

Midterm Examination

Information and Computer Science 121

Professor Debra J. Richardson
Winter 1998

Name: _____

Signature : _____

Student ID: _____

Instructions

1. This exam is closed book, closed notes, closed neighbors.
2. The exam has 15 pages (don't panic ... there are lots of pictures).
3. Print your name and student ID number and sign the exam where indicated above. Print your name on each page of the exam.
4. For *true/false* questions, a statement is false unless it is entirely true; circle true or false in the left margin.
5. For the *problems*, draw your solution models neatly in the space provided.
6. For *very brief answer* questions, please respond in just a few words; write your answer neatly in the space provided.
7. For *short answer* questions, please be clear and concise in your answers; write your answer neatly in the space provided.
8. All answers are to be given on these pages. Use the back sides of the pages only if necessary, with a reference as to where any extra portion of the answer is located.
9. The maximum score for this exam is 100 points. Each true/false question is worth 1 point, each problem is worth 7 points, each very short answer is worth 2 points, and each short answer question is worth 4 points.
10. You have one hour and 15 minutes to complete the exam – that is, until 12:15. Plan your time accordingly.

question type	true/false	problem	very short answer	short answer
# of questions	10	4	15	8
worth (%)	1/# = 10	7/# = 28	2/# = 30	4/# = 32
grade (%)				

total grade	%
-------------	---

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

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

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

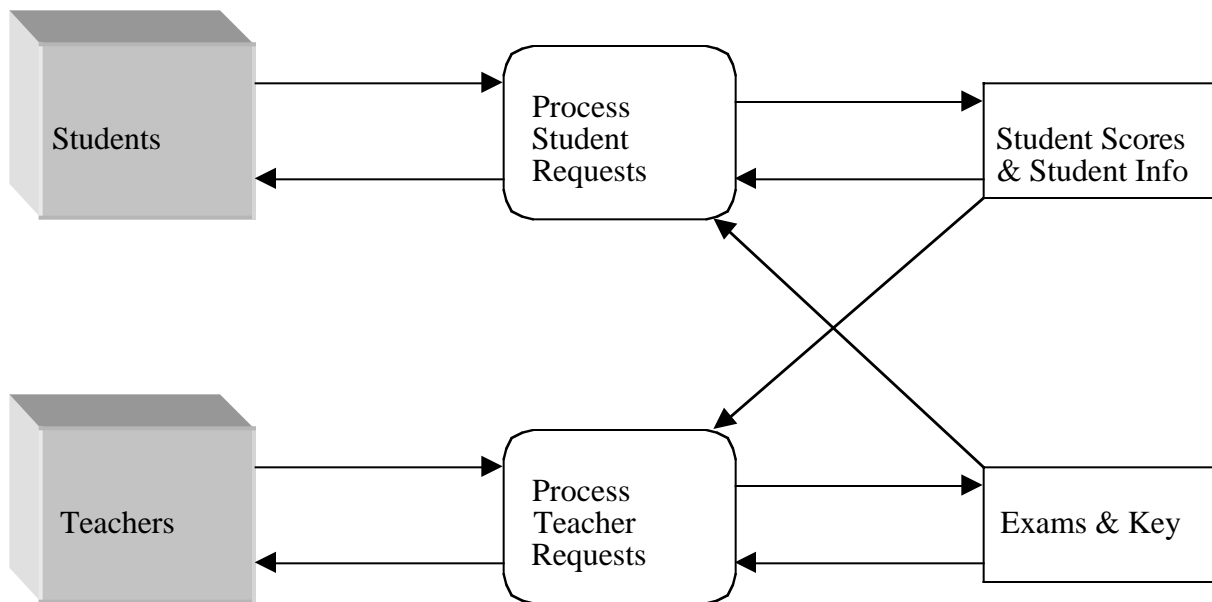
11. Data flow diagrams

Suppose you are tasked to build an Automated Examination System for a high school, with the following desired functionality:

- 1 - Teachers shall input an exam with the key (answers) to the exam;
- 2 - Students shall request an exam to take;
- 3 - Students shall provide answers for a granted exam;
- 4 - The system shall automatically grade the exam;
- 5 - Both teachers and students shall be able to view scores.

