APL - Assignment n°5

By Arthur ADAM and Robin LOTODE
Please note that every file mentioned in this PDF is provided in the Canvas Submission.

Part 1 - Tweedledum & Tweedledee

This problem is called "Tumbledum & Tumbledee". The rules of that challenge are the following:

- We have 2 unknown person. Each gives us an affirmation. Unfortunately, one of those information is false.
- We must find who is giving each affirmation knowing that:
  - Tweedledum lies on Monday, Tuesday and Wednesday.
  - Tweedledee lies on Thursday, Friday and Saturday.
  - Both tells the truth on Sunday.

We tried to understand the logic between the possibility tree that each ressource was giving us. Unfortunately, we didn't manage to find any logic tree branching. We also didn't understand how to manage the affirmation in our code.

We just submitted the file setting up the possible initial state that problem can have.

Part 2 - Traveling Salesman

This problem is the following:

- A salesman should travel to 6 different cities.
- Our goal is to find the shortest path to go to travel to the 6 cities.

Traveling Salesman Solving

In order to solve that program, we had to create the `salesman.smt2` (file submitted with that PDF) as an input file.

That implementation is an implementation of the Kruskal's Algorithm (seen during Math3 lectures).

We also chose to make sure that every city is visited once using a distinct statement.

Firstly, we setted up our initial state. We used functions to set our connections between cities and the path cost between each city.

Then, we made a assertion with a forall statement in order to make our salesman travels across cities.

Finally, in order to make our path the shortest, we chose to use optimization option of Z3. We choose to use the `minimize` function of z3 to make sure that our path would be the shortest.

Traveling Salesman Output

The following input file provides the following output:

```
WARNING: optimization with quantified constraints is not supported
sat
(((pathCost maxNumberCity) 16)
 ((seenCities 1) CIT1)
 ((seenCities 2) CIT2)
 ((seenCities 3) CIT3)
 ((seenCities 4) CIT6)
 ((seenCities 5) CIT5)
 ((seenCities 6) CIT4))
```