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

February 1st, 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. **There are 2 variants of Exercise 3, both dealing with Project Management. You are required to answer only one of the two variants; if you answer both of them we will grade the answer to question 3.b. Please note that this last one gives you less points than the other.**
8. **Total available time: 2h**

Scores of each question:

Question 1 (MAX 7) _________

Question 2 (MAX 6) _________

Question 3 (MAX 3) **a** __________**b** _____________

Question 1 Alloy (7 points)

Consider the following Alloy signatures:

```
abstract sig Boolean {}
one sig True extends Boolean {}
one sig False extends Boolean {}

sig TCPData {
  processed: one Boolean
}

sig TCPServer {
  received: set TCPData
}
```

`TCPData` represents information being transferred through a TCP network and `TCPServer` represents servers able to receive such data. When `processed` is true, this means that the corresponding `TCPData` has been processed by a server. Of course, this happens only if the server receives such data.
Assume to observe the final state of the system composed of some data and some servers. In this final state, all data have been received by the corresponding servers (there are no data at the sources nor in transit) and, possibly, processed. Under this assumption, extend the above Alloy model as follows:

a) Model a TCP channel that maps `TCPData` to `TCPServers` through a routing relation. For the sake of simplicity, focus only on the destination of the routing, not on the source and make sure that the channel can route each `TCPData` to at most one `TCPServer`. Keep in mind that you are representing the situation you observe in the final state, so, at the end of the routing activity. This means that the routing relation indicates to which server a channel has routed a certain piece of data.
b) Model as a predicate the following domain assumption (which is enforced thanks to the guarantees offered by the TCP protocol): *data routed toward servers have been received by them*.
c) Model as a predicate the following requirement: *all data received by servers have been processed.*
d) Model as a predicate the following goal: *data routed toward servers should have been processed*.
e) Define the following two assertions:
    a. `GoalFulfillment` that aims at checking whether the goal is fulfilled when the requirement and the assumption are true.
    b. `GoalNeedsAssumptions` to check the following: even if the requirement holds, the goal cannot be satisfied if the assumption does not hold.

Solution
A possible solution for the exercise is the following. Other ones are also possible, of course.

```
sig TCPChannel {
  routed: TCPData -> lone TCPServer
}

pred TCPChannelAssumption {
  all d: TCPData, s: TCPServer |
        d in TCPChannel.routed.s implies d in s.received
}

pred TCPServerRequirement {
  all d: TCPData, s: TCPServer |
        d in s.received implies d.processed = True
}

pred Goal {
```

```
  all d: TCPData |
     d in (TCPChannel.routed).TCPServer implies d.processed = True
}

assert GoalFulfillment {
  TCPServerRequirement and TCPChannelAssumption implies Goal
}

assert GoalNeedsAssumption {
  TCPServerRequirement implies
    (not TCPChannelAssumption implies not Goal))
}
```

Question 2 Testing (6 points)

Consider the following C program (lines are numbered for your convenience, please refer to these numbers in your solution):

```
1  main() {
2    int a, h, f, q;

3    scanf(“%d”, &a);
4    scanf(“%d”, &q);
5    h = q - 2;
6    while (a > 0){
7      if (q == h + 2)
8        f = a;
9      else if (a > f)
10         f = a;
11     scanf(“%d”, &a);
12     h = h+1;
13   }
14   printf(%d”, f);
15 }
```