Here is an incomplete top-Level data flow diagram for this system. You must create the next iteration DFD showing the functionality listed above. . At this level of iteration, do not worry about logins or security.

NOTE: Each function must be shown as an individual process. Be sure to uniquely label each item in your DFD, including edges. As you can see, the edges in this top level DFD have been left un-labelled. You must decide how to label the edges in the second iteration DFD.



NOTE: there is a blank page immediately following this one for your answer.

This page left intentionally blank (space for problem 11).

12. Entity-relationship diagrams

Suppose you are building an Elevator System to control P elevators serving L floors. Draw an entity-relationship diagram defining the relationship between the Elevator_Controller, Elevators, Elevator_Buttons (the elevator buttons reflect the floor numbers to which an elevator is requested to go), and Floor_Buttons (up and down buttons on the floors outside the elevator used to call the elevator).

NOTE: As stated, assume there are P elevators and L floors.

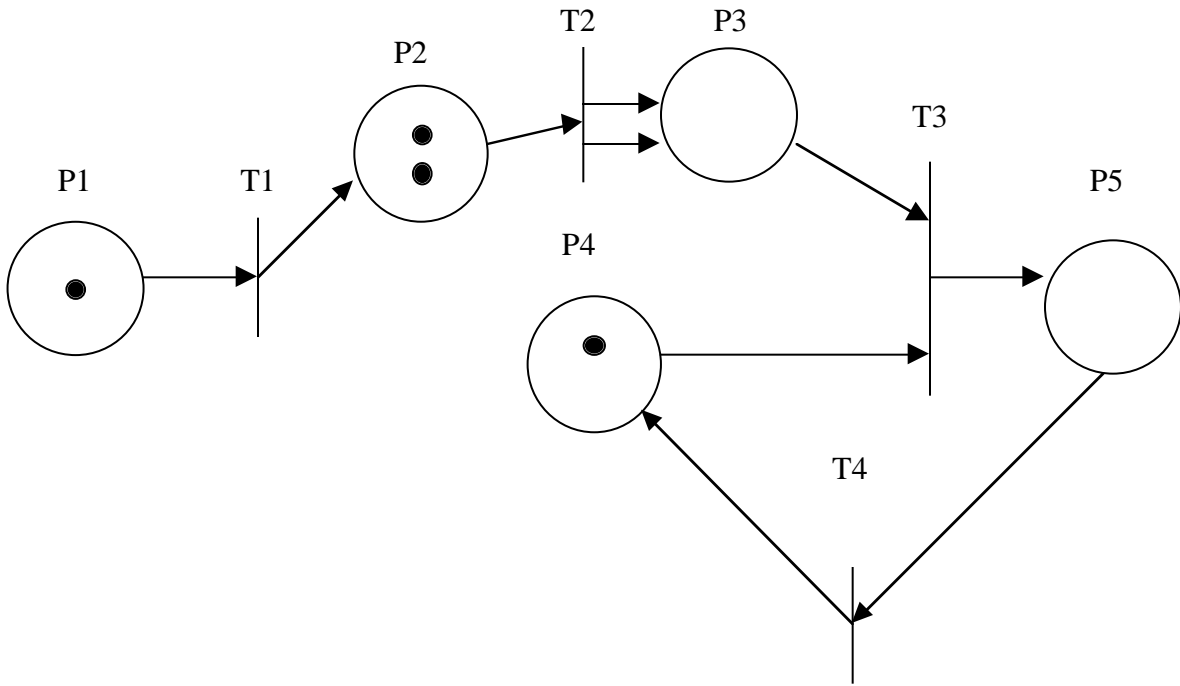
13. Finite State Machine (State Transition Diagram)

Create a Finite State Machine describing a point-of-sale system (POS) at a grocery store. A point-of-sale system allows you to use your ATM card to pay for groceries. Assume that POS starts with the amount of sale, which can be increased by a cash back request. If a valid ATM card and matching PIN are input, the system checks to see that the account has sufficient funds for the transaction (the sale plus any cash back). If approved, the system conducts the appropriate transaction. The transaction is cancelled either by an invalid ATM/PIN, an insufficient account balance, or upon request of the customer at any time.

