UNIVERSITY OF LONDON
IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 2004

MSc in Computing Science
for Internal Students of the Imperial College of Science, Technology and Medicine

# PAPER M1

# PROGRAM DESIGN AND LOGIC

Thursday 13 May 2004, 10:00
Duration: 120 minutes

*Answer THREE questions*

Paper contains 4 questions
Calculators not required

***Section A*** *(Use a separate answer book for this Section)*

1a Formalise the following sentences in predicate logic using the predicates:

| | |
|---|---|
| onboard(X, Y) | to mean person X was onboard ship Y |
| journey(S, Dest, D) | to mean ship X sailed towards destination Dest on date D |
| private_cabin(X) | to mean X had a private cabin |
| passenger_ship(X) | to mean X was a passenger ship |

*and* any additional predicates you find necessary.

i) Titanic was a passenger ship that sailed form England to the US on 10 April 1912. This was its one and only journey. Not all the passengers reached the US.

ii) The passengers on the Titanic were in the first class, second class or third class. Everyone in first class had a private cabin. Some, but not all, in second class had private cabins. None in third class had private cabins.

b

i) Show $\neg A \rightarrow B \vdash A \vee B$
using syntactic techniques only. At each stage of the proof clearly state the inference rule and the wffs used.

ii) Consider sentences S1-S5 below

S1 $\forall X (\neg P(X) \rightarrow Q(X))$
S2 $\forall X (P(X) \rightarrow \exists Y\ R(X,Y))$
S3 $\forall X (Q(X) \rightarrow S(X))$
S4 $\forall X \neg (S(X) \wedge T(X))$
S5 $\forall X (T(X) \rightarrow \neg\exists Y\ R(X,Y))$

Show S2, S5 $\vdash P(a) \rightarrow \neg T(a)$, where a is a constant,
using syntactic techniques only. At each stage of the proof clearly state the inference rule and the wffs used.

iii) Show S1,S2,S3,S4,S5 $\vdash \forall X \neg (T(X))$
using only syntactic techniques and, if required, the results of b(i) and b(ii), above. Sentences S1-S5 are as given above. At each stage of the proof clearly state the inference rule, or b(i), b(ii), and the wffs used.

*Parts a and b carry 45% and 55% of the marks, respectively.*

2 a

i) Show $(A \rightarrow B) \rightarrow C \equiv (\neg A \rightarrow C) \wedge (B \rightarrow C)$ using any semantic or syntactic techniques except truth tables. Explain your reasoning at every step.

ii) Show $\neg(M \wedge N) \vee P \rightarrow Q,\ Q \rightarrow R \ \vdash\ P \rightarrow R$ using a(i), above, if you wish, and any semantic or syntactic techniques except truth tables. Explain your reasoning at every step.

iii) Consider the action of a person borrowing a book from a library, *borrow(Borrower, Book)*. Express the preconditions and postconditions of this action in the situation calculus. You can assume that the library is known and you do not need to mention it explicitly in your formalisation. For a borrower to be able to borrow a book, they must be a member of the library and the book must be in stock. You can assume that the only properties we are interested in are library membership and what is in stock. You can also assume that this particular library has no more than one copy of any book in stock, and that membership is not affected by the action of borrowing.

b Suppose a Prolog database contains facts for the relations:

| | |
|---|---|
| student(S) | S is the name of a student |
| enrolled(S,C) | S is enrolled on course C |
| lectures(L,C) | L lectures on course C |
| lecturer(L) | L is a lecturer |
| topic(C,T) | C is a course with some lectures on topic T |

Assume it gives complete information about these relations.

i) Give the Prolog query to find a student who is enrolled on course 'C123' and is not enrolled on course 'C321'.

ii) Give the Prolog query to find a lecturer who lectures on at least two different courses (Hint: use \= to check that the courses are different.) What does it tell you about the data if this query immediately fails, i.e. returns no answers?

iii) Define the relation: prog_qualified(S)
Meaning: Student S is enrolled on a course that includes 'C++' or 'Java' as a topic

iv) Define the relation: taught_by(S,L)
Meaning: There is a course C that student S takes on which L lectures

v) Define the relation: one_lecturer(C)
Meaning: C is a course for which there is exactly one lecturer

