INF 212 Analysis of Programming Languages
Project 2 – Operational Semantics
Due date: 4/15
Demo date: 4/16

This is a pen-and-paper project!

Part 1. "While" Language Parallelism Extension

During the class, we saw a simplified "While" language. Now we define a new language: While^{par}, which supports parallelism command (C1 par C2) operations. In this construct, no particular preference is given to one or the other; this means that their executions are to be interleaved, which will lead to non-deterministic behavior.

a) Specify the inference rules for While^{par}

b) Considering the "Conditional Critical Regions" mechanism, implement a simple "Conditional Critical Regions" mechanism (for example, a command waits for execution when the condition is not true, then the command will be executed without interruptions) for While^{par} . Specify the inference rules.

Part 2. Proof

In class, we introduced how to prove the correctness of language implementation. Finish the proof of soundness of the implementation of "While language" of the lecture.

Some hints:
To simplify the task, we assume the correctness for Arithmetic and Boolean expressions have been asserted.
We also provide the following Lemma here for you to use directly without need to provide a proof:

if $\langle c^1, e^1, s\rangle \rhd^k \langle c', e', s'\rangle$ then
$\langle c_1{:}c_2, e_1{:}e_2, s\rangle \rhd^k \langle c'{:}c_2, e'{:}e_2, s'\rangle$