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

July 10th, 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 7) _____

Question 2 (MAX 6) _____

Question 3 (MAX 3) _____

Question 1 Alloy (7 points)

Consider a social network that has 2 types of users: (unregistered) visitors, and registered users. Registered users can create posts and comment on other people's posts; visitors, instead, can only retrieve posts from the system. A post is a mix of text and pictures and it is created by a registered user. A post can have comments by registered users, where a comment consists of a text. Comments can be added in reply to other comments. Registered users can be "friends" with other registered users. The "friend" relationship is asymmetric, in the sense that a user *u1* can be "friends" with user *u2*, but the vice versa does not necessarily hold.

Consider the following signature defining the social network:

```
sig SocialNetwork {
  users: set RegisteredUser,
  posts: set Post,
  comments: set Comment
}
```

1. Define the signatures modeling visitors, registered users, posts, and comments, making sure that they capture the relationships that are described in the text above. Add any other signature you may need to model the required ones.
2. Define constraints (for example through facts) that ensure the following properties hold:
    a. All comments are connected to a post.
    b. All comments and posts have been issued by a registered user.
    c. Comments introduced in reply to other comments refer to the same post as the comment replied to.
3. Define a predicate `addPost` that models the insertion in the social network of a new post (which has no associated comments), provided that it is issued by a user that is registered in the social network.
4. Define an assertion to check that the "friend" relationship is not symmetric.

Solution

```
sig Text {}
sig Picture {}

sig Visitor {}

sig RegisteredUser extends Visitor {
  friends : set RegisteredUser
}

sig Post {
  text : Text,
  pics : set Picture,
  creator : RegisteredUser,
  comments : set Comment
}


sig Comment {
  text : Text,
  creator : RegisteredUser,
  reply_to : lone Comment
}
```

```
fact EachCommentIsOnPost {
  all c : Comment | one p : Post | c in p.comments
}
```

The second constraint is actually already enforced by the signature.

The first and second constraints, indeed, could be expressed with respect to a social network. So, an alternative formulation could be the following:

```
fact EachCommentIsOnPosts2 {
  all sn : SocialNetwork, c : Comment |
      c in sn.comments implies (one p : Post | p in sn.posts and c in p.comments)
}

fact CommentsAndPostsIssuedByRegisteredUser {
  all sn : SocialNetwork |
      (all c : Comment | c in sn.comments implies c.creator in sn.users)
      and
      (all p : Post | p in sn.posts implies p.creator in sn.users)
}


fact RepliesOnlyToSamePost {
  all c1, c2 : Comment |
      c2 in c1.reply_to implies
           (one p : Post | c1 in p.comments and c2 in p.comments)
}


pred addPost [ s, s' : SocialNetwork, p : Post ] {
  not p in s.posts
  p.creator in s.users
  p.comments = none
  s'.users = s.users
  s'.comments = s.comments
  s'.posts = s.posts + p
}

assert friendNotSymmetric {
  friends != ~friends
}
```

Question 2 Requirements and Function Points (6 points)

A milk vending machine accepts 3 types of coins: 25¢ (cents), 50¢, and 1$. It accepts coins only in ascending order, for example, after inserting a 50¢ coin, it can only accept 50¢ or 1$ coins and it cannot accept 25¢ coins until the process is restarted by asking for the remaining change.
Every time the amount of money in the machine reaches (or surpasses) 1$, it produces a bottle of milk, subtracts 1$ from the amount of money that is in the machine, and leaves the rest in the machine.

At any time, the user can ask for the money still in the machine (and not used to buy a bottle of milk) to be returned (this can occur even if there is no money remaining in the machine); the effect of this is that the process is restarted, so smaller coins can be introduced again.
The vending machine also accepts fidelity cards. When the user inserts a fidelity card, he/she can buy a bottle of milk for 75¢ instead than 1$. Thus, if the machine receives the fidelity card and the current amount of money is equal to or surpasses 75¢, it produces the bottle of milk, subtracts 75¢ from the amount of money that is in the machine, and leaves the rest in the machine.

