Exercise 1 (Parallel Composition in ParWHILE): (3 Points)

In the lecture we extend the **while** language with a parallel composition operator $c_1 \parallel c_2 \in \mathbf{Cmd}$, which expresses that both c_1 and c_2 have to be executed in parallel and the execution steps from c_1 and c_2 are interleaved.

An equivalence relation $\cong$ w.r.t a language is said to be a congruence relation if it is preserved by the language constructs, for example for command constructs, that is if $\forall c_1, c_2, c_3 \in \mathbf{Cmd}\ ;\ \forall b \in \mathbf{BExp}$ such that $c_1 \cong c_2$ then

$$\begin{aligned} c_1; c_3 &\cong c_2; c_3 \\ \textbf{if } b \textbf{ then } c_1 \textbf{ else } c_3 &\cong \textbf{if } b \textbf{ then } c_2 \textbf{ else } c_3 \\ \textbf{while } b \textbf{ do } c_1 &\cong \textbf{while } b \textbf{ do } c_2 \\ c_1 \parallel c_3 &\cong c_2 \parallel c_3 \\ &\dots \end{aligned}$$

and analogously for the boolean and axiomatic expressions. Let the binary relation $\approx$ be defined as

$$c \approx c' \text{ if only if } \forall \sigma, \sigma' \in \Sigma.\ \langle c, \sigma \rangle \to^* \sigma' \iff \langle c', \sigma \rangle \to^* \sigma'$$

Show that $\approx$ w.r.t the extended **while** language is *not* a congruence relation.

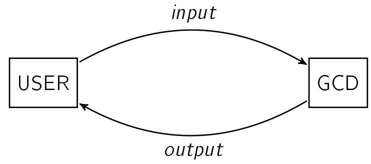
Exercise 2 (CSP Semantics): (2+2 Points)

a) Consider the following CSP program c:

$c :=$
$y := 4; \textbf{if}\,(y > 0) \to ((x := y) \parallel (x := 3))\,\textbf{fi}$
$\textbf{do}\,(x == 3 \wedge \alpha?x \to \beta!x) \,\square\, (x == 3 \to \alpha!y)\,\textbf{od}$

Provide all "meanings" of c using the formal semantics of CSP as given in the lecture.

b) Consider following system (illustrated in following figure) which the USER can give an input (two integers) over the channel *input* to a GCD and the GCD will compute the greatest common divisor based on the USER's input and feedback the result over the channel *output*. Please model this system by using CSP language.



Exercise 3 (Fairness in CSP): (3 Points)

a) Does strong unfairness implies weak unfairness?

b) Now we relax the requirement of CSP in the lecture and assume we can have common variables in parallel composition of commands. Consider following mutual exclusion algorithm in CSP. Is the fairness requirement for two processes P_1 and P_2: "each process enters its critical section infinitely often" satisfied?

```
b1 := false; b2 := false; P1 ∥ P2;
where
P1 :
   do true →
           ... noncritical section ...
           b1 := true;
           x := 2;
           if (x = 1) ∨ (¬b2) →
           ... critical section ...
           fi
           b1 := false;
   od

P2 :
   do true →
           ... noncritical section ...
           b2 := true;
           x := 1;
           if (x = 2) ∨ (¬b1) →
           ... critical section ...
           fi
           b2 := false;
   od
```