Soft Eng 3M04
Mid-Term II 2003

Dr. Jacques Carette

Name: _____

Student No.: _____

- This midterm contains 5 questions on 3 double-sided pages (including this one).
- This midterm will be marked out of 50. There are 54 total marks available.
- Answer the question in the space provided.
- Make sure that your name is on all sheets.
- You may seperate the last page with the MIS/MID and the definition of the mod function.
- Make sure that you do not get stuck on one question; use your time wisely.

---

1. Give the value of all state variables, output and/or exception after each call in the calling sequence for the MIS and MID on the last page. On the first line, write down the complete initial state. Assume that 'a=int. *[32]*

	state		output		exceptions	
call	MIS	MID	MIS	MID	MIS	MID
isempty()						
* init()						
addback(6)						
addfront(5)						
removefront()						
addback(4)						
addfront(3)						
isempty()						
* addback(2)						
* addback(1)						
removefront()						
removeback()						
removefront()						
removeback()						
isempty()						

2.

$$t \rightarrow ((t = (num \geq 0)) \wedge |s| = num \wedge$$
$$(\forall i : int.0 \leq i \leq num - 1 \rightarrow s[|s| - i - 1] = d[(start + i) \bmod \text{Max}]))$$

Evaluate the abstraction function given above for the starred (*) states indicated in question 1. Please show all relevant details. *[8]*

init()	
addback(2)	
addback(1)	

3. (true/false) It is possible to write an expression using the state variables of the MID on the last page which will always be equal to the MIS state variable *out*. *[1]*

4. Explain what your answer to question 3 means with respect to the abstraction function from question 2 from a verification and validation perspective. *[2]*

5. Given the declarations

val s : ‘a seq
val f : (‘a → bool) → ((‘a → ‘b) → (‘a seq → ‘b seq))
val g : ‘a → bool
val h : ‘a → ‘b

(a) Write a formula that states that all the elements of s are different. *[2]*

(b) What is the type of f(g) ? *[2]*

(c) f(h)(g)(s) is an invalid expression - explain why. *[1]*

(d) Write a formula which asserts that all elements of s satisfy the predicate g. *[1]*

(e) Let ‘a=int for this question. Write down a value for s which would make the following formula true. *[1]*

$$|s| \geq 5 \wedge (\forall i : int.0 \leq i < |s| - 1 \rightarrow s[i] < s[i+1]) \wedge (\exists j : int.0 \leq j < |s| \wedge s[j] < -100)$$

(f) Write an expression which would be found in the MIS for f which would assert that $f(g)(h)(s)$ is a sequence of elements y_j of type ‘b, and where $y_j = h(x_i)$ for some $x_i \in s$, and where we select exactly those elements x_i for which $g(x_i)$ is true. *[4]*

The following MIS and MID will be used for several questions.

Common information:

Used External Functions: NONE

Used External Data Types: ‘a

Exported Constants: Max:int = 4

Exported Functions:

Name	Input Types	Output Types	Exceptions
init			
addfront	‘a		notinit, full
addback	‘a		notinit, full
removefront		‘a	notinit, empty
removeback		‘a	notinit, empty
isempty		bool	notinit

MIS

State Variables:

s : ’a seq

t : bool := false

out : ‘a

Transition Functions:

init()

Transition: $s :=<>$

$t := \text{true}$

addfront(n:’a)

Exception: $\neg t \Rightarrow$ notinit

$|s| \geq \text{Max} \Rightarrow$ full

Transition: $s := s \parallel n$

addback(n:’a)

Exception: $\neg t \Rightarrow$ notinit

$|s| \geq \text{Max} \Rightarrow$ full

Transition: $s := n \parallel s$

‘a removefront()

Exception: $\neg t \Rightarrow$ notinit

$|s| = 0 \Rightarrow$ empty

Transition: $out := s[|s| - 1]$

$s := s[0..|s| - 2]$

Output: out

‘a removeback()

Exception: $\neg t \Rightarrow$ notinit

$|s| = 0 \Rightarrow$ empty

Transition: $out := s[0]$

$s := s[1..|s| - 1]$

Output: out

bool isempty()

Exception: $\neg t \Rightarrow$ notinit

Output: $|s| = 0$

MID

Variables:

’a d[0..Max-1]

int num := -1, start := 0

Exported Functions:

```
init()
    num := 0;
    start := 0;
addfront(n:'a)
    if num < 0 then ERROR(notinit)
    else if num ≥ Max then ERROR(full)
    else start := (start-1) mod Max;
        d[start] := n; num := num + 1
addback(n:'a)
    if num < 0 then ERROR(notinit)
    else if num ≥ Max then ERROR(full)
    else d[(start+num) mod Max] := n;
        num := num + 1
'a removefront()
    if num < 0 then ERROR(notinit)
    else if num = 0 then ERROR(empty)
    else start := (start+1) mod Max;
        num := num - 1;
        RETURN(d[(start-1) mod Max])
'a removeback()
    if num < 0 then ERROR(notinit)
    else if num = 0 then ERROR(empty)
    else num := num - 1;
        RETURN(d[(start+num) mod Max])
bool isempty()
    if num < 0 then ERROR(notinit)
    else RETURN(num=0)
```

(see back of page for more information)

Convention: if s is of type 'a seq, then s[a..b] with $b < a$ is understood to be <>.

Reminder:

x	...	-4	-3	-2	-1	0	1	2	3	4	5	6	...
x mod 4	...	0	1	2	3	0	1	2	3	0	1	2	...