Constraint Programming

Scheduler for Advising Students

Done By:

Omar Mohamed Saleh 28-6741

Omar Mohamed Ali 28-1120

List of Main Predicates:

scheduleCourse(CourseList, L, Schedule, Groups, LectureGroup):

This predicate takes as input CourseList, which is a list of courses to be taken by the student and scheduled by this predicate. L is the list of slots that is scheduled by the predicate. Groups and LectureGroups are both lists of groups that the student take courses with. Schedule is the student's schedule generated by the predicate and it is a list of slots. These slots hold information for the number of the slot, the ID of the course and whether it is a lecture, tutorial or a lab. The predicate makes valid configurations of each course so that its lectures, tutorials and labs are scheduled with valid configurations.

scheduleCourses(CollectiveList, BestConfig):

This predicate takes as input BestConfig, which is the best configuration of the courses that should be taken by the student. This list is originally generated by the GenerateScheduleConfiguration predicate. CollectiveList is the output of the predicate and is a list of pairs. The first element of the pair is a schedule in a readable form. The second element is the same schedule that is used in further processing.

generateScheduleConfiguration(L2, Probation):

Probation is a boolean that indicates whether the student is under probation or not. L2 is the output of the predicate and the input of scheduleCourses predicate which is BestConfig. This list is the best configuration of schedule tat can be taken by the advising student. Obligatory courses are taken into consideration when generating this list.

generateAllowedCourseList(L):

This Predicate has one output. L is a list of course IDs of all the courses that this student is eligible to take taking into consideration his past courses and prerequisites of the courses.

There are other supporting predicates used such as:

countUnique:
Determines the number of unique tutorial groups in a student's schedule.

obligatoryCourses/optionalCourses:
Generates the list of obligatory/optional courses that the student should take.

filterCourses:
Removes the courses that the student is ineligible to take.

countDaysOff:
Determines if a given day is off.

server/handle_api:
responsible for starting and running the prolog server and replying to external requests made to the prolog server, running the scheduling predicates and replying with the highest rated schedule to the node server to be displayed to the user.

Connection with Node:

Node is used as a backend for the web application. The csv file is uploaded through node and parsed by javascript in order to generate the prolog facts and written in a prolog file. A prolog server is started to receive the parsed schedule. When the schedule is generated it is sent to Node. Node then applies processing on the schedule in order to be passed as a parameter to pug so that the schedule is built on the webpage. Pug is a language used in order to write dynamic html content so that it can be presented on a webpage.