UNIVERSITY OF DUBLIN

CS4B1A1

TRINITY COLLEGE

FACULTY OF ENGINEERING & SYSTEMS SCIENCES

DEPARTMENT OF COMPUTER SCIENCE

B.A. (Mod.) Computer Science **Trinity Term 2004**
Degree Examination

4BA1 INFORMATION SYSTEMS

Friday, 4th June 2004 Goldsmith Hall 9.30 – 12.30

Mr. Vincent P. Wade and Dr. Mícheál Mac an Airchinnigh

Attempt **five** questions, at least **two** from each section.
Students may avail of the HANDBOOK OF MATHEMATICS of Computer Science.
The partition of marks is noted in the margin and correlated with effort.
Please use separate answer books for each section.

SECTION A

1. Consider an airline relational database which describes airline flight information. Each flight is identified by a flight number, and is flown by a particular airline. Each flight consists of one or more 'flight legs' (i.e. hops between different airports to a destination airport). Therefore, for each flight, the database records one or more flight leg numbers (e.g. $1, 2, 3\ldots$). Each flight leg has scheduled arrival and departure times, as well as departure and arrival airports. However, each of these flight legs can be flown on different dates. Therefore the database also records a 'flight leg instance' (where each instance of a flight leg has a date on which the flight travels, as well as specific information for that flight leg instance e.g. number of seats available, the airplane used on the leg instance and the actual arrival and departure dates and airports. An airplane is identified by an airplane id and is of a particular airplane type. The database also records what types of airplanes can land at what airports. An airport is identified by an airport code. Fares are recorded for each flight leg instance. A customer reservation is made by recording the customer name and phone number when the customer is booking a particular leg instance of a flight.

(a) Draw the functional dependency diagram for this database indicating any assumptions you have made and any extra information you deem appropriate for the database. [7/20]

(b) Derive the relational structure for the database. [4/20]

(c) Consider an update for an airline database so as to enter a reservation on a particular flight leg instance on a given date:

(i) Give the SQL command for the creation of two of the tables in your relational structure — one which would hold reservations of customers on a flight leg instance, and the other showing permitted airplane types in particular airports.

(ii) Give the SQL for the customer reservation i.e. it should record the booking of a customer on a leg instance of a flight.

(iii) What types of constraints would you expect to check e.g. entity constraint etc.? [9/20]

2. Consider the following relational model which describes employees working on projects, within departments in a company. The model also describes the locations of the departments, and some details as to the dependents of the employees. The primary keys of the relations are underlined.

Employee (Fname, MiddleInitial, LastName, <u>EmployeeSSN</u>, DateOfBirth, Address, Sex, Salary, MgrSSN, DeptNo)

Department (DeptName, <u>DeptNo</u>, MgrSSN, MgrStartDate)

DepartmentLocation (<u>DeptNo, Dlocation</u>)

Project (PName, <u>PNumber</u>, PLocation, DeptNo)

WorksOn (<u>EmployeeSSN, PNumber</u>, Hours)

Dependent (<u>EmployeeSSN, DependentName</u>, Sex, DateOfBirth, Relationship)

(a) Give the SQL statement to create the employee table. Include in your answer any foreign key(s) and at least two table constraints which could be appropriate. [4/20]

(b) Give the SQL statement to create a VIEW of employee names, their dependents, the names of the projects on which they work and the hours they worked for that project. [4/20]

(c) Delete employee whose EmployeeSSN is 7666666, based on your definition of that table in (a). [4/20]

(d) Create an assertion that no employee can work on more than five projects. [4/20]

(e) Suppose there is another table called ProjectsTooLong which contains projects whose total number of hours exceed 1000. The table has the form:

ProjectsTooLong (PNumber)

Define an appropriate trigger to automatically insert a project number into this table if the total number of hours for that project exceeds 1000. [4/20]

3.

