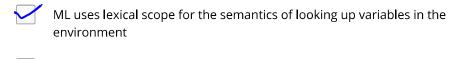
Exam for Part A

测验, 15 个问题

8 points

1。

Check a box if and only if it is an accurate description of ML



ML has no language constructs for creating mutable data

ML has a REPL as part of the definition of the language

ML is statically typed

12 points

2.

Here is a particular list of pairs in ML:

For each *pattern* below, check the box if and only if this pattern matches the value above.



x::(y::z)

(a,b,c)::d

(a,b)::(c,d)::(e,f)::[]

(a,b)::(c,d)::(e,f)::g

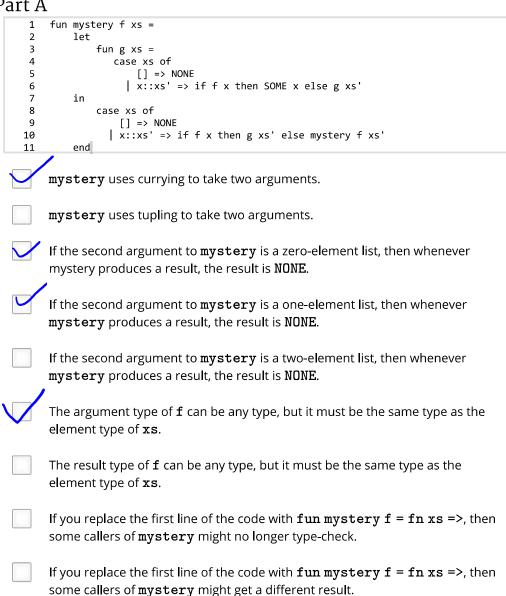
31 points

3.

For each of the statements below, check the box if and only if the statement is true regarding this ML code:

Exam for Part A

测验, 15 个问题



 \mathbf{g} is a tail-recursive function.

For the entire computation of a call like mystery someFun someList, the total number of times someFun is called is *always* the same as the length of someList (for any someFun and someList).

For the entire computation of a call like mystery someFun someList, the total number of times someFun is called is *sometimes* the same as the length of someList (depending on someFun and someList).

For the entire computation of a call like mystery someFun someList, the total number of times someFun is called is *never* the same as the length of someList (for any someFun and someList).

8 points

4.

The **null** function is predefined in ML's standard library, but can be defined in many ways ourselves. For each suggested definition of **null** below, check the box if and only **Exam for Piatre A**nction would behave the same as the predefined null function whenever the function below is called.

Note: Consider only situations where calls to the functions below type-check.

fun null xs = case xs of [] => true | _ => false

fun null xs = xs=[]

fun null xs = if null xs then true else false

fun null xs = ((fn z => false) (hd xs)) handle List.Empty => true

3 points

5.

The next four questions, including this one, relate to this situation: Suppose somebody has written a library for a collection of strings (perhaps implemented as some sort of linked list of strings or tree of strings, but the details do not matter). The library includes higher-order functions map, filter, and fold that operate on these collections and have their conventional meanings. For each problem below, decide which of these library functions is the best to use for implementing the desired function.

(For those needing a precise definition of best: On this exam, the best function, given appropriate arguments, returns the final result you need, meaning you need no more computation after calling the function. If multiple functions can do this, choose the one that can be used by passing it the function argument that itself does the least amount of work.)

Desired function: Take a collection of strings and produce a new collection where each string in the output is like a string in the input except the string has any space characters removed.

map filter

fold

3 points

6

Desired function: Take a collection of strings and return a string that is the concatenation of all the strings in the collection.

filter

fold

测验, 15 个问题

points

7。

Desired function: Take a collection of strings and a number n and return how many strings in the collection have a length that is a multiple of n.

- map
- filter
- fold

3 points

8.

Desired function: Take a collection of strings and return a collection containing the strings in the input collection that start with a capital letter.

- map
- filter
- fold

4 points

9.

This datatype binding and type synonym are useful for representing certain equations from algebra:

Which of the mathematical equations below could not be elegantly represented by a value of type equation?

- x+y=z
- $(x+4) + z = 7 \cdot y$
- $\int x^3 \cdot y^2 = z^0$
- 14.2 + 3 = 17.2
- $\int x^y = z$

Exam for Part A

测验, 15 个问题

10。

Here is a particular polymorphic type in ML:

```
1 'a * 'b -> 'b * 'a * 'a
```

For each type below, check the box if and only if the type *is* an instantiation of the type above, which means the type above is more general.

```
string * int -> string * int * int
int * string -> string * int * int
int * int -> int * int * int

foo : int, bar : string} -> {a : string, b : int, c : int}

'a * 'a -> 'a * 'a * 'a
```

3 points

11.

