CS_376 PROGRAMMING WITH ABSTRACT DATA TYPES

(Attempt 2 questions out of 3)

Question 1.

(a) (i) What is an abstract data type in programming?

[2 marks]

(ii) Discuss briefly the benefits of abstract data types in programming.

[2 marks]

(iii) What is the mathematical definition of an abstract data type?

[2 marks]

(iv) How are the programming and the mathematical definition of an abstract data type related?

[2 marks]

(v) What are the advantages of the mathematical definition of abstract data types over the programming definition?

[2 marks]

(b) Discuss the advantages and disadvantages of defining abstract data types in a functional and in an object-oriented programming language. What are the differences with respect to the addition of generators and observers?

[6 marks]

(c) (i) What is a *congruence* on a Σ-algebra?

[2 marks]

(ii) Let $\Sigma = (S, \Omega)$ be a signature and let $\varphi\colon A \to B$ be a homomorphism between Σ-algebras A and B. Let $\sim$ be a congruence on B. For every sort $s \in S$ define a binary relation $\approx_s$ on A_s by

$$a \approx_s a' \quad :\Longleftrightarrow \quad \varphi_s(a) \sim_s \varphi_s(a')$$

Show that

$$\approx \; := \; (\approx_s)_{s \in S}$$

is a congruence on A.

[7 marks]

Question 2.

(a) Explain the differences between minimal, intuitionistic and classical logic. What can you say about the behavior of these logics with respect to program synthesis from proofs?

[5 marks]

(b) (i) Derive in minimal logic the formula $(((P \to Q) \to Q) \to R) \to P \to R$.

[4 marks]

(ii) Write down the proof term corresponding to the derivation you found in (i).

[3 marks]

(iii) Explain why the following 'derivation' is not correct:

$$\dfrac{\dfrac{\dfrac{\overline{x = x}\ \text{refl}}{\exists y\,(x = y)}\,\exists^+ \qquad \dfrac{\dfrac{\dfrac{x = y}{\forall y\,(x = y)}\,\forall^+}{x = y \to \forall y\,(x = y)}\,\to^+}{\forall y\,(x = y \to \forall y\,(x = y))}\,\forall^+}{\forall y\,(x = y)}\,\exists^-}{\exists x\,\forall y\,(x = y)}\,\exists^+$$

[3 marks]

(c) (i) State the axioms for the data type of boolean values.

[2 marks]

(ii) Let x be a variable of sort boole. Derive from the axioms given in (i) the formula

$$\exists x\,((x = \mathrm{T} \to P) \land (x = \mathrm{F} \to Q)) \to (P \lor Q)$$

in minimal logic.

[8 marks]

Question 3.

(a) (i) Define what it means for a Σ-algebra to be *initial* in a class $\mathcal{C}$ of Σ-algebras.

[2 marks]

(ii) What is a *loose specification*? What is a model of a loose specification?

[2 marks]

(iii) What is an *initial specification*? What is a model of an initial specification?

[2 marks]

(iv) How can a model for an initial specification be constructed?

[2 marks]

(v) The class of all models of a loose or initial specification forms an abstract data type. In which case (loose or initial) is this abstract data type monomorphic? (no proof required)

[2 marks]

(b) Produce an initial specification for the algebra of *queues of natural numbers* that contains (among others) a sort queue, a constant $\mathsf{emptyqueue}$ for the empty queue and the following operations:

- add: adds a number at the end of the queue;
- front: returns the front element of a nonempty queue (if the queue is empty the number 0 shall be returned);
- back: removes the front element from a nonempty queue (if the queue is empty the empty queue shall be returned);

[7 marks]

(c) Let nat be a sort, $0\colon \mathsf{nat}$ a constant, $+, *\colon \mathsf{nat} \times \mathsf{nat} \to \mathsf{nat}$ operations (both used in infix notation), and $x, y, z\colon \mathsf{nat}$ variables. Consider the term rewriting system given by the following rules:

$$x + 0 \mapsto x$$

$$x * 0 \mapsto 0$$

$$x * (y + z) \mapsto x * y + x * z$$

Prove that this term rewriting system is terminating and confluent.

[8 marks]