THE UNIVERSITY OF NEW SOUTH WALES

Sample Examination

June 2007

COMP2111
System Modelling and Design

Time allowed: 1 hours

Total number of questions: 20

Questions are not necessarily of equal value.

Questions must be answered in pencil on the Generalised Answer Sheet.

Be careful: if you wish to delete a selection, make sure that you completely erase the old selection.

Answers written in this book or the accompanying work book will not be read.

A summary of the B mathematical toolkit is attached to this paper. During the examination, it may be detached from the paper if desired.

This paper may not be retained by the candidate.
Instructions for this paper

Please read carefully

This is a multiple-choice examination in which there may be more than one correct answer to any question.

Marking scheme: If there are \( N \) correct answers to all questions in the paper then each correct selection to a question will earn you \( \frac{1}{N} \) of the total marks for the paper, and each incorrect selection will result in \( \frac{1}{2N} \) of the total marks for the paper being deducted.

For any one question, your cumulative mark for that question will never be less than zero. So, if \( C \) and \( W \) are the number of correct and incorrect selections, respectively, in a question then the mark for that question will be \( \max(C - \frac{W}{2}, 0) / N \times 100\% \) of the total marks for the paper.

Examples: Assume a paper with 50 questions and five choices in each question. Assume the total number of correct choices across the whole paper is 100 and suppose the total mark for the paper is 100.

In one question you select:

- 1 correctly and make no other choice: your mark will be \( (1/100) \times 100 = 1 \) mark.
- 1 correctly and 1 incorrectly: your mark will be \( (1 - 0.5)/100 \times 100 = 0.5 \) mark.
- 2 correct and 1 incorrect: your mark will be \( (1 + 1 - 0.5)/100 \times 100 = 1.5 \) mark.
- 1 correct and 2 incorrect: your mark will be \( (1 - 0.5 - 0.5)/100 \times 100 = 0 \) mark.
- 1 correct and 3 incorrect: your mark will be \( \max(1 - 1, 5, 0)/100 \times 100 = 0 \) mark.

Note

1) Even though the phrasing of a question may imply more than one answer, there could be only one correct answer.

2) Answers that are only conditionally correct are not considered correct for the purposes of this examination.
The number of correct answers in this paper is 48.

Note carefully: all of the B mathematics in the following questions are in marked-up form, unless otherwise stated.

1) Consider the following partial specification of a StockControl machine.

MACHINE StockControl (maxstock)
CONSTRAINTS maxstock ∈ N
SETS STOCK
CONSTANTS EmptyBag, BagCount
PROPERTIES
   card(STOCK) = maxstock ∧ 
   EmptyBag = STOCK × {0} ∧ 
   BagCount ∈ Bag(STOCK) → (STOCK → N) ∧ 
   ∀bag.(bag ∈ Bag(STOCK) ⇒ BagCount(bag) = EmptyBag < bag)
VARIABLES warehouse
INARIANT warehouse ∈ Bag(STOCK)
INITIALISATION warehouse := {}
DEFINITIONS Bag(X) ≡ X → N
END

In this machine we are modelling a warehouse as a bag of STOCK.

We wish to specify an operation AddStock(stock,quantity) that will add some quantity of some stock to the warehouse, either adding new stock or adding to existing stock in the warehouse. The specification of the operation is as follows:

   AddStock(stock,quantity) ≡
   PRE...
   THEN warehouse(stock) := BagCount(warehouse)(stock) + quantity
   END

Which of the following are sufficiently strong preconditions?

A) stock ∈ dom(warehouse) ∧ quantity ∈ N.
B) stock ∈ STOCK ∧ quantity ∈ N.
C) stock ∈ STOCK ∧ quantity ∈ N.
D) stock ∈ dom(warehouse) ∧ quantity ∈ N.
E) stock ∈ dom(warehouse) ∧ quantity ∈ ran(warehouse).
2) Continuing the example in question 1 above.

We wish to specify an operation \( \text{RemoveStock}(stock, quantity) \), which will remove some quantity of stock from the warehouse. The specification of the operation is as follows

\[
\text{RemoveStock}(stock, quantity) \triangleq \\
\text{PRE} stock \in \text{dom}(\text{warehouse}) \land quantity \in \mathbb{N}_1 \land quantity \leq \text{warehouse}(stock) \\
\text{THEN} \ldots \\
\text{END}
\]

Which of the following would be satisfactory substitutions for the body of the operations?