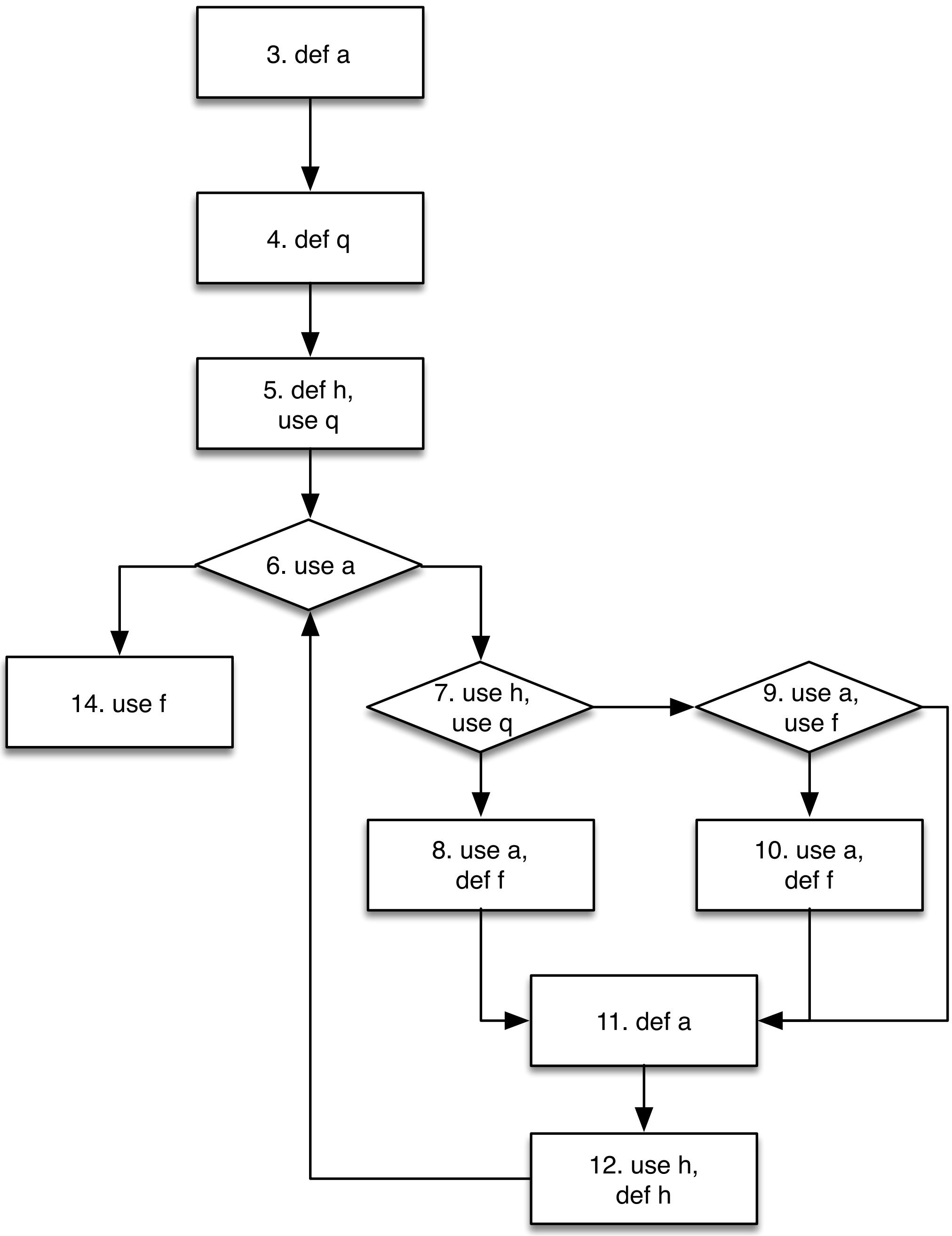
1. Draw the control flow graph of the program for def-use analysis (highlight in the graph where variables are defined and used) and write the def-use pairs in the table below.
Def-use analysis shows that there some potential problems in the program; explain what these problems are.

