DUS 2020 – Bonus assignment

The goal of the individual bonus assignment is to create a command line program in the Python programming language. The program must be able to process an input file with a list of executable sequences of transitions in a Petri net and then generate two sets of inequalities. The first set must contain all the inequalities that ensure that all of the sequences can be executed. The second set must contain all the inequalities that ensure that none of the wrong continuations is executable. Up to 5 bonus points can be awarded for a working solution.

The programs receives two arguments from the command line. The first argument gives the relative path to the input file. The second argument gives the relative path to the output file.

The input file contains all the executable sequences of transitions in some Petri net. Each line contains one sequence. The sequences are represented by a string of lowercase characters, where each letter represents one transition.

Example of an input file:

```
abc
acb
cab
```

The output file must have the following format. Each line of the output file contains one inequality. The file starts with all the inequalities that ensure that all the input sequences are executable. These inequalities are followed by one empty line, that separates the two sets of inequalities. The inequalities that prevent all the wrong continuations from the input sequences follow this empty line.

The order of the inequalities in either set doesn’t matter.

If two inequalities are equal to each other because of the commutative property, it is not necessary to output both of them. However outputting both of them is not considered as an error.

The output file must not contain superfluous entries. The first set of inequalities must not contain inequalities that would ensure execution of sequences that are not part of the input file. The second set of inequalities cannot prevent the execution of either executable sequences from the input file, or not executable sequences that are already covered by one of the wrong continuations.

All the output inequalities must have the following format. The inequalities must start with a lowercase letter **m**, that represents the initial marking of a place in a Petri net. Then follows the left side of the inequality with entries in the form **+a_p** or **-a_c**. The letter **a** can be replaced by any letter of the alphabet and represents a transition in a Petri net. The symbols **+** and **–** can have an empty space on one or both sides. If the left side should contain the same variable multiple times (**+2a_p**), it must be written out multiple times (**+a_p+a_p**). The output file must not contain any number characters. The left side of the inequality is followed by an inequality symbol. The allowed symbols are (**>**, **<**, **>=**, **<=**). The inequality symbol can have an empty space on either or both sides. The inequality symbol is followed by the right side of the inequality in the form **a_c**. The letter **a** can be replaced by any letter of the alphabet.

Examples of correctly formatted output lines:

```
m>=a_c
m - b_c + b_p < a_c
m +c_p -c_c -c_c +c_p < a_c
```

Examples of incorrectly formatted output lines:

```
m> =a_c
m - b_c + b_p < +a_c
m + 2c_p - 2c_c < a_c
d_c + d_p >= a_c
m + e_c - e_p >= a_c
m - f_c + f_p < a_c + b_c
```

The program does not have to solve the system of inequalities. The program must be able to generate the sets of inequalities that are necessary to perform the synthesis (process mining) of a Petri net. If the program is able to create the necessary systems of inequalities, solve them and construct the input and output matrices as well as the initial marking of the synthesized Petri net, more that 5 bonus points can be awarded.

Example input and output files can be downloaded from the course website.

When submitting the assignment follow all instructions given in the “Coursework submissions” section in the AIS.