Computer Science & Engineering Department
I. I. T. Kharagpur

Principles of Programming Languages: CS40032

Elective

Assignment – 7: Denotational Semantics *Marks: 25*

Assign Date: *6^{th} April, 2020* Submit Date: *23:55, 12^{th} April, 2020*

1. Study the algebra of Finite Lists, given in Module 13 and simplify the following expressions. Show all the steps in your simplification. **[4]**

   (a) $(hd(one\ cons\ (two\ cons\ nil))cons\ nil)$

   (b) $(\lambda l.\ (null\ l) \rightarrow (zero\ cons\ nil)[](one\ cons\ nil))(tl(one\ cons\ nil))$

   (c) $((one\ cons\ (two\ cons\ nil))cons\ nil)$

   (d) $(\lambda l.hd\ l)(tl(zero\ cons\ (tl(two\ cons\ nil))))$

2. Consider the Programming Language of Binary Numerals with Addition in Module 13. Show that the axioms of the Axiomatic Semantics are logical consequences of the Denotational Semantics. Further give axiomatic semantics of a queue. Syntactic Domain and Functions are given below: **[3+3]**

   - Syntactic Domains:
     Int: Integers –integers
     nat: natural numbers –natural numbers
     boolean: true,false
   - Signatures/Functions:
     new: $\rightarrow$ QUEUE
     enq: QUEUE x E $\rightarrow$ QUEUE
     deq: QUEUE $\rightarrow$ QUEUE
     front: QUEUE $\rightarrow$ Int
     back: QUEUE $\rightarrow$ Int
     size: QUEUE $\rightarrow$ nat (naturals, int ¿=0)
     empty: QUEUE $\rightarrow$ boolean

3. Given below is an incomplete denotational semantic of a simple language of nonnegative integer numerals. This definition requires two auxiliary functions defined in the semantic world, where Number x Number denotes the Cartesian product.

   plus : Number x Number $\rightarrow$ Number
   times : Number x Number $\rightarrow$ Number

   Denotational Semantic:

   - Syntactic Domains:
     N: Numeral –nonnegative numerals
     D: Digit –decimal digits
   - Abstract Production Rules:
     Numeral ::= Digit | Numeral Digit
     Digit ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
   - Semantic Domain:
     Number = { 0, 1, 2, 3, 4, .. } –natural numbers
   - Semantic Functions:
     value : Numeral $\rightarrow$ Number
     digit : Digit $\rightarrow$ Number

For the given incomplete Denotational Semantic, define it's complete Semantic Equations and use it to evaluate the following: **[4+2]**

(a) $value[[98]]$

(b) $value[[0021]]$

4. Refer to the below language - PostFix, a simplestack-based language. **[9]**

- $Abstract\ Syntax$:
  $P \in Program$
  $Q \in Commands$
  $C \in Command$
  $A \in ArithmeticOperator \ = \{add,\ sub,\ mul,\}$
  $N \in Intlit \ = \ \{...,\ -2,\ -1,\ 0,\ 1,\ 2,\ ...\}$
  $P ::= (postfix\ N_{numargs}\ Q_{body})$
  $Q ::= C^*$
  $C ::= N|pop|swap|A|(Q)$

- $Semantic\ Domains$:
  $t \in StackTransform \ = \ Stack \ \rightarrow Stack$
  $s \in Stack \ = \ Value^* \ + \ Error$
  $v \in Value \ = \ Int \ + \ StackTransform$
  $r \in Result \ = \ Value + Error$
  $a \in Answer \ = \ Int + Error$
  $Error \ = \ error$
  $i \in Int \ = \ \{\ldots, -2, -1, 0, 1, 2, ...\}$
  $b \in Bool \ = \ \{true, false\}$

- $Semantic\ Operations$:

  $errorResult \ : \ Result$
  $errorAnswer \ : \ Answer$
  $errorStack \ : \ Stack$
  $errorTransform \ : \ StackTransform$
  $push \ : \ Result \rightarrow StackTransform$
  $pop \ : \ StackTransform$
  $top \ : \ Stack \rightarrow Result$
  $arithop \ : \ (Int \rightarrow Int \rightarrow Result) \rightarrow StackTransform$
  $transform \ : \ Result \rightarrow StackTransform$
  $resToAns \ : \ Result \rightarrow Answer$

- $Valuation\ Functions$:
  $P \ : \ Program \rightarrow Int^* \rightarrow Answer$
  $P[[(postfix\ N_{numargs}\ Q)]]$
  $= \lambda i^*.$ **if** $(length\ i^*) \ = N[[N_{numargs}]]$
  then $(resToAns\ (top\ (Q[[Q]]\ (Value^* \rightarrow Stack\ (map\ Int \rightarrow Value\ i^*)))))$
  else $errorAnswer$

  $Q \ : \ Commands \rightarrow StackTransform$
  $Q[[C\ .\ Q]] \ = \ Q[[Q]] \circ C[[C]]$
  $Q[[]] \ = \ \lambda s.\ s$
  $Q[[Q]] \circ C[[C]] \ = \ \lambda s.\ (Q[[Q]]\ (C[[C]]\ s))$

  $C \ : \ Command \rightarrow StackTransform$
  $C[[N]] \ = \ (push\ (Value \rightarrow Result\ (Int \rightarrow Value\ (N[[N]]))))$
  $C[[(Q)]] \ = \ (push\ (Value \rightarrow Result\ (StackTransform \rightarrow Value\ Q[[Q]])))$
  $C[[pop]] \ = \ pop$
  $C[[swap]] \ = \ \lambda s.\ (push\ (top\ (pop\ s))\ (push\ (top\ s)\ (pop\ (pop\ s))))$
  $C[[A]] \ = \ (arithop\ A[[A]])$

  $A \ : \ ArithmeticOperator \rightarrow (Int \rightarrow Int \rightarrow Result)$

$A[[sub]] \;=\; \lambda i_1 i_2.\ (Value \rightarrow Result\ (Int \rightarrow Value\ (i_1 -_{Int} i_2)))$
Similarly for add, mul

$N\ :\ Intlit \rightarrow Int$
N maps integer numerals to the integer numbers they denote.

Using the above semantics, write the evaluation of:

(a) $P[[(postfix\ 3\ 3\ sub\ swap\ pop)]]\ [56,\ 90]$

(b) $P[[(postfix\ 2\ 5\ swap\ sub\ pop)]]\ [7,\ 8]$