lydia Chen, George Iosifidis, Marco Zuniga

Question:	1	2	3	4	5	6	7	Total
Points:	10	7	8	20	15	15	15	90
Score:								

- This is an online (open book) exam
- Write your answers with a black or blue pen, not with a pencil
- **Always justify your answers, unless stated otherwise**
- Upload your answers as one file per question in PDF, PNG, or JPEG format

- The exam covers the following material:
  (a) the paper “Basic Concepts and Taxonomy of Dependable and Secure Computing” by A. Avizienis ; J.-C. Laprie ; B. Randell ; C. Landwehr
  (b) chapters 18-20,22-23 (DoE), and 30-33 (Queueing Theory) of the book “The Art of Computer Systems Performance Analysis” by R. Jain
  (c) the paper “Petri nets: Properties, analysis and applications” by T. Murata
  (d) chapters 11.2 (DTMC), and 11.3 (CTMC) of the book “Introduction to probability, statistics, and random processes” by H. Pishro-Nik
  (e) the paper “Exploring Exploration: A Tutorial Introduction to Embedded Systems Design Space Exploration” by A.D. Pimentel

Operational Laws	
Utilization law	$U = XS$
Little's law	$N = XR$
Forced-flow law	$X_k = V_k X$
Bottleneck law	$U_k = D_k X$
Operational Bounds	
Througput	$X \leq \min\left(\frac{1}{D_{max}}, \frac{N}{D+Z}\right)$
Response time	$R \geq \max\left(D, N \times D_{max} - Z\right)$
Queueing Theory M/M/1	
Utilization	$U = XS = \lambda/\mu = \rho$
Probability of n clients in the system	$P_n = \rho^n(1-\rho)$
Mean #clients in the system	$N = \rho/(1-\rho) = \lambda/(\mu-\lambda)$
Mean #clients in the queue	$N_Q = N - \rho$
Mean response time	$R = N/\lambda = 1/(\mu - \lambda)$
Mean waiting time	$W = R - S = \rho/(\mu - \lambda)$
Basic Math	
Geometric series	$\sum_{k=0}^{\infty} r^k = \frac{1}{1-r}$, for $\lvert r \rvert < 1$

ANOVA Table for One Factor Experiments

Component	Sum of Squares	%Variation	DF	Mean Square	F-Comp.	F-Table
y	SSY=$\sum y_{ij}^2$		ar			
$\bar{y}_{..}$	SS0=$ar\mu^2$		1			
y-$\bar{y}_{..}$	SST=SSY-SS0	100	ar-1			
A	SSA $= r\Sigma\, \alpha_i^2$	$100\left(\frac{\text{SSA}}{\text{SST}}\right)$	a-1	MSA $= \frac{\text{SSA}}{a-1}$	$\frac{\text{MSA}}{\text{MSE}}$	$F_{[1-\alpha; a-1, a(r-1)]}$
e	SSE=SST- SSA	$100\left(\frac{\text{SSE}}{\text{SST}}\right)$	$a(r-1)$	MSE $= \frac{\text{SSE}}{a(r-1)}$		

Washington University in St. Louis CSE567M ©2008 Raj Jain

Question 1 [10 points]

(a) 5 points Avizienis et al. [reading list] distinguish 3 important **threats** to the dependability of embedded systems: faults, errors, and failures. Discuss the relationship between these concepts **and** illustrate them with an example from the embedded systems domain.

Solution: See Section 3.5 from the "Basic Concepts and Taxonomy of Dependable and Secure Computing" paper describing how the **activation** of a fault leads to an error, which **propagates** to the service interface resulting in an observable failure. Example could be the software controlling a drone, which is typically full of bugs (faults). When a particular set of conditions apply (wind gust at full moon) the execution takes a rare path including one of these bugs, which leads to an erroneous internal state ("we have landed") that propagates into a catastrophic failure when the motors are subsequently shutdown (the drone falls from the sky, ouch).

(b) 5 points Mention 3 **means** to address the above-mentioned threats **and** link them to your example.

