Final Exam -- ELG 7187C

Winter 2007, April 17 (17:30 – 20:30)

No documentation allowed

Student's name: _____ Id. Number: _____

Please answer:

(A) 5 questions out of the questions 1 through 6 (18 points each), and

(B) 2 questions out of the questions 7 through 11, but not the one related to your project (5 points each)

Question 1: Requirements and Activity diagrams

(a) One usually distinguishes between functional requirements and non-functional requirements. Give three examples of non-functional requirements.

(b) Is the formalism of finite state machines a formalism that is suitable to describe non-functional requirements ? – Please explain in a few words. If your answer is negative, please indicate whether certain extensions of finite state machines make the formalism suitable for describing certain non-functional requirements.

(c) Please design and write down an Activity Diagram that represents the process of getting accepted as a graduate student at your university (UofO or UCarleton). Note: Concerning the notation of Activity Diagrams you should follow the example given in Annex 1.

Question 2:

Part A: We consider the following information (of a car vendor) to be stored in the form of XML: For each client of the car vendor, the name, address and telephone number; as well as the cars that were sold during the last 5 years (name of manufacturer, name of the model, and year of sale) and information about which client bought which car (if any).

A DTD for this information is given below:

```
<!ELEMENT clients (client*)>
<!ELEMENT client (name, address, phone, car*)>
<!ELEMENT car (manuf, model, year)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT address (#PCDATA)>
<!ELEMENT phone (#PCDATA)>
<!ELEMENT manuf (#PCDATA)>
<!ELEMENT model (#PCDATA)>
```

```
<!ELEMENT year (#PCDATA)>
```

Please complete the following XML document in such a form that it conforms to the DTD above and represents the client John Hopps with the address 300 Stewart, Ottawa, and telephone 613 434 5667 who has bought two cars, namely a Maxima built by Nissan in 1999 and a Fiesta built by Ford in 2000.
XML document:

```
<?xml version='1.0'  encoding='ISO-8859-1' ?>
```

Part B: We consider the following information (of a construction company) to be stored in the form of XML: For each employee, characterized by his name and employee number, indicate on which projects he has worked and during which time interval (starting date and ending date). Please write down a DTD for an XML document that represents this information for all employees of the company.

Question 3: Labeled transition systems

Figure 1 shows a labeled transition system (LTS) A with the alphabet {a, b, c} .

(a) A is non-deterministic. Please write down a deterministic LTS B that is equivalent to A. First explain what it means that B is equivalent to A. Then write down your LTS B.

(b) Figure 1 also shows another LTS C. Let us assume that the LTSs A and C perform their actions in rendezvous. Write down an LTS that represents the behavior of A and C executing in rendezvous.

(c) Same question as under (b), but now we assume that the transition from 3 to 1 labelled c in C does not exist. Is there any problem in the behavior of A and C performing in rendezvous ?

