Ian Hanagan

JC Helm

Model-Driven Software Development

7 April 2020

Alloy Project - Part 1

Model Specification:

1. Abstractions:
    a. The abstractions we have in our Alloy model include *Time*, *Class*, *Student*, and *Professor*. These are the necessary abstractions to create the desired relation that were specified in the project proposal, and they were therefore created as *sig*s.
    b. The *Time sig* was created as an *abstract sig* because all of its possible atoms exists as extending atoms. This is done so that the separate *Time* sigs are comparable.

2. Relations:

    The relationships we defined between signatures can be seen below:

    sig Class { happens: some Time, taughtBy: one Professor, takenBy: Student }

    This relationship simply specifies that a *Class* happens at some *Time,* taught by a *Professor*, and taken by some *Students*.

    sig Professor { teaches: some Class }

    This relationship states that a *Professor* teaches a *Class*.

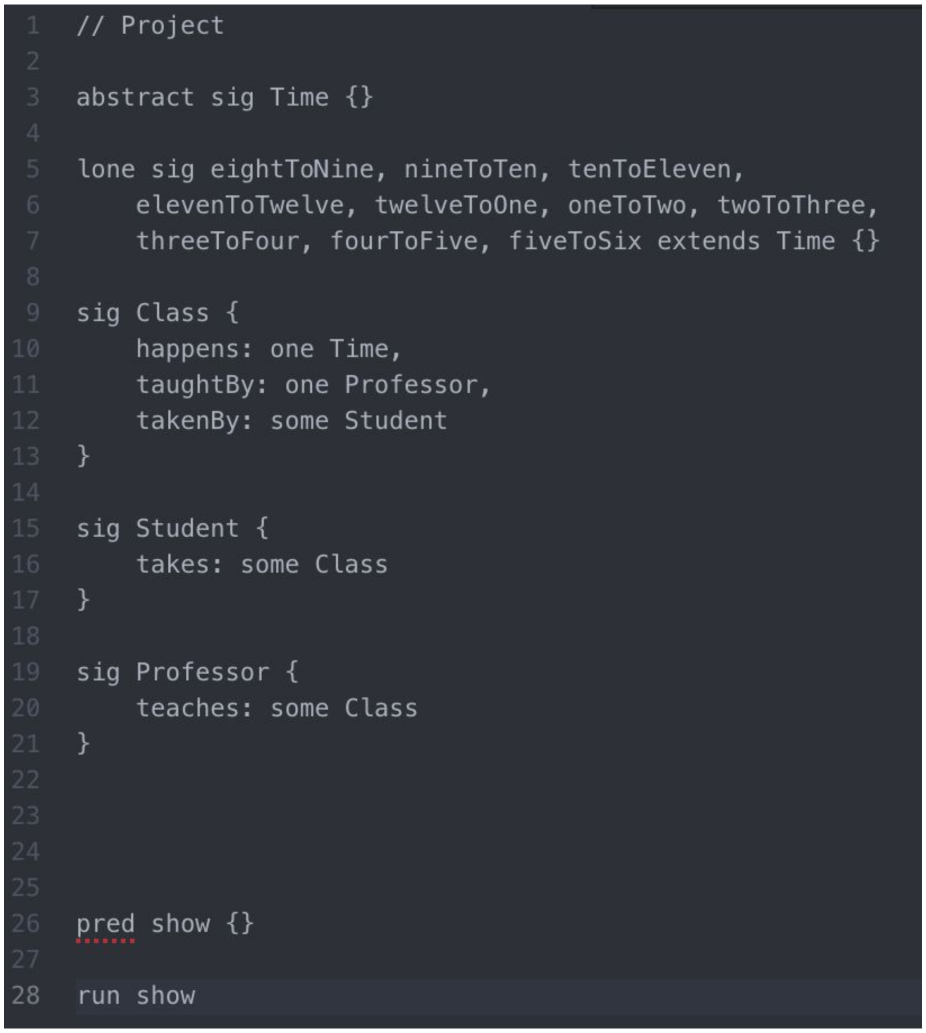
    sig Student { takes: some Class }

    This relationship states that a *Student* takes a *Class.*

3. Multiplicities:

The multiplicities of each relationship are as follows: a *Class* can be taught at one or more times, and taken by one or more *Students*; however, the *Class* is taught by exactly one *Professor*. A *Professor* teaches at least one *Class*, and finally a *Student* takes at least one *Class*.

