COMPUTER SCIENCE MOCK (King's) Part IA

NATURAL SCIENCES TRIPOS Part IA (Paper CS/1)

January 2021 0.00 to 0.00

COMPUTER SCIENCE MOCK EXAM (KING's) 1

*Answer **one** question from each sections.*

*Submit the answers in five **separate** bundles, each with its own cover sheet. On each cover sheet, write the numbers of **all** attempted questions, and circle the number of the question attached.*

You may not start to read the questions printed on the subsequent pages of this question paper until instructed that you may do so by the Invigilator

STATIONERY REQUIREMENTS
Script paper
Blue cover sheets
Tags

SPECIAL REQUIREMENTS
Approved calculator permitted

SECTION A

1 Foundations of Computer Science

Three alternative representations for non-negative integers, n, are:

- **Peano**: values have the form `S(... S(Z) ...)`, applying `S` n times to `Z` where `S` and `Z` are constructors or constants of some data type.
- **Binary**: values are of type `bool list` with 0 being represented as the empty list, and the least-significant bit being stored in the head of the list.
- **Church**: values have the form `fn f => fn x => f(... f(x) ...)`, applying `f` n times to `x`

(a) Write ML functions for *each* of these data types which take the representation of an integer n as argument and return n as an ML `int`. [6 marks]

(b) Write ML functions for *each* of these data types which take representations of integers m and n and return the representation of $m + n$. Your answers must *not* use any value or operation on type `int` or `real`. [*Hint:* you might it useful to write a function `majority: bool*bool*bool -> bool` (which returns true when two or more of its arguments are true) and to note that the ML inequality operator '`<>`' acts as exclusive-or on `bool`.] [10 marks]

(c) Letting *two* and *three* respectively be the Church representations of integers 2 and 3, indicate whether each of the following ML expressions give a Church representation of some integer and, if so what integer is represented, and if not giving a one-line reason.

(i) *two three*

(ii) *three two*

(iii) *two* $\circ$ *three*

(iv) *three* $\circ$ *two*

[4 marks]

2 Foundations of Computer Science

(a) We are interested in performing operations on nested lists of integers in ML. A nested list is a list that can contain further nested lists, or integers. For example: `[[3, 4], 5, [6, [7], 8], []]`

We will use the datatype:

```
datatype nested_list = Atom of int
                     | Nest of nested_list list;
```

Write the code that creates a value of the type `nested_list` above. [1 mark]

(b) Write the function `flatten` that flattens a nested list to return a list of integers. [3 marks]

(c) Write the function `nested_map f n` that applies a function `f` to every `Atom` in `n`. [4 marks]

(d) What is the type of `f` in Part (c)? [1 mark]

(e) Write a function `pack_as xs n` that takes a list of integers and a `nested_list`; the function should return a new `nested_list` with the same structure as `n`, with integers that correspond to the integers in list `xs`. *Note: It is acceptable for the function to fail when the number of elements differ.* Example:

```
> pack_as [1, 2, 3] (Nest [Atom 9, Nest [Atom 8, Atom 7]]);
val it = Nest [Atom 1, Nest [Atom 2, Atom 3]]: nested_list
```

[6 marks]

(f) What does the data type `nested_zlist` correspond to? [2 marks]

```
datatype nested_zlist  = ZAtom of int
                       | ZNest of (unit -> nested_zlist list);
```

(g) Write the function that converts a `nested_zlist` to a `nested_list`. [3 marks]

SECTION B

3 Object-Oriented Programming

(*a*) You are given the following implementation for an element of a list:

```
class Element {
  int item;
  Element next;

  Element(int item, Element next) {
    super();
    this.item = item;
    this.next = next;
  }

  @Override
  public String toString() {
    return item + " " + (next == null ? "" : next);
  }
}
```

(*i*) What does the statement `super()` mean? [1 mark]

(*ii*) What is the meaning of `this` in the line `this.item = item`? [1 mark]

(*iii*) What is the purpose of the annotation `@Override`? [2 marks]

(*iv*) Rewrite the class to be immutable. You may assume that there are no sub-classes of Element. [2 marks]

(*b*) Use the immutable `Element` class to provide an implementation of an immutable class `FuncList` which behaves like an `int list` in ML. Your class should include a constructor for an empty list and methods `head`, `tail` and `cons` based on the following functions in ML. Ensure that your class behaves appropriately when the list is empty. [6 marks]

```
fun head x::_ = x;
fun tail _::xs = xs;
fun cons (x,xs) = x::xs;
```

(*c*) Another developer changes your implementation to a generic class `FuncList<T>` that can hold values of any type T.