A) \( \text{warehouse}(stock) := \text{warehouse}(stock) - quantity. \)
B) \( \text{warehouse}(stock) := \text{BagCount}(\text{warehouse})(stock) - quantity. \)
C) \( \text{IF} \) quantity \( < \) \( \text{warehouse}(stock) \)
\( \text{THEN} \) \( \text{warehouse}(stock) := \text{warehouse}(stock) - quantity \)
\( \text{ELSE} \) \( \text{warehouse} := \{stock\} \triangleleft \text{warehouse} \)
\( \text{END} \)
D) \( \text{BagCount}(\text{warehouse})(stock) := \text{BagCount}(\text{warehouse})(stock) - quantity. \)
E) \( \text{warehouse} := (\text{warehouse} \leftarrow \{stock \mapsto \text{warehouse}(stock) - quantity\}) \triangleright \{0\} \)

3) Which of the following rules are correct?

A) \( f \in X \Rightarrow Y \Rightarrow f \in X \Rightarrow Y \)
B) \( f \in X \Rightarrow Y \Rightarrow f \in X \Rightarrow Y \)
C) \( f \in X \Rightarrow Y \Rightarrow f^{-1} \in Y \Rightarrow X \)
D) \( (P \Rightarrow Q) \land P \Rightarrow Q \)
E) \( f \in X \Rightarrow Y \Rightarrow f^{-1} \in Y \Rightarrow X \)

4) If \( Op_A \) is refined by \( Op_C \), which of the statements are consistent with refinement?

A) \( Op_A \) and \( Op_C \) must have identical behaviour.
B) The precondition of \( Op_C \) is no stronger than the precondition of \( Op_A \).
C) The precondition of \( Op_C \) is no weaker than the precondition of \( Op_A \).
D) Outside of the precondition of \( Op_A \), \( Op_C \) can do whatever it likes.
E) If \( Op_A \) is feasible then \( Op_C \) is guaranteed to be feasible.

5) The following statements state similarities between \text{INCLUDES} and \text{SEES} clauses. Which of the following statements are true?

A) Machine that use either clause are not excluded from refinement.
B) Both clauses are transitive.
C) Both clauses provide write access to the included and seen machine variables.
D) Both clauses provide access to all the included and seen machine operations.
E) The machine parameters in both included and seen machines must be instantiated in the respective \text{INCLUDES} or \text{SEES} clause.
6) Which of the following are correct?
   A) between them sets, sequences and bags are aggregates that provide membership, ordering and multiplicity.
   B) sequences provide ordering.
   C) sequences and bags provide ordering.
   D) bags provide multiplicity and sequences provide ordering.
   E) sets provide membership.

7) Which of the following statements about guarded substitutions are correct?
   A) If the guard is satisfied then the substitution will be correct.
   B) The substitution will proceed on the assumption that the guard was satisfied before the substitution was invoked.
   C) A guard ensures that the state invariant is maintained.
   D) Maintenance of the state invariant requires the guard to be satisfied.
   E) If a guard is not satisfied then the substitution is skipped.

8) Which of the following statements about preconditions are correct?
   A) Satisfying the precondition ensures that the operation will be correct.
   B) Whether or not the precondition is satisfied, the operation will proceed.
   C) A precondition ensures that the state invariant is maintained.
   D) Maintenance of the state invariant requires the precondition to be satisfied.
   E) The operation will proceed on the assumption that the precondition was satisfied before the operation was invoked.

9) Given \( xx \in \mathbb{N} \), what predicate is specified by the following?
   \[
   \begin{array}{l}
   \text{IF } xx = a \text{ THEN } xx := xx + b \text{ END } \quad xx > c
   \end{array}
   \]
   A) \( xx = a \Rightarrow xx + b > c \)
   B) \( xx + b > c \vee xx > c \)
   C) \( xx + b > c \wedge xx > c \)
   D) \( xx = a \Rightarrow xx + b > c \wedge xx \neq a \Rightarrow xx > c \)
   E) \( xx = a \Rightarrow xx + b > c \vee xx \neq a \Rightarrow xx > c \)

10) Given a function \( f \) that is defined as \( f \in X \rightarrow X \), which of the following are correct?
    A) \( \text{dom}(f) \in X \)
    B) \( \text{dom}(f) \in P(X) \)
    C) \( \text{dom}(f) = X \)
    D) \( \text{dom}(f) \subset X \)
    E) \( \text{ran}(f) \subseteq \text{dom}(f) \)
MACHINE Coin_TYPE