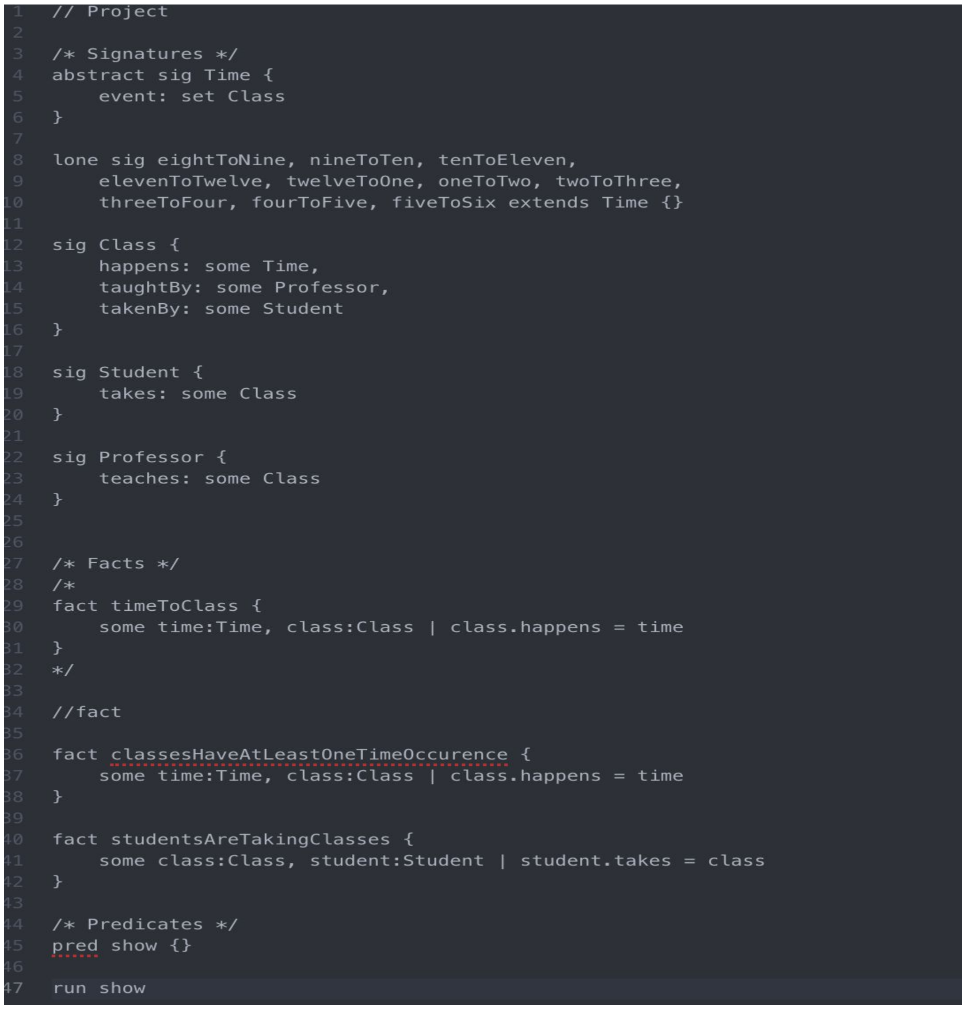
4. Incremental Modeling:
    a. The following three code snippets represent our first version, our second version, our third version, and our git diff logs between each of them.