Solution: Following Section 2.4, the dependability threats can be addressed by:

- Fault prevention – apply proper (model-based) software development methods.
- Fault tolerance – include checks safeguarding operations (e.g., 'speed==0 required for shutting off motors').
- Fault removal – during development validation tools can be used to uncover tricky faults (e.g., synchronization faults may be uncovered by Petri Nets).
- Fault forecasting – extensive testing provides some insight into the quality of the drone software, assessing the risks involved / need for further fault removal.

Question 2 [7 points]

Here is a 2-way ANOVA analysis with 2 factors being A and B. You need to figure out the missing values to answer the questions below.

	DF	SS	MS	F
A	2	80	40	**
B	3	100	33.33	**
Interaction	**	40	6.67	0.8
Error	**	100	**	
Total	23	320		

(a) 1 point How many replicates of experiments were performed?

Solution: 2 replications.

(b) 3 points We want to know which factors (A, B, and AB) are significant. What statistics should we compare against the computed F statistics for factor A,B, and AB, respectively? Assume the significant level is 0.05.

Solution: Following the notation of the cheatsheet (F[1-α; a-1, a(r-l)]) we find

- For factor A, compare it with F[0.95; 2, 12].
- For factor B, compare it with F[0.95; 3, 12].
- For factor AB, compare it with F[0.95; 6, 12].

The alternative notation of F(a-1, a(r-l), α) is also accepted.

(c) [2 points] The P-value is another way to check if factors are significant. We obtained the following P-values (0.0097, 0.56, 0.53) for factor A, B, and AB, respectively. OOPs! we actually made a mistake in one of the P values. Which factor is mistaken? Give your reason!

Solution: The F statistics of factor B is very similar to that of factor A, so their P values should be quite close also. The F statistics of factor AB is quite different, so the P value of B cannot be similar to the P value for AB. Thus, the P value for factor B is wrong.

(d) [1 point] If you want to fit a regression model, which factors shall be included? And why?

Solution: You should only include the significant factors. This rule needs to be applied to the (bogus) answers from above; your mileage may vary.

Question 3 [8 points]

You want to know the significant parameters that affect the image recognition software on your cell phone. You use DoE to figure that out. You consider the following parameters 4 parameters: batch size (A), learning rate (B), and memory size (C), and image sizes (D).

(a) [2 points] You start with the 2k full factorial design. Write down the sign table for only factor A, B, C, and D.

Solution:

A	B	C	D
-1	-1	-1	-1
1	-1	-1	-1
-1	1	-1	-1
1	1	-1	-1
-1	-1	1	-1
1	-1	1	-1
-1	1	1	-1
1	1	1	-1
-1	-1	-1	1
1	-1	-1	1
-1	1	-1	1
1	1	-1	1
-1	-1	1	1
1	-1	1	1
-1	1	1	1
1	1	1	1

(b) [1 point] With 3 replication per configuration, how many experiments do you need to run?

Solution: 48

(c) 2 points However, it's too time consuming to run the full factorial design. And, you decide to run a half factorial design by following the generator function of I=ABCD . Based on this, write down the sign table for only factor A, B, C, and D.?

Solution:

A	B	C	D
-1	-1	-1	-1
1	-1	-1	1
-1	1	-1	1
1	1	-1	-1
-1	-1	1	1
1	-1	1	-1
-1	1	1	-1
1	1	1	1

(d) 2 points Tell us what kind of design this is and its resolution?

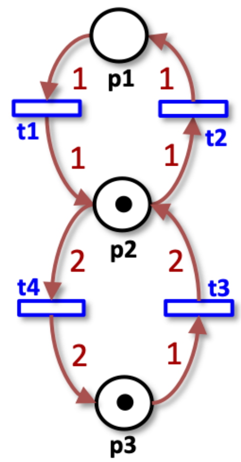
Solution: 2^{4-1} design, with resolution IV

(e) 1 point Is this a good generator function, compared to I=ABD? And, why?

