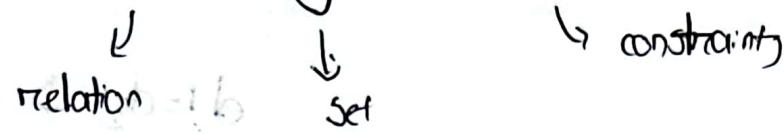


Requirement ACQ validate করতে: we use ~~assertion~~ ACQ

Inside the function: No. I need to

Field, Signature, Fact



→ No person has a parent that is also a sibling

no p: Person | p.siblings in p.parent

sum of children p.parent in p.siblings : No

function name

(return type) ~~return~~ ~~int~~ ~~String~~ return type

fun sisters [p: Person]; ~~women~~ female {

no same ID's ~~return~~ ~~String~~ ~~String~~

parameter(s)

key word

return DB & O = language | biggest p.siblings - Male

}

p → siblings :

all p: Person | p.siblings = { all q: Person | q.parent = p.parent } - F

abstract sig FSObject {
 parent : lone Dir
}

sig Dir extends FSObject {
 entries: set Entry

sig File extends FSObject { }
sig Name { }
sig Entry {
 name: one Name
 object: one FSObject
}

one sig Root { }

a) all f: FSObject | (f in file \Rightarrow f not in Dir) and
(f in Dir \Rightarrow n " , f, b)

b) all d: Dir | all e: d.entries | e.object.parent = d

3) b assert check-1

all d: Dir | all e: d.entries | all d₂: Dir |

d ≠ d₂ ⇒ no (e ∈

d₂.entries)

all d₁, d₂: Dir | no (d₁.entries & d₂.entries)

② assert check-4

all d: Dir | all d₁, d₂: d.entries | e₁.name ≠ e₂.name

d.entries থেকে দুটি entry (e₁ আর e₂)

pick ক্ষেত্রে যাগো same না

③ One fs. FSObject | fs.parent = 0 → ধার্য জন্তু true

জ্ঞান জন্ম

④ assert check-5

q = fsmap.insert(q, 10) } add(q) | const(q, 10)

30/1/24

Testing

Unit testing in black box method

black box testing

black box testing methodology

Test suite

क्षेत्रीय test case

क्षेत्रीय

31/1/24

Mutation testing :

If source code changes, how well would our test suite perform

To test quality of our test data

meaning we have access to our

source code (:

white box testing)

Mutation Testing example:

Types of Mutants

Equivalent Mutants \rightarrow syntactical change only

'Nothing happens' } yellow color হলু, yellow হালে
 }
 and "syntax error",
 সাক্ষী না

Cost of Mutation Testing

mitot - mitosis

Slow test two black fish and expect about same %
Speeding up:

profession → stub test → to develop test of

~~Observation~~

卷之三

Lab 02 solution

abstract Person{

has-A
father : lone Man
mother : lone Woman

} has-A

sig ~~Abse~~ Man extends Person{

wife : lone Woman

sig ~~Abse~~ Woman extends Person{

husband : lone Man

Brotherhood {

, has-A

parent

} has-A brother

, has-A

has-A

no p:Person | p in p.^ (father + mother)

Footprint {

} has-A check Footprint {

Quiz 1 solutionSet B

sig Bank {

number : Int,

name : String

branches : set Branch

transactions : set Transaction

sig Branch {

number : Int,

name : String;

number : Int,

accounts : set Account

abstract sig Account {

acc_number : Int,

balance : Int,

owner : one Accountholder

sig Accountholder {

id : Int,

name : String

}

sig Individual extends Account {}

,, Business , , , {}

sig Transaction {

id : Int,

status : String,

amount : Int

date : Date

from : one Account

to : one Account

Facts: \exists bank \exists acc-number \exists name \exists branch \exists account