NOTE: be sure to indicate what are the initial and final states.

14. Petri-nets

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



Can T1 fire?

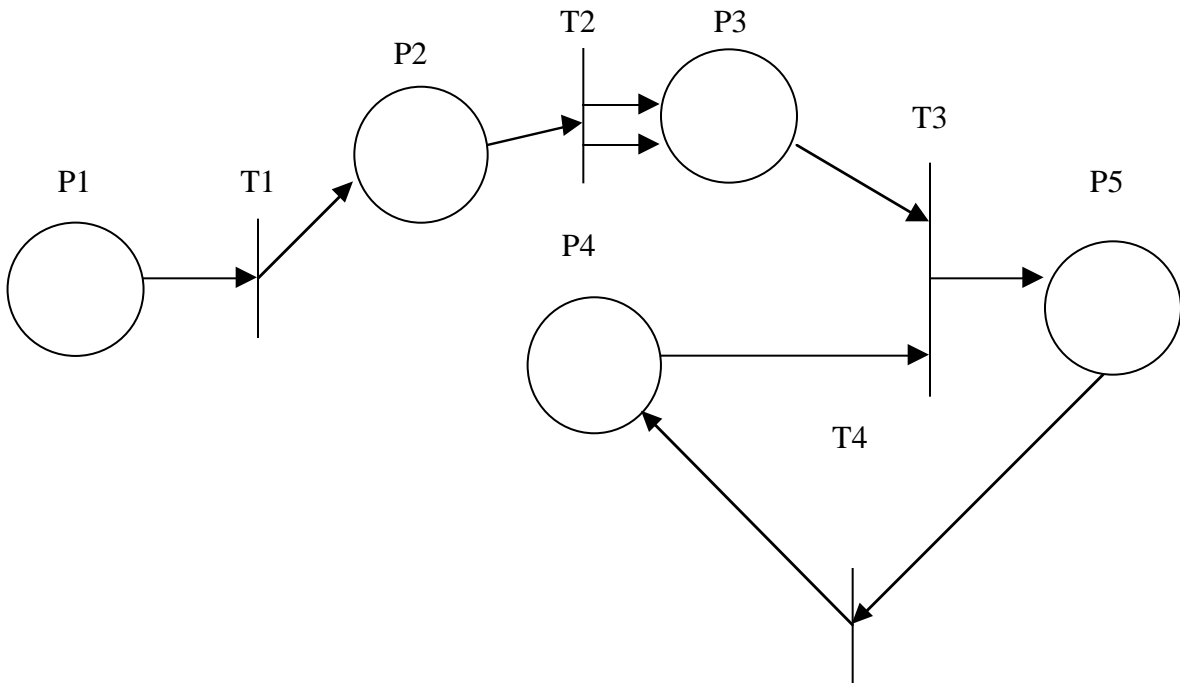
Circle

YES