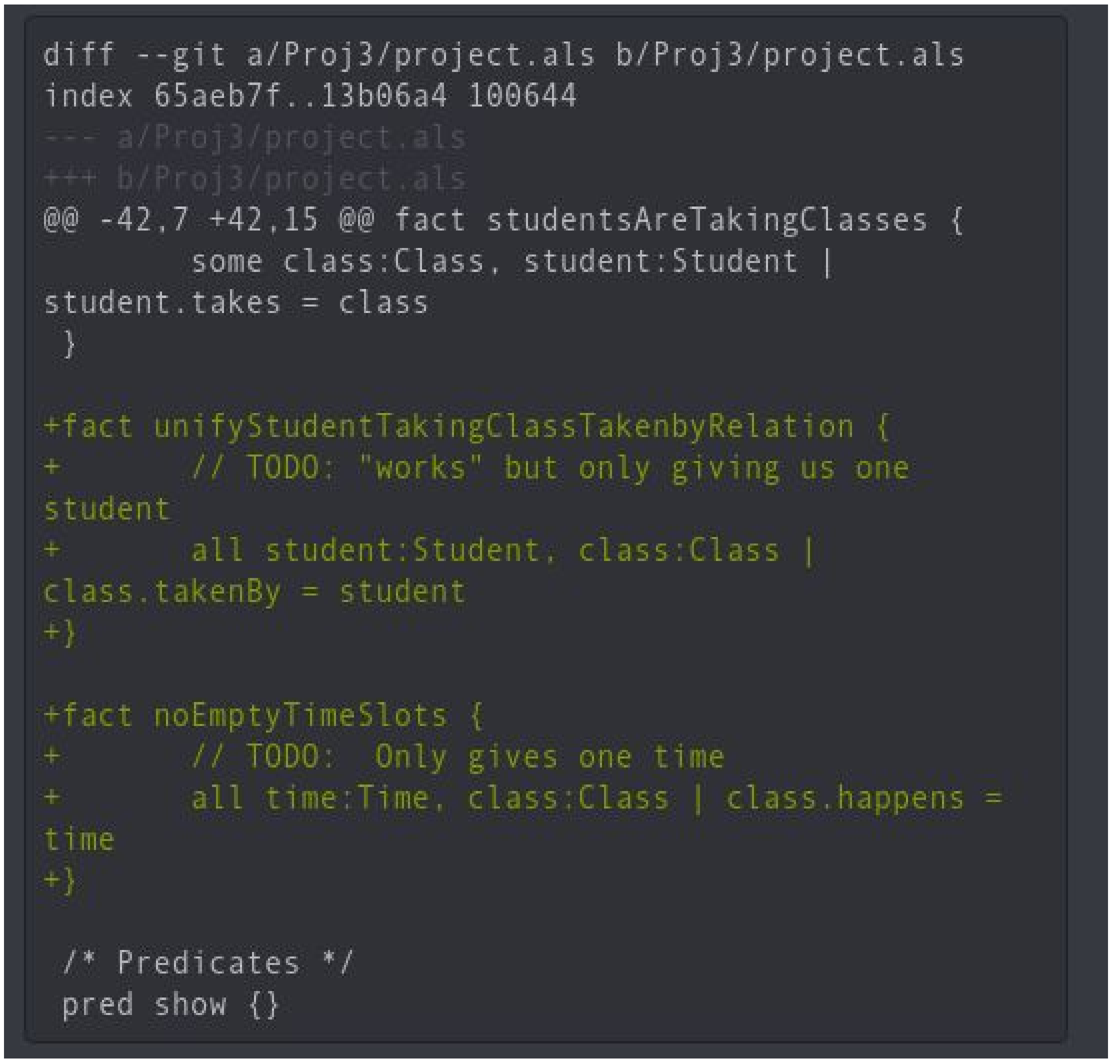
```
// Project

abstract sig Time {}

lone sig eightToNine, nineToTen, tenToEleven,
    elevenToTwelve, twelveToOne, oneToTwo, twoToThree,
    threeToFour, fourToFive, fiveToSix extends Time {}

sig Class {
    happens: one Time,
    taughtBy: one Professor,
    takenBy: some Student
}

sig Student {
    takes: some Class
}

sig Professor {
    teaches: some Class
}




pred show {}

run show
```

```
diff --git a/8ca2d5a b/13b06a4
index 8ca2d5a..13b06a4 100644
--- a/8ca2d5a
+++ b/13b06a4
@@ -1,14 +1,18 @@
-// Project
+// Project

-abstract sig Time {}

-lone sig eightToNine, nineToTen, tenToEleven,
+/* Signatures */
+abstract sig Time {
+       event: set Class
+}
+
+lone sig eightToNine, nineToTen, tenToEleven,
        elevenToTwelve, twelveToOne, oneToTwo,
twoToThree,
        threeToFour, fourToFive, fiveToSix extends
Time {}

 sig Class {
-       happens: one Time,
-       taughtBy: one Professor,
+       happens: some Time,
+       taughtBy: some Professor,
        takenBy: some Student
 }

@@ -21,8 +25,36 @@ sig Professor {
 }


+/* Facts */
+/*
+fact timeToClass {
+       some time:Time, class:Class | class.happens =
time
+}
+*/
+
+//fact

+fact classesHaveAtLeastOneTimeOccurence {
+       some time:Time, class:Class | class.happens =
time
+}
+
```

