Dipartimento di Elettronica, Informazione e Bioingegneria
Politecnico di Milano

Prof. Elisabetta Di Nitto

20133 Milano (Italia)
Piazza Leonardo da Vinci, 32
Tel. (39) 02-2399.3400
Fax (39) 02-2399.3411

Software Engineering for HPC – Written Exam

18th of June, 2024

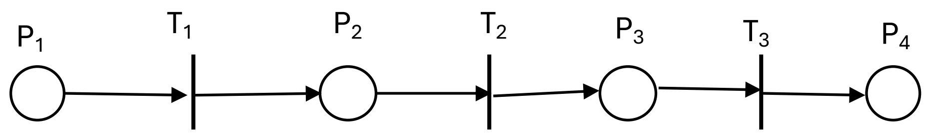
Last name, first name and Id number (Matricola or person code)

Number of paper sheets you are submitting as part of the exam

Rules for the exam

A. Remember to write your name and Id number on each piece of paper that you hand in.
B. You may use a pencil.
C. Incomprehensible handwriting is equivalent to not providing an answer.
D. The exam is open book. You can use your notes, the course slides and additional papers or documents you have collected.
E. You can use the material in printed or in electronic form. In this last case, you can use a plan ebook reader not connected to the internet.
F. You can also use a plain calculator.
G. Read carefully all points in the text!
H. The exam score is up to 16 points. The minimum score to pass the exam is 8 points
I. Total available time at the exam: 1h and 30 mins

Petri nets (10 points)
The Petri Net in the figure below represents a typical workflow in a hospital emergency department.
Place P_1 is the starting point. A token in this place represents a patient that has entered the hospital. Transition T_1 moves the patient to the triage assessment (place P_2), after that the patient moves to the medical examination (place P_3) via transition T_2. At the end, transition T_3 moves the patient to place P_4 where he/she waits for the examination results.



PN_1: Complete the diagram so that:

- As soon as the patient moves to the medical examination, the triage can start a process that produces the necessary documents.
- Examination results are produced after the end of the examination.
- The patient waiting in place P_4 is enabled to receive the examination results only after the documents generated by the triage are ready.

- At this point, there are two possible outcomes: the patient can be either hospitalized or sent home.

Describe your extensions to the initial Petri net and the choices you have made.

PN_2: Referring to the Petri net obtained in your answer to question PN_1, assume that the initial marking includes a single token in P1 representing a single patient going through the medical check. Draw the reachability graph for the Petri net and demonstrate that the following property holds: *No patient can be hospitalized and sent home at the same time.*

PN_3: Extend the Petri net to model the fact that no more than three patients at a time can be under medical examination and demonstrate that this property actually holds.

Solution

PN_1

The Petri net is extended as described in Figure 1. The places and transitions in red model the preparation of the documentation by the triage doctors. The places and transitions in green model the arrival of the examination result (place P_8) after their elaboration (transition T_8) and the two possible outcomes (places P_9 and P_{10} defined as output places of transitions T_6 and T_7, respectively).