(a) What is meant by concurrent execution of database transactions in a multi-user system? Describe three types of interference which can occur across concurrent transactions. [7/20]

(b) Compare and contrast predominantly lock based concurrency control algorithms (round-wait & wait-die) with predominantly timestamp oriented methods. In your answer explain each of these algorithms and contrast the different ways they achieve concurrency. [11/20]

(c) Under what conditions would you suggest lock based versus time stamp based concurrency? [2/20]

4.

(a) Consider the three transactions T1, T2, T3 and the schedules S1 and S2 given below. State whether each schedule is serialisable or not. If a schedule is serializable write down the equivalent serial schedule. If it is not serializable, state what the precedence conflict is which has caused it to be non-serializable.

T1: r1(x), r1(z), w1(x);
T2: r2(z), r2(y), w2(z), w2(y);
T3: r3(x), r3(y), w3(y);

S1: r1(x), r2(z), r1(z), r3(x), r3(y), w1(x), w3(y), r2(y), w2(z), w2(y);
S2: r1(x), r2(z), r3(x), r1(z), r2(y), r3(y), w1(x), w2(z), w3(y), w2(y); [8/20]

(b) Compare and contrast OODBMS and Relational DBMS under the following headings: Information Model, Representational power (ability to represent simple and complex relations), ease of use and programming, query and query optimisation. [12/20]

SECTION B

5. A hotel may be considered to be an arrangement of rooms. Each room is given a unique label which for convenience might be thought of as a room number. Rooms are grouped together into disjoint physical regions. For example in the classical small-footprint high-rise architecture a region is typically called a floor. Other architectural models are possible. We model this basic hotel structure as follows.

$$\alpha \in ROOMS = LABEL \longrightarrow ROOM$$
$$\beta \in ZONES = REGION \longrightarrow (LABEL \longrightarrow ROOM)$$

Associated with each room there is a corresponding information base, which contains its label, name or description, purpose, and capacity. Certain rooms are for sleeping, others are for dining, etc. We model this abstractly by

$$\gamma \in INFOBASE = LABEL \longrightarrow INFO$$

In conclusion, the model of the hotel ($SYSTEM$) may expressed as the product:

$$SYSTEM = ROOMS \times ZONES \times INFOBASE$$

(a) Write precise mathematical expressions for each of the following:

(i) "the labels of the $ROOMS$ are the same as those used in the $ZONES$ and in the $INFOBASE$";

(ii) "a new room x labelled l is sited in the region r and the associated information y recorded";

(iii) "the state of the hotel after a major fire destroyed all rooms $S = \{x_1, x_2, \ldots, x_n\}$". [8/20]

(b) Classes of hotel may also be distinguished by their rooming policy with respect to the model $REGISTER = PERSON \longrightarrow ROOM$, $OCCUPANCY = ROOM \longrightarrow \mathbb{N}$ and $CAPACITY = ROOM \longrightarrow \mathbb{N}$.

(i) Write a formal specification (including signature and pre-condition) for the operations of CheckIn, CheckOut and ChangeRoom ignoring capacity and occupancy.

(ii) Prove that ChangeRoom can be defined in terms of CheckIn and CheckOut. [6/20]

(c) In general, a model of the form $\mu \in M = X \longrightarrow Y$ is said to be *fruitfully ambiguous* in the sense that with respect to computing it is constructive, algebraic and faithful and with respect to semantics, within topos theory for example, it can be multiply explained. Consider the operation $\mu \sqcup [x \mapsto y]$ which models the entering of new information to an existing model. Both μ and $\mu \sqcup [x \mapsto y]$ are usually considered to be partial maps. Illuminate their meaning within (i) the topos of sets $\mathcal{S}$ and (ii) the topos of maps $\mathcal{S}^{\downarrow}$. [6/20]

6. A spelling checker dictionary ($DICT_0$) may be modelled as a set of words. Implementation of *set* as *list* suggests the reification ($DICT_1$). A further elaboration of the spelling checker dictionary ($DICT_2$) may be modelled by hashing words into pages to obtain a specialized form of distributed dictionary.