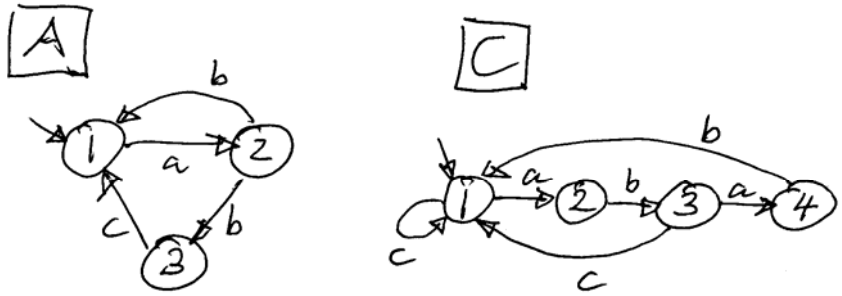


Figure 1

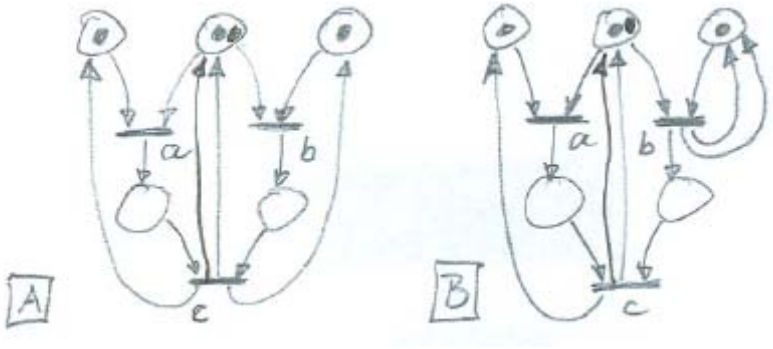
Question 4: Petri nets (17 points)

The figure below shows two Petri nets A and B which are quite similar. The transitions are labeled a, b and c.

(a) What are the possible sequences of transition executions that could be performed by Petri net A ? – Please explain in words.

(b) Write down an LTS with transitions labeled a, b and c such that the possible execution sequences of the LTS are the same as the sequences that can be performed by Petri net A.

(c) How does Petri net B compare with Petri net A ? – Please discuss: possible sequences of transition executions, and other properties.

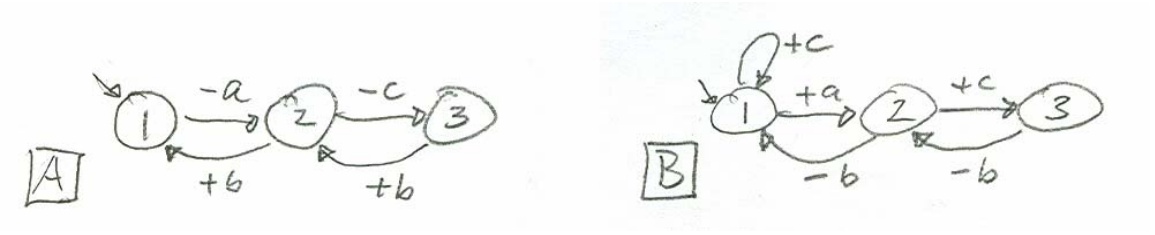


Question 5: Reachability analysis

(a) Below are the specifications of two protocol entities A and B in the form of two IOA. Does the specification of entity A make any assumptions about its environment ? - Does the specification of entity B make any assumptions about its environment ? – Please explain.

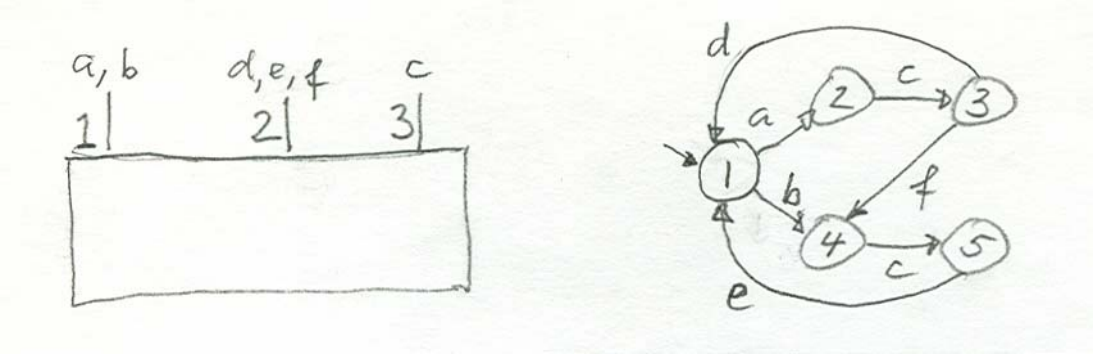
(b) Please, perform a reachability analysis for the two protocol entities A and B, assuming that they communicate with one another through buffered message passing. The initial states are the states labeled 1.