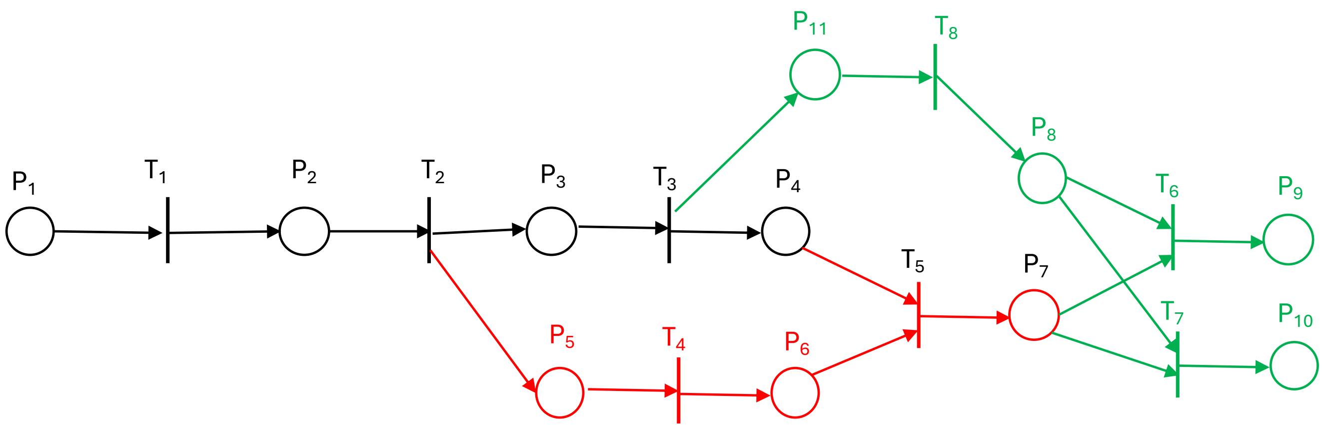


Figure 1. Extended Petri net representing the preparation of documents and the possibility to have two outcomes.

PN_2

The reachability graph, built assuming that the initial marking includes a single token in P_1, is shown in Figure 2. By inspecting it, we can easily see that there is no node in the graph that indicates that two tokens in places P_9 and P_{10} can coexist, leading to a situation where the patient both goes home and is hospitalized.

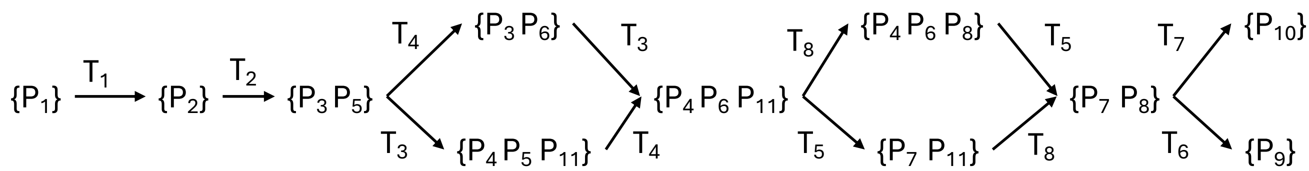


Figure 2. Reachability graph.

PN_3

In this case we have a shared resource which is represented by the number of slots available for the medical check. This can be represented through a place (P_{12} in the two figures below) having as initial marking three tokens and connected to transition T_2 to model the lock of a slot when someone goes through the medical check. To model the fact that slots return to be available at the end of the medical check, one of the transitions in the net should return the token back to P_{12}. In Figure 2 the token is returned in P_{12} by transition T_3 right after the end of the medical check. In Figure 3, instead, we model the fact that, after the end of the medical check, the staff remains busy to produce the outcome. In this case, P_{12} is an output place of transition T_8.

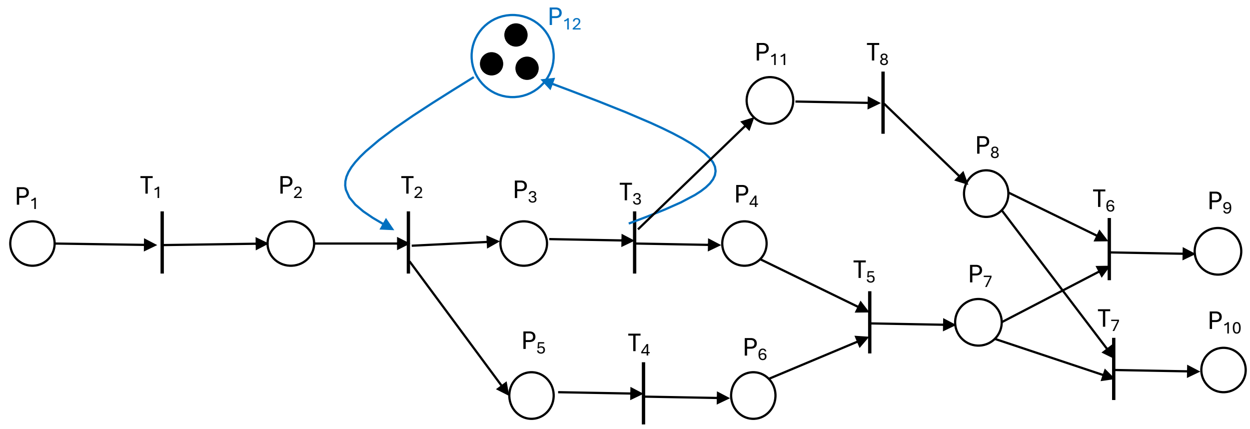


Figure 3. Limited slots for the medical examination - v1.