*Parts a, b each carry 50% of the marks.*

**Section B** (*Use a separate answer book for this section*)

3 Consider the following simplified description of cinema booking:

- A film has a title (string), a basic price (float), and a rating (universal or restricted).
- A cinema has a price factor (float), a maximal number of seats (int), a bank account, a schedule determining the film to be played on a given date and time, and keeps track of the number of seats available on a given date and time.
- A seat may be booked from a cinema, for a date and time, by giving a credit card, and the age of the customer:

  No action is taken if a) no film is scheduled for that date and time, or b) the customer is under 12 years old, and the film for that date and time is restricted, or c) no seats are available for that date and time.

  Otherwise, the number of seats available for that date and time is decremented by 1. If the customer is under 12, then half the film price multiplied by the cinema price factor is debited from the credit card. If the customer is 12 or above, then the film price multiplied by the cinema price factor is debited from the credit card.

a Write C++ class declarations (i.e. no function bodies) to support the above.
You may assume the following predefined classes:

```
template <class T1, class T2>
class Table { // a look up table associating T1-s to T2-s
public:
   Table();                    // an empty table
   void set(T1 x, T2 y);
        // inserts an association form x to y into the table
        // overwrites any previous association of x
   T2* get(T1 x);
      // returns the T2 associated with x in the table
      // returns NULL if no associated element to be found
   ....}
class Account{ ... }
class CreditCard{
public:
    debit(Account& a, float amount); // pays amount into a
    ...  }
class DateAndTime{ ... }
```

b Write a test function, where, assuming that `acc1` is an `Account`, `dt1` is a `DateAndTime`, and `cc` is a credit card:

i) `f1` is a film called `Internal Affairs`, priced at `33.3`, with restricted rating,

ii) `c1` is a cinema with account `acc1`, price factor `2.0`, and `300` seats,

iii) film `f1` is scheduled in cinema `c1` on date and time `dt1`,

iv) a seat for an `11` year old is booked in `c1` on date and time `dt1` using credit card `cc`, a seat for a `22` year old is booked in `c1` on date and time `dt1` using credit card `cc`.

c Write the body of the function implementing booking (ignoring concurrency issues).

*The three parts carry, respectively, 40%, 20%, 40% of the marks.*

4 Consider the following simplified description of a mobile phone billing system:

- A mobile phone has a number and an associated payment account, which may be "prepay" or "contract". Contract accounts have an outstanding balance and a discount rate (expressed as a percentage from 0 to 100); any amounts billed to a contract account are discounted by this percentage and added to the outstanding balance. Prepay accounts have an available balance; any amounts billed to a prepay account are subtracted from the available balance (unless the available balance would drop below zero, in which case the billing action is rejected).
- Mobile phones can send messages, of which there are two types: text messages and video clips. Both types of messages are associated with a destination phone number. Text messages have a text payload which may be up to 1600 characters long; they are billed at 12p for every 160 characters in the message. Video clips have an associated video clip filename; they are billed according to the duration of the video (25p for every second of video). The maximum cost of any message is configurable, and is initially set to be 100p.
- When a mobile phone tries to send a message, it first attempts to bill the cost of the message to its associated payment account. If this is successful, the message is sent, otherwise the message is rejected. A mobile phone can be used to view the balance of its associated payment account.

a Draw a UML class diagram to describe the above.

b Write C++ class declarations (i.e. no function bodies) to support the above.

c Write a test function where:

- Buffy's mobile has number 0777666555 and a contract payment account giving a 20% discount.
- Willow's prepay mobile has number 0777888999.
- Buffy's mobile sends the text message "HELP!" to 0777888999.
- The maximum message cost is set to 120p.
- Willow's mobile sends the video clip "vampire_slaying_tutorial.mpg" of length 50 seconds to 0777666555.
- Willow's phone requests the balance on its account.

d Write the bodies of the functions from part (b).

*The four parts carry, respectively, 25%, 25%, 20%, and 30% of the marks.*