SETS COIN = { One, Two, Five }

CONSTANTS

CoinValue, MoneyValue, SubMoney

PROPERTIES

CoinValue ∈ COIN → N ∧
CoinValue = { One → 1, Two → 2, Five → 5 } ∧
MoneyValue ∈ MONEY → N ∧
MoneyValue = λ money . ( money ∈ MONEY |
    ∑ coin . ( coin ∈ dom ( money ) |
        money ( coin ) × CoinValue ( coin ) ) ) ∧
SubMoney ∈ MONEY ↔ MONEY ∧
∀ ( m1, m2 ) . ( m1 ∈ MONEY ∧ m2 ∈ MONEY ∧ m1 → m2 ∈ SubMoney ⇒
    dom ( m1 ) ⊆ dom ( m2 ) ∧
    ∀ coin . ( coin ∈ dom ( m1 ) ⇒ m1 ( coin ) ≤ m2 ( coin ) ) )

DEFINITIONS MONEY ≜ COIN → N

END

Figure 1: Coin type machine

11) Consider the machine Coin_TYPE in figure 1. This machine defines a set COIN as a set of coin denominations, a “type” MONEY as essentially a bag of COIN, and provides a number of constants useful for determining some properties of coins and money.

Which of the following are correct?

A) \{ One → 1, Two → 1, Five → 1 \} → \{ One → 2, Two → 2, Five → 2 \} ∈ SubMoney
B) \{ One → 2, Two → 2, Five → 2 \} ∈ SubMoney[\{ One → 1, Two → 1, Five → 1 \}]
C) \{ One → 1, Two → 1, Five → 1 \} ∈ SubMoney[\{ One → 2, Two → 2, Five → 2 \}]
D) \{ One → 1, Two → 2 \} → \{ One → 2, Two → 3, Five → 2 \} ∈ SubMoney
E) \{ One → 2, Two → 2, Five → 2 \} → \{ One → 2, Two → 2 \} ∈ SubMoney

5
12) Continuing from question 11.

Another machine SEES Coin_TYPE and has a variable cash of type MONEY. An operation

\[ \text{change} \leftarrow \text{Purchase}(\text{price}, \text{payment}) \]

models a customer paying for some goods with a price \( \text{price} \) and paying with money represented by \( \text{payment} \).

The precondition contains \( \text{price} \in \mathbb{N} \land \text{payment} \in \text{MONEY} \). Which of the following should also be present as conjuncts in the precondition:

A) \( \text{MoneyValue}(\text{payment}) \geq \text{price} \)
B) \( \text{payment} \geq \text{price} \)
C) \( \text{cash} \geq \text{payment} - \text{price} \)
D) \( \text{MoneyValue}(\text{cash}) \geq \text{MoneyValue}(\text{payment}) - \text{price} \)
E) \( \text{MoneyValue}(\text{cash}) \geq \text{payment} - \text{price} \)

13) Continuing with the operation \( \text{Purchase} \), shown in question 12. We need to specify the change given to the customer and it is going to be specified using the following ANY construct:

\[ \text{ANY} \; \text{ch} \; \text{WHERE} \; \text{ch} \; \in \; \text{MONEY} \land \ldots \; \text{THEN} \; \text{change} := \text{ch} \; \text{END} \]

Assume that the money represented by the variable \( \text{cash} \) is adequate to cover the required change. What conjuncts should be in the remaining part of the WHERE constraint.

The change represented by \( \text{change} \) should be exact, and all the money for \( \text{change} \) should come from \( \text{cash} \).

A) \( \text{ch} = \{ \text{One} \mapsto \text{MoneyValue}(\text{payment}) - \text{price} \} \)
B) \( \text{ch} \mapsto \text{cash} \in \text{SubMoney} \)
C) \( \text{MoneyValue}(\text{ch}) = \text{MoneyValue}(\text{payment}) - \text{price} \)
D) \( \text{ch} = \{ \text{Five} \mapsto ((\text{MoneyValue}(\text{payment}) - \text{price}) \mod 5)/5, \)  
   \( \text{Two} \mapsto ((\text{MoneyValue}(\text{payment}) - \text{price}) \mod 5)/2, \)  
   \( \text{One} \mapsto ((\text{MoneyValue}(\text{payment}) - \text{price}) \mod 5) \mod 2 \} \)
E) \( \text{ch} = \text{payment} - \text{price} \)

14) Which of the following are correct?