(*i*) This means that `FuncList<T>` is no longer immutable. Explain why and what could be done to remedy this. [2 marks]

(*ii*) Java prohibits covariance of generic types. Is this restriction necessary in this case? Explain why with an example. [6 marks]

4 Object-Oriented Programming

(*a*) What is an object? [2 marks]

(*b*) Give four examples of how object-oriented programming helps with the development of large software projects and explain why each one is helpful. [8 marks]

(*c*) Explain the meaning of the Open-Closed principle. [2 marks]

(*d*) Draw a UML diagram for a design satisfying the Open-Closed principle and explain why it satisfies it. [8 marks]

SECTION C

5 Digital Electronics

(*a*) (*i*) Show that

$$A + B.C = (A + B).(A + C)$$ [2 marks]

(*ii*) Using the distributive law in Part (*a*)(*i*), express the following equation in product of sums form with 4 product terms, each with a sum of 3 variables:

$$F = H + I.\overline{J} + \overline{K}.L$$ [4 marks]

(*b*) (*i*) Write down the truth table for the logic unit (LU) defined in the following table that can execute one of 4 logical operations at a time on 2 data inputs (Y_1, Y_0), to yield output (Z). The LU is under the control of inputs (I_1, I_0) and the logical operations are encoded to 2-bit instruction codes as follows:

Operation	OR	AND	XOR	NOP
Instruction code $(I_1 I_0)$	00	01	10	11

Note that operations OR, AND and XOR have their usual meanings and that the execution of NOP implies Z can take any binary value.

[3 marks]

(*ii*) Use a Karnaugh Map to determine a simplified expression for Z in Part (*b*)(*i*). [2 marks]

(*c*) (*i*) Show using a circuit diagram how W can be implemented in 2-level sum of products form using AND gates followed by OR gates. Remember to include any NOT gates required since only uncomplemented input variables are available:

$$W = \overline{B}.\overline{C} + \overline{A}.B.C + A.C.\overline{D}$$

[2 marks]

(*ii*) Consider the implementation in Part (*c*)(*i*). Assume that the gates have finite propagation delay. Describe what happens at W when inputs $\{A, B, C, D\}$ change from $\{1, 0, 1, 0\}$ to $\{1, 0, 0, 0\}$.

[3 marks]

(*iii*) Determine the other single input variable change that will give a similar problem to that observed in Part (*c*)(*ii*).

[2 marks]

(*iv*) Determine a modified sum of products expression for W that will eliminate the problems observed in Part $(c)(ii)$ and Part $(c)(iii)$.

[2 marks]

6 Digital Electronics

(*a*) (*i*) Briefly describe the ways in which sequential logic differs from combinational logic.

[2 marks]

(*ii*) Describe the main features that differentiate synchronous from asynchronous sequential logic.

[4 marks]

(*b*) A synchronous 3-bit counter implemented using D-type Flip-Flops has a mode control input M. When $M = 0$, the counter output sequence represented in decimal form is 0, 1, 2, 3, 4, 5, 6, 7, and repeat. When $M = 1$, the counter output sequence represented in decimal form is 0, 1, 3, 2, 6, 7, 5, 4, and repeat. The Flip-Flop outputs are $\{Z_2 Z_1 Z_0\}$ where Z_0 represents the least significant bit of the counter output.

(*i*) Draw a state diagram that describes this counter.

[4 marks]

(*ii*) Write down the state transition table corresponding with the state diagram in Part (*b*)(*i*).

[2 marks]

(*iii*) Determine the excitation combinational logic in sum of products form for D-type Flip-Flop input D_0, i.e., the input of the Flip-Flop that represents the least significant bit of the counter. Show that the required combinational logic can be implemented using a 2-input XNOR gate plus some other combinational logic gates.

[3 marks]

(*c*) Use row matching to reduce the number of states required to represent the single input (X), single output (Z), Mealy finite state machine described in the following state transition table:

Current State	Next State		Output (Z)	
	$X = 0$	$X = 1$	$X = 0$	$X = 1$
A	B	C	1	0
B	F	D	0	0
C	D	E	1	1
D	F	E	0	1
E	A	D	0	0
F	B	C	1	0

Draw the resulting state diagram. [5 marks]

SECTION D

7 Discrete Mathematics

(*a*) Let n be a positive natural number. Show $x \equiv y \,(\mathrm{mod}\, n)$ determines an equivalence relation between integers x and y. [3 marks]

(*b*) Describe the extended Euclid algorithm which given a pair of positive natural numbers (m, n) returns not only their gcd, $\gcd(m, n)$, but also its expression as a linear combination, $j.m + k.n$, for integers j and k. [7 marks]

(*c*) Assume positive natural numbers m and n are coprime, so $\gcd(m, n) = 1$ with associated linear combination $j.m + k.n = 1$, for integers j and k.

(*i*) Show that for any natural numbers r and s there is a solution to

$$x \equiv r \,(\mathrm{mod}\, m) \;\wedge\; x \equiv s \,(\mathrm{mod}\, n)\,.$$

[*Hint:* Take $x = s.j.m + r.k.n$.] [4 marks]

(*ii*) Show the solution is unique mod $m.n$, *i.e.* $x \equiv y \,(\mathrm{mod}\, m.n)$ for any two solutions x and y. [6 marks]

8 Discrete Mathematics

A binary relation $\prec$ on a set A is *well-founded* iff there are no infinite descending chains $\cdots \prec a_i \prec \cdots \prec a_1 \prec a_0$.

