

Midterm Examination Key

Information and Computer Science 121

Professor Debra J. Richardson
Winter 1998

True/false questions (1 point each. Circle true or false in the left margin.)

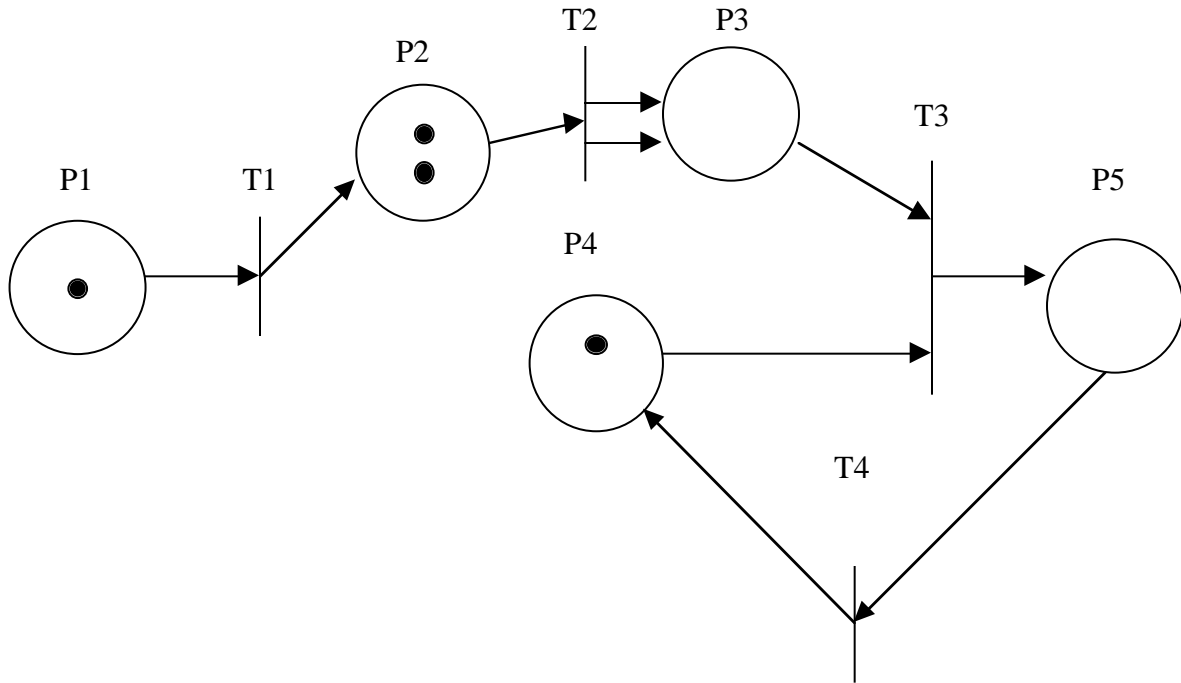
- | | | |
|-------|-----|-------------------------------------------------------------------------------------------------------|
| false | 1. | A requirements document only focuses on defining the functionality of the system. |
| true | 2. | The majority of faults in software engineering are requirements are design faults. |
| false | 3. | The CMM focuses on process improvement, while ISO 9000 focuses on process maturity. |
| false | 4. | Requirements analysis focuses on how the system will be built. |
| false | 5. | “Build and Fix” is a lifecycle model. |
| true | 6. | Software Engineering is a part of Systems Engineering. |
| false | 7. | Adding staff will alleviate schedule delays when developing large software products. |
| false | 8. | The finite state machine system modeling notation is less formal than the data flow diagram notation. |
| true | 9. | Software tools are automated implements for methods. |
| false | 10. | High cohesion is better than low coupling. |

Problems (7 points per question. Please provide diagrams in the space provided.)

Answers to 11-13 were so varied that we are not providing a key. If you would like your grade explained, please see Michele or Debra.