(c) Are there any design flaws for these protocol entities ? -- Please explain. How would you change the definition of these protocol entities in order to eliminate the design problems identified ?

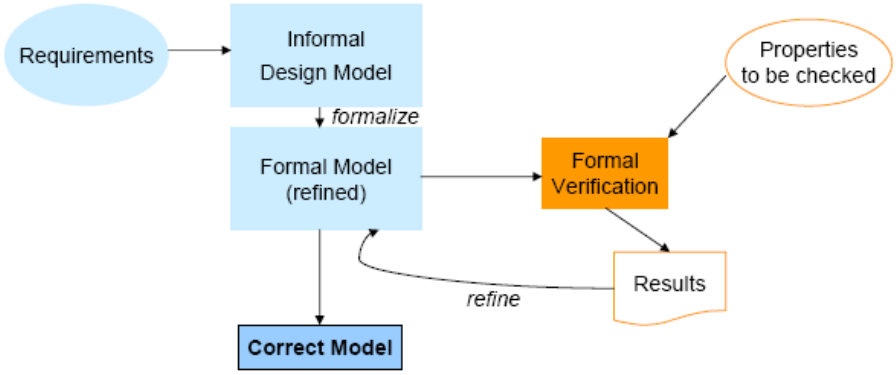


Question 6: Protocol derivation

Please design a protocol including 3 entities that realize the communication service defined below. Please write down the behavior of the three entities M1, M2, and M3 that perform the operations of the protocol at the access points 1, 2 and 3 shown in the architectural diagram below. As indicated in the figure, the interactions a and b take place at access point 1, d, e and f take place at access point 2, and c takes place at access point 3.


a, b
d, e, f
c
1
2
3
d
a
2
c
3
1
b
f
4
5
c
e

Question 7 (Related project "3-party-conference in SDL (Siyuan He)" : Explain in a few words what the meaning of the following diagram is.



Question 8 (Related project " Modeling with LTSA (Alexander Kutman) " : Explain in a few words what the behavior of COUNTDOWN [2] is. See definition on the slide below.

- A countdown timer which beeps after N ticks, or can be stopped (adapted from [2])

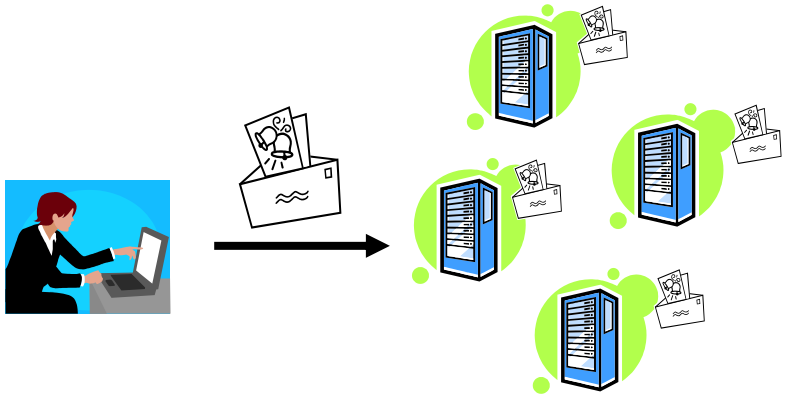
```
COUNTDOWN (N=3)   = (start->COUNTDOWN[N]),
COUNTDOWN[i:0..N] =
        (when(i>0) tick->COUNTDOWN[i-1]
        | when(i==0) beep->STOP | stop->STOP)
```

```
Process:
        COUNTDOWN
States:
        6
Transitions:
        COUNTDOWN = Q0,
        Q0              = (start -> Q1),
        Q1              = (tick -> Q2
            |stop -> Q5),
        Q2              = (tick -> Q3
            |stop -> Q5),
        Q3              = (tick -> Q4
            |stop -> Q5),
        Q4              = ({beep, stop} ->
        Q5),
        Q5              = STOP.
```

- *COUNTDOWN*: process (sequence of actions)
- *N*: process parameter
- *[i:0..N]:* process variable
- *start, stop, tick, beep:* actions
- *when(i>0):* guarded action
- *beep->STOP* | *stop->STOP* : choice
- *STOP*: elementary FSP process - deadlock

Question 9 (Related project " Modeling the practical Byzantine fault tolerance service with SDL (Ying Qiao) " : Explain in a few words what the purpose of this Byzantine Fault Tolerant database service is and against what kinds of faults (in the communication medium and in the servers) the protocol of this service must be tolerant.

