The University of Texas at Dallas
Computer Science Program

Test 2 April 27, 2000

**Conditions:** Closed book Duration: 70 minutes

**Write legibly; unreadable answers are not answers!**

**Each answer should be concise and precise! Clearly indicate any assumptions**

**Name:** _____

**{Please underline last name}**

**Student Number:** _____

**1.** _____ /30

**2.** _____ /30

**3.** _____ /10

**4.** _____ /10

**5.** _____ /10

**6.** _____ /10

**Total** _____ /100

**1. [30 marks]** **page 2/11**

For each of the following ten statements, indicate whether it is true (mark T) or false (mark F). (No penalty for a wrong answer)

__T__ This is the second test for CS6361.501.

___ **1.** As opposed to structural requirements analysis, behavioral requirements analysis involves the inputs to the system, the outputs from the system, and structural relationships between them.

___ **2.** Entity Relationship Model was originally proposed as an extension of relational data model in order for optimizing algorithms and analyzing their computational complexity.

___ **3.** When using IDEF0X, the first step in the development process is the construction of a model of the system to be built, consisting of entities, their life states and actions.

___ **4.** Unlike Petri-net, a StateChart is an abstract machine which can be in more than one state at any given time.

___ **5.** In a product-oriented approach to dealing with non-functional requirements, the central concern is how to integrate the enforcement of NFRs with the development process.

___ **6.** An evolutionary prototype is done quickly, hence usually showing poor performance due to the use of, for example, inefficient data structures and algorithms.

___ **7.** Unlike many other object-oriented methodologies, a novel feature of a formal object-oriented requirements modelling language such as RML/Telos is the treatment of relationships between objects also as objects, hence increasing expressiveness and extensibility of the language.

___ **8.** In the bottom-level of SADT, non-functional requirements should be denoted by diamonds, and clearly explained in a Data/Requirements Dictionary.

___ **9.** Unlike decision tables, finite state machines do not force a requirements engineer to consider operational characteristics of the needed algorithm such as repetition of common actions and do-while loop.

___ **10.** One of the meta-NFRs on both functional and non-functional requirements is "non-ambiguity", i.e., a requirement cannot be interpreted in more than one way.

___ **11.** Conceptual modelling is a key to conquering complexity inherent in just about any reasonably large software systems, and it encourages modelling things as "perceived" by the programmers, hence involving both the knowledge and cognitive aspects.

___ **12.** As often done in SADT, the primary purpose of enterprise modelling is to find ways to effectively automate the software development process.

___ **13.** In the original ERD, the central concern was the uniformity in organizational primitives such as classification/instantiation, aggregation/decomposition, and specialization/generalization.

___ **14.** In using Statecharts, the first step is the construction of a model of the (projected) system by the use of teleological reasoning.

___ **15.** Unlike functional requirements, non-functional requirements are used to describe the structure of information to be maintained by the system and how such structure helps meet the goals of the agents in the intended application.

**2. [30 marks]** 

Circle the best answer to each of the following questions.

**1.** Which is *not* included in the intellectual origins of Object-Oriented problem analysis?

1. formal specification
2. knowledge representation
3. mobile agents
4. information modelling
5. programming languages

**3.** Which is the *least* relevant to non-functional requirements?

1. contextuality;
2. subjectivity
3. events
4. softgoals
5. satisficing

**4.** Which is *not* relevant to RML/Telos?

1. axiomatization of epistemological primitives
2. treatment of assertions as objects
3. interval calculus
4. behavioral modelling
5. theory of types

**4.** Which is *not* relevant to StateCharts?

1. reduction in the state space explosion
2. modeling of concurrent processes
3. multiple tokens
4. organization of processes via Superstates.
5. default entry point

**5.** Which is *not* relevant to SADT?

1. ambiguities in temporal relationship between inputs and outputs
2. agent-oriented modelling
3. a sub-box as an instance, attribute, sub(super-)class of its parent box
4. actigrams and datagrams
5. maximum 6 (sub-)boxes

**6.** Which is *not* relevant to Church's theory of types?

1. self reference
2. logical paradox
3. infinite hierarchy of types
4. probabilistic reasoning
5. the set of all sets

**7.** Which is *not* the role of non-functional requirements?

1. selection criteria in mapping functional requirements into a design.
2. provision of guidance for process requirements specification
3. an important part of the basis for software design.
4. proving the completeness of test data
5. rational design

**8.** Which is *not* relevant to requirements validation?

1. completeness of functional requirements and non-functional requirements
2. prototyping
3. consistency between enterprise models and system models
4. measuring the degree to which non-functional requirements are satisfied
5. inspection