(a) Show a binary relation $\prec$ on a set A is well-founded iff any nonempty subset Q of A has a minimal element, *i.e.* an element m such that

$$m \in Q \ \wedge \ \forall b \prec m.\ b \notin Q \ .$$

[5 marks]

(b) Show that defining

$$(n_1, n_2) \prec (n_1', n_2') \ \Leftrightarrow \ (n_1, n_2) \neq (n_1', n_2') \text{ and } n_1 \leq n_1' \text{ and } n_2 \leq n_2'$$

determines a well-founded relation between pairs of positive natural numbers.

[7 marks]

(c) Let $\longrightarrow$ be a binary relation between pairs of positive natural numbers for which

$$(m, n) \longrightarrow (m, n - m) \text{ if } m < n\,, \quad \text{and} \quad (m, n) \longrightarrow (m - n, n) \text{ if } n < m\ .$$

Using (a) and (b), or otherwise, show that for all pairs of positive natural numbers (m, n), there is a natural number h such that

$$(m, n) \longrightarrow^* (h, h)\ .$$

[8 marks]

SECTION E

9 Databases

Suppose that we have a relational database with the following tables.

Table	Primary Key
`Movies(mid, title, year)`	`mid`
`People(pid, name)`	`pid`
`Genres(gid, genre)`	`gid`
`ActsIn(pid, mid)`	`pid, mid`
`HasRole(pid, mid, role)`	`pid, mid, role`
`HasGenre(gid, mid)`	`gid, mid`

In tables `ActsIn` and `HasRole`, `pid` is a foreign key into `People` and `mid` is a foreign key into `Movies`. In table `HasGenre`, `mid` is a foreign key into `Movies` and `gid` is a foreign key into `Genres`.

Note that this database is similar to, but not the same as, the examples used in lectures and the database used for practicals.

(a) For the table `ActsIn`, carefully explain what is meant by saying that `pid` is a foreign key into `People`. [2 marks]

(b) Discuss potential problems this database might suffer due to data redundancy. [2 marks]

(c) Write an SQL query that produces triples of the form `genre1`, `genre2`, `total` that count the number of movies associated with a pair of distinct genres. Each pair of genres should only appear once in the result. That is, if the triple `genre1`, `genre2`, `total` appears in the result, then the triple `genre2`, `genre1`, `total` should not. [5 marks]

(d) Suppose that `kid` is the `pid` associated with Kevin Bacon. Write SQL that returns every `pid` for actors with a Bacon number of 2. This SQL should not include views. [5 marks]

(e) Simplify the SQL of Part (d) using views. [6 marks]

10 Databases

This question develops an Entity-Relationship (ER) model for a new database. The database will be called Meta-ER because it contains Entity-Relationship models! The entities of our ER model are

entity name	description
Model	each Model represents an ER model
Entity	each Entity represents an ER entity
Relationship	each Relationship represents an ER relationship
Attribute	each Attribute represents an attribute

Each of our entities will have an **id** attribute (the primary key) and a **name** attribute. In addition, the Attribute entity will have a **type** attribute indicating the data type of the Attribute:

entity name	attributes
Model	**id**, **name**
Entity	**id**, **name**
Relationship	**id**, **name**
Attribute	**id**, **name**, **type**

(*a*) We start with one many-to-many relationship ModelHasEntity between Model and Entity that indicates which entities belong to the Model. For example, we may have a model called "MoviesModel" related to the entities presented in lecture, or a model "Trucks-R-Us" for a transportation company. ModelHasEntity is many-to-many to allow different models to share entities. Your task now is to complete this ER model and consider implementing it in a relational database.

(*b*) Define a relationship between Entity and Attribute called EntityHasAttribute. What cardinality should this relationship have? Justify your answer. [2 marks]

(*c*) Define a relationship between Relationship and Attribute called RelationshipHasAttribute. What cardinality should this relationship have? Justify your answer. [2 marks]

(*d*) Define a relationship called RelationshipRelatesEntity between Relationship and Entity. What cardinality should this relationship have? Justify your answer. [2 marks]

(*e*) Should the relationship RelationshipRelatesEntity itself have attributes? Justify your answer. Let us assume that all of our relationships are binary. [2 marks]

(*f*) Describe a relational implementation of your ER model, including keys and foreign keys. [4 marks]

(*g*) Given your relational implementation, write an SQL query that takes a model name `mname` and returns all triples `ename1`, `rname`, `ename2` where `ename1` and `ename2` are names of entities in the model `mname`, and `ename1` is related to `ename2` via the relationship with name `rname`. [8 marks]

END OF PAPER