The next 5 questions, including this one, are similar. Each question uses a slightly different definition of an ML signature **COUNTER** with this same structure definition:

```
structure NoNegativeCounter :> COUNTER =
 2
    struct
 3
    exception InvariantViolated
 6
    type t = int
    fun newCounter i = if i <= 0 then 1 else i
9
10
    fun increment i = i + 1
11
12
    fun first larger (i1,i2) =
13
        if i1 <= 0 orelse i2 <= 0
14
        then raise InvariantViolated
15
        else (i1 - i2) > 0
16
17
    end
18
```

In each problem, the definition of **COUNTER** matches the structure definition **NoNegativeCounter**, but different signatures allow clients to use the structure in different ways. You will answer the same question for each **COUNTER** definition by choosing the best description of what it allows clients to do.In this question, the definition of **COUNTER** is:

```
1 signature COUNTER =
2 sig
3   type t = int
4   val newCounter : int -> t
5   val increment : t -> t
6   val first_larger : t * t -> bool
7   end
```



This signature allows (some) clients to cause the

NoNegativeCounter.InvariantViolated exception to be raised.

Exam for F 测验, 15 个问题	This signature makes it impossible for any client to call Part A NoNegativeCounter.first_larger at all (in a way that causes any part of the body of NoNegativeCounter.first_larger to be evaluated).
	This signature makes it possible for clients to call NoNegativeCounter.first_larger, but never in a way that leads to the NoNegativeCounter.InvariantViolated exception being raised.
	3 points
	12。 In this question, the definition of COUNTER is:
	<pre>1 signature COUNTER = 2 sig 3 type t = int 4 val newCounter : int -> t 5 val first_larger : t * t -> bool 6 end 7</pre>
	This signature allows (some) clients to cause the NoNegativeCounter.InvariantViolated exception to be raised.
	This signature makes it impossible for any client to call NoNegativeCounter.first_larger at all (in a way that causes any part of the body of NoNegativeCounter.first_larger to be evaluated).
	This signature makes it possible for clients to call NoNegativeCounter.first_larger, but never in a way that leads to the NoNegativeCounter.InvariantViolated exception being raised.
	3 points
	13。 In this question, the definition of COUNTER is:
	<pre>1 signature COUNTER = 2 sig 3 type t 4 val newCounter : int -> int 5 val increment : t -> t 6 val first_larger : t * t -> bool 7 end 8</pre>
	This signature allows (some) clients to cause the NoNegativeCounter.InvariantViolated exception to be raised.
	This signature makes it impossible for any client to call NoNegativeCounter.first_larger at all (in a way that causes any part of the body of NoNegativeCounter.first_larger to be evaluated).
	This signature makes it possible for clients to call NoNegativeCounter.first_larger, but never in a way that leads to the

 ${\bf NoNegative Counter.Invariant Violated}\ {\bf exception}\ being\ raised.$

Exam for Part A

测验, 15 个问题

3 points

14。

In this question, the definition of **COUNTER** is:

```
1 signature COUNTER =
2 sig
3    type t
4    val newCounter : int -> t
5    val increment : t -> t
6    val first_larger : t * t -> bool
7 end
```

This signature allows (some) clients to cause the NoNegativeCounter. InvariantViolated exception to be raised.

This signature makes it impossible for any client to call

NoNegativeCounter.first_larger at all (in a way that causes any part

of the body of NoNegativeCounter.first_larger to be evaluated).

0/

This signature makes it possible for clients to call NoNegativeCounter.first_larger, but never in a way that leads to the NoNegativeCounter.InvariantViolated exception being raised.

3 points

15。

In this question, the definition of **COUNTER** is:

```
1 signature COUNTER =
2 sig
3 type t = int
4 val newCounter: int -> t
5 val increment: t -> t
6 end
```

This signature allows (some) clients to cause the NoNegativeCounter.InvariantViolated exception to be raised.

This signature makes it impossible for any client to call

NoNegativeCounter.first_larger at all (in a way that causes any part

of the body of NoNegativeCounter.first_larger to be evaluated).

This signature makes it possible for clients to call

NoNegativeCounter.first_larger, but never in a way that leads to the

NoNegativeCounter.InvariantViolated exception being raised.

我(Feitao YI)了解提交不是我自己完成的作业将永远不会通过此课程或导致我的		
Coursera 帐号被关闭。	了解荣誉准则的更多信息	

Submit Quiz