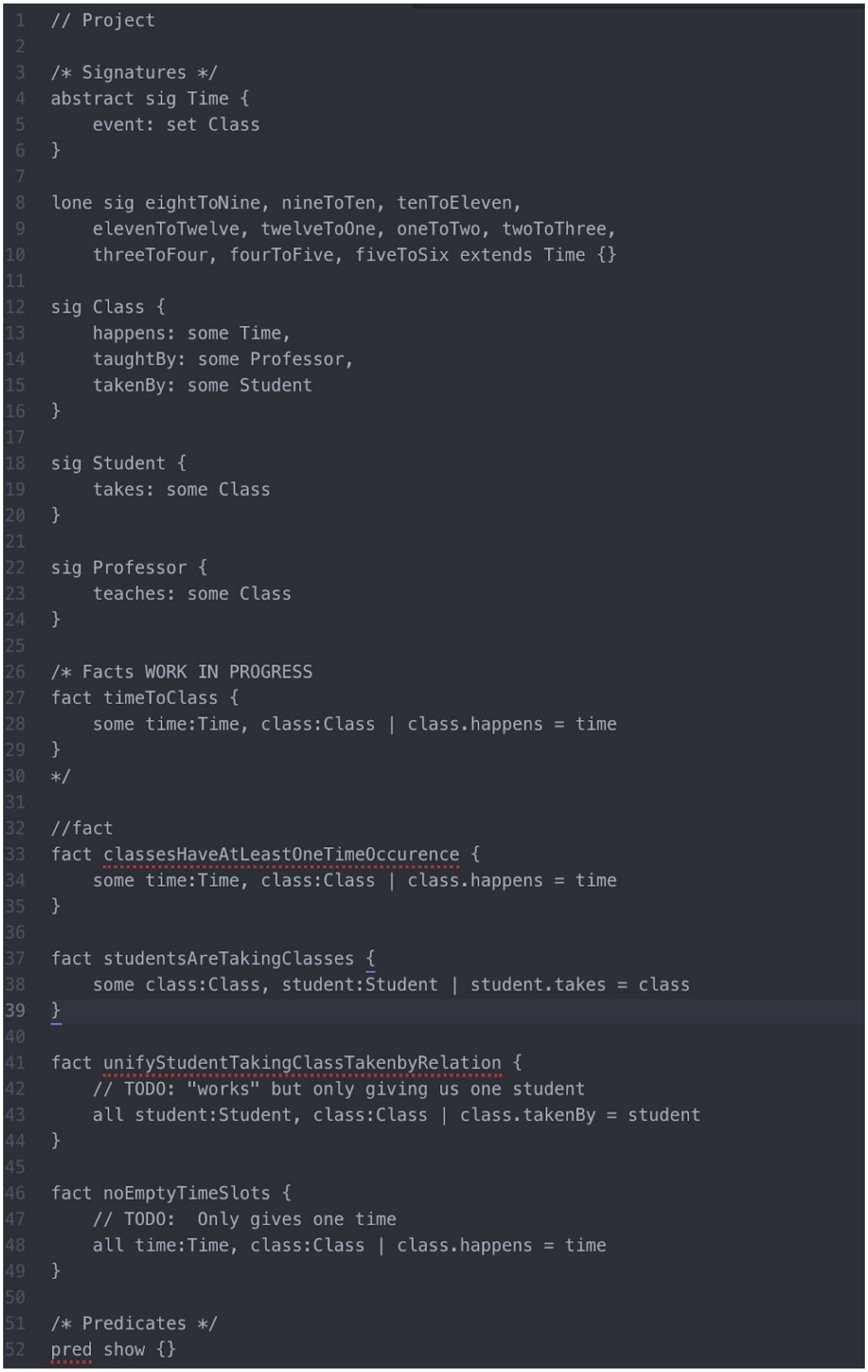
Snapshot 2:



```
// Project

/* Signatures */
abstract sig Time {
    event: set Class
}

lone sig eightToNine, nineToTen, tenToEleven,
    elevenToTwelve, twelveToOne, oneToTwo, twoToThree,
    threeToFour, fourToFive, fiveToSix extends Time {}

sig Class {
    happens: some Time,
    taughtBy: some Professor,
    takenBy: some Student
}

sig Student {
    takes: some Class
}

sig Professor {
    teaches: some Class
}


/* Facts */
/*
fact timeToClass {
    some time:Time, class:Class | class.happens = time
}
*/

//fact

fact classesHaveAtLeastOneTimeOccurence {
    some time:Time, class:Class | class.happens = time
}

fact studentsAreTakingClasses {
    some class:Class, student:Student | student.takes = class
}

/* Predicates */
pred show {}

run show
```