Solution: I=ABCD is good (to be preferred), because this design results into a resolution 4 design and I=ABD results into resolution 3 design.

Question 4 [20 points]

(a) 5 points Is the following Petri Net 6-bounded? Please justify your answer.



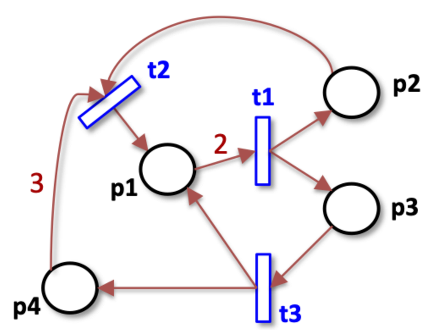
Solution: No, it is not. For example, the sequence of firings t_3, t_4, t_3, t_3, t_4, t_3, t_3 results in the marking m=(0,7,0).

(b) Consider the following incidence matrix:

$$A = \begin{bmatrix} -2 & 1 & 1 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \\ 0 & -3 & 1 \end{bmatrix}$$

i. 3 points Draw the Petri Net that is associated with incidence matrix A.

Solution:



ii. 1 point Is there any initial marking with at least 4 tokens in total, for which some of the transitions cannot be fired in any firing sequence? Justify your answer with a proof or an example.

Solution: An example of a marking that includes 4 tokens in total and prevents t_2 from firing is (1,1,0,2).

iii. 1 point If the initial marking m_0 is as follows:

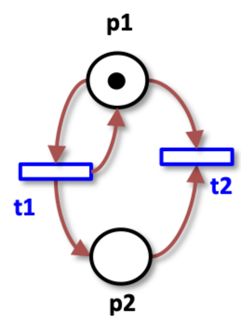
$$m_0 = \begin{bmatrix} 3 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

and t_1 fires, calculate analytically, i.e., using algebraic equations, what the obtained marking m_1 will be. Please include the calculations in your answer.

Solution: The calculation of m_1 involves a matrix multiplication and addition:

$$m_1 = \begin{bmatrix} 3 \\ 1 \\ 1 \\ 0 \end{bmatrix} + \begin{bmatrix} -2 & 1 & 1 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \\ 0 & -3 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 2 \\ 0 \end{bmatrix}$$

(c) Consider the following Petri Net:

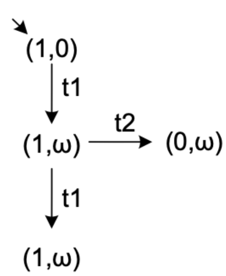


i. 2 points What is the purpose (or, usage) of a coverability tree?

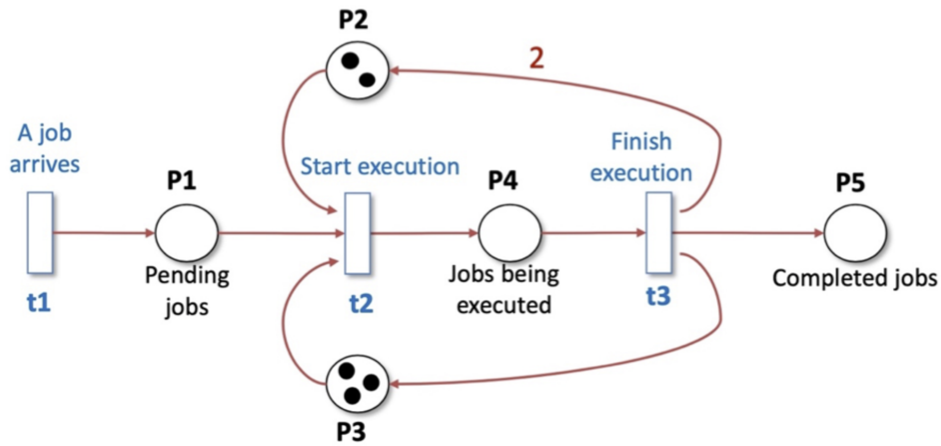
Solution: A coverability tree (a) allows to analyze unbounded Petri Nets and (b) can be used to find out if the reachability graph is infinite.

ii. 3 points Construct the coverability tree for the above Petri Net .