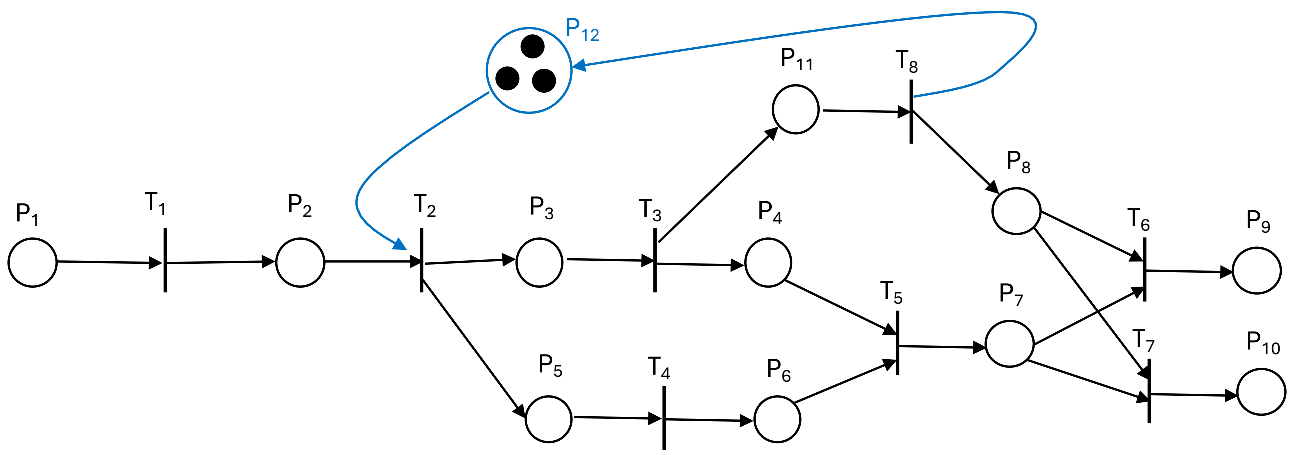


Figure 4. Limited places for the medical examination - v2.

Referring to the Petri net of Figure 3, demonstrating that no more than three patients can go through the medical check at the same time can be obtained as follows: we can observe that the portion of the net involved in controlling the access of patients in the medical check is composed of transition T_2, which grants the patients access and T_3, which frees the resources and moves the patients to the next step. Let's assume that P_2 contains an infinite number of tokens and, therefore, can potentially overflood the medical check with patients. Under this condition, the firing of T_2 is regulated by the number of tokens in P_{12}. The firing of T_3 is regulated exclusively by the number of tokens in P_3. Therefore, the two transitions can fire in any of the sequences that can be obtained by following the arcs of the partial reachability graph in Figure 5, starting from node $\{3P_{12}\}$. No other sequences are possible, and we can note that the number of tokens in P_3 never exceeds 3, as we wanted to demonstrate, and that the sum of tokens in P_{12} and P_3 is always 3. Note that in the partial graph we are not taking into account the marking of P_2 as we have assumed that it is infinite and the marking of P_4 and P_{11} because they do not have an effect on limiting the access of patients to the medical check, so they are not interesting for this analysis.

