Discrete structuren – Computer assignment resit

Name: Rodi Scheele
Student number: 2088262
Date: 26-04-2018

Used libraries

I use three main libraries in my code

Sympy

For most of my formulas I need symbols. I use the symbol functionality of sympy to achieve this. Sympy can use the symbols in formulas to calculate the outcome for them.
The other sympy functionalities that I use are .roots to find the roots of an equation and linsolve to solve systems of linear equations generated with the found roots and initial values.
The last sympy functionality I use is .simplify to simplify the F(n) notation.

Pandas

The only reason I import pandas is to write to a csv file. I have a build in solution checker which writes its results to a csv file.

Math

I import math to replace all my 'sqrt' values with 'math.sqrt'. This is done because using 'eval()' doesn't work on 'sqrt' and requires a library to understand what to do with it, hence 'math.sqrt'.

Code

The code is based on the template given within the course. Most of the reading of files formatting and reformatting of equations is done by the template.

The code written by myself are the 'solve homogeneous equation' and 'solve nonhomogeneous equation' functions. The solver for homogeneous relations is completely automated and just requires files as input, it will use the input and transform it to a formula as taught in the course, step by step.

The solver for non-homogeneous relations requires user input to define whether it is a 'exponential', 'constant' or 'polynomial' function, input as 'e' 'c' or 'p'. Currently only the exponential and constant functions work. Both of them transform the recurrence relation step by step to a formula, as done in the course.

I had some problems with the polynomial part. Main problem being that I was not able to do the algebra to calculate the pt, pt-1, …, p0 from theorem 6 from the book. The code goes as far as to formulate the equation with theorem 6 but the code can't find solutions for the p's.

Last but not least I have written the 'check_solution' function. This function takes the recurrence relation and the output formula and calculates the first 20 values after the initial values with both functions. It then compares both outputs and checks whether they are the same with a difference of maximum 1/1000. The output is a csv file. This has been implemented as a check for myself.