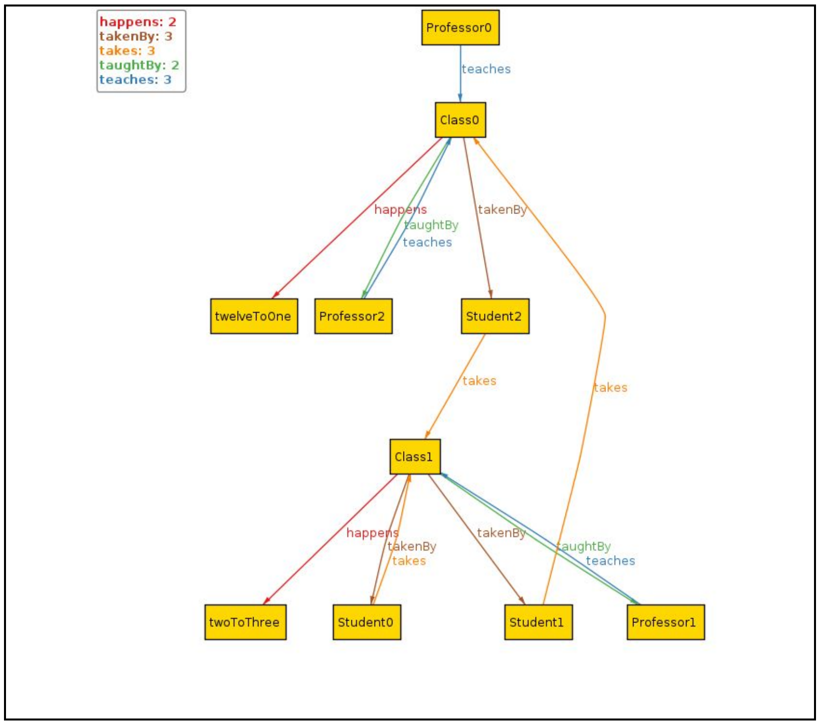
```
diff --git a/Proj3/project.als b/Proj3/project.als
index 65aeb7f..13b06a4 100644
--- a/Proj3/project.als
+++ b/Proj3/project.als
@@ -42,7 +42,15 @@ fact studentsAreTakingClasses {
        some class:Class, student:Student |
student.takes = class
 }

+fact unifyStudentTakingClassTakenbyRelation {
+       // TODO: "works" but only giving us one
student
+       all student:Student, class:Class |
class.takenBy = student
+}

+fact noEmptyTimeSlots {
+       // TODO:  Only gives one time
+       all time:Time, class:Class | class.happens =
time
+}

 /* Predicates */
 pred show {}
```



