Midterm 6892

Theodore S. Norvell

2012 Oct 11

Your name please:

The following abstract definiton of DynamicArray will be of use in Q3

class $DynamicArray[T]$
    public readonly var $s : \mathrm{Seq}[T] := [\,]$
    public method $getLength(\ ) : \mathrm{int}$
        postcondition $result = length(s)$
    public method $get(\ i : \mathrm{int}\ ) : T$
        precondition $0 \le i < length(s)$
        postcondition $result = s(i)$
    public method $set(\ i : \mathrm{int},\ v : T\ )$
        precondition $0 \le i \le length(s)$
        changes s
        postcondition $s = s_0[0,..i]\hat{}[v]\hat{}s_0[i+1,..length(s)]$
    public method $clipTo(\ i : \mathrm{int}\ )$
        precondition $0 \le i \le length(s)$
        changes s
        postcondition $s = s_0[0,..i]$
end $DynamicArray$

Q0 [5] Express the following pre- and postcondition using quantifier notation (or other mathematical notation). You may assume that the character set is such that the 26 lowercase letters of English are all greater or equal that 'a' and less or equal to 'z', and that no other character has this property.

procedure lessThan($n : \mathbb{N}, a : \text{Array[Char]}, b : \text{Array[Char]}$) : Bool

precondition: a and b have the same length which is n and all characters in both arrays are lower-case letters of the English alphabet.

changes nothing

postcondition: The result is true if a is prior to b in alphabetic (lexicographic) order. Otherwise the result is false.

Do not implement the procedure, just specify it. Note that "fgh" is prior to "ggg", for example.

Q1 (a) [5] Supply an invariant for the following loop. Function f is the Fibonacci function defined by

$$f(0) = 1 \qquad f(1) = 1 \qquad f(n+2) = f(n) + f(n+1), \text{ for all } n \in \mathbb{N}$$

$\{\ k = K \geq 0\ \}$
var $s := 1$
var $t := 1$
$\{\ I :$

$\}$
while $k > 0$ do
    var $u := t$
    $t := s + t$
    $s := u$
    $k := k - 1$
end while
$\{\ s = f(K)\ \}$

(b) [5] List all *boolean expressions*, that should be shown to be valid in order to show that this proof outline is valid.

(c) [3] Give a variant expression that can be used to show that the loop terminates.

Q2 [10] Design the missing part of this proof outline. The running time should be roughly proportional to n. Do not change n nor the items of a. Assume that a and b are arrays of numbers of the same type. Be sure to write the invariant.

$\{\ a.\text{length} = n \wedge b.\text{length} = n + 1\ \}$

$\{\ \forall i \in \{0, .., n\} \cdot b(i) = \sum_{k \in \{0,..i\}} a(k)\ \}$

Q3 [10] Reminder: $\mathbb{N}$ is the type (or set) of all natural numbers, $\mathbb{Z}$ is the type of all integers, and $\mathbb{N} \overset{\text{tot}}{\to} \mathbb{Z}$ is the type of infinite sequences of integers. Suppose *zeros* is the function such that $zeros(n) = 0$, for all $n \in \mathbb{N}$. Suppose that we have the following abstract data type

---

class InfSeq

var $s : \mathbb{N} \overset{\text{tot}}{\to} \mathbb{Z} := zeros$
// abstract invariant: only a finite number of items are not 0
abstract invariant $|\{i \in \mathbb{N} \mid s(i) \neq 0\}| \in \mathbb{N}$

procedure get($i : \mathbb{N}$)
precondition $true$
postcondition $result = s(i)$

procedure put($i : \mathbb{N}, v : \mathbb{Z}$)
precondition $true$
changes s
postcondition $s(i) = v \wedge \forall j \in \mathbb{N} \cdot j \neq i \Rightarrow s(j) = s_0(j)$

end InfSeq

We want to represent a mutable infinite sequence InfSeq using a dynamic array (see first page)
(a) Give the declaration of all concrete fields

(b) Give the linking invariant. (Recall that the linking invariant should imply the abstract invariant.)
