Вопросы к экзамену по курсу “Формальная спецификация и верификация программ” 2012/2013

Версия 1.0

Время выполнения всей работы - 120 минут. В билете будет 10 задач тех типов, которые представлены в этом документе.

RSL и RAISE метод

“практические” задания

A1. По данной явной спецификации построить эквивалентную неявную спецификацию без использования рекурсии и циклов.

Пример:

```
variable x : Int, y : Int
value f : Unit -~-> write x, y Nat
      f() is (if x + y > 0 then y := x end;x+y)
```

A2. По данной неявной спецификации функции f построить явную спецификацию этой функции, являющуюся уточнением данной неявной спецификации.

Пример:

```
value f : Nat-list >< Nat -~-> Nat
f(nums, max) as n
post n < len nums /\ sum(nums, n) < max /\ sum(nums, n+1) >= max
pre max > 0 /\ max <= sum(nums, len nums)
value sum : Nat-list >< Nat -> Nat
sum(s, n) is if n = 0 then 0 else hd s + sum(tl s, n-1)
pre n <= len s
```

A3. По данной алгебраической спецификации вычислить значение выражения, если это возможно сделать. В ответе представить процесс вычисления в виде переписывания термов.

Пример: выражение: first(add(add(add(add(add(empty, 5), 100), 0), 2), 20)).

алгебраическая спецификация:

```
scheme A0 = class
      type S
      value empty : S,
            add : S >< Nat -> S,
            first : S -~-> Nat
      axiom all s : S, e : Nat :-
            first(add(s, e)) is if e > 10 then 2 * first(s)
```

```
                    elsif e > 5 then first(s)
                    else 0 end
end
```

A4. Закончите аксиому алгебраической спецификации типа данных S, формализующую поведение указанных далее операций над этим типом.
Пример: S - очередь ограниченного размера capacity, add - добавление в очередь, size - количество элементов в очереди.

```
scheme QUEUE = class
        type S, E
        value capacity : Nat
        value empty : S,
                add : S >< E -~-> S,
                size : S -> Nat
        axiom all s : S, e : E :- size(add(s, e)) is  . . .
```

A5. Указать, какие из *стилевых* характеристик относятся к данной спецификации (явная, неявная, алгебраическая, моделеориентированная, аппликативная, императивная, формальная, неформальная).
Пример: (в качестве спецификации может быть любой текст, который встретился в курса, а именно RSL-спецификация, PVS-теоремы)

```
scheme S = class
      type T, E
      variable x : Nat
end
```

A6. Запишите на RSL *полный* инвариант указанного типа согласно заданию, не меняя определений типов. При необходимости можете добавить новые разделы спецификации. Полный инвариант типа - тот, из которого следуют все возможные инварианты этого типа.
Пример: `type E, S = E-list`
S является стеком ограниченного размера

A7. Чему равно значение каждого из следующих выражений на RSL? Если ответ зависит от каких-либо переменных, записать эквивалентное выражение, максимально упростив исходное выражение (типы всех переменных таковы, что все написанные выражения семантически корректны).
Пример:
а) <. x**2 | x in <.1 .. 3.> :- abs x < 10 .>
б) {x, y, x} \ { x }
в) [1 +> 2] !! [1 +> 3] union [2 +> 2]
г) x > 0 /\ x + y < 1 /\ y > 2
д) x isin s => x isin (s union t)

A8. Приведите пример инвариантного свойства состояния указанной подсистемы/системы или её объекта (объектов) управления, которое она должна поддерживать (на русском языке).
Пример: диспетчер виртуальной памяти.

A9. Для данной подсистемы/системы привести *один* пример её операции, которую имеет смысл на ранних этапах формальной спецификации определить частично (т.е. не тотально). Укажите условие, при котором эта операция будет не определена в такой спецификации.
Пример: диспетчер виртуальной памяти.

A10. Опишите возможное отношение соответствия между двумя моделями заданных видов.
Пример: неявная спецификация и явная спецификация.

