FIT5216: Modelling Discrete Optimization Problems
Inclass Task 8: RetailRoster

1 Problem Statement

In a retail roster for a set of people `PERSON` we need to assign people to shifts { `opening`, `morning`, `lunch`, `afternoon`, `closing` } for each of seven week days `MON, TUE, WED, THU, FRI, SAT, SUN`. Each person has preferences for each day and shift given by data in the form

```
enum SHIFT = { opening, morning, lunch, afternoon, closing };
enum DAY = { MON, TUE, WED, THU, FRI, SAT, SUN };
enum PERSON;
array[PERSON,DAY,SHIFT] of int: pref;
```

The following constraints apply:

- Each person can take at most two shifts a day
- A person who is `closing` cannot be on `afternoon`.
- A person who is on `morning` cant be on `opening`.
- A person who is on `closing` cant be on `opening`.
- There needs to be at exactly one person on `opening` and `closing`, at least 2 people on `morning` and `lunch` and at least 3 people on `afternoon`
- There needs to be at least 8 people in total assigned to `opening`, `morning` and `lunch`.
- There needs to be at least 8 people in total assigned to `lunch`, `afternoon` and `closing`.
- No person can be assigned a shift on a day when their preference for that shift is 0.

The aim is to maximize the sum of preferences of the assigned shifts in the roster.

Build a MiniZinc model `retailroster.mzn` to find a solution. Use the variable declarations and output:

```
array[PERSON,DAY] of var set of SHIFT: r;
output ["r = array2d(PERSON,DAY,\(r));\n"];
```

Note that you may want to use a different representation for defining the constraints. You may want to just create these variables when a solution is found declaring them as

```
array[PERSON,DAY] of set of SHIFT: r :: output_only = ...
```

where the missing expression `...` calculates the value of this array from your actual representation of the decisions, once they are all fixed!

Extend your MiniZinc model to maximize the sum of preferences of the selected shifts. I.e. if person P1 is assigned shifts `morning` and `lunch` on the first day and their preferences for these shifts are 3 and 2, then we add 5 to the total objective. The total objective is the sum of preferences of selected shifts for all people and all days.

2 Instructions

Edit the provided `mzn` model files to solve the problems described above. Your implementations can be tested locally by using the *Run* icon in the MiniZinc IDE or by using,

```
minizinc ./modelname.mzn ./datafile.dzn
```

at the command line.