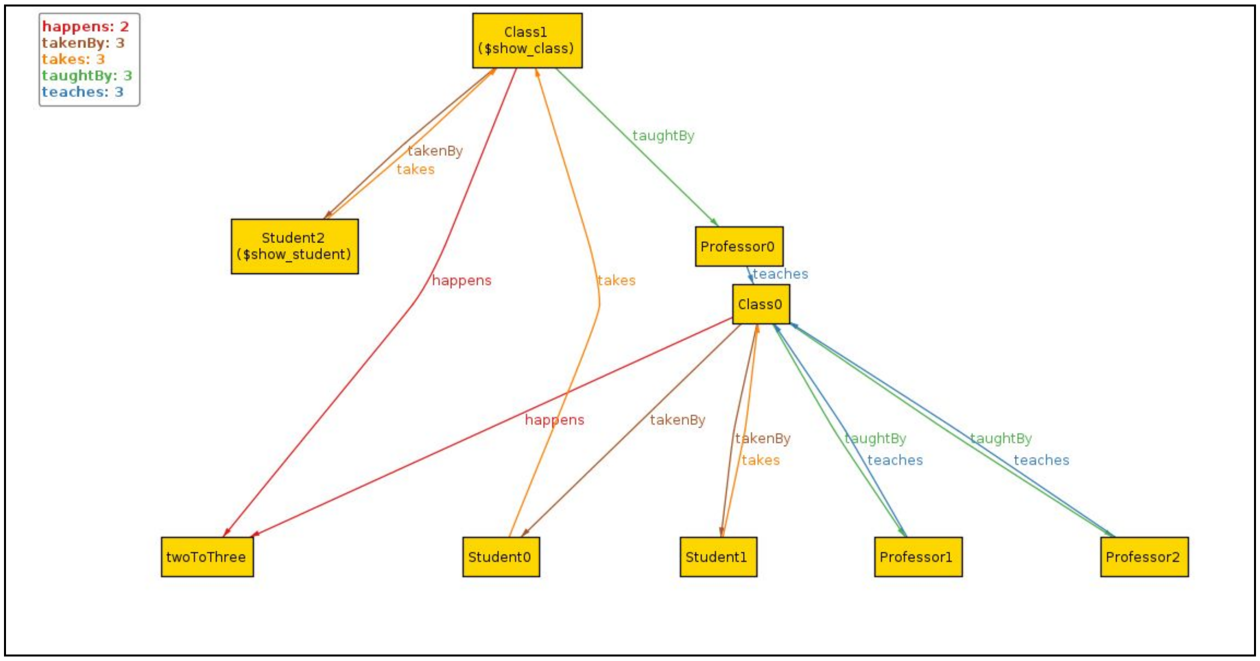
```
// Project

/* Signatures */
abstract sig Time {
    event: set Class
}

lone sig eightToNine, nineToTen, tenToEleven,
    elevenToTwelve, twelveToOne, oneToTwo, twoToThree,
    threeToFour, fourToFive, fiveToSix extends Time {}

sig Class {
    happens: some Time,
    taughtBy: some Professor,
    takenBy: some Student
}

sig Student {
    takes: some Class
}

sig Professor {
    teaches: some Class
}

/* Facts WORK IN PROGRESS
fact timeToClass {
    some time:Time, class:Class | class.happens = time
}
*/

//fact
fact classesHaveAtLeastOneTimeOccurence {
    some time:Time, class:Class | class.happens = time
}

fact studentsAreTakingClasses {
    some class:Class, student:Student | student.takes = class
}

fact unifyStudentTakingClassTakenbyRelation {
    // TODO: "works" but only giving us one student
    all student:Student, class:Class | class.takenBy = student
}

fact noEmptyTimeSlots {
    // TODO:  Only gives one time
    all time:Time, class:Class | class.happens = time
}

/* Predicates */
pred show {}
```

b. Below is an instance from each of the three versions listed above.

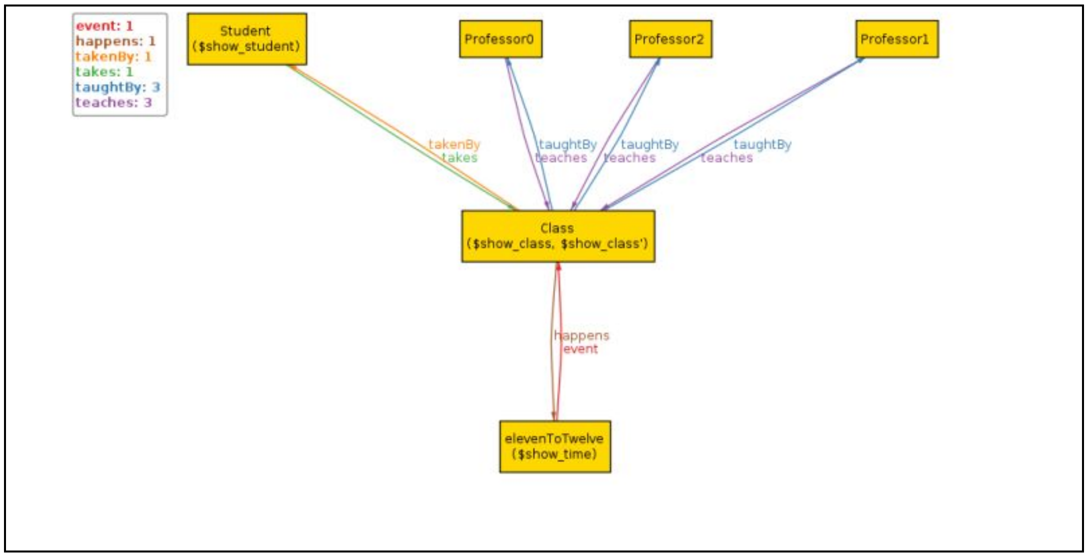
Snapshot 1



Snapshot 2



Snapshot 3



5. Facts:

   a. ClassesHaveAtLeastOneTimeOccurence: a *Class* must be taught at at least one *Time*.

      StudentsAreTakingClasses: *Students* must take at least one *Class*.

      NoEmptyTimeSlots: All *Time* slots are taken up by at least one *Class*.

      UnifyStudentTakingClassTakenByRelation: make sure that every *Student* that is taking a given *Class* is included in that *Class*'s takenBy relation.