2. Use symbolic execution to show whether the problems highlighted by def-use analysis can occur or not during execution of the program.

variable	**⟨ def, use ⟩ pairs**

Solution

1.
The control flow graph of the program for def-use analysis is the following:



variable	**⟨ def, use ⟩ pairs**

a	⟨3, 6⟩, ⟨3, 8⟩, ⟨3, 9⟩, ⟨3, 10⟩, ⟨11, 6⟩, ⟨11, 8⟩, ⟨11, 9⟩, ⟨11, 10⟩
h	⟨5, 7⟩, ⟨5, 12⟩, ⟨12, 7⟩, ⟨12, 12⟩
f	⟨??, 9⟩, ⟨??, 14⟩, ⟨8, 9⟩, ⟨8, 14⟩, ⟨10, 9⟩, ⟨10, 14⟩
q	⟨4, 5⟩, ⟨4, 7⟩

Def-analysis highlights 2 potential uses of variable `f` (at lines 9 and 14) before it is initialized.

2.
The paths leading to the potential uses of `f` before its initialization are the following: 3, 4, 5, 6, 14 and 3, 4, 5, 6, 7, 9.
Symbolic execution through path 3, 4, 5, 6, 14 gives the following result:
3: a = A
4: q = Q
5: h =Q-2
6: $A \leq 0$
Hence, the path is indeed feasible with condition $A \leq 0$, so this is a real potential problem that can occur in the program.

Path 3, 4, 5, 6, 7, 9, instead, is not feasible. In fact, symbolic execution gives the following result:
3: a = A
4: q = Q
5: h =Q-2
6: $A > 0$
7: $Q \neq Q-2+2$
which yields the contradictory condition $Q \neq Q$. Hence, this is a false positive that def-use analysis produces.

Question 3, Alternative a: Project management (3 points)

A project has been set up to build an application composed of three software components.
The budget assigned to the project is 10000 euros
At a planned review the project manager (PM) calculates the following parameters according to the Earned Value Analysis:

- Earned Value (EV) is 800 euros.
- Planned Value (PV) is 1500 euros.
- Actual Cost (AC) is 1000 euros.

As PM of the project please answer to the following questions:

1. Inform the stakeholders about the situation of the project in terms of schedule and cost.
2. Estimate the budget at completion (EAC) of the project considering the following two options:
   a. The project will progress spending at the actual rate.
   b. The project will progress spending at the original rate.

Please provide a short motivation to your answers.

Solution

1.
The project is running over budget because the value produced is less than the cost (EV < AC); in addition, it is running behind schedule because the value produced is less than the value planned (EV < PV).

2.
The cost performance index of the running project is CPI = EV/AC = 0.8

a. using the actual rate EAC = BAC/CPI = 10000/0.8 = 12.500 euros
b. continuing at the original rate EAC = AC + (BAC – EV) = 1000 + (10000-800) 10200 euros

Question 3, Alternative b: Project management (1.5 points)

A company offers a bike rental service. Bikes are distributed in the city based on where they have been left by the previous user. The company wants to develop a software system to allow users to rent and ride bikes. Rental fees are computed according to the usage time. In particular, through the system, a user should be able to:

1) Log in.
2) Display the list of available bikes organized in different categories (e.g., mountain bike, city bike, racing bike) and their rental price.
3) Make a reservation for a bike from the list and obtain back a OTP (One-Time Password) to be used to unlock the bike.
4) Start the rental by inserting the OTP (One-Time Password) received at reservation time on the bike numeric keypad.
5) Terminate the rental.

Referring to the Function Points analysis, how would you classify function type number 3 (make reservation)? Which complexity would you assign to it? Provide a short motivation for both answers.

Solution
Make a reservation can be classified as an external input. The OTP may be sent back to the user as a reply to the make reservation request or through a different mechanism, for instance, by sending an SMS or an email message to the user. In the second case, it should be accounted as a different function type offered by the system, in particular, as an external output. Both function types can be considered of simple complexity. In particular, make reservation requires the interaction with two related entities in the database, a reservation and a bike. Sending the OTP requires the interaction with an external email or SMS delivery service. In the case the OTP is returned as a result of make reservation, we could assign to this function type a medium complexity to account for the OTP generation.