CSE 341 Programming Languages

Homework 4 Report

Part 1:

We have a flight list. We have some paths and we need to write some facts for these flights. But these paths should be bidirectional. We write paths and a predicate for this bidirectional. It returns a true or false result. For example we write a flight from Istanbul to Izmir this is fact. We write a predicate and we have a flight from Izmir to Istanbul.

Part 2:

In this part , we calculate path lenght. For example, we want to find distance from Istanbul to Isparta. We find from Istanbul to Izmir and from Izmir to Isparta. Then we add these results.

Part 3:

We have 2 tables. For first table, we write two different facts. One is class hours and lessons, two is lesson and the class room. Second table we have students and their lessons.

Schedule: The student receives class and time information. It shows us where and at what time the student arrives.

Usage: getting class and time information. It shows us which classes are full at what time or which classes are full at the time entered. Usage (X, 16) shows us which class is full at 16 o'clock.

Conflict: shows the class or courses with time conflicts. He takes 2 courses and compares their classes and times. For example, if we give the same lesson to both inputs, it returns true because these courses have same time and same classes.

Meet: Take two students as input. It checks whether two students have attended the same classes. We take students' course numbers and check if they are in the same place at the same time.

Part 4:

Element: Take a set and an element. Check the elemet is in this set. I use member function for this predicate.

Union: We receive 3 inputs. If all 3 are empty, we return the result empty. If the first and second set are empty, we still return the empty result. If the first element of the first set exists in the second set, we add the first set to the second set without the first element. But if it is not in the second set, then we add the first set to the second set completely.

Intersect: we take the first element of the first set and add it to our list if it exists in the second set. But if we don't find it on the second list, we continue to search again after that element.

Equivalent: I'm looking at whether two sets are equal. I checked it if the input sets were equal. I don't understand what kind of equality it should be. If two sets as input (1,2,3) and (2,1,3), I sort them and if they are equal, I said they are the same set.

Ahmet Yuşa Telli

151044092