Solution:



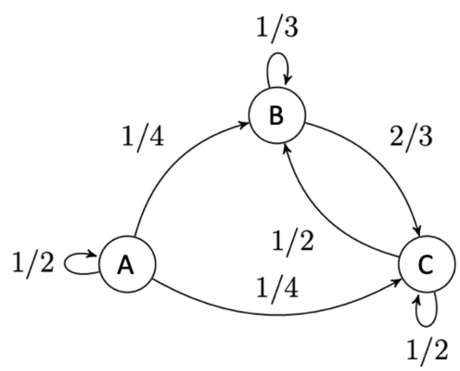
(d) 5 points Reason the maximum number of jobs that can be executed concurrently in the system that the following Petri Net models.



Solution: The maximum number of concurrent jobs is determined by the maximum number of tokens that can be placed at P4. This number is 3, and a firing sequence that gives this result is $\{t_1, t_2, t_3, t_1, t_1, t_1, t_2, t_2, t_2\}$. Note that the number of tokens at P2 increases after every "loop", while the number of tokens at P3 remains "fixed" at 3.

Question 5 [15 points]

Consider the Discrete Markov Chain below with state-space $S = \{A, B, C\}$:



Assume that $X_t = X_0, X_1, \cdots, X_n, \cdots$ denotes the state of the system at time t, and the initial distribution is $\pi_0 = [1, 0, 0]$.

(a) 1 point $P(X_5 = B \mid X_4 = C, X_3 = B, X_2 = A, X_1 = A) =?$

Solution: As Markov systems are memoryless we find:
$P(X_5 = B \mid X_4 = C, X_3 = B, X_2 = A, X_1 = A) = P(X_5 = B \mid X_4 = C) = 1/2$

(b) 1 point $P(X_3 = B, X_2 = B \mid X_1 = A) =?$

Solution: $P(X_3 = B, X_2 = B \mid X_1 = A) = P(X_3 = B \mid X_2 = B) \times P(X_2 = B \mid X_1 = A) = 1/3 \times 1/4 = 1/12$

(c) 4 points $P(X_2 = B) =?$

Solution: We need to find all paths from $X_0 = A$ to $X_2 = B$:
$P(X_2 = B) = P(X_2 = B \mid X_1 = A) + P(X_2 = B \mid X_1 = B) + P(X_2 = B \mid X_1 = C) = 1/4 \times P(X_1 = A) + 1/3 \times P(X_1 = B) + 1/2 \times P(X_1 = C) = 1/4 \times 1/2 + 1/3 \times 1/4 + 1/2 \times 1/4 = 1/8 + 1/12 + 1/8 = 1/3$

(d) 1 point Explain why the chain is reducible.

Solution: It's reducible because once mass escapes A (either to B or C) there is no possibility for it to return (there is no path from B nor C to A).

(e) 1 point In general, with reducible chains, we cannot guarantee that the steady-state is always the same (independently of the initial state). Explain why that is the case.

Solution: In a reducible chain there must be one class U that cannot be reached from all other classes. If mass can stay within that class then we can end up with at least two different steady states: one with mass in U if the initial state contains mass in U, and the other without mass in U when starting without any mass in U.

(f) 1 point How many classes does the chain have? Explain why.

Solution: A forms its own class, and B and C form a second class as they can reach each other in a single step. Thus the chain has 2 classes.

(g) 1 point Are the classes recurrent or transient? Explain why.