- State machine approach
- Number of server nodes $\geq 3f+1$



Question 10 (Related project " Modeling CSMA-CA in SDL (Kambiz Frounchi) " : Explain in a few words the terms used in the following slide (such as "deadlock", "unspecified receptions", "other errors", "performing bit-state exploration", etc.

Validation and Simulation

- Models are validated using the Telelogic SDL suite validator to avoid :
  - Deadlocks
  - Unspecified Receptions
  - Other errors
  - Validation done by:
    - Performing bit-state exploration
    - Verifying MSCs that demonstrate the expected behavior of the system
- The Telelogic SDL suite simulator is used to obtain the metrics
  - Closed-system simulation

Question 11 (Related project " Aircraft Landing Protocol in Alloy (Panesar-Walawege)" : Explain in a few words the meaning of "leader" based on the definition given in the lower part of the following slide:

Examples

- Signature: sig Aircraft {mahf : Sides, leader : Aircraft}
- Fact:

```
fact {all x: zone map |
  x.aircraftsQueue.noDuplicates
  }
```

- Predicate:

```
pred leader [a,b : Aircraft , queue : AircraftSequence]
{ not AircraftSequence.isEmpty
a in AircraftSequence.elems
b in AircraftSequence.elems
a = (AircraftSequence.prev[b])
}
```

I hope you enjoyed the course. I wish you a nice summer !

Annex: 1

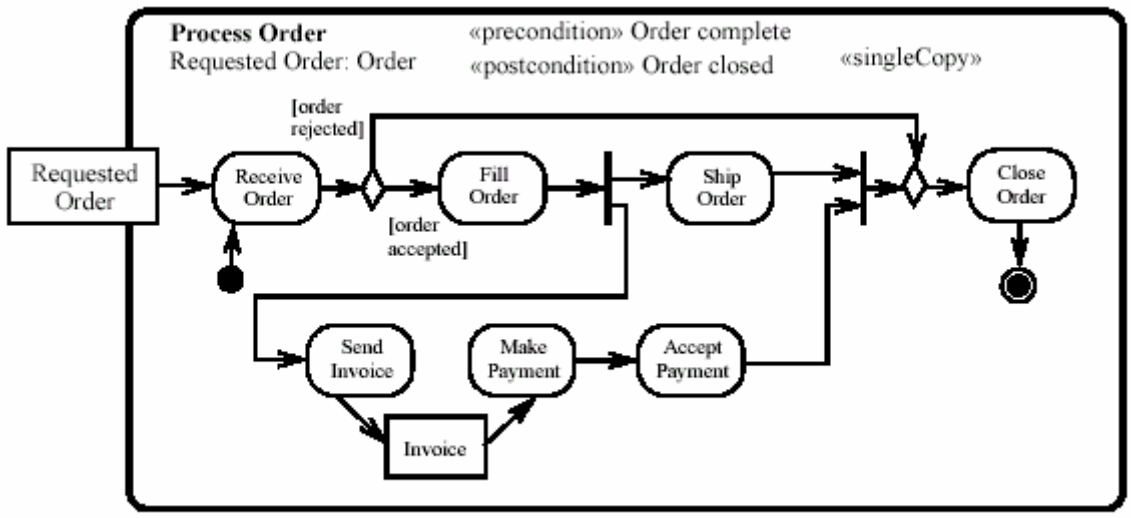


Figure 6-29. Example of an activity with input parameter.