Logic and Computability SS22, Practical Bonus Assignment 4

Due: TBD

[**5 Points**] *Seating Arrangement Problem.* For our final programming exercise we take a look at the seating arrangement problem. The problem that you are facing is the following: There are only a few hours left before your weeding and you have lost the seating plan for the big table. You and your friends have gathered a list of all to be seated at the big table, but you cannot simply place them in any order since this could lead to some unpleasent situations. Alongside the list of members at the table you have also come up with a pairs of people which need to be seated next to each other and some which need to have some other guests in between.

For this exercise we will use our knowledge about custom Z3 datatypes and make use of *uninterpreted functions*.

You will be given a program skeleton which prepares the parsing of the input file. An input file consists of multiple lines which may either have

- a pair of friends: `Bob likes Alice`,
- a pair of foes: `Ada dislikes Bob`,
- a guest without preferences: `John` or
- a comment: `#John likes Ada`.

Your tasks are to:

(a) Extend the parsing of the skeleton in such a way that the prepared data structures get filled with Z3 variables.

(b) Define a Z3 uninterpreted function which maps our guests to enumerated positions at the table.

(c) Define a function `neighbours` which returns whether two guests are neighbours.

(d) Each guest should sit at exactly one position at the table.

(e) Friends need to be seated next to each other and foes must not be seated next to each other.

We represent the positions at our table with an uninterpreted function which maps to the integer positions at the table. Note that the table wraps around on both edges of this integer list that we are mapping to, i.e. position `0` is right next to position `len(guests)`.

If we can find a proper seating plan which adheres to our given constraints we will visualize it otherwise Z3 will tell us that this is not satisfiable. Note that Z3 will not fully define our uninterpreted function but we can get around that by extending our call to `model.evaluate(...)` to `model.evaluate(...,model_completion=True)`, which will assing all free variables. The seating plan will be visualized with a table in your terminal and as a list of the mapping in the very end of our program call.

```
Patric dislikes Ada
Patric dislikes Katie
Patric dislikes Bob
Ada
John likes Alice
Bob likes Andrea
Andrea
Alice
Ada likes Julia
Ada likes Katie
Robert
```

Figure 1: The constraints and member list that you have come up with.

```
       Patric Robert
       ---------------------
Andrea|<<<<<<<<<<<<<<<<<<<<<|Katie
      |>>>>>>>>>>>>>>>>>>>>>|
   Bob|<<<<<<<<<<<<<<<<<<<<<|Ada
      |>>>>>>>>>>>>>>>>>>>>>|
  John|<<<<<<<<<<<<<<<<<<<<<|
       ---------------------
       Alice Julia

Seating plan:['Patric', 'Robert', 'Katie', 'Ada', 'Julia',
'Alice', 'John', 'Bob', 'Andrea']
```

Figure 2: A possible seating plan for the given input file from Figure 1.