14. Petri-nets

a. Given the initial marking of the Petri-net shown below:

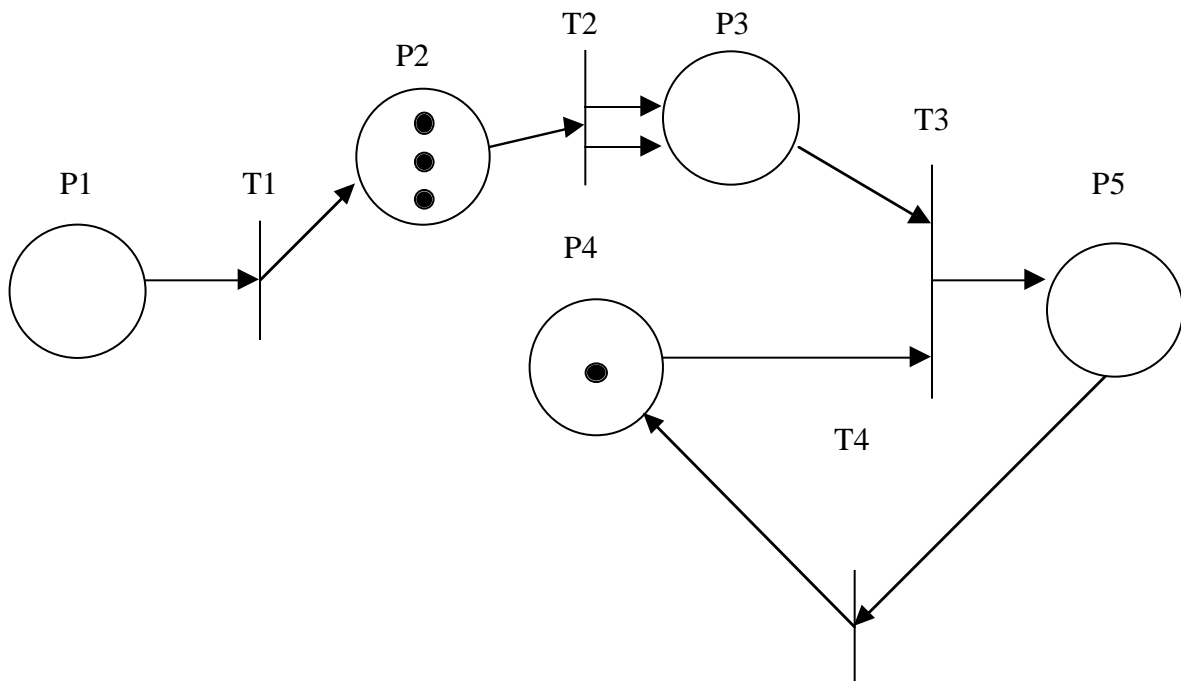


Can T1 fire?

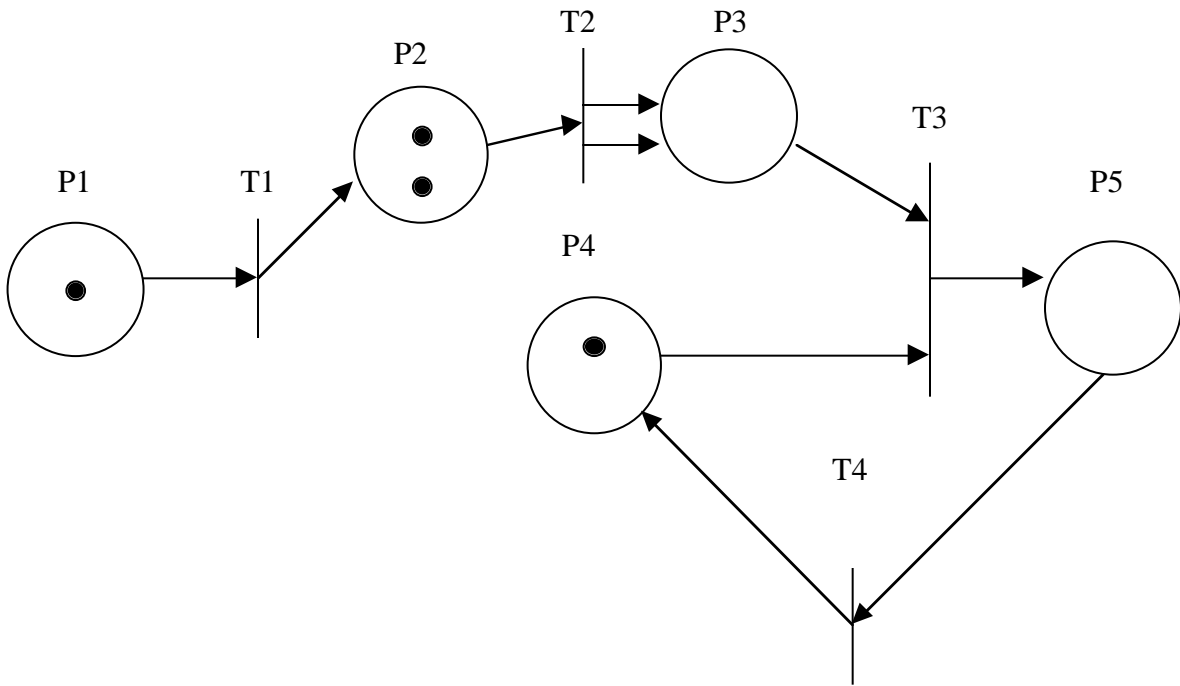
Circle

YES

If so, what does the Petri-net look like after it does?



