Logic

Project 2018-2019

Write a program in Java that solves the class scheduling problem. You will be given a list of classes and a list of teachers, where each class is assigned to a teacher. Furthermore, you are also given a list of rooms and timeslots. The goal is to assign each class to a room and a timeslot, while guaranteeing that no teacher has to teach two classes at the same time and that two classes do not simultaneously take place in the same room.

The program takes two arguments, the name of an input file, containing the input data, and the name of an output file that will contain the solution.

The following file formats will be used. The first line of the input file will contain four numbers: the number of classes C, the number of teachers T, the number of rooms R and the number of timeslots S. The second line will contain C integers $(T_1, \ldots, T_C)$, between 1 and T, indicating that T_i teaches course i.
The output file will contain C rows, each containing four columns with the course, the teacher, the room and the timeslot.

You must use the Sat4J SAT-Solver (`http://www.sat4j.org`) which can easily be used as a Java library (`http://download.forge.ow2.org/sat4j/getting-started-sat4j-0.3.pdf`). The goal of the project is to encode the class scheduling problem as a logical formula that is then passed on to the SAT-Solver, which determines whether the formula is satisfiable or not. More information about the DIMACS format (input format of the SAT-Solver) can be found here: `http://www.domagoj-babic.com/uploads/ResearchProjects/Spear/dimacs-cnf.pdf`.

The main part of the project is the modelization of the problem as a logical formula. This modelization should be discussed in depth in the project report. (E.g.: How did you translate your problem into a formula? Why did you make certain choices? Did the solver work more efficiently with a certain modelization?)

You are welcome to add more constraints to the problem in order to make it more realistic, as long as they are well documented in the project report. For example, it could be that there is more than one possible scheduling, you could therefore define a way of choosing a best one (e.g. uses the least number of rooms or timeslots). Or you could add more parameters to the problem, such as adding a capacity to each room and indicating how many students are enrolled in each course, and the scheduling should additionnally guarantee that if course C_i takes place in room R_j, then the capacity of R_j is bigger than the number of students enrolled in C_i.

Example: A file `data.in` that contains

```
3 2 2 4
1 2 1
```

should output a file `data.out` containing for example

```
1 1 1 1
2 2 2 1
3 1 1 2
```

The file `data.in` indicates that there are three courses (C_1, C_2, C_3), two teachers (T_1, T_2), two rooms (R_1, R_2) and four timeslots (S_1, S_2, S_3, S_4). Furthermore, courses C_1 and C_3 are taught by T_1 and course C_2 is taught by T_2.

A possible solution to the class scheduling problem is contained in `data.out` indicating that course C_1, taught by T_1, takes place in room R_1 during timeslot S_1, course C_2, taught by T_2, takes place in room R_2 during timeslot S_1 and course C_3, taught by T_1, takes place in room R_1 during timeslot T_2.

Remarks:

- You will work individually.
- Your code should compile and work on Linux; ms800 computers will be used as reference.
- Pack your work in a compressed archive named **logic_name.tar.gz**.
- Your archive should contain your source code and your report in pdf format.
- Mail your work to **Pascal.Gribomont@uliege.be** by 18th December 2018, 23:59.
- The project must be provided as a maven project. It should compile with the command `mvn package`. SAT4J is available for maven here:
  `https://mvnrepository.com/artifact/org.ow2.sat4j/org.ow2.sat4j.core/2.3.5`
- Your project must run from maven; you can do that with the `exec` plugin
  `http://www.mojohaus.org/exec-maven-plugin/index.html`.
  It should then run with
  `mvn exec:java -Dexec.mainClass="Main" -Dexec.args="data.in data.out"`

Project submission: before 18th December 2018, 23:59