A11. Сформулировать на RSL возможную функцию абстракции от типа Simpl в тип Sspec.
Пример: type A, B, Simpl = A -m-> B, Sspec = (A><B)-set

A12. В процессе уточнения моделей была построена следующая цепочка типов: S1 ~~~> S2 ~~~> S3 (S2 - уточняет S1, S3 - уточняет S2). Для данных S1 и S3 приведите S2, нетривиально отличающийся от S1 и S3.
Пример: type A, B, C, S1 = (A >< B >< C)-set, S3 = A -m-> (B >< C)

A13. Верифицируйте описанную в задании реализацию программной системы. Т.е. кратко сформулируйте формальную модель реализации (поместите его в ответ), формальной сформулируйте требование корректности (поместите его в ответ) и проведите верификацию (в ответ поместите краткое описание, как вы это делали, и вердикт).
Пример: Разрабатывается управляющая микросхема аппарата по размену купюр и монет. Аппарат принимает одну купюру (10р, 50р, 100р) или монету (10р) и выдает монеты достоинством 5р на ту же сумму. У аппарата есть лампочка, которая моргает в том случае, если размен купюры невозможен (например, закончились монеты или обнаружена внутренняя проблема), и горит непрерывно, если невозможен как размер купюры, так и размен монеты. Разработчики решили завести в аппарате 2 регистра. Если у автомата запрашивают размен монеты, то из второго регистра вычитается число 2 и его значение копируется в первый регистр. Если у автомата запрашивают размен купюры, то из первого регистра вычитается 2, 10 или 20 (в зависимости от достоинства купюры) и его значение копируется во второй регистр. Если в первом регистре число меньше 20, лампочка начинает моргать. Если во втором регистре число меньше 2, лампочка переводится в режим постоянного горения.

A14. Для данной неявной спецификации ответить на вопрос, определяет ли она однозначно возвращаемый результат. Ответ обосновать.

Пример:
value find : Nat-list >< Nat -~-> Nat
find(xs, x) as k
post k isin inds xs /\ xs(k) >= x /\ xs(k) < 2*x

А15. Опишите на естественном языке из каких частей будет состоять контракт приведенного ниже метода на языке Java и какие свойства будут составлять каждую из частей.
Пример:
class java.io.InputStream
public abstract int read() throws IOException
Reads the next byte of data from the input stream. The value byte is returned as an int in the range 0 to 255. If no byte is available because the end of the stream has been reached, the value -1 is returned. This method blocks until input data is available, the end of the stream is detected, or an exception is thrown. A subclass must provide an implementation of this method.
Returns: the next byte of data, or -1 if the end of the stream is reached.
Throws: IOException - if an I/O error occurs.

А16. Задание на понятие абстракции.
Пример: Ваша команда разрабатывает модуль почтового веб-клиента для автоматизации работы с контактами корпоративной почты. Модуль нужен для упрощения выбора адресатов писем и ведения списка писем. Опишите на RSL необходимую и достаточную абстракцию почтового веб-клиента (набор сигнатур и сортов) для разработки и исследования модуля. Прокомментируйте на естественном языке семантику введенных вами имён функций и типов данных.

А17. Вопрос по теме “Formal Testing”.
Пример: Достаточно ли для выполнения критерия MC/DC обеспечить покрытие всех конъюнктов в СДНФ разложении постусловия функции и почему?

А18. Вопрос по теме “UniTESK”.
Пример: приведите не менее двух критериев полноты на основе программных контрактов.

“теоретические” задания
Б1. Дать определение термина в одном-двух предложениях. Определение должно пояснять смысл термина.
Пример: предусловие операции.

Б2. Привести *одно* отличие двух данных понятий.
Пример: VDM и RAISE.

Методы Флойда

“практические” задания

A1. Для данных пред- (pre) и пост- (post) условия, записанных на RSL, предложите реализацию функции P, которая является частично корректной относительно них. Для записи функции так же используйте RSL. Функция должна завершаться на максимальном числе входных данных, удовлетворяющих предусловию. Функция P должна быть представлена в явном виде (в частности, без неявных конструкций таких как кванторы, неявный let, сокращенные (comprehended) выражения), без использования бесконечных значений целых чисел, бесконечных множеств, chaos и т. п.). Функция P должна быть корректно определена (в частности, все частично определенные операции, например, hd, должны вызываться только для тех аргументов, на которых они определены).

