1. (15 points)

Consider the following CCS system Sys, which is intended to implement a simple communications protocol containing a sender S and a receiver R. The sender accepts messages to send, then gives them to the receiver and awaits an acknowledgement. For reasons of fault-tolerance, the sender may also “time-out” by executing a $\tau$ and resend.

$$
\begin{align*}
S & \triangleq \text{send}.S' \\
S' & \triangleq \text{out}.(\tau.S' + \text{ackin}.S) \\
R & \triangleq \text{in}.\text{recv}.\text{ackout}.R \\
\text{Sys} & \triangleq (S[c/\text{out},\text{ackin}] | R[c/\text{in},\text{ackout}]) \{c\}
\end{align*}
$$

(a) (5 points) What are the initial transition(s) of this system (give actions and target states)?

(b) (10 points) Call a CCS expression deadlocked if it is incapable of any transitions. Can the above system reach a deadlocked state? Explain your answer.

2. (20 points)

Although CCS is a communication-based formalism, it may also be used to model shared memory. The idea is to use a separate server process to handle each variable. Supposing the variable to be modeled is $V$, the associated server would use the actions $sV_i$ and $rV_i$ for every value $i$ that $V$ can assume. Action $sV_i$ corresponds to setting $V$ to $i$. Action $rV_i$ is enabled with $V$ has value $i$.

(a) (4 points) Give a CCS process implementing a boolean variable $V$. Assume that the values $V$ may assume are 0 and 1.

(b) (4 points) Construct the labeled transition system for this process.

(c) (2 points) How would you make a variable $V$ local to a particular system in CCS? How would you give read-only access to the environment?

(d) (10 points) Write a CCS system corresponding to the following piece of pseudo-code.

(Here cobegin ... coend is used to spawn threads. The idea is that each statement separated by the $||$ runs concurrently. Assume that memory accesses are atomic.)

Construct your system so that the environment has only read access to $V$.

```plaintext
bool V = 1;
cobegin
  V := not V
  || V := not V
coend
```

What can one infer about the value(s) of $V$ upon termination using the operational semantics of CCS?
3. (20 points)
Using whatever means you like, prove or disprove the following equivalences.

(a) (5 points) \( a.b.0 + a.c.0 \sim a.b.0 + a.c.0 + a.(b.0 + c.0) \)
(b) (5 points) \( a.0 + \tau.b.0 \approx_C a.0 + \tau.b.0 + b.0 \)
(c) (10 points) Let

\[
\begin{align*}
A & \triangleq a.A \\
B & \triangleq b.B \\
C & \triangleq a.C + b.C
\end{align*}
\]

Then \( C \sim A|B \).

4. (20 points) Consider the following definitions.

\[
\begin{align*}
A & \triangleq \text{startA}.a.\text{startB}.A \\
B & \triangleq \text{startB}.b.\text{startC}.B \\
C & \triangleq \text{startC}.c.\text{startD}.C \\
D & \triangleq \text{startD}.d.\text{startA}.D \\
S' & \triangleq (A \mid B \mid C \mid D) \setminus \{\text{startB, startC, startD}\} \\
S & \triangleq (\text{startA}.0 \mid S') \setminus \{\text{startA}\} \\
R & \triangleq a.b.c.d.R
\end{align*}
\]

Components A, B, C and D are processes in a system. Upon receiving its start signal, each performs its action and sends a start signal to its “neighbor”.

(a) (10 points) Show that \( S \approx R \).
(b) (5 points) Show that \( S \not\approx_C R \).
(c) (5 points) Is \( S \approx_C \tau.R \)?