A) \( \text{Abort} \) can never be implemented.
B) An infeasible construct cannot be refined to a feasible construct.
C) \( \text{Abort} \) can be refined to anything.
D) A feasible construct can be refined to an infeasible construct.
E) A feasible construct can always be refined to a feasible construct.
15) Given two operations, \( Op_A \) and \( Op_C \), in which \( Op_C \) is a refinement of \( Op_A \), which of the following are consistent with the requirements of refinement:

A) In all states the behaviour of \( Op_A \) and \( Op_C \) is identical.
B) When the precondition of \( Op_A \) is false \( Op_C \) does anything.
C) The precondition of \( Op_C \) is stronger than the precondition of \( Op_A \).
D) When \( Op_C \) fails (aborts) \( Op_A \) also fails.
E) The precondition of \( Op_C \) is weaker than the precondition of \( Op_A \).

16) Given the operation:

\[
\text{result} \leftarrow Op \equiv \begin{cases} 
\text{result} : \in \{1, 2, 3\} 
\end{cases}
\]

Which of the following would be satisfactory in the body of a refinement of \( Op \)?

A) \( \text{result} : \in \{1, 2, 3\} \)
B) \( \text{result} : \in \{1, 3\} \)
C) \( \text{result} : \in \{0, 1, 2, 3\} \)
D) \( \text{result} : \in \{\} \)
E) \( \text{result} := 1 \)

17) Consider the following:

i) \[\text{SELECT } \arg \in \{1, 2, 3\} \text{ THEN } \text{result} := 0\]
\[\text{WHEN } \arg \in \{3, 4, 5\} \text{ THEN } \text{result} := 1\]
\[\text{END}\]

ii) \[\text{SELECT } \arg \in \{1, 2\} \text{ THEN } \text{result} := 0\]
\[\text{WHEN } \arg \in \{3, 4, 5\} \text{ THEN } \text{result} := 1\]
\[\text{END}\]

iii) \[\text{SELECT } \arg \in \{1, 2\} \text{ THEN } \text{result} := 0\]
\[\text{WHEN } \arg \in \{4, 5\} \text{ THEN } \text{result} := 1\]
\[\text{END}\]

Assume \( \arg \in 1 \ldots 5 \)

Which of the following are correct?

A) ii) is refined by i).
B) i) is refined by ii).
C) ii) is refined by iii), but the refinement is infeasible.
D) i) is refined by iii), but the refinement is infeasible.
E) i) is refined by iii), and the refinement is feasible.
18) Consider the Sequence machine shown in Figure 2. Sequence is refined by SequenceR shown in Figures 3 and 4. SequenceR gives a list model of a sequence using functions to model “pointers”. Which of the following describes a property specified by the invariant of SequenceR.

A) if the sequence has more than one item then the pointer to the last the last item is not in \(\text{dom}(\text{seqnext})\).
B) if the sequence has more than one item then the pointer to the first item is in \(\text{dom}(\text{seqnext})\).
C) seqnext is a bijective function.
D) seqnext is an injective function.
E) seqlast is always in the \(\text{ran}(\text{seqnext})\).

19) Continuing the example in question 18. Which of the following could be in the Refinement relation?

A) \(\text{ran}(\text{seqval}) = \text{ran}(\text{myseq})\)
B) \(\text{dom}(\text{seqval}) = \text{dom}(\text{myseq})\)
C) \(\text{seqsize} = \text{size}(\text{myseq})\)
D) \(\forall \ pos. (\ pos \in \text{dom}(\text{myseq}) \Rightarrow \text{seqval}(\ \text{seqnext}^{\ pos}(\ \text{firstseq})) = \text{myseq}(\ pos)\)
E) \(\forall \ pos. (\ pos \in \text{dom}(\text{myseq}) \Rightarrow \text{seqval}(\ \text{seqnext}^{\ pos-1}(\ \text{firstseq})) = \text{myseq}(\ pos)\)

20) Continuing the example in question 18. Which of the following comments on the refinement SequenceR are correct?

A) The initialisation is correct.
B) AppendVal is correct.
C) ModVal is correct.
D) GetVal is correct.
E) None of the operations is correct.
MACHINE Sequence (VAL, maxseq)

VARIABLES myseq

ININVARIANT

myseq ∈ seq (VAL) ∧ size (myseq) ≤ maxseq

INITIALISATION myseq := []

OPERATIONS

pos ← AppendVal (val) ≜
    PRE val ∈ VAL ∧ size (myseq) ≠ maxseq THEN
        myseq := myseq ∥ val ||
    END
    pos := size (myseq) + 1

END;

ModVal (pos, val) ≜
    PRE pos ∈ dom (myseq) ∧ val ∈ VAL THEN
        myseq (pos) := val
    END;

val ← GetVal (pos) ≜
    PRE pos ∈ dom (myseq) THEN
        val := myseq (pos)
    END

END

Figure 2: Sequence machine
REFINEMENT SequenceR
REFINES Sequence
SETS ITEM
VARIABLES items, seqfirst, seqlast, seqnext, seqsize, seqval