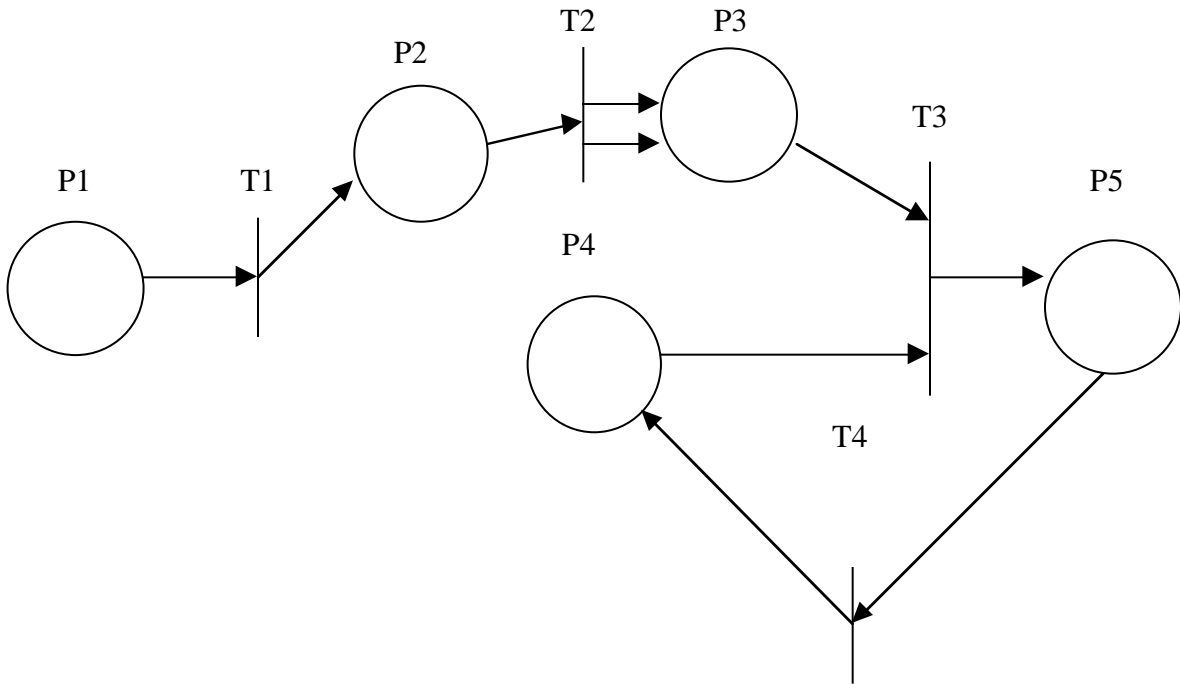
b. Given the initial marking of the Petri-net shown below:



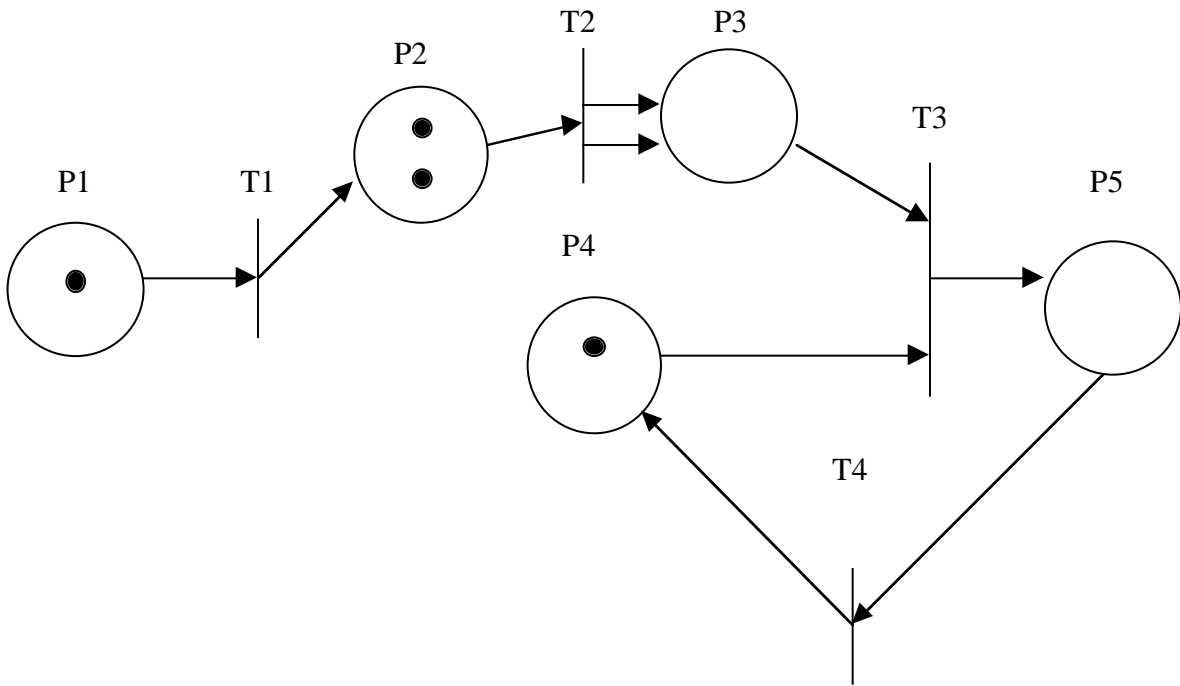
Can T3 fire?

Circle NO

If so, what does the Petri-net look like after it does?



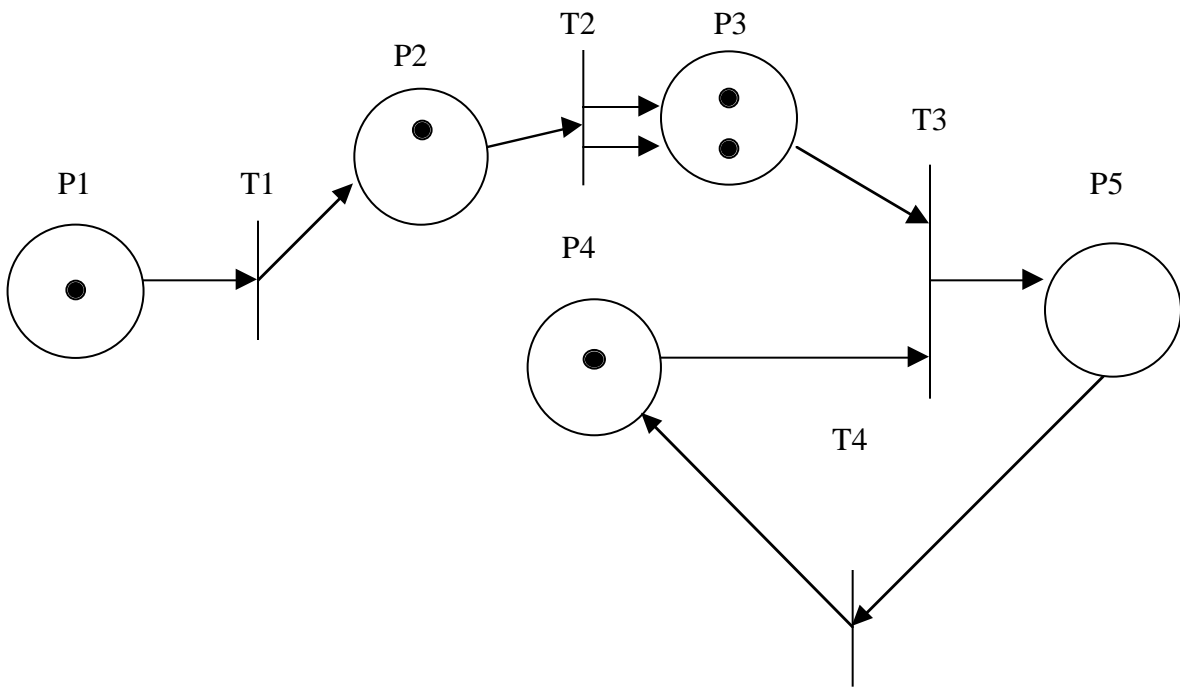
c. Given the initial marking of the Petri-net shown below:



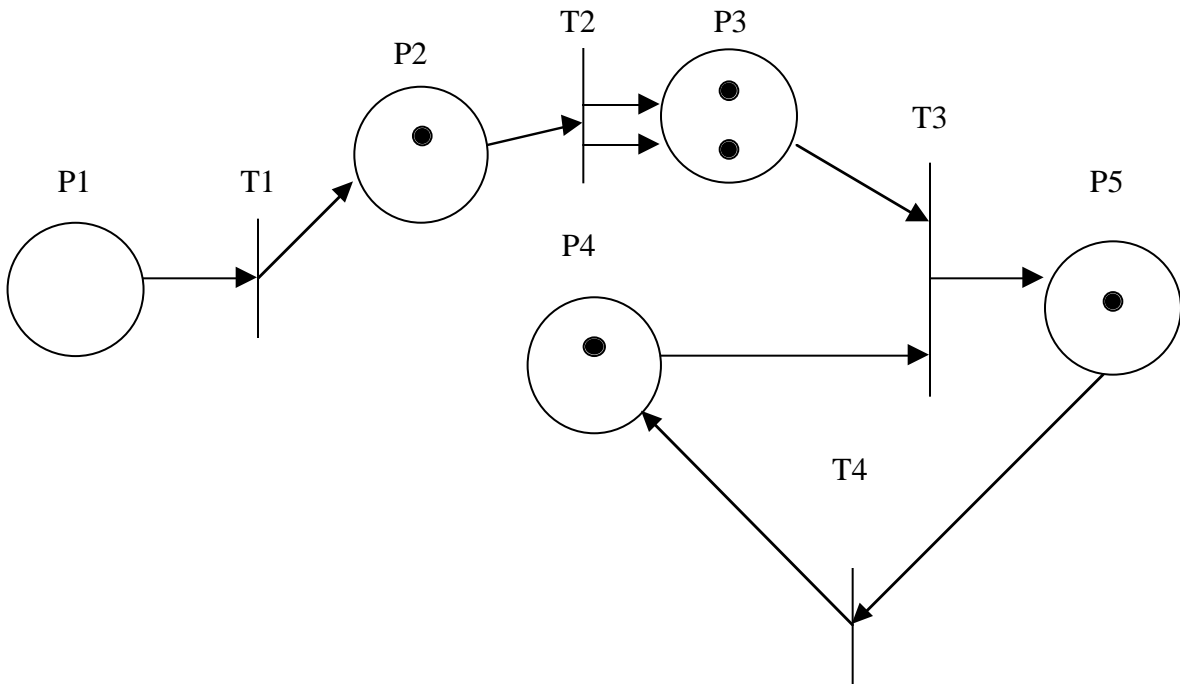
Can T2 fire?

Circle YES

If so, what does the Petri-net look like after it does?



d. Given the initial marking of the Petri-net shown below:

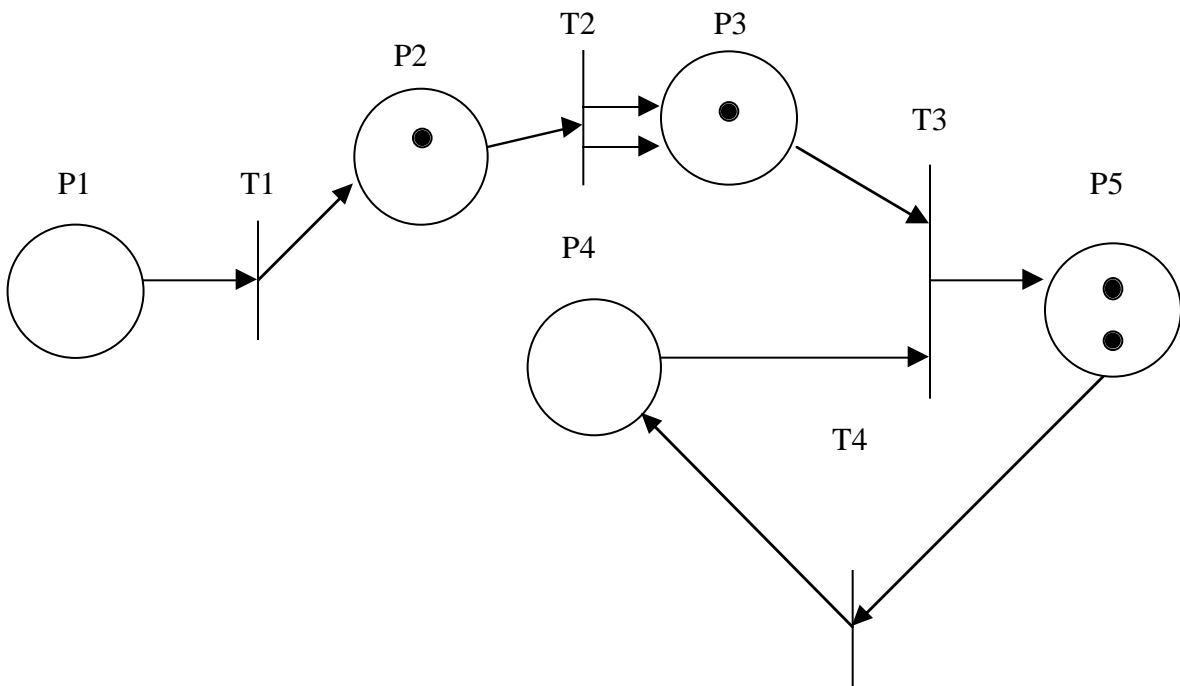


Can T3 fire?

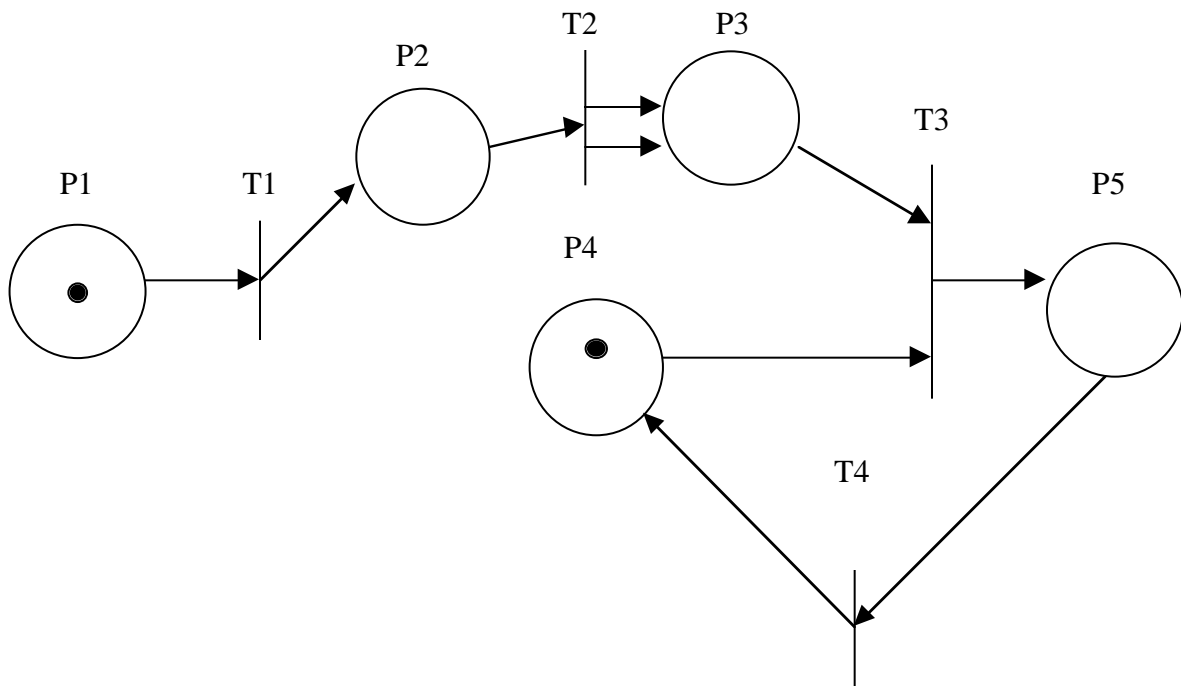
Circle

YES

If so, what does the Petri-net look like after it does?



e. Given the initial marking of the Petri-net shown below:



How many transitions must fire to reach the final state of this Petri-net? _____

Petri-nets have no final state.

Very brief answers (2 points each. Please answer each question in 1-6 words in the space provided. You will not be penalized for more words, as long as they are directly related to the question; but be aware that you should be able to answer the question in a few words.)

