X CS4B1A1

UNIVERSITY OF DUBLIN
TRINITY COLLEGE

Faculty of Engineering & Systems Sciences

Department of Computer Science

B.A. (Mod.) Computer Science
Degree Examination

Trinity Term 2005

4BA1 INFORMATION SYSTEMS

Monday, 13th June 2005 | Exam Hall | 9.30–12.30

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

Instructions to Candidates:

Attempt five questions, at least two from each section.

All questions carry equal marks.

The partition of marks is noted in the margin and correlated with effort.

Please use separate answer books for each section.

Materials permitted for this examination:

1. Handbook of Mathematics for Computer Science (2005 Edition)

2. Sheet on Description Logic for Computer Science (2005 Edition)

SECTION A

Q1.

(a) Explain how "pure time stamping" can be used to provide concurrency control (i.e. a concurrency control mechanism with no locking mechanisms). In your answer give algorithms to support both retrieval from a database, and writing to a database. [10/20]

(b) Suppose two transactions T1 and T2 were allowed to execute without safeguarding against interference, where their respective operations would be

$$\{r1(X), r1(Y), w1(X), r2(Y), w2(Y), w1(Y), r1(Z)\}$$

Illustrate how, using the time stamping method you've described above, the operations listed would be executed so as to ensure non-interference of transactions. Give the actual schedule executed when executing the operators using the time stamping concurrency control algorithms. In your example execution of these operations indicate any usage of caches, timestamps, waiting, starting/restarting, etc. [6/20]

(c) "Time stamping techniques are potentially effective for concurrency control where the database and their transactions are distributed." Indicate if you agree or disagree with this statement giving two reasons for your answer. [4/20]

Q2. A metro tourist information system contains information about the tourist attractions and the metro rail lines linking them in the city. For each tourist attraction, the database maintains the attraction's name, the name of the metro station nearest to the attraction, and the distance from the city centre. Also associated with each attraction is the name of the street on which the attraction resides. The database also contains each rail line name (e.g. RedLine, BlueLine), and the station names served by each rail line.

(a) Draw a functional dependency diagram of the relationships between the attributes needed for the database. State any assumptions used in your dependency diagram and explain any extra information you deem necessary for your model. [6/20]

(b) Derive a relational model in BoyceCodd normal form based on the functional dependency diagram, indicating primary and foreign keys. [5/20]

(c) Outline the SQL queries you would use to solve the following:

(i) Find out the name of the metro station nearest to the 'Wax Museum.' [3/20]

(ii) What are the names of the rail lines over which a tourist must travel when starting from the 'Wax Museum' and arriving at the 'Concert Hall.'
You may assume that no more than two rail lines are needed to link any two attractions. [3/20]

(iii) Convert from miles to kilometers, the distance from the centre of the city to each attraction on the 'BlueLine.' You may assume a conversion factor of 1:1.6 (miles to kilometers). [3/20]

Q3.

(a) What are semantic constraints in a Relational Database? Explain how semantic constraints can be defined in SQL, giving examples of such SQL constructs and indicating under which conditions they are tested or invoked. [12/20]

(b) Suppose the following relational model represents the values of cars and the insurance costs for those cars:

Cars (CarID, CarValue)
Insurance (CarId, InsuranceCost)

Give the (Trigger) SQL command to lower the insurance wherever a car depreciates in value. You may calculate the amount of change in insurance cost as follows:

Depreciation = (old value of the car – new value of the car) *.3

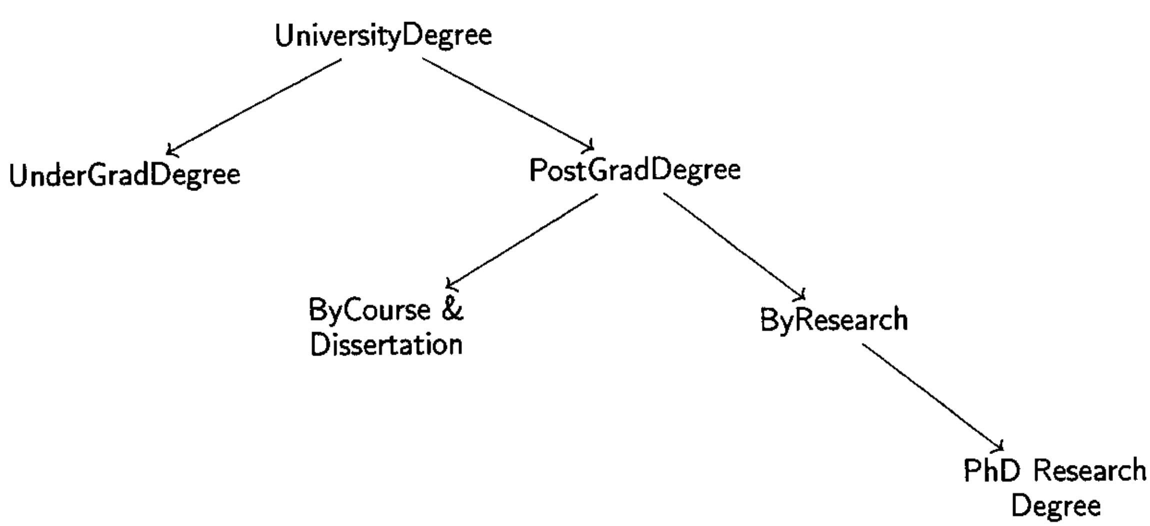
[4/20]

(c) Explain how Triggers can be used to provide extra support for managing security (access permissions for users on tables). In your answer give an example of such a trigger. [4/20]

Q4.

(a) Compare and contrast OODBMS versus Relational DBMSs. In your answer refer to representation of data types supported, schema extensibility, query languages, integration with programming languages, standardization and performance. [14/20]

(b) Suppose a university used an OODBMS to represent their courses with the following classes:



Using suitable attributes of your own choice, derive a relational schema for the above OO schema. Comment on the problems you envisage in (i) retrieving information about PhDResearch Degrees, and (ii) deletion of a university degree from the relational database you have designed. [6/20]

SECTION B

Q5. Let us imagine a city to be an arrangement of regions for leisure, residential, industrial, business, ... purposes. Each region is considered to be a collection of uniquely addressed places each of which might be occupied by street, house, hotel, park, factory, ... Associated with each addressed place there is a corresponding information record, which contains its (i) name or description, (ii) purpose, (iii) footprint in square metres, (iv) location in latitude, longitude, height, and (v) geometrical shape. Let us model the city abstractly by

$$\begin{aligned} \langle \alpha, \beta, \gamma \rangle \in \mathit{CITY} &= \mathit{PLACES} \times \mathit{ZONES} \times \mathit{INFOBASE} \\ \alpha \in \mathit{PLACES} &= \mathit{ADDR} \longrightarrow \mathit{PLACE} \\ \beta \in \mathit{ZONES} &= \mathit{REGION} \longrightarrow \mathit{PLACES} \\ \gamma \in \mathit{INFOBASE} &= \mathit{ADDR} \longrightarrow (\mathit{NAME} \times \mathit{PURPOSE} \times \mathit{FPRINT} \times \mathit{LOC} \times \mathit{SHAPE} \times \ldots) \end{aligned}$$

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

(i) "Each place in the city has a unique address."

(ii) "The places in the city are exactly all those in the zones."

(iii) "All the places in region r are for residential purposes only." [8/20]

(b) To increase industrial capacity, a new region r is added to the city. In this new region places such as a ring road, a number of access roads, open spaces, and industrial units of varying sizes are addressed. Sketch **very briefly** the operations needed to update the city $\langle \alpha, \beta, \gamma \rangle$. [6/20]

(c) The Web Ontology Language restricted to Description Logic (OWL-DL) is ideally suited for the formal conceptual modelling of the real world knowledge which corresponds to the formal mathematical model used in programming. Sketch **very briefly** an ontology for places in the city. [6/20]

Q6. A *Pizza* is considered to be composed of two parts: a *PizzaBase* and a set of *PizzaToppings*. This is conveniently modelled as a product:

$$\langle \beta, \sigma \rangle \in \mathit{Pizza} = \mathit{PizzaBase} \times \mathcal{P}\mathit{PizzaTopping}$$

(a) (i) Draw the corresponding category-theoretic product diagram.