INVARIANT
\[
\begin{align*}
\text{items} & \subseteq \text{ITEM} \land \text{seqfirst} \in \text{ITEM} \land \\
( \text{seqsize} \neq 0 \Rightarrow \text{seqfirst} \in \text{items} ) \land \\
\text{seqfirst} & \notin \text{ran} ( \text{seqnext} ) \land \\
( \text{seqsize} = 1 \Rightarrow \text{seqfirst} = \text{seqlast} ) \land \\
( 1 < \text{seqsize} \Rightarrow \text{seqfirst} \in \text{dom} ( \text{seqnext} ) ) \land \\
\text{seqnext} & \in \text{items} \land \\
( \text{seqsize} \neq 0 \Rightarrow \text{seqlast} \in \text{items} ) \land \\
( 1 < \text{seqsize} \Rightarrow \text{seqlast} \notin \text{dom} ( \text{seqnext} ) ) \land \\
\text{seqnext} & \in \text{items} \Rightarrow \text{items} \land \\
( \text{seqsize} \leq 1 \Rightarrow \text{dom} ( \text{seqnext} ) = \{ \} ) \land \\
\text{seqsize} & \in \mathbb{N} \land \\
\text{card} ( \text{items} ) & = \text{seqsize} \land \\
\text{seqval} & \in \text{items} \rightarrow \text{VAL}
\end{align*}
\]

Figure 3: Refinement of Sequence Machine (part 1)

INITIALISATION
\[
\begin{align*}
\text{items} , \text{seqnext} , \text{seqsize} , \text{seqval} & := \{ \} , \{ \} , 0 , \{ \} \parallel \text{seqfirst} :\in \text{ITEM} \parallel \text{seqlast} :\in \text{ITEM}
\end{align*}
\]
OPERATIONS

\[ pos \leftarrow \text{AppendVal}(val) \triangleq \]
\[ \text{ANY item WHERE item} \in \text{ITEM} - \text{items} \text{ THEN} \]
\[ \text{items} := \text{items} \cup \{ \text{item} \} \parallel \]
\[ \text{IF } \text{seqsize} = 0 \text{ THEN } \text{seqfirst} := \text{item} \]
\[ \text{ELSE } \text{seqnext}(\text{seqlast}) := \text{item} \]
\[ \text{END } \parallel \]
\[ \text{seqlast} := \text{item} \parallel \text{seqval}(\text{item}) := \text{val} \parallel \]
\[ \text{seqsize} := \text{seqsize} + 1 \parallel \]
\[ pos := \text{seqsize} + 1 \]
\[ \text{END ;} \]

\[ \text{ModVal}(pos, val) \triangleq \]
\[ \text{seqval}(\text{seqnext}^{\text{pos}} - 1 (\text{seqfirst})) := \text{val} ; \]

\[ val \leftarrow \text{GetVal}(pos) \triangleq \]
\[ \text{val} := \text{seqval}(\text{seqnext}^{\text{pos}} (\text{seqfirst})) \]
\[ \text{END} \]

Figure 4: Refinement of Sequence Machine (part 2)
Answers

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>B</td>
</tr>
<tr>
<td>2</td>
<td>C, E</td>
</tr>
<tr>
<td>3</td>
<td>B, C, D</td>
</tr>
<tr>
<td>4</td>
<td>B, D</td>
</tr>
<tr>
<td>5</td>
<td>A</td>
</tr>
<tr>
<td>6</td>
<td>A, B, D, E</td>
</tr>
<tr>
<td>7</td>
<td>E</td>
</tr>
<tr>
<td>8</td>
<td>B, E</td>
</tr>
<tr>
<td>9</td>
<td>D</td>
</tr>
<tr>
<td>10</td>
<td>B, C, E</td>
</tr>
<tr>
<td>11</td>
<td>A, B, D</td>
</tr>
<tr>
<td>12</td>
<td>A, D</td>
</tr>
<tr>
<td>13</td>
<td>B, C</td>
</tr>
<tr>
<td>14</td>
<td>B, C, D, E</td>
</tr>
<tr>
<td>15</td>
<td>B, D, E</td>
</tr>
<tr>
<td>16</td>
<td>A, B, E</td>
</tr>
<tr>
<td>17</td>
<td>B, C, D</td>
</tr>
<tr>
<td>18</td>
<td>A, B, D</td>
</tr>
<tr>
<td>19</td>
<td>A, C, E</td>
</tr>
<tr>
<td>20</td>
<td>A, B, C</td>
</tr>
</tbody>
</table>