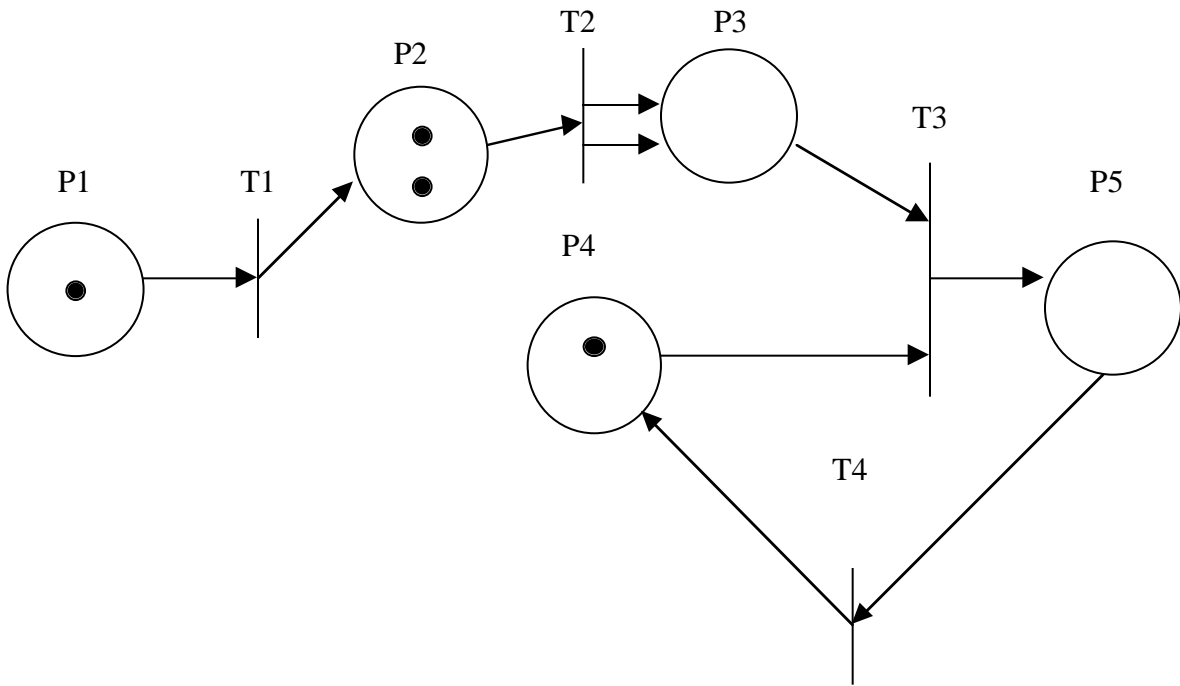
or

NO

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



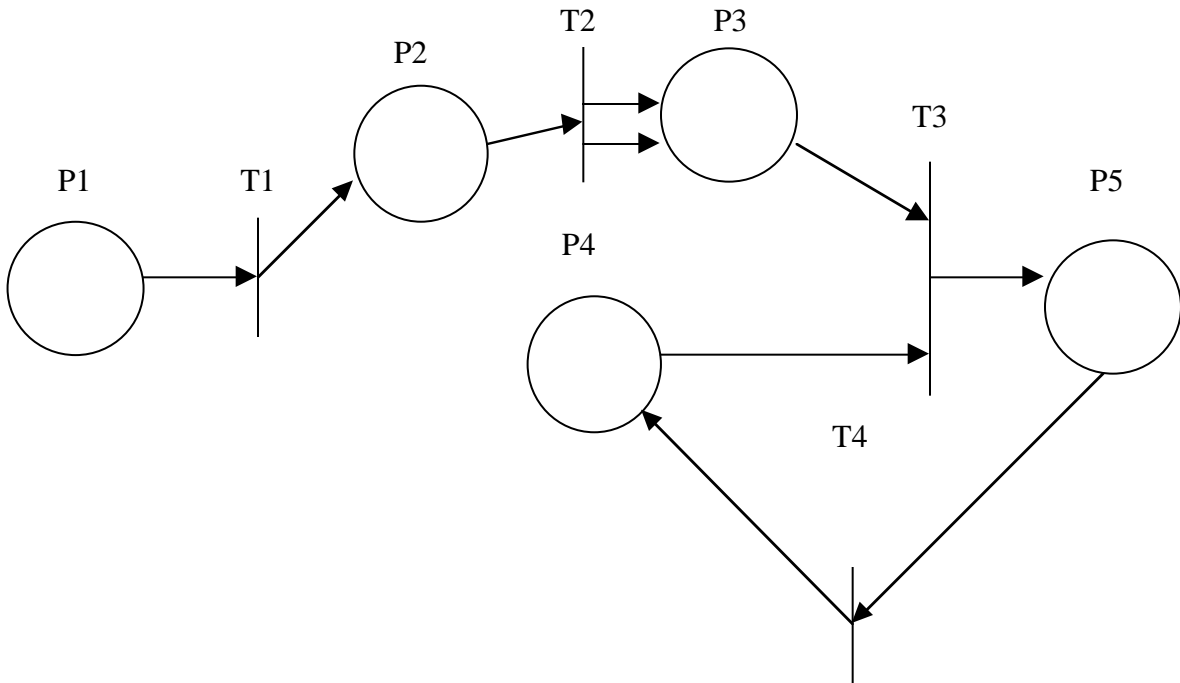
b. Given the initial marking of the Petri-net shown below:



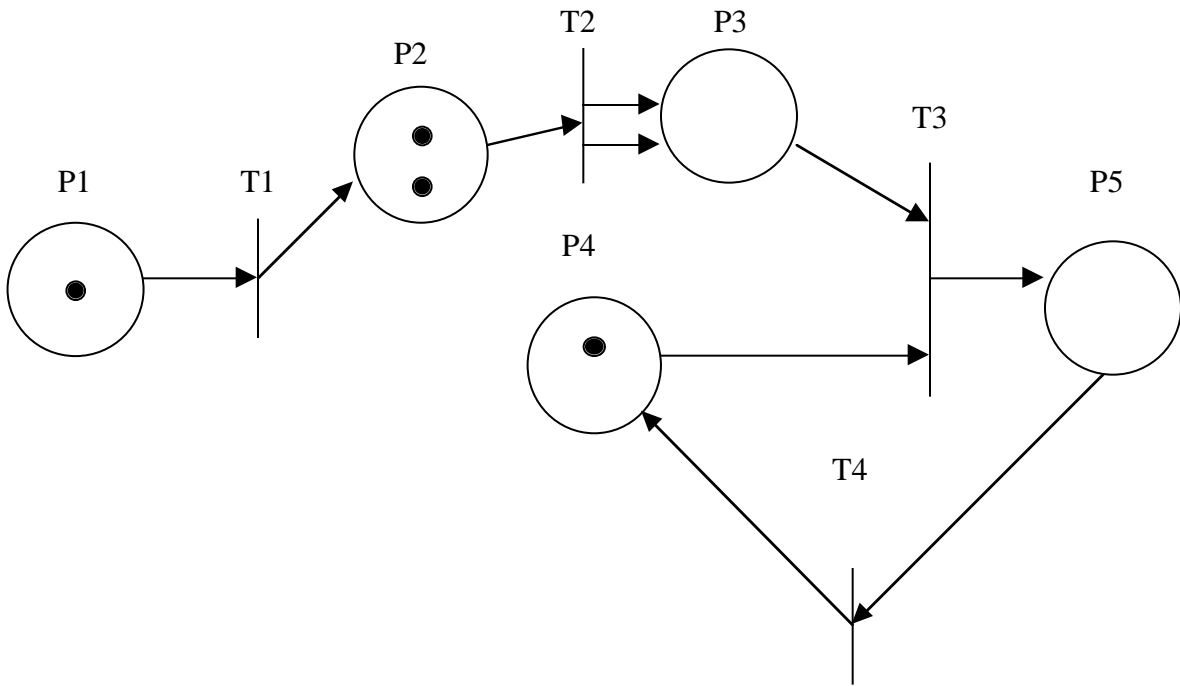
Can T3 fire?