(ii) What exactly is the relationship between $\langle \beta, \sigma \rangle \in \mathit{Pizza}$ and $1 \xrightarrow{\langle \beta, \sigma \rangle} \mathit{Pizza}$?

(iii) Write a complete formal specification for the addition of a new *PizzaTopping* t to an existing *Pizza* description. Use a pre-condition that has the "form" of Description Logic subsumption to capture ontological "newness".

(iv) Write a formal specification for testing whether or not a set of *PizzaToppings* T is part of a given *Pizza* description and show clearly how it may be used to give "three-valued logic" within the topos of sets $\mathcal{S}$. [8/20]

(b) The set of *PizzaToppings* may be classified into four disjoint *Types*: Cheese, Meat, Vegetable and Fish. Show how fibering *PizzaToppings* over these *Types* and then sectioning does **not** define the notion of a *NamedPizza* such as *MargheritaPizza* which may be deemed to have exactly only *TomatoTopping* and *MozzarellaCheeseTopping* [6/20]

(c) Consider the existential restriction $\exists$ *hasTopping.MozzarellaCheeseTopping* which describes an anonymous class of individuals that have at least one topping that is Mozzarella cheese. A binary relation such as *hasTopping* may be modelled by a mapping of the form $A \xrightarrow{\beta} \mathcal{P}M$. Show with the aid of a suitable example such as $\{a \mapsto \{x\}, b \mapsto \{y, z\}\}$ how such binary relations may also be modelled by a span, i.e., a pair of mappings of the form $A \xleftarrow{\sigma} X \xrightarrow{\tau} M$. [6/20]

Q7. The partial mapping is ubiquitous in computing. A deterministic finite state machine (fsm) $(Q, \Sigma, \delta, q_0, F)$ consists of a set of states Q, an input alphabet Σ of letters, a start state q_0, a set of final states F and state transitions are given by the action:

$$\delta \colon \Sigma \times Q \longrightarrow Q$$

There is a natural extension of the state transition map to the free monoid over Σ:

$$\delta^{\star} \colon \Sigma^{\star} \times Q \longrightarrow Q$$

(a) Consider the fsm with $Q = \{0, 1, 2\}$, $\Sigma = \{l, d, u\}$, $q_0 = 0$, $F = \{1\}$ which captures something of the syntactic structure of a program variable. The action is

$$\delta = \begin{bmatrix} \langle l,0 \rangle & \mapsto & 1, & & & & & & \\ \langle l,1 \rangle & \mapsto & 1, & \langle d,1 \rangle & \mapsto & 1 & \langle u,1 \rangle & \mapsto & 2 \\ \langle l,2 \rangle & \mapsto & 1, & \langle d,2 \rangle & \mapsto & 1 & & & \end{bmatrix}$$

(i) Give the state transition table for the action δ and draw the corresponding state transition diagram.

(ii) Considering an input letter as the name of a map $Q \longrightarrow Q$, it is clear that the action consists of three maps two of which are partial. There are two classical techniques by which such partiality is removed. What are they? [8/20]

(b) Words in $\Sigma^{\star}$ may also be considered as maps $Q \longrightarrow Q$ which together give the action $\delta^{\star}$. There are only a finite number of such maps which may be determined by computing the syntactic transformation monoid. Demonstrate this by completing a suitable portion of this table

$\star$	l	d	$\dots$
l	$\dots$	$\dots$	$\dots$
d	$\dots$	$\dots$	$\dots$
$\vdots$	$\dots$	$\dots$	$\dots$

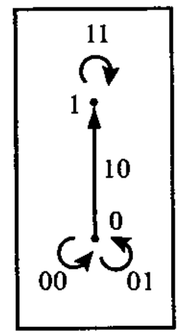
where $x \star y = xy$ denotes the map composition $y \circ x$. [6/20]

(c) Forgetting start and final states, the state transition diagram δ may be regarded as an object in the topos of graphs $\mathcal{S}^{\dot{\downdownarrows}}$. Let $\gamma = [\langle l,0 \rangle, \mapsto 1, \langle l,1 \rangle \mapsto 1, \langle d,1 \rangle \mapsto 1]$ denote a part of δ. Construct the corresponding subobject classifier $\delta \xrightarrow{\varphi_\gamma} \Omega$. [6/20]

