19114056

INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
Department of Computer Science and Engineering
Spring Semester 2021-22 End Term Examination 26.4.2022
CSN312 Principles of Programming Languages FM: 40 Duration: 3hour

Answer **all the** questions. Answers should appear in **the order** of the questions.
Answer to each question **must begin** in a **new page**.
Parts of the same question should appear in a contiguous manner.
No mark will be given **if** multiple **answers are written** for a question.
Answers **written in PENCIL shall not be evaluated. Use PEN for diagrams.**
Answers that **violate** the above instructions will be awarded **zero** mark.
Zero mark would be given for correct answers with no steps/unjustified steps/incorrect justifications.
For each question, the final answer should be explicitly stated as: Final Answer: ....
DO NOT WRITE ANYTHING ON THE QUESTION PAPER

1. Give a recursive definition of multiplication. Let the name of the function be **multiply**.
Give a recursive lambda term **R** corresponding to **multiply**.
Obtain a non-recursive lambda term **T** from **R** such that **T = multiply**.
Show the complete computation to multiply two numbers 2 and 3, that is, (T 2 3).
All the steps of the computation must be justified. [1+1+2+4]

Remark: Well-known functions (those discussed in the class) can be used. For example, if *succ* is used then write *succ* is the successor function. Similarly, if the symbol 1 is used then write 1 denotes the number 1. There is no need to give lambda terms for these functions, here *succ*, 1. If other functions (those not discussed in the class) are used then their definitions as lambda terms must be given.

2. A list can be represented as x : l where x is the Head and l is the Tail. For example, a list L = [2,3,1] can be represented as 2 : [3,1] where Head is 2 and Tail [3,1].

Design simple lambda terms from scratch (without using any well-known functions) for (i) : , (ii) Head, and (iii) Tail such that the following conditions hold.
C1. Head(x : l) ->*$_\beta$ x (->*$_\beta$ means beta reduction in zero or more steps)
C2. Tail(x : l) ->*$_\beta$ l
Show that the designed lambda terms satisfy both the conditions. [6+2]

Remark: Merely giving a lambda term will not get any credit. The reasoning behind the term must be clearly given.

3. (a) Design a parametric composition function to compose two functions *f* and *g*. State the type of the parametric function. Obtain an instance of the parametric function by applying it to actual types like int, bool, nat, etc. [4]

Remark: Merely giving a function will not get any credit. The steps leading to the answer must be clearly justified.

(b) Stack has three operations `push`, `pop`, and `top`—returns the topmost element. Assume that Stack is implemented using a list. Let intStack be a data type for Stack whose elements are integers. Let **genericStack** be a data type for Stack that can work for any type of element. State the type of **genericStack**. [4]

Remark: Use appropriate notations. No explanation is needed.

4. (a) Construct a proof tree for $\vdash \{n > 0\}\ \mathbf{P}\ \{j = 1 + 2 \times n\}$ where P is given below, by using a suitable loop invariant. Justify the choice of the invariant. [5]

```
P:
L1. k := 0;
L2. j := 1;
L3. while (k ≠ n)  do
L4.     k := k + 1;
L5.     j := j + 2; od
```

(b) Derive a rule for the **repeat-until** command in the Hoare Logic framework, where
(**repeat** C **until** S) ≡ (C ; **while** ¬S **do** C **od**) [3]

5. Define a lifted set for a given set *X*.
Define an information ordering relation on the Cartesian product of two lifted sets.
Let A′, B′ be the lifted sets corresponding to A = {true, false} and B = {up, down} respectively.
Give a suitable diagram to represent the information ordering relation on $A' \times B'$.
Give one monotonic mapping and one non-monotonic mapping from A′ to B′. Give diagrams for each. [8]
Use PEN for diagrams. No marks would be given for diagrams using pencil.

END