Q1: Given the system described above and referring to the Jackson-Zave distinction between the world and the machine, identify:

- At least two world phenomena that are not shared with the machine.
- At least two shared phenomena controlled by the world.
- At least two shared phenomena controlled by the machine.
- At least one machine phenomenon not shared with the world.

Please use the table below to answer this question (the number of rows does not necessarily correspond to the number of phenomena you can identify for this case).

Phenomenon	**Shared nor not**	**Who controls it**	**Short explanation (if the name you assign to the phenomenon is not self-explanatory, or you need to specify some conditions on the phenomenon)**

Q2: Referring to the function points analysis, identify the relevant function types and classify them as ILF, ELF, External Inputs, External Outputs, External Inquiry. Please use the table below to answer this question. You do not need to compute function points.

	Function types
ILF	
ELF	
External inputs	
External inquiry	
External outputs	

Solution

Phenomenon	**Shared nor not**	**Who controls it**	**Short explanation (if the name you assign to the phenomenon is not self-explanatory, or you need to specify some conditions on the phenomenon)**
User wants to buy some milk	N	W	
User inserts a coin in the machine	Y	W	
The machine compares the inserted coin with the last received one	N	M	
The machine rejects the inserted coin	Y	M	
The machine accepts the inserted coin	Y	M	
User inserts a fidelity card	Y	W	
The machine checks and accepts the fidelity card	Y	M	This phenomenon could also be split into two different ones: machine checks fidelity card which is not shared and machine accepts fidelity card which is shared and controlled by the machine.
The machine sees that amount needed to buy a bottle of milk is reached	N	M	
The machine delivers the bottle of milk	Y	M	
The machine updates the current amount of money	Y	M	We assume here that the user sees the update to the amount of money. This is not strictly needed so this phenomenon could be even classified as internal to the machine. In this case the machine would not be too friendly with its users.
The user goes home with the milk	N	W	
The user wants to receive the money back	N	W	
The user asks for the money back	Y	W	
The machine delivers the amount of money to the user	Y	M	

The machine resets the money count	N	M	
The operator sets the current number of bottles in the machine	Y	W	This and the following phenomena are not directly derived from the problem description, but can be inferred by thinking at the initialization and maintenance of the machine
A milk sensor signals the milk in the machine is finishing	Y	W	
The machine decreases the counter of the current number of bottles	N	M	
The machine goes out of service	Y	M	This happens if the milk level is below threshold or if there are no more bottles in the machine

Function points analysis

Function type	**Functions**
ILF	currentAmountOfMoney lastCoin milkPrice fidelityCardDiscount currentAmountOfBottle milkLevelBelowThreshold (this can be a boolean) fidelityCards (user data, cardID) registeredSales (amount paid, numberOfMilkBottles, fidelityCardID)
ELF	None
External inputs	Insert Coin Ask For Money Back Insert Fidelity Card Insert Current Number of Bottles Amount of Milk Below Threshold
External inquiry	Query the current amount in the machine
External outputs	Deliver Milk Bottle Deliver the Money Back Show the Updated Amount

Question 3 Testing (3 points)

Consider the following program fragment in C:

```
0 program(int x, int y, int z)
1  int i;
2  if(x*y<z-y)
3     y = z*z;
4  else
5     y = z-x;
6  while(x<z+y) {
7     x = y*i;
```

```
8     i = i+1;
9  }
10 return x;
```

Identify the def-use pairs for this program fragment and draw the conclusion from this analysis. Please use the table below to answer this question.

Variable name	**<def, use> pairs**
x	
y	
z	
i	

Solution

Variable name	**<def, use> pairs**
x	<0, 2> <0, 5> <0, 6> <0, 10> <7, 6> <7, 10>
y	<0, 2> <3, 6> <3, 7> <5, 6> <5, 7>
z	<0, 2> <0, 3> <0, 5> <0, 6>
i	< _ , 7> <8, 7> <8, 8> < _ , 8>

As it emerges from the analysis, variable *i* is used before being defined.