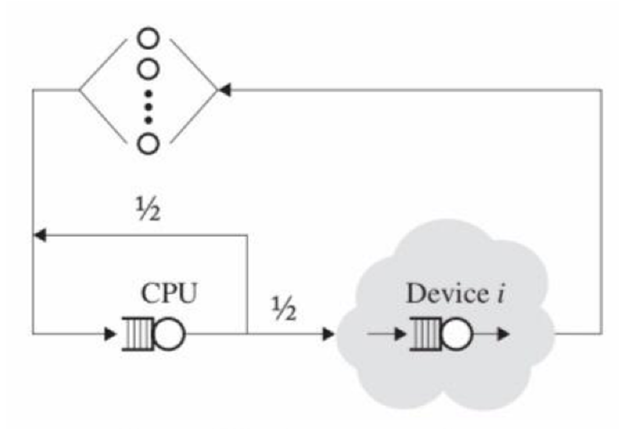
Solution: Class A is transient; class B+C is recurrent.

(h) 5 points Even though the above chain is reducible, we can still obtain the steady-state for any initial state. Explain why that is the case and calculate the steady-state.

Solution: All mass that starts out in state A will eventually "escape" into B or C, thus for the steady state we can simply reduce the system to the subgraph spanned by B and C. That subgraph is irreducible and non-periodic, so we can find the steady state. In steady state the flows between B and C must be equal, so:
$P_B \times 2/3 = P_C \times 1/2$, hence, $PB = 3/4 P_C$. The total mass should equal one, so $P_B + P_C = 7/4 P_C = 1$. Thus $P_C = 4/7$, and $P_B = 3/7$. The steady state distribution therefor equals $\{0, 3/7, 4/7\}$.

Question 6 [15 points]

Consider the following interactive system:



in which users generate requests that need somewhat complex computations as well as post processing in the cloud that involves a Device i amongst others. Through an extensive measuring campaign the following facts were uncovered:

mean user think time = 30 seconds
mean service time at Device i = 15 ms
utilization of Device i = 0.3
#visits-per-request to Device i = 10
#jobs in the cloud subsystem = 31
mean total time in the system (including think time) = 50 seconds

(a) 3 points Compute the overall throughput of the system. Show your calculations and mention which operational law(s) you apply.

Solution: The **utilization law** ($U = X \cdot S$) gives us that the throughput X_i for Device i equals $U_i/S_i = 0.3/0.015 = 20$ jobs/s. Using the **forced-flow law** ($X_i = X \cdot V_i$) we find that the overall throughput X equals $X_i/V_i = 20/10 = 2$ requests/s.

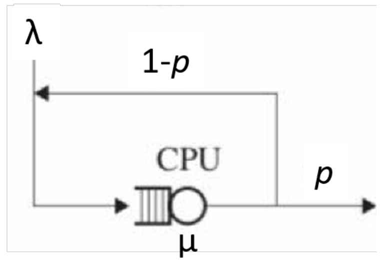
(b) 3 points What is the average number of users that is eagerly waiting for the results of their request? (Mention the operational law(s) that you apply).

Solution: Following **Little's law** ($N = X \cdot (R + Z)$) we find that the overall number of requests in the system equals $2 \times 50 = 100$. The requests are proportionally distributed over 'being serviced' and 'being thought of' as 20:30, thus the number of waiting users equals $20/50 \times 100 = 40$.

(c) 2 points What is the mean number of requests in the processing part (CPU + queue) of the system?

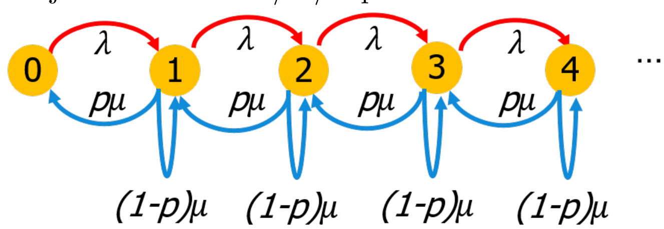
Solution: Of the 40 outstanding requests, 31 are in the cloud, so the remaining 9 must be in the processing part of the system.

Consider the processing part of the system in which a user request may be processed multiple times as jobs only exit the system with a certain probability, say p:



(d) 5 points Reason that the loop-back shown above is equivalent to an M/M/1 system with service rate $p\mu$.

Solution: We can model the system as an infinite chain of states encoding the number of jobs in the system just like for an M/M/1 queue.



