Dipartimento di Elettronica e Informazione

Politecnico di Milano

prof. ***Carlo Ghezzi e Elisabetta Di Nitto***

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

Software Engineering II

Part I

2 February 2007

Last name

First name

Identification number (Matricola)

Section (specify the professor you are associated with, either Prof. Ghezzi or Prof. Di Nitto)

Notes

1. Missing identification data invalidate the exam.
2. Return **only** these pages. Extra sheets will be ignored. You may use a pencil.
3. The exam is in 2 parts. For part 1 you will not be allowed to use books or class notes. For part 2 you will be allowed to use books and class notes. The first part must be in 30 min., the second part in 55 min. The final result is the sum of the scores obtained in both parts.
4. Any use of electronic devices (computers, calculators, cell phones, …) is forbidden.
5. You cannot keep a copy of the exam when you leave the room.

Question 1 – (2 points)

Consider the following statement, rephrased from a well-known statement by F. Brooks "Adding manpower to a late project further delays the project”. Can you provide two reasons to justify why and when the statement would be true? Can you also identify some practical situation in which the statement would be false?

The problem is that late manpower adds communication overhead in a team that may already be large. Also, these people might not know the project, nor the application area. It may instead be ok if the task assigned to these people is perfectly well specified, like implementation of a certain well defined functionality.

Question 2 –(3 points)

Publish-subscribe architectures may differ in the way the middleware guarantees an ordering relation among messages. One particular order is with respect to the sender. If the middleware does not guarantee it, is it possible to implement it explicitly? How?

Before sending out the message, a sender may add its identifier and a timestamp, or a sequential number, so that recivers may reconstruct the correct sequence.

Question 3 – (2 points)

Briefly describe what are the main features that Statechart adds to a conventional finite-state machine.

And and or (sub)states.

Question 4 – (2 points)

Define what “man-month” mean as measure of effort (and cost) of a software project.

It is a unit of cost, defined by the average monthly cost of a software engineer. A project may cost, say, 30 man-month. The measure does not say how many months will the project take. However, it does not mean that we can interchangeably allocate 10 persons for 3 month each or 2 persons for 15 months each.

Software Engineering II

2 February 2007

Last name

First name

Identification number (Matricola)

Section (specify the professor you are associated with, either Prof. Ghezzi or Prof. Di Nitto)

Part II

Question 1 (6.5 points)

Provide an Alloy specification of the following problem that deals with classes taught at the university. A class is a set of students and a professor. A class has associated a precedence set that identify the classes that are a prerequisite for taking that class. No student can take a class A if he has not given all the exams for the classes that are in A precedence set.

Elisabetta Di Nitto 25-9-07 18:10

Deleted: Each class is taught twice a week, and has start and an end time. Students can only take classes that do not overlap in time. Each professor must teach at least one and at most three classes, also non-overlapping in time.

```
sig String{}
sig Student{
        name: String,
        takenClasses: set class,
        currentClasses: set class

}
sig Professor{
        name:String,
        taughtClasses: set class
}

sig class{
        name : String,
        teacher: lone Professor,
        students: set Student,
        precedence: set class
}

fact noSelfPrecedence{

        all c: class | c !in c.precedence
}


fact enrolling{
        all c: class| all s : c.students |  c in s.currentClasses

}

fact noRepeating{

        all s: Student | no c: s.currentClasses | c in s.takenClasses
}


fact respectPrecedence{

        all s : Student | all c : s.currentClasses | all cp : c.precedence | cp in s.takenClasses

}


pred show(){}

assert precResp{
        all s: Student| all c: s.currentClasses| all p : c.precedence | p in s.takenClasses

}


check precResp for 20
```

N.B.
This solution is executable using the alloy analyzer and the assertion is evaluated without counter examples for 20 elements or less.
Simpler solutions can be accepted if they conform to the exercise request, even if they are not executable.

Question 2 (6.5 points)

1. Identify a path through the following program such that function calcPower returns -1.
2. Define the path condition corresponding to the path.
3. Show if the path condition is satisfiable.

```
int x, y;

int testForError() {
        if ( y > 0)
                return 1;
        else {
                print("Error in Exponent");
                return -1;
        }
}
int calcPower() {
        if ( y == 1)
                return x;
        else {
                y = y-1;
                return x*calcPower();
        }
}

int power() {

        if ( testForError() == -1)
                return -1;
        else
                return calcPower();
}
main() {
        x = 2;
        y = 2;
        print( power() );
}
```

1) A possible path that makes calcPower return -1 is: y = 3; x = -1

2) path condition: Y odd and > 0; x = -1. the path condition is calculated for calcPower function.

If we consider the whole program we have that for any y < 0 the program returns -1, but the calcPower function is not executed

3) Considering the program with a fixed main given, the path condition is not satisfiable,since y = 2 , so it is >0 but even, and x != -1.

If we don't consider the input then one of the possible paths that satisfy the path condition is the one reported in solution of question 1)

Elisabetta Di Nitto 25-9-07 15:46

Deleted: testForError() {
if (y > 0)
return 1;
else {
print("Error in Exponent");
return -1;
}
}
calcPower() {
if (y == 1)
return x;
else {
y = y-1;
return x*calcPower();
}
}