$$DICT_0 = \mathcal{P}\,WORD \qquad DICT_2 = PAGE^p$$
$$DICT_1 = WORD^\star \qquad PAGE = WORD^\star$$
$$\sigma\colon WORD^\star \longrightarrow WORD_{\leq}{}^\star, \quad \text{the usual sorting map.}$$
$$h\colon WORD \longrightarrow \mathbb{Z}_p, \quad \text{a hashing map where } p \text{ is prime.}$$

The two models, $DICT_1$ and $DICT_2$, are subject to the following invariants:

$$\mathsf{inv}\text{-}\,DICT_1\colon DICT_1 \longrightarrow \mathbb{B}$$
$$\mathsf{inv}\text{-}\,DICT_1(\delta) := (\sigma\delta = \delta) \wedge (|\delta| = |\,\mathsf{elems}\,\delta|)$$
$$\mathsf{inv}\text{-}\,DICT_2\colon DICT_2 \longrightarrow \mathbb{B}$$
$$\mathsf{inv}\text{-}\,DICT_2(\delta) := \text{“properties inherited from } \mathsf{inv}\text{-}\,DICT_1\text{”}$$

(a) (i) Write a *complete* specification for the operation LookUp of a word w with respect to each of the three models.

(ii) State clearly in plain English the meaning and intended purpose of the invariant inv- $DICT_1$. [8/20]

(b) Determine the retrieve map $\mathcal{R}\colon DICT_2 \longrightarrow DICT_1$ and hence show *in broad outline* that the following diagram commutes.

$$\begin{tikzcd}
DICT_1 \arrow[r, "Ent_1{[w,\sigma]}"] & DICT_1 \\
DICT_2 \arrow[u, "\mathcal{R}"] \arrow[r, "Ent_2{[w,h,\sigma]}"'] & DICT_2 \arrow[u, "\mathcal{R}"']
\end{tikzcd}$$

where $Ent_1[w,\sigma]$ and $Ent_2[w,h,\sigma]$ denote the entering of a new word w into a dictionary with respect to the appropriate model. [6/20]

(c) (i) Draw a sketch of the fibering of the hashing map $h\colon WORD \longrightarrow \mathbb{Z}_p$ and comment on the *practical computing* consequences of a unique section

$$s\colon \mathbb{Z}_p \longrightarrow WORD$$

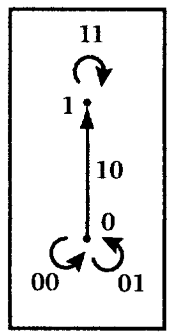
(ii) From a geometric perspective how might one use a torus to examine and/or develop a hashing map $h\colon WORD \longrightarrow \mathbb{Z}_p$? [6/20]

7. A topos may be considered to manifest computational structure together with the corresponding intrinsic logic. Modelling the same computational construct in different though related topoi provides a different perspective on the same reality. For example, the notion of *state change* may be modelled in the topos of sets, $\mathcal{S}$, by a pair of parallel maps $\langle f, g \rangle$ where $X \overset{f}{\underset{g}{\rightrightarrows}} Y$ and which we read as "first f and then g". The state change may be made more explicit by the mapping $\varphi \xrightarrow{h} \gamma$ between map objects $\varphi = \boxed{X \xrightarrow{f} Y}$ and $\gamma = \boxed{X \xrightarrow{g} Y}$ in the topos of maps $\mathcal{S}^{\downarrow}$. Finally, the state change and the directionality of the change may both be hardwired into a graph object encoding in the topos of graphs $\mathcal{S}^{\downdownarrows}$.

(a) (i) Consider the swopping of the values of two variables x and y given by the sequence of simple assignment statements $t := x$; $x := y$; $y := t$. Assuming an initial assignment of $[x \mapsto a, y \mapsto b, t \mapsto c]$, illustrate the incremental state change by constructing the three corresponding parallel pairs of maps and then the associated three graph objects in the topos of graphs $\mathcal{S}^{\downdownarrows}$.