**9.** Which *is* considered as the central difference between IDEF0 and IDEF1X?

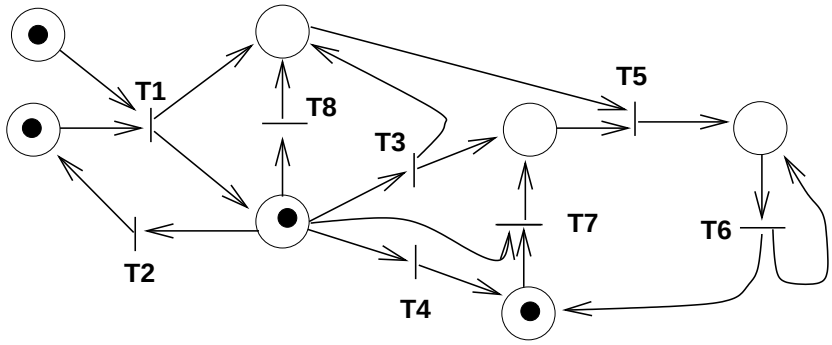
1. functional requirements vs. non-functional requirements
2. finite states vs. infinite states
3. composition vs. aggregation
4. activity modelling vs. information modelling
5. data modelling vs. knowledge modelling

**10.** Which is *not* relevant to formal semantics in the context of requirements engineering?

1. reasoning
2. consistency checking
3. derivation of new facts
4. modus ponens
5. first order logic

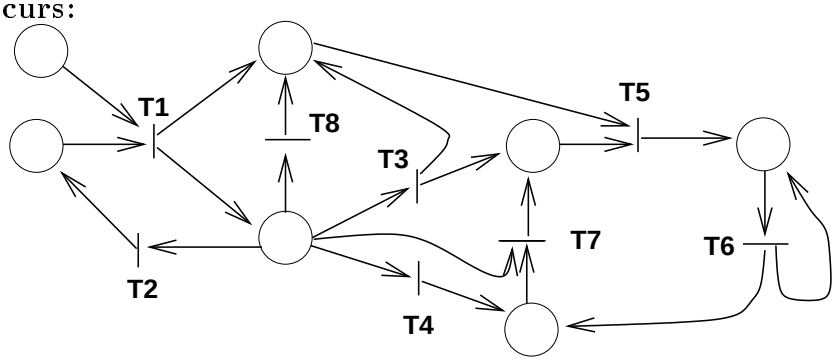
**3. [10 marks]** 

Consider the following Petri-net:

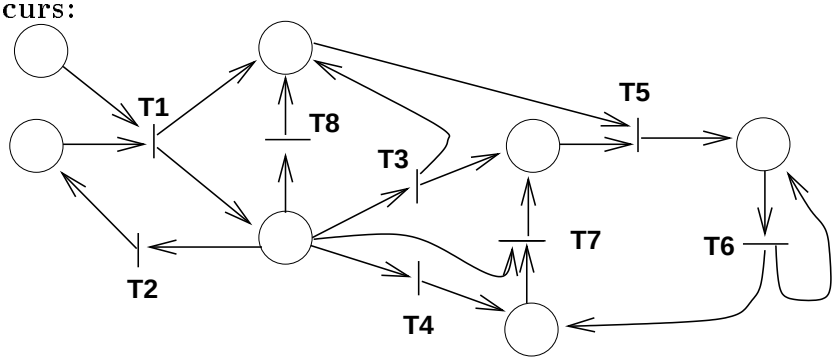



Suppose that stimuli occur in the following order: **T1**, **T3**, **T5**, **T6**, **T7**. Show the configurations that the above Petri-net goes through according to the stimuli, taking the simplified convention as discussed in class.