Пример:

pre(x, y) = hd x > hd y

post(x, y, z) = z(2) isin elems tl tl x

A2. В блок-схеме программы выделена подсхема, в которую входит ровно одна дуга и из которой выходит ровно одна дуга. Известно, что про эту схему доказана полная корректность относительно некоторых пред- и пост- условий. При этом дуге, ведущей в подсхему, была сопоставлена точка сечения A и оценочная функция uA, а дуге, исходящей из подсхемы, точка сечения B и оценочная функция uB. Фундированное множество обозначено как W. Для данных uA, uB, W предложить содержимое подсхемы (в виде программы на RSL) и утверждение в точке A, гарантирующее достижимость B из A на максимальном числе данных. x - входная переменная программы, y1, y2 - промежуточные переменные.

Пример:

uA = len y1 + len y2

uB = uA

W = (Nat, <)

A3. Какое требуется минимальное число точек сечения, чтобы доказать частичную корректность *хотя бы одной* программы, имеющей указанные особенности. Ответ кратко обосновать.

Пример:

блок-схема содержит 2 вложенных цикла

A4. Что из перечисленного может быть индуктивным утверждением?(все свободные переменные целочисленные, эквивалентные преобразования формул не производились, за исключением, быть может, опускания несущественных скобок)

Пример:

а) t > 0
б) t
в) all t : Int :- (t > 0 => t+1 > 0)
г) (Nat U { -2 }, <)
д) all y : Int :- exists t : Int :- (y > t => t > 0)

А5. Что из перечисленного может быть оценочной функцией ? (все указанные переменные целочисленные, эквивалентные преобразования формул не производились, за исключением, быть может, опускания несущественных скобок)
Пример:
а) t > 0
б) t
в) all t : Int :- (t > 0 => t+1 > 0)
г) (Nat U { -2 }, <)
д) all y : Int :- exists t : Int :- (y > t => t > 0)

А6. Что из перечисленного может быть фундированным множеством ? (все указанные переменные целочисленные, эквивалентные преобразования формул не производились, за исключением, быть может, опускания несущественных скобок)
Пример:
а) t > 0
б) t
в) all t : Int :- (t > 0 => t+1 > 0)
г) (Nat U { -2 }, <)
д) all y : Int :- exists t : Int :- (y > t => t > 0)

А7. Что из перечисленного может быть условием корректности ? (все указанные переменные целочисленные, эквивалентные преобразования формул не производились, за исключением, быть может, опускания несущественных скобок)
Пример:
а) t > 0
б) t
в) all t : Int :- (t > 0 => t+1 > 0)
г) (Nat U { -2 }, <)
д) all y : Int :- exists t : Int :- (y > t => t > 0)

А8. Что из перечисленного может быть условием верификации ? (все указанные переменные целочисленные, эквивалентные преобразования формул не производились, за исключением, быть может, опускания несущественных скобок)
Пример:
а) t > 0
б) t
в) all t : Int :- (t > 0 /\ true => t+1 > 0)

г) (Nat U { -2 }, <)
д) all y : Int :- exists t : Int :- (y > t => t > 0)

А9. Что из перечисленного может быть условием завершимости ? (все указанные переменные целочисленные, эквивалентные преобразования формул не производились, за исключением, быть может, опускания несущественных скобок)
Пример:
а) t > 0
б) t
в) all t : Int :- (t > 0 => t+1 > 0)
г) (Nat U { -2 }, <)
д) all y : Int :- exists t : Int :- (y > t => t > 0)

А10. Сколько необходимо операторов языка блок-схем для записи модели следующей функции языка Си ? Ответ обосновать. Что делает эта функция?
Пример:

```
int P(int x) {
        int y1 = x, y2 = 0;
        if (x < 0) {
                return 0;
        } else {
                while (y1 > y2) {
                        y1 --; y2--;
                }
                return y1;
        }
}
```