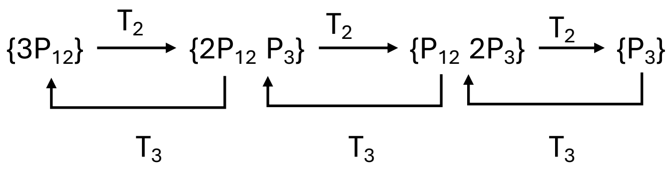
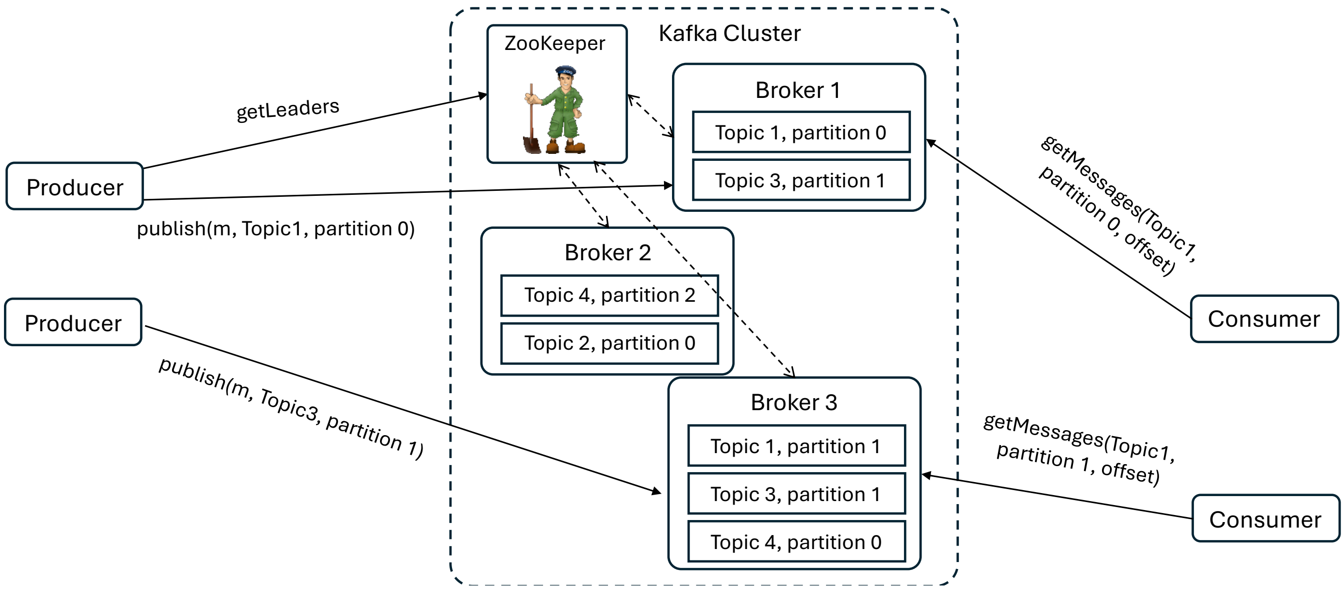


Figure 5. Partial reachability graph referring to P_{12} and P_3.

Considering the net in Figure 4, the demonstration can follow the same line, but we should focus our attention on the sub-net composed of transitions T_2, T_3, T_8 and places P_3, P_{11} and P_{12}.

Availability (6 points)
Consider the following instance of a Kafka cluster. Suppose that the servers on which the 3 *Brokers* are hosted have the following availability values.

- Server Broker 1: 97%
- Server Broker 2: 94%
- Server Broker 3: 98%



As illustrated by the schema, *Consumers*, retrieve various messages concerning certain topics. Messages are obtained through the `getMessages` operation, which can retrieve messages from any broker handling a topic and partition. For the purpose of this exercise, assume that the messages have already been published. Assume also that the schema shows all the existing partitions for the depicted topics.

Availability_1: What is the total availability of an operation that needs to retrieve 2 messages, one related to Topic 3 partition 1 and the other to Topic 2 partition 0?

Availability_2: What is the total availability of an operation that needs to retrieve 2 messages, one related to Topic 1 partition 1 and the other to Topic 4 partition 0?

Availability_3: Assuming that each broker can handle only a single partition for each topic and that the number of available servers is fixed, define a new configuration of the Kafka Cluster so that the operation of point Availability_2 has a total availability greater or equal to 99%.

Solution

Availability_1

This is the series of the availability of Topic 2 partition 0 (which is not replicated, so the availability of the partition is the availability of the corresponding broker) and Topic 3 partition 1 (which is replicated, so its availability is the parallel of the corresponding brokers).

A = 0.94*(1-(1-0.97)*(1-0.98)) = 0.939

Availability_2

This is the series of the availability of Topic 1 partition 1 (which is not replicated, so the availability of the partition is the availability of the corresponding broker) and Topic 4 partition 0 (also not replicated). The two partitions happen to be on the same broker:
A = 0.98*0.98 = 0.960

Availability_3

Trying to minimize the changes in the current configuration, we can replicate Topic 1 partition 1 on Broker 2 and Topic 4 partition 0 on Broker 1. This is possible under the assumption that we can introduce new partitions in each broker while respecting the constraint of not having partitions of the same topic in a broker. This will result in the following availability, which fulfills the required constraint:
A = (1-(1-0.94)*(1-0.98)) * (1-(1.0.97)*(1-0.98)) = 0.998