Circle YES or NO

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

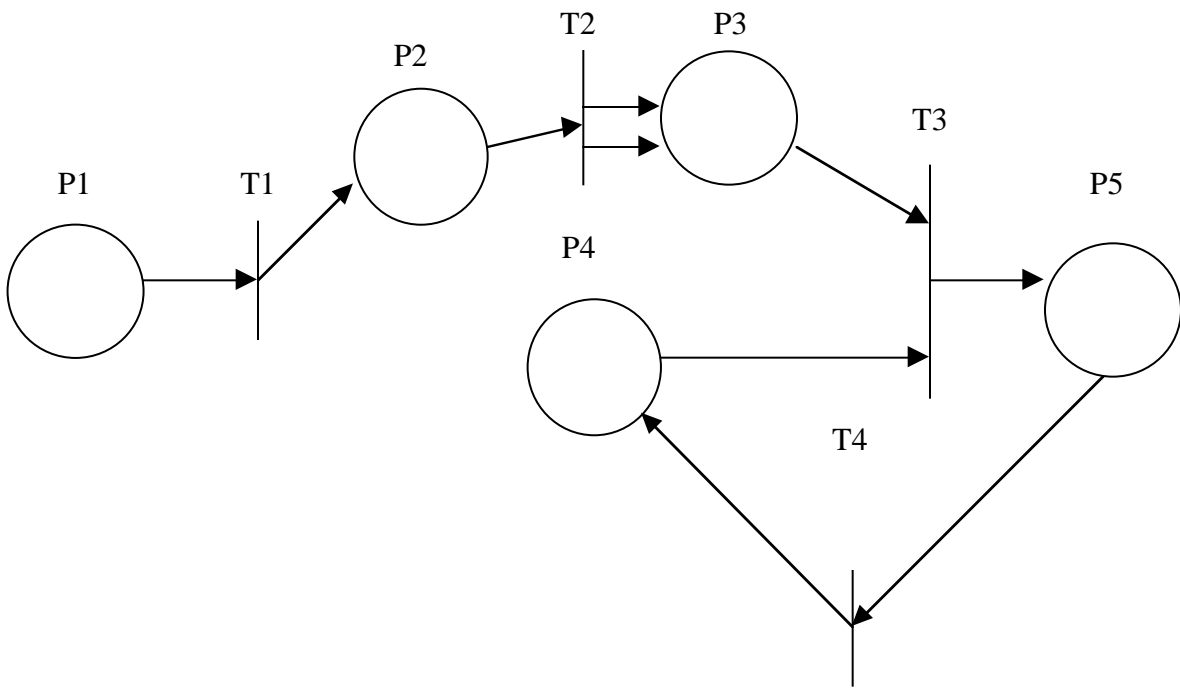


c. Given the initial marking of the Petri-net shown below:

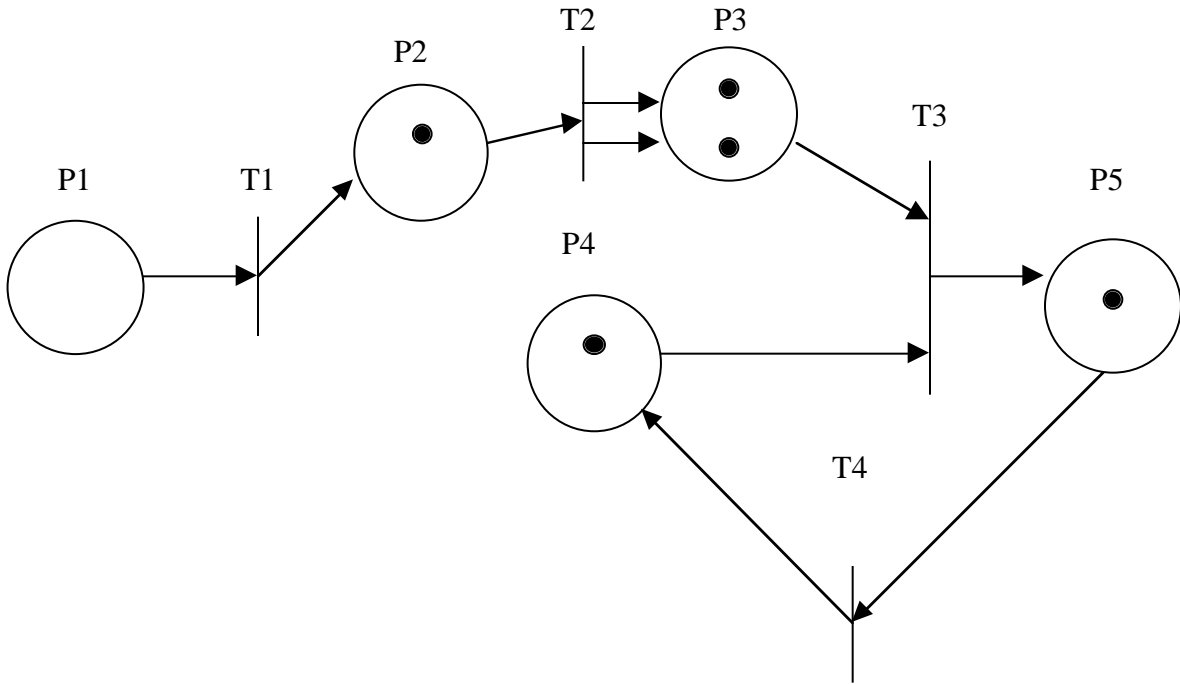


Can T2 fire? Circle YES or NO

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

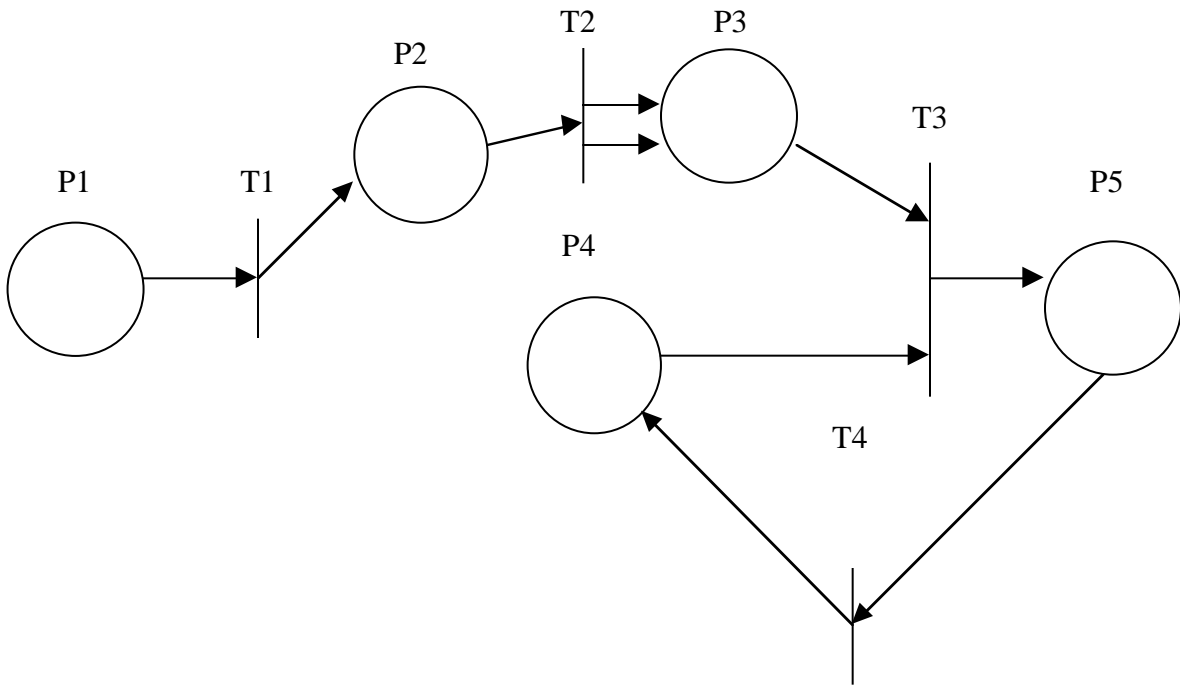


d. Given the initial marking of the Petri-net shown below:

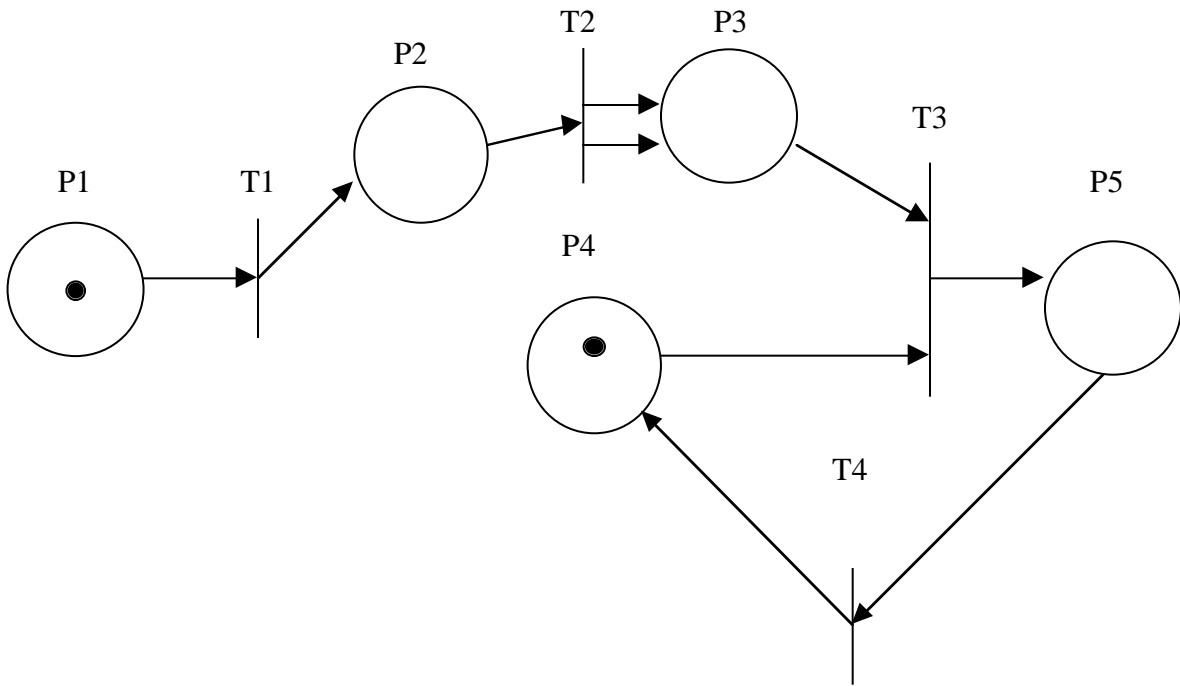


Can T3 fire? Circle YES or NO

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? _____

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*?

16. Which lifecycle phase defines the *acceptance test plan*?

17. What three problems in software production led to the *software crisis*?

18. What activity was added in the *spiral model* that was not present in the *waterfall model*?

19. What is the primary objective of *prototyping*?

20. Requirements should be complete. Name five other desirable characteristics of a *requirements document*.

21. What technique or method is specifically directed toward *usability testing*?

22. User satisfaction is one important attribute addressed by *usability testing*. Name two more.

23. *Human-computer interaction* focuses the system's interface. Name one more foci of HCI.
24. Completeness is one desirable characteristic of a good *design*. Name three more.
25. _____ abstraction is used in both *structured design* and *object-oriented design*, while
_____ abstraction is only used in *object-oriented design*.
26. According to *Brooks*, what is the Number One cause for software products to go awry?
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.
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.
29. What is the primary cause of *inconsistency in design*?

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?

31. Define and differentiate *information hiding* and *abstraction*?

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?

33. Why is *maintenance* so expensive?

34. What is meant by the term *man-month*, and what does Brooks discuss as the problem with its use?

35. What are the three components of a *lifecycle model* and why is each important?

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.

37. What is the difference between *verification* and *validation*?