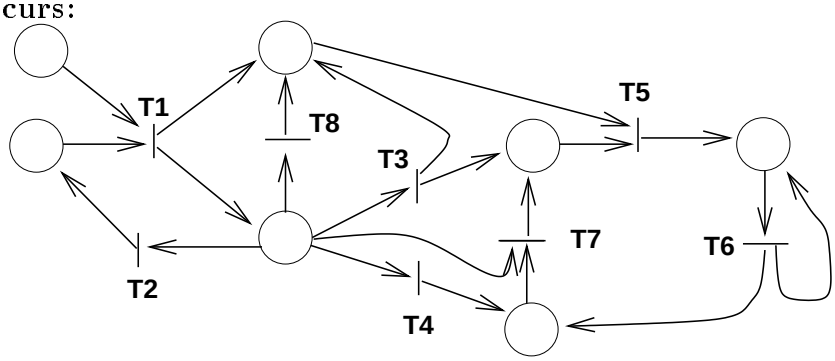
**After T1 occurs:**



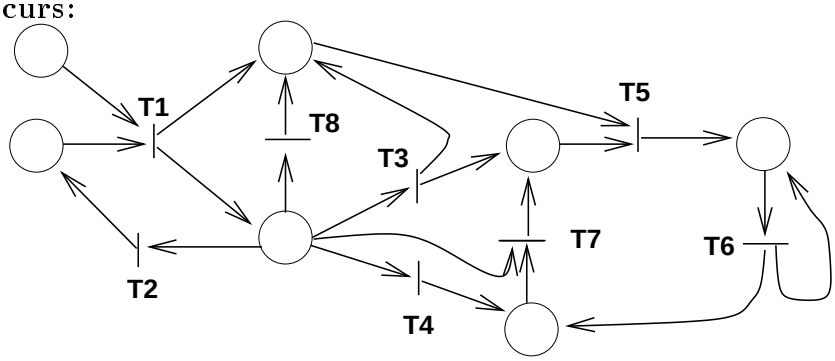

**After T3 occurs:**



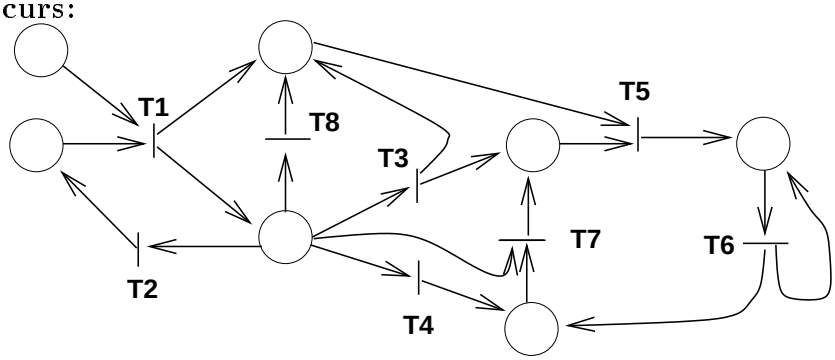
After T5 occurs:




After T6 occurs:




After T7 occurs:



**4. [10 marks]** 

Consider the following definitions:

| | |
|---|---|
| *IN (i, C, t)*: | predicate asserting that token $i$ is an instance of class $C$ at time $t$; |
| *IS-A ($C_1$, $C_2$)*: | time-independent predicate asserting that class $C_1$ is a subclass of class $C_2$; |
| *PROFDEF (C, a)*: | time-independent function which gives the class to which the value of attribute $a$ for instances of $C$ must belong. |

- **4.1 [5 marks]** Using the predicates above, axiomatize the following:

  Each instance of a metaclass is also in the superclass of the metaclass.

- **4.2 [5 marks]** Using an example, precisely describe what the following axiom means:

  *(PROPDEF (C, a) = E ∧ IS-A (D, C))*
  *⇒ ∃ F [PROPDEF (D, a) = F ∧ IS-A (F, E)]*

Consider the following process of dealing with some non-functional requirements:

> The developer states the non-functional requirement *"Accounts should be secure"*, refines it into three sub-goals, for Integrity (guarding against unauthorized update or tampering), Confidentiality (guarding against unauthorized disclosure), and Availability (guarding against interruption of service) of the account. The developer further refines the Integrity goal into two sub-goals, for Accuracy and Completeness of the account.
>
> Now focusing on the Confidentiality Requirement in moving towards a secure target design, the developer considers a list of confidentiality assurance techniques and decides to use an Authorization Technique. Repeating this process, the Authorization Technique is further refined to goals for Identification, Authentication, and Access Rule Validation. At this moment, the developer notices that Access Rule Validation has a positive impact on the accuracy of accounts, as ill-intentioned users can be denied access and prevented from committing forgery. The developer now further refines Authentication into a goal for "Require additional ID", but feels that this goal would have a negative impact on User-friendly access. Although User-Friendly access hasn't yet been posted, it turns out that it is fairly important and should have been posted in the very beginning. So, the developer now posts it as a goal, although an Authorization may be seen negatively towards User-Friendly access if the Authorization involved something personal.

- **5.1. [5 marks]** Using a dependency graph (softgoal interdependency graph), represent the process described above as precisely as possible.

- **5.2 [3 marks]** Suppose we have a decision table with 3 input conditions, where the first input condition can have 5 different values, the second can have 4, and the third can have 3. Also suppose that the decision table has 5 different actions. Compute the maximum number of rules.

- **5.3 [2 marks]** Explain the differences between metaclasses and classes using set theory. Give an example also.

**6. [10 marks]** 

**Precisely and concisely** describe

**6.1 [7 marks ]** what *uniformity* means in object-oriented behavioral requirements modelling which uses the Augmented Petri Net (APN) formalism.

**6.2 [3 marks ]** the major differences between sets and classes. Provide an example.