all b_1, b_2 : Bank | $(b_1.\text{number},)$ = $b_2.\text{number}$

all $\text{disj } br_1, br_2$: Branch | $br_1.\text{name} \neq br_2.\text{name}$

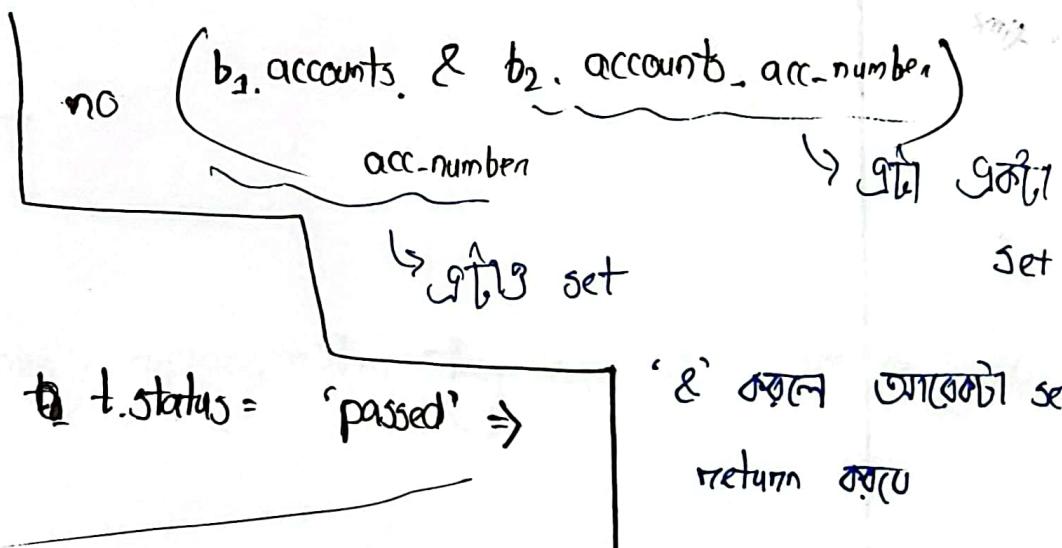
④

all $\text{disj } a_1, a_2$: Account | \dots bank मिलाए गए हैं

all $\text{disj } t_1, t_2$: Transaction | \dots transaction गया है

इति प्राप्ति करा

② all $\text{disj } br_1, br_2$: Branch | \dots branch



③ all t : Transaction | $t.t.status = \text{'passed'}$ \Rightarrow $\&$ इसलिए आवेदनी set return करा

$\hookrightarrow (t.from \text{ in Individual} \Rightarrow t.amount \leq 50,000)$

and

$(t.from \dots \Rightarrow \dots \leq 150,000)$

④ all $t: Transaction$ ($t.from \in Ind \& t.to \in Bus \& t.amount > 15000$
 $\Rightarrow t.status = 'failed'$) | and
and $t.amount / t.balance < 0.1$

⑤ all $t: Transaction$ | $t.status = 'passed' \Rightarrow t.from.balance - t.amount$
and
 $t.amount / t.to.balance < 0.1$

need
concept
'time'

Seven myths :

- ⑤ In the beginning, the cost is high (as requirements are vague), completeness, ambiguity, consistency are achieved through formal method thereby reducing bugs in the long run. Consequently, it reduces cost.
- ⑥ Requirements are easy to understand by the customer whereas understanding the final product (the codebase) is extremely difficult. Formal method is in the middle ground.
- ⑦ List of successful softwares are provided.
- ⑧
- ⑨
- ⑩ F.M doesn't completely replace, rather they support formal systems.
- ⑪
- ⑫ If correctness is of concern, F.M is very recommended.
- ⑬ Supported by many study resources.
- ⑭ IBM did project to hire formal-method people from, but IBM (IBM) 110th of the project.

14/12/24

Ten Commandments:

Go through the writings

Quiz 1 solution

Set A

sig Exam {
 date : Date,
 courses : some Course,
 info : some SeatingArrangement
}

sig SeatingArrangement {
 who : one Student
}

 room : one Room

 labeledSeat : one Seat

sig Sett {
 number : Int
}

sig Room {
 roomNo : Int
}

 building : String
 capacity : Int

 labelledSeats : Set Seat
 some

sig Student {

id : Int,

dept : Department,

batch : Int,

prog : Program,

currSem : Semester,

takenCourses : some Course

}

sig Course {

code : Int,

name : String

sig Semester {

program : Program

offeredCourses : some Course

}

sig
abstract ^ Department {

programs : some Program

}

one sig CSE, EE .. extends Department {}

abstract sig Program {

requiredCourses : some Course

}

one sig BSC in CSE, BSc in SWE extends Program {}

CSE has two programs

Facts:

all d: Department | d = CSE \Rightarrow d.programs = {CSE, SWE}

all s:

i) all e: Exam | ^ e.info. who | s.dept in Dept and

this has already been ensured by the signatures

s.prog in Program

One

at c:

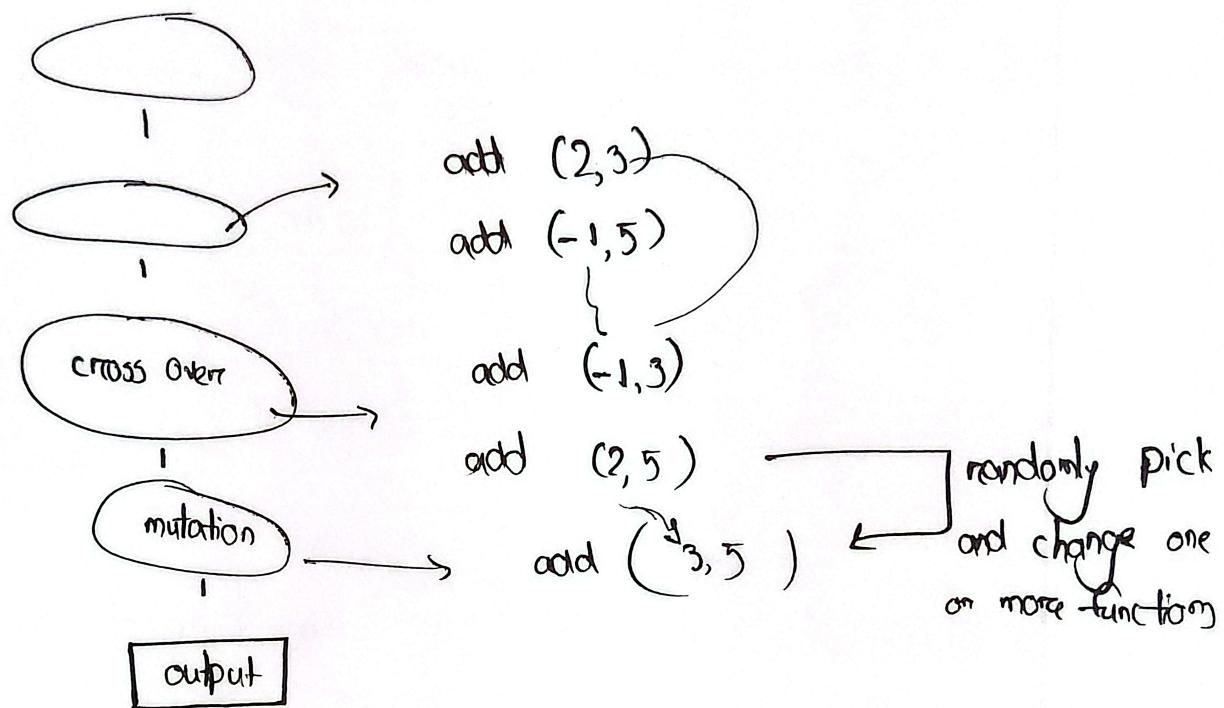
ii) all e: Exam | all s: e.info. who | ^ s.takenCourses |

previously
manually

2012/24

Generating test cases / suites
automatically

Genetic algorithm:



population

↓
selection (based on fitness function)

fitness function depends on our goal-

do we want to do branch

cover or

For exam: Enosite & Smart Select शिखाए ताके पर
↳ उत्तम research questions फॉर्मिंग थिए

Papers:

1) "On the Effectiveness of Manual and Automatic".

Three tools. - each has its own advantages

2) Selectively combining . . .

↳ Step tool go test selection better

3rd contribution

→ Enquiry go way (figure 2)

卷之三

ଏହି ପ୍ରୟୁକ୍ତିଗୁଡ଼ ମାଧ୍ୟମରେ ଆଜି
test case କିମ୍ବା 'cluster' ଏବଂ

Initialize \rightarrow Initial Test case \rightarrow fitness function \rightarrow Genetic
algo (known) Algo

↳ 8 types of labelling / clustering : প্রাণীর একই similar label

проверка
~~табл~~

ଜ୍ଞାନ (known)

Algo

↳ Selected গুলি যদি goal achieve করতে না পারে, তাইনি

unselected „ দিয়ে „ „, ক্ষমানা try করে,

CSE department, IUT, Subsidiary organ of OIC
SWE: 4805: Software Verificaiton and Validation

QUIZ-1

DURATION: 30 MINUTES

Student ID:

SUMMER SEMESTER, 2022-2023

FULL MARKS: 15

SET: A

-
1. IUT maintains a system to maintain the examination routine. However, it is not properly verified and validated. The specification of the system is given below:

An **exam** comprises a collection of details, including the courses that will be conducted, the date on which it will be held, and information pertaining to seating arrangements. The **seating arrangement** contains three pieces of information: the student taking the exam, the room, and the assigned seat where the student will be seated for the exam. Every seat has a unique number. **Rooms** are identified by a unique room number, belong to an academic building, have a capacity, and have a set of labelled seats. Each room is configured in a manner where ten students can sit in a single row. A **student** is identified by a unique ID, belonging to a specific department, program, batch, current semester, and has taken some courses. A **course** is identified by a code and name, while a **semester** is associated with a program which has a list of offered courses. The **department** can be CSE, EEE, CEE, and MPE. Every department has some **programs**, where every program has some required courses. For example, CSE department has two programs, B.Sc. in CSE and B.Sc. in SWE.

- a) Analyze the specification and write the signatures with their associated fields and facts. 5

(CO2)

(PO2)

- b) Write the following constraints in Alloy as fact, considering the signatures and fields that you have designed in Question 1.a). 5

(CO2)

(PO2)

- i. All students must belong to a valid department and program for an exam.
- ii. In the exam hall, students of every program can sit only for the courses that are assigned in their current semester.
- iii. All students in an exam must fit in the room's capacity.
- iv. No two programs of any department have the identical offered courses.
- v. In any given room, it is prohibited to seat two students of the same program of any department consecutively. ?

- c) Write the following statements as assertions to verify and validate in Alloy. Justify whether a counterexample is found or not for each one. Assume all the facts mentioned in Question 1.a) are incorporated. (CO)
(PO)

- i. No student is assigned to two exams at the same day
- ii. It is allowed to have 70% similarities of the offered courses of any two programs of the same department, but 30% for the different departments.
- iii. For an assigned seat of a program's student, there is no any other students around him/her. ?
- iv. Each room can only be occupied by students from a single batch. ?
- v. For any program, there must not any duplication of courses of different semesters.