

BEE

We arranged our solution so the BEE folder is located in the same folder as the solution. The way you introduced in the class on how to invoke and integrate with BEE did not work, we managed to make it work using the use_module predicate accessing the pl files:

```
:- use_module('./bee/bApplications/auxs/auxRunExpr', <imported predicates>).
```

```
:- use_module('./bee/bApplications/auxs/auxMatrix', <imported predicates>).
```

Please note the correct path to the relevant files is set when testing our solution.

Scheduling

In our solution we create a list of n variables where n is the number of the exams. Each variable X_i 's value is a number between 1 and n which represents the day on which the i th exam should take place, similarly to the solution's structure.

First we find look the the conflict argument as graph where each exam is a vertex and each conflict $c(i, j)$ is an edge in that graph between the vertex of exam i and the vertex of j in the graph.

We find the greatest clique in that graph. This clique is a list of exams that each and every one of them should take place in a separate day. For each exam in the clique with the size of K we set a different day between 1 and K .

For the rest $n-K$ exams we create BEE's int variables representing the day of the exam. We apply BEE constraints that force the conflict. Another set of BEE constraints bind the variables with M - the number used for comparing solutions and holds the amount of days used for distributing the exams.

Running the solution

For some of the problems the default allocated memory was not enough. With the consult of Prof. codish we run the prolog application with the following terminal command, allocating 10GB RAM, avoiding a failure due to "out of memory" error:

```
"prolog -G10g -L10g -T10g"
```

For our comfort (and yours) we implemented a bash script and a prolog wrapper that runs our implementation with the instances provided, printing statistics, and applying a reasonable timeout of 100 minutes. We added to our submission the relevant files and an output of our run that was applied remotely on a lab computer via ssh:

- testInstances.pl - a prolog wrapper, allowing us to run a specific instance
- testRun.sh - the bash script
- results.txt - the output of the run
- Instances folder containing the provided instances of the problem

We, Yarden Chen (305509069) and Gal Tfilin (303097109), assert that the work we submitted is entirely our own. We have not received any part from any other student in the class (or other source), nor did we give parts of it for use to others. We have clearly marked in the comments of our program any code taken from an external source. We assert that the division of work between us is: Yarden 50% and Gal 50%.