Dipartimento di Elettronica, Informazione e Bioingegneria
Politecnico di Milano

Prof. Elisabetta Di Nitto and Matteo Rossi

20133 Milano (Italia)
Piazza Leonardo da Vinci, 32
Tel. (39) 02-2399.3400
Fax (39) 02-2399.3411

Software Engineering II

September 6st, 2019

Last Name
First Name
Id number (Codice Persona or Matricola)

Note

1. The exam is not valid if you do not fill in the above data.
2. Write your answers on these pages. Extra sheets will be ignored. You may use a pencil.
3. Incomprehensible hand-writing is equivalent to not providing an answer.
4. The use of any electronic apparatus (computer, cell phone, camera, etc.) is strictly forbidden.
5. You cannot keep a copy of the exam when you leave the room.
6. The exam is composed of three exercises. Read carefully all points in the text!
7. **Total available time: 2h**

Scores of each question:

Question 1 (MAX 8) _____

Question 2 (MAX 4) _____

Question 3 (MAX 4) _____

Question 1 Alloy (8 points)

The city of San Francisco has deployed a system supporting the crowd-sourced monitoring of parking violations. The system offers to citizens a mobile app to take pictures of vehicle license plates that are blocking bike lanes in some areas of the city. Before sending the picture to a server, the app annotates it with information about the time when the picture was taken and the geographical position. When the server receives a picture (and related information) it identifies the vehicle, thanks to an image recognition service, and issues a fine to the vehicle owner. The citizen sending the picture receives an acknowledgement together with the fine issued.

1. Given the following signature representing the image recognition service:
```
one sig ImageRecognitionService {
    mapsPicture: Picture -> lone LicensePlate
}
```
Define in Alloy a set of signatures and associated constraints/facts that model citizens, vehicles, vehicle owners, fines, pictures, license plates and the system itself.
2. Define a fact to ensure that only the owners of vehicles for which a violation has been signaled receive a fine.
3. Define an assertion to check that a license plate cannot correspond to two different vehicles.
4. Explain whether, given the model you have defined, the above assertion is valid or not.

Solution

```
/* Answer to point 1*/
sig LicensePlate {}
sig Picture {}
sig Fine {}

sig Citizen {
	pictures: set Picture
}

sig Owner extends Citizen {
	fines: set Fine
}

sig Vehicle {
	l: LicensePlate,
	o: Owner
}

one sig System {
	ir: ImageRecognitionService,
	fines: LicensePlate lone -> Fine
}

/*
the following fact ensures that two vehicles cannot share the same license
plate. This can be considered a domain assumption given the case under analysis
*/
fact noTwoVehiclesShareLicensePlate {
	all disj v1, v2: Vehicle | v1.l & v2.l = none
}

/*
the following fact ensures that exactly the same picture cannot be taken by two
different citizens. Of course, they could take two different pictures of the
same vehicle
```

```
*/
fact noTwoCitizensTakeTheSamePicture {
      all disj c1, c2: Citizen | c1.pictures & c2.pictures = none
}

/* The following fact ensures that no two different vehicle owners can receive
exactly the same fine. Of course, they could receive fines of the same type.
In this model, we do not distinguish between different types of fines as
it is not needed for the problem at hand */
fact noTwoOwnersForTheSameFine {
      all disj o1, o2: Owner | o1.fines & o2.fines = none
}

/* Answer to point 2 */
fact onlyOwnersOfSignaledVehiclesReceiveFine {
      all f: Fine | f in LicensePlate.(System.fines) and ((System.fines).f
       in Picture.((System.ir).mapsPicture))
}

/* Answer to point 3 */
assert coherence {
      all lic: LicensePlate | #(l.lic) <= 1
}

check coherence
pred show[]{}
run show
```

Answer to point 4: In the above model, the assertion is valid thanks to the presence of fact `noTwoVehiclesShareLicensePlate`. When commenting this fact, which ensures that license plates univocally identify a vehicle, the assertion is invalid.

Question 2 JEE (4 points)

Consider the system described in the previous question. Assume that a microservice supporting the recognition of license plate numbers, given a picture of the plate, is already available.

A. Which operation should this microservice offer?
B. List the **entities** that are needed to implement the system, define their attributes, and highlight the relationships between them, indicating the corresponding annotations in JEE (e.g., @OneToMany).

Solution

The microservice will offer the operation mapsPicture that will receive as an input the picture containing the license plate and will provide as an output a license plate number or an error if no license plate has been recognized with a reasonable level of accuracy.

The entities are the following:

Citizen with fields: name (string), surname (string), birth date (date) and location (string or geo coordinates), address (string or geo coordinates), fiscal code (string), ownedVehicles (@OneToMany relationship with Vehicle), issuedPictures (@OneToMany relationship with Picture).

Picture with fields: time (date), position (geo coordinates), picture content (bulk of bits), recognizedLicensePlate (@ManyToOne relationship with Vehicle), author (@ManyToOne relationship with Citizen).

Vehicle with fields: licensePlate (string), owner (@ManyToOne relationship with Citizen)

Fine with fields: date (date), vehicle (@ManyToOne relation to Vehicle), associatedPicture (@OneToOne relationship with Picture).

Question 3 Verification (4 points)

Consider the following C program (lines are numbered for your convenience, please refer to these numbers in your solution), assume that x and y are non-negative:

```
0 int funct(int x, int y)
1  if (y % 2 == 0)
2    return -1;
3  while (x % 2 == 0)
4    x = x / 2;
5  if ((x + y) % 2 != 0)
6    x = x  - 1;
7  return x+y;
```

1. Symbolically execute the program covering the following paths, highlighting the path condition associated with each path:
    1. 0, 1, 3, 4, 3, 5, 6, 7
    2. 0, 1, 3, 5, 7
    3. 0, 1, 3, 4, 3, 4, 3, 5, 7

2. Are all instructions in the code reachable through some path, even one not listed in the text, or not? Motivate your answer.

Solution
Path 0, 1, 3, 4, 3, 5, 6, 7:
0: x = X, y = Y,
1: Y must be odd (Y % 2) != 0
3: X must be even (X % 2) == 0
4: x = X/2
3: X/2 must be odd. The only possibility is that X = 2 so that X/2 = 1
5: X/2 + Y = 1+Y should be odd. This is not possible given that Y is odd
Thus, this path cannot be followed.

Path 0, 1, 3, 5, 7
0: x = X, y = Y,
1: Y must be odd (Y % 2) != 0
3: X must be odd (X % 2) != 0
5: X + Y must be even, which is always true if X and Y are odd (an odd number can be written as the previous even number + 1, so the sum between two odd numbers is equal to the sum of the two previous even number + 2. This is certainly even).
7: the value X+Y is returned
In summary, following the path above requires the following condition to be true: x and y must be assigned to odd numbers.

Path 0, 1, 3, 4, 3, 4, 3, 5, 7
0: x = X, y = Y,
1: Y must be odd (Y % 2) != 0

3: X must be even (X % 2) == 0
4: x = X/2
3: X/2 must be even
4: x = (X/2)/2=X/4
3: X/4 must be odd. This is possible if X=4 so that X/4 = 1
5: X/4+Y = 1+Y should be even, which is always the case
7: The value 1+Y is returned
In summary, following the path above requires the following condition to be true: Y must be odd and X must be 4 times an odd number (for example X = 4).

Notice that instruction number 6 can never be reached as Y will always be odd and, regardless of its initial value, x at line 5 will have assumed an odd value.