(ii) The graph object X in the topos of graphs $\mathcal{S}^{\downdownarrows}$ shown opposite encodes a well-known binary logical operator $\Omega \times \Omega \longrightarrow \Omega$ in the topos of sets $\mathcal{S}$. Construct the corresponding parallel pair of maps and identify their functionality. [8/20]

X



(b) Construct the equalizer $E \xrightarrow{e} \Omega \times \Omega$ for the pair of parallel maps $\Omega \times \Omega \overset{f}{\underset{g}{\rightrightarrows}} \Omega$ of (a)(ii) above and hence show that $x \wedge y = x$ is equivalent to $x \leq y$ with respect to E. [6/20]

(c) For some set X with suitable parts $S \overset{i}{\hookrightarrow} X$ and $T \overset{j}{\hookrightarrow} X$ show how to construct the part $S \Rightarrow T$ which is classified by $X \xrightarrow{\to \circ \langle \varphi_S, \varphi_T \rangle} \Omega$ [6/20]

8. "This is the key move in the [Hegelian] dialectic: the negation of negation, the annihilation of nothingness ... in any creative act." (George Steiner, *Grammars of Creation*. Faber and Faber Limited, London, p. 98, 2001). Let us seek to illuminate an aspect of this dialectic by considering *the construction* of the truth value object Ω in a given topos.

On the construction of Ω, that is to say on the elaboration of the internal structure of Ω, we may proceed at first by the following method based on the general principle that *logic emerges from structure.*

- let us pick a *point* of Ω, usually denoted by $\mathbf{1} \xrightarrow{\mathbf{t}} \Omega$, which we consider to denote "true".
- since $S = \mathbf{1}$ is a *part* of $X = \Omega$ then it has the obvious classifier φ_S. Let $\mathsf{not}\, S$ denote the largest part of X which is *separate from* S. Corresponding to $\mathsf{not}\, S$ there must be another point of Ω, $\mathbf{1} \xrightarrow{\mathbf{f}} \Omega$, which is *separate from* $\mathbf{t}$ and which we usually call "false".
- formally there is a negation map $\Omega \xrightarrow{\neg} \Omega$ with $\neg\mathbf{t} = \mathbf{f}$.
- the rest of the internal structure of Ω may be exposed by a judicious choice of part maps $S \overset{i}{\hookrightarrow} X$.

(a) Considering parts of the graph object G shown in Fig. 1 show a step by step construction of the truth value object $\Omega^{\downdownarrows}$ in the topos of graphs $\mathcal{S}^{\downdownarrows}$.

G

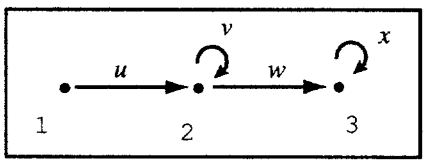


Figure 1: A simple model of three linked www pages.

[8/20]

(b) (i) For an appropriate choice of part $S \hookrightarrow G$ show that $\mathsf{not}\,\mathsf{not}\, S \neq S$.

(ii) In contrast to $\mathsf{not}\, S$ there is also the notion of $\mathsf{non}\, S$ which is the smallest part of G that together with S makes up the whole of G. For your choice of S above sketch $\mathsf{non}\, S$ and hence illustrate the boundary of S, denoted ∂S, which is that part common to both S and $\mathsf{non}\, S$ (with respect to G). [6/20]

(c) The topos of dynamical systems $\mathcal{S}^{\circlearrowleft}$ has as objects a set X together with an endomap $X \xrightarrow{\alpha} X$ usually denoted by $X^{\circlearrowleft\alpha}$. Construct the truth value object $\Omega^{\circlearrowleft\omega}$ of $\mathcal{S}^{\circlearrowleft}$. Since an object $X^{\circlearrowleft\alpha}$ may be regarded as a graph object, why is it that the truth value object $\Omega^{\circlearrowleft\omega}$ is so different from $\Omega^{\downdownarrows}$? [6/20]