Q8. To construct the truth-value object Ω of a topos, we may use a method based on the general principle that *logic emerges from structure* (and *vice versa*):

- pick a *point* $\mathbf{1} \xrightarrow{\mathbf{t}} \Omega$, which denotes "true".
- there must be another point of Ω, $\mathbf{1} \xrightarrow{\mathbf{f}} \Omega$, which is *separate from* $\mathbf{t}$ and which denotes "false".
- formally there is a negation map $\Omega \xrightarrow{\neg} \Omega$ with $\neg\mathbf{t} = \mathbf{f}$.
- the rest of Ω may be exposed by a judicious choice of part maps $S \overset{i}{\hookrightarrow} X$.

X



(a) The graph object X in the topos of graphs $\mathcal{S}^{\dot{\downdownarrows}\cdot}$ shown opposite encodes a well-known binary logical operator $\Omega \times \Omega \longrightarrow \Omega$ in the topos of sets $\mathcal{S}$.

With the aid of X and its parts, show a step by step construction of the truth value object $\Omega^{\dot{\downdownarrows}\cdot}$ in the topos of graphs $\mathcal{S}^{\dot{\downdownarrows}\cdot}$.

[8/20]

(b) The intrinsic logic of a functional programming language is said to be naturally three-valued. Demonstrate the validity of this claim by constructing the truth value object $\Omega^{\downarrow\omega}$ of the topos of mappings $\mathcal{S}^{\dot{\downarrow}\cdot}$. [6/20]

(c) Every object in the topos of graphs $\mathcal{S}^{\dot{\downdownarrows}\cdot}$ may be considered to be the encoding of a pair of parallel mappings in the topos of sets $\mathcal{S}$. Obtain the parallel pair corresponding to the truth value object $\Omega^{\dot{\downdownarrows}\cdot}$. Since X is a part of $\Omega^{\dot{\downdownarrows}\cdot}$ show, with the aid of a coproduct (sum) diagram or otherwise, the relationship between the truth value objects of $\mathcal{S}$, $\mathcal{S}^{\dot{\downarrow}\cdot}$ and $\mathcal{S}^{\dot{\downdownarrows}\cdot}$. [6/20]

Description Logic for Computer Science

Dr. Mícheál Mac an Airchinnigh

2005

OWL Element	Symbol	Example	Meaning of example
allValuesFrom	$\forall$	$\forall$ children Male	All children must be of type Male
		$\forall$ hasChild Male	individuals all of whose children are Male
someValuesFrom	$\exists$	$\exists$ children Lawyer	At least one child must be of type Lawyer
		$\exists$ hasChild Lawyer	individuals having child who is Lawyer
hasValue	$\ni$	rich $\ni$ true	The rich property must have the value true
cardinality	$=$	children $= 3$	There must be exactly 3 children
minCardinality	$\geq$	children ≥ 3	There must be at least 3 children
maxCardinality	$\leq$	children ≤ 3	There must be at most 3 children
complementOf	$\neg$	$\neg$ Parent	Anything that is not of type Parent
intersectionOf	$\sqcap$	$\mathsf{Human} \sqcap \mathsf{Male}$	All Humans that are Male
unionOf	$\sqcup$	$\mathsf{Doctor} \sqcup \mathsf{Lawyer}$	Anything that is either Doctor or Lawyer
enumeration	$\{\ldots\}$	{male female}	The individuals male or female

Table 1: Protégé OWL Syntax

Concept	Symbol	Example	Meaning of example
Specialization	$\sqsubseteq$	$\mathsf{Woman} \sqsubseteq \mathsf{Person}$	Every Woman is Person
Definition	$\equiv$	$\mathsf{Woman} \equiv \mathsf{Person} \sqcap \mathsf{Female}$	Woman is defined to be female Person
Normalization		$\mathsf{Woman} \equiv \overline{\mathsf{Woman}} \sqcap \mathsf{Person}$	Woman is Person with womanly qualities
		$\overline{\mathsf{Woman}}$	Qualities that distinguish Woman among $\mathsf{Persons}$

Table 2: Terminology