We go to the right when a new job arrives (with rate λ), go left when a jobs completes (with rate μ) **and** exits the systems with probability p, so with an effective rate of $p\mu$. The fraction $1-p$ of jobs returning back to the queue can be ignored as that amounts to staying in the same state; the looping back can be regarded as increasing the job length, which causes (matches) the effective slowdown of the CPU.

(e) 2 points What is the CPU utilization in the complete interactive system?

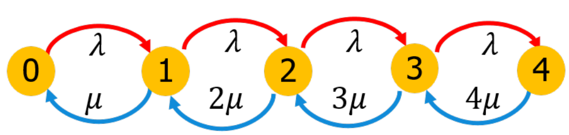
Solution: Let's use the M/M/1 result that $N = \rho/(1-\rho)$; it follows that $\rho = N/(N+1)$. We know that $N = 9$, so the the load $\rho = 9/10$. Thus the CPU utilization is 0.9

Question 7 [15 points]

A small telephone switchboard has four lines. Calls during the busiest part of the day arrive in a Poisson process with rate 80 per hour and the lengths of the calls are independently exponentially distributed with mean length 3 minutes. Any call that is made when all the lines are busy is lost.

(a) 3 points Draw the state transition diagram for the calls connected by the switchboard.

Solution:



(b) 5 points Determine the stationary distribution of the number of ongoing calls.

Solution: $\lambda = 80$ and $\mu = 60/3 = 20$, so $\rho = \lambda/\mu = 4$.

$\lambda P_0 = \mu P_1 \quad P_1 = \rho P_0 = 4P_0$

$\lambda P_1 = 2\mu P_2 \quad P_2 = \frac{1}{2}\rho P_1 = 2P_1 = 8P_0$

$\lambda P_2 = 3\mu P_3 \quad P_3 = \frac{1}{3}\rho P_2 = \frac{4}{3}P_2 = \frac{32}{3}P_0$

$\lambda P_3 = 4\mu P_4 \quad P_4 = \frac{1}{4}\rho P_3 = P_3 = \frac{32}{3}P_0$

Factoring in that $\sum_{n=0}^{4} P_n = 1$ yields the following expression:
$P_0 = \frac{1}{1+4+8+32/3+32/3} = 3/103.$

As such the distribution becomes:

$$P_0 = 3/103$$
$$P_1 = 12/103$$
$$P_2 = 24/103$$
$$P_3 = 32/103$$
$$P_4 = 32/103$$

(c) 2 points What proportion of calls are lost?

Solution: Calls are lost when all lines are busy, that is, when the number of ongoing calls equals 4. The proportion of lost calls is thus $P_4 = 31\%$.

(d) 4 points It is proposed to increase the size of the switchboard to six lines. What will be the proportion of lost calls in this case?

Solution: We need to extend the state diagram with two more states, compute the stationary distribution, to arrive at P_6.

$\lambda P_4 = 5\mu P_5 \quad P_5 = \frac{1}{5}\rho P_4 = \frac{128}{15} P_0$
$\lambda P_5 = 6\mu P_6 \quad P_6 = \frac{1}{6}\rho P_5 = \frac{256}{45} P_0$

Factoring in that $\sum_{n=0}^{6} P_n = 1$ yields the following expression:
$P_0 = \frac{1}{103/3+128/15+256/45} = 45/(1545 + 384 + 256) = 45/2185 = 9/437.$

Thus the fraction of lost calls decreases to $P_6 = \frac{256}{45} P_0 = 256/2185 = 12\%$.

(e) 1 point Will the alternative of splitting the district into two parts, each being served by a 3-line switchboard reduce the fraction of lost calls even further?

Solution: The Loo example from the lectures showed that two M/M/2 queues at half the capacity perform worse than one M/M/1 queue at full capacity, so NO the alternative is not an improvement. (This intuition van be verified by solving the 3-line system with arrival rate $\lambda/2$, resulting in a lost-call probability of 21% (4/19), which exceeds that of the 6-line system.)