Scientific paper summary and coding test

Submission deadline: Friday 25th of July at 17.00 Seoul time

Material provided: This description and the journal article from Bhaskar and Nigam.

Coding language: Python

General remarks: The use of Large Language Models (LLMs) for this test is forbidden and can be detected by us. Finding information on the net is allowed.

Description of the tasks:

1- Using the article shared in the email, read the article carefully and make a summary (in English) about the *Π calculus as well as reasoning with dimensionless numbers* parts of the paper in a separate pdf document.

2- Using this knowledge from the paper, develop an algorithm (using a flowchart representation) and code this algorithm for the pressure regulator Figure 7 copied below using Python as a coding language.

   The user of your code should be able with your code to:

   a. Define qualitative objectives on the variables of the problem. The variables in the model can be Increased (I), Decreased (D) or kept Constant (C). In the pressure regulator example, the goal is to keep *Pout* constant (C), while the input pressure Pin is decreasing or increasing (D or I). Nevertheless, your code should be able to let the user also define objectives on the other variables of the model.

   b. Propagate those objectives on the other variables of the model using the Pi calculus and the equations presented in Figure 7 of the article and copied below.

   c. Generate a list of the variables where contradictions have been detected by your algorithm. In your algorithm and code, **a contradiction is discovered if** two **contradictory objectives (**I, D or C) are obtained on a single variable **after propagation of the initial objectives in the equations of the pr**essure regulator.

      In the example of the pressure regulator, all the propagation of changes in the pressure regulator starts from a change in the pressure *Pin*. A change in *Pin* is **initiating the propagation to all the** other variables in the model.

      Note: In this exercise, **a contradiction** emerges from the transient of the pressure *Pin* (The transients are the **short variations** that would affect the **variations of** *Pout* due **to the variation of** *Pin*). **After this** transient, steady-state behavior will appear. The **transition from transient to** steady state is not studied in this exercise. Only the results **of the initial variations (i.e.** transient) of *Pin* and its **propagation** are part of the exercise.

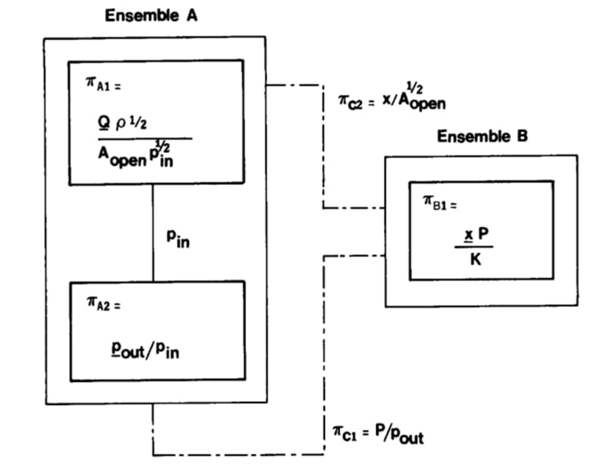
Ensemble A
$\pi_{A1} = \frac{Q \rho^{1/2}}{A_{open} p_{in}^{1/2}}$
p_{in}
$\pi_{A2} = p_{out}/p_{in}$
$\pi_{C2} = x/A_{open}^{1/2}$
Ensemble B
$\pi_{B1} = \frac{x P}{K}$
$\pi_{C1} = P/p_{out}$

Fig. 7. Inter-ensemble analysis of the pressure regulator.

Material to be provided at the end of the test:

1- A pdf document integrating the summary and the flowchart of the algorithm.
2- A Python code that we can analyze and test.

Key evaluation criteria:

3- Clarity of the summary
4- Clarity and simplicity of the algorithm
5- Quality of the coding in terms of code architecture, coding style and comments provided to the code for clarity and reuse, ability to integrate your code into other applications (modularity) and finally speed performance of the code generated.

Questions: If you have any questions or concerns regarding the clarity of the task described above, please contact me by email. This is perfectly acceptable to do it.