15. What is the average percentage of lifecycle costs spent during *maintenance*?
60-70 %
16. Which lifecycle phase defines the *acceptance test plan*?
Requirements analysis and specification
17. What three problems in software production led to the *software crisis*?
Low quality software, late delivery of software, over-budget software.
18. What activity was added in the *spiral model* that was not present in the *waterfall model*?
Risk analysis
19. What is the primary objective of *prototyping*?
Requirements acquisition
20. Requirements should be complete. Name five other desirable characteristics of a *requirements* document.
Any five of the following:
abstract, consistent, concise, appropriate, even
feasible, modifiable, precise, readable, unambiguous,
verifiable, testable, free of implementation bias
21. What technique or method is specifically directed toward *usability testing*?
Cognitive walkthrough
22. User satisfaction is one important attribute addressed by *usability testing*. Name two more.
Any two of the following:
learnability, memorizability, efficiency, error recovery or robustness
23. *Human-computer interaction* focuses the system's interface. Name one more foci of HCI.
One of the following:
expertise, experience, user tasks
24. Completeness is one desirable characteristic of a good *design*. Name three more.
Any three of the following:
consistency, non-ambiguity, uniformity,
confirmability, flexibility, desensitized to change
25. Procedural abstraction is used in both *structured design* and *object-oriented design*, while data abstraction is only used in *object-oriented design*.

26. According to *Brooks*, what is the Number One cause for software products to go awry?
Time, schedule
27. Rank the following types of coupling from worst to best: *data coupling*, where one module passes only homogenous data items to another; *control coupling*, where one module passes an element of control to another; *content coupling*, where one module directly references the contents of another.
Content -> control -> data
28. Rank the following types of cohesion from worst to best: *temporal cohesion*, where a module consists of a series of actions related in time; *procedural cohesion*, where a module performs a series of actions related by the sequence of steps to be followed; *informational cohesion*, where a module consists of a number of independent actions performed on the same data.
Temporal -> procedural -> informational
29. What is the primary cause of *inconsistency in design*?
Miscommunication

Short answers (4 points each. Please answer each question in two or three sentences in the space provided.)

30. What is the primary purpose of *acceptance testing*? How does it differ from system testing?
The primary purpose of acceptance testing is to provide a validation to the client that the right system was built at the time of delivery. System testing, on the other hand, is done against requirements specification. In fact, the system test plan should be more comprehensive as the requirements specification is more than just the acceptance test plan.
31. Define and differentiate *information hiding* and *abstraction*?
Information hiding dictates that each module should hide the information that is not required by the module's clients. Abstraction says that we should identify important aspects and ignore the details; as such it is a way to handle complexity. Information hiding is a way of making abstraction explicit. Abstraction puts off details, while information hiding hides them. Abstraction says there is one specification and many realizations; information hiding makes the specification visible and the realizations hidden.
32. When developing software with *prototypes*, should you keep the prototype and build upon it? State yes or no and discuss why or why not?
No, because 1) it's too easy to slip into the build and fix model; 2) the prototype is built quickly with very little if any concern for its quality; 3) maintenance costs rise dramatically; and 4) a prototype cannot serve as a legal contract.
33. Why is *maintenance* so expensive?
Because defects that have been around this long are very expensive to correct, because original developers are most likely no longer around. Software typically has not been well-documented and well-developed.

34. What is meant by the term *man-month*, and what does Brooks discuss as the problem with its use?
Man-month is a unit of measure for the size of a job.
Men and months are not interchangeable because of the communication that takes place between developers working on the same project.
Thus, additional men will not necessarily reduce time to delivery (although obviously additional women will ☺).
35. What are the three components of a *lifecycle model* and why is each important?
Intermediate software products mark the end of phases, enable reviews, and facilitate visibility;
Phases promote manageability and provide organization;
Reviews ensure satisfaction of requirements.
36. Compare and contrast the *CMM* and *ISO 9000[-3]*? Discuss overall purpose and philosophy of the models as well as the basic emphasis of their approaches.
CMM is a process improvement model with an emphasis on project management;
ISO-9000 is more focused on quality improvement and stresses documenting the process.
37. What is the difference between *verification* and *validation*?
Verification compares [the implementation] to the specification, checking to see if we are building the product right;
Validation compares [the implementation] to user needs, checking to see if we are building the right product.