2AA4 - Assignment 1

Please read this document very carefully. Follow instructions exactly. If you have any questions please post them to MS Teams or ask during office hours.

This assignment is due Feb 18th, by 11:59pm. I will accept the assignment up to 48 hours late without penalty.

I have created an Assignment 1 channel in Teams. If you have questions about the assignment, please post them there. Thank you.
Unless specifically stated, assume you are not allowed to import external libraries.

Purpose

The primary objective of this assignment is to assess your ability to interpret and implement a mathematical specification.

Note:

- I have made the specifications intentionally unintuitive at times. That is, I have given functions and variables generic names, and may have stated some things in way which are less elegant than they could have been stated.
- Myself and the TAs are happy to answer questions regarding notation, but questions such as "should $f3$ return true if there are... " will most likely be answered with: "Please read the specification."

Overview

There are no other files associated with this document. You are responsible for submitting two files:

1. `Mystery1.java`
2. `Mystery2.java`

See below for details on what you are responsible for completing.

Your Tasks

At the end of this document are two separate MIS's (one right now, one more to come in a few days). You are responsible for implementing these two specifications. Mystery1 and Mystery2 are not associated with each other in any way. Think of them as two separate questions. Submit these implementations as `Mystery1.java` and `Mystery2.java` respectively. You may import/use ArrayLists and HashMaps if you wish.

Submitting and Grading

This assignment will be submitted electronically via Avenue. Part of your assignment will be auto graded, part will be done manually. A rough breakdown is given below.

- `Mystery1.java`: 50%
  - f1: 5%

  - f2: 20%
  - f3: 25%

- `Mystery2.java`: 50%
  - add() and getNext() will be tested in conjunction for the 50% this module is worth.

Code which does not compile will be heavily penalized. Code which has incorrect method/module names will be penalized up to 20% Good luck!

Academic Dishonesty Disclaimer

All of the work you submit must be done by you, and your work must not be submitted by someone else. Plagiarism is academic fraud and is taken very seriously. The department uses software that compares programs for evidence of similar code.

Please don't copy. The TAs and I want you to succeed and are here to help. Here are a couple of general guidelines to help you avoid plagiarism:

Never look at another assignment solution, whether it is on paper or on the computer screen. Never show another student your assignment solution. This applies to all drafts of a solution and to incomplete solutions. If you find code on the web that solves part or all of an assignment, do not use or submit any part of it! A large percentage of the academic offenses in involve students who have never met, and who just happened to find the same solution online. If you find a solution, someone else will too.

Mystery1 Module Interface

Uses

None

Syntax

Exported Types

Mystery1 = ?

Exported Access Routine

Routine Name	In	Out	Exceptions
new Mystery1		Mystery1	None
f1	String, String		None
f2	String	$\mathbb{N}$	None
f3	String, String	$\mathbb{B}$	None

Semantic

Local Types

X = tuple(s_1:String, s_2:String)

State Variables

S : X{}

Assumptions

- All inputs are of the proper type.
- The notation $X(s_1, s_2)$ is shorthand for saying (s_1, s_2) is a tuple of type X.

Access Routine Semantics

Mystery1():

- transition: $S = \{\}$
- output: out := this

f1(s_1, s_2):

- transition:
  $s_1 \neq s_2 \Rightarrow S := S \cup \{X(s_1, s_2)\}$
  $s_1 = s_2 \Rightarrow S := S$

f2(s):

- output: out := $+(x : X | x \in S \wedge x.s_2 = s : 1)$

f3(s_1,s_2):

- output: out :=
  $\exists(L$:String[ ]$|| \ L[0] = s_1 \wedge L[|L| - 1] = s_2 : P(L))$

Local Functions

$P(L)$:String[ ]$\rightarrow \mathbb{B}$
$P(L) = \forall(i : \mathbb{N}|0 < i < |L| : X(L[i-1], L[i]) \in S) \wedge X(L[|L|-1], L[0]) \in S$

Mystery2 Module Interface

Uses

None

Syntax

Exported Types

Mystery2 = ?

Exported Access Routine

Routine Name	In	Out	Exceptions
new Mystery2		Mystery2	None
add	$\mathbb{R}, \mathbb{N}$		None
getNext		$\mathbb{R}$	None

Semantic

Local Types

X = tuple($x_1 : \mathbb{R}$, $x_2 : \mathbb{N}$, $x_3 : \mathbb{Z}$)

State Variables

S : X{}

Assumptions

- All inputs are of the proper type.
- The notation $X(x_1, x_2, x_3)$ is shorthand for saying (x_1, x_2, x_3) is a tuple of type X.
- add(value, n) is never called when $n >$MAX_X2.
- getNext is never called when $|S| = 0$.

Access Routine Semantics

Mystery2():

- transition: $S = \{\}$
- output: out := this

add(value, n):

- transition: $S := S \cup \{X(\text{value}, n, z)\}$, where $\forall(x : X \mid x \in S \wedge x.x_2 = n : z < x.x_3)$

getNext():

- output: out := $x'.x_1$, where $x' \in S \wedge$
  $\forall(x : X \mid x \in S : x'.x_2 \leq x.x_2) \wedge$
  $\forall(x : X \mid x \in S \wedge x.x_2 = x'.x_2 : x'.x_3 \leq x.x_3)$

- transition: $S := S - \{x'\}$, where $x' \in S$
  $\forall(x : X \mid x \in S : x'.x_2 \leq x.x_2) \wedge$
  $\forall(x : X \mid x \in S \wedge x.x_2 = x'.x_2 : x'.x_3 \leq x.x_3)$

Local Constants

MAX_X2 = 10