А11. Каково минимальное и максимальное число утверждений, которые необходимо доказать при использовании лишь одной точки сечения в рамках доказательства частичной корректности заданной функции P (она задана на языке Си) ? Ответ обосновать. Если одной точки сечения недостаточно, обосновать это.
Пример:

```
int P(int x) {
        int y1 = x, y2 = x;
        do {
                if (y1 > y2 + y2) {
                        y2 ++;
                }
                y1 += y2;
        } while (y1 < y2);
        return y1 + y2;
}
```

A12. Каково минимальное и максимальное число утверждений, которые необходимо доказать при использовании лишь одной точки сечения в рамках доказательства полной корректности заданной функции P (она задана на языке Си) ? Ответ обосновать. Если одной точки сечения недостаточно, обосновать это.
Пример:

```
int P(int x) {
        int y1 = x, y2 = x;
        do {
                if (y1 > y2 + y2) {
                        y2 ++;
                }
                y1 += y2;
        } while (y1 < y2);
        return y1 + y2;
}
```

A13. Доказать или опровергнуть теорему о частичной или полной корректности. Заглавными буквами обозначены программы, строчными - формулы.
Пример:
Верно ли, что для любых a, b если {a}P{b}, то {b}P{a} ?

A14. Дано выражение. Может ли оно быть оценочной функцией и, если может, то при каких условиях? Свой ответ кратко аргументируйте по определению.
Пример: выражение: x (x - очередь из целых чисел)

“теоретические” задания
Б1. Дать определение термина в одном-двух предложениях. Определение должно пояснять смысл термина.
Пример: частичная корректность.

Б2. Приведите *одно отличие и одно сходство* данных двух понятий.
Пример: полная корректность и частичная корректность.

Инструмент PVS и язык ACSL

“практические” задания

A1. Каково минимальное количество листовых вершин в дереве доказательства следующей теоремы (не используя prop, assert и grind) ?
Пример: (A IMPLIES B AND C) AND A IMPLIES B

A2. Приведите PVS теорему, в которой бы встречался указанный кусок доказательства (последовательность команд PVS, все команды завершаются результативно).
Пример: (skosimp) (assert)

A3. Можно ли использовать PVS для приведенной цели? Если можно, напишите *одно* предложение, как.
Пример: формальная спецификация функциональных свойств программ.

A4. Перепишите на языке PVS аксиому с языка RSL.
Пример:
axiom forall x : X, y : Y :- f(x) + f(y) > 0 => f(x) = 2*f(y) /\ f(x+y) > f(x)

A5. Задание на дерево доказательства PVS.
Пример: Приведите пример теоремы на PVS, в дереве доказательства которой будет не менее 2 листовых вершин. Ответ обосновать.

A6. Сформулируйте на PVS любую *одну* из теорем для данной программы на языке Си, снабженной спецификацией на ACSL. Теоремы строятся для доказательства полной корректности реализации относительно спецификации согласно методам Флойда..
Пример:

```
/*@ ensures if (x < 0) \result == 0; else \result == x/2; */
int P(int x) {
        int y1 = x, y2 = 0;
        if (x < 0) {
                return 0;
        } else {
                /*@ loop variant y1; */
                while (y1 > y2) {
                        y1 --; y2--;
                }
                return y1;
        }
}
```

A7. Сформулировать на ACSL контракт функции, описанной в задании.
Пример: максимум массива, сигнатура функции int max(int *values, int length);

А8. Удовлетворяет ли заданная реализация на языке Си своему контракту? Ответ обосновать.

Пример:

```
/*@     requires x > 3 && y > 0;
        ensures \result > 1;  */
int P(int x, int y) {
        if (x < y+1) {
                return y - x;
        } else {
                return x;
        }
}
```

“теоретические” задания

Б1. Что делает указанная команда PVS ?

Пример: split

Б2. Напишите два сходства приведенных двух понятий.

Пример: антецедент и консеквент.