Concurrent Programming - Exercícios 3

CCS

1. Which of the following expressions of CCS are correct?.
   (a) $(a.0 + \overline{a}.A)\backslash\{a, b\}$
   (b) $(a.0 + \overline{a}.A)\backslash\{a, \tau\}$
   (c) $\tau.\tau.B + 0$
   (d) $(a.b.B + \overline{a}.0)|B$
   (e) $(a.b.B + \overline{a}.0).B$

2. Let $A := b.a.B$, using the inference rules show that the following transitions exist:
   - $(A|\overline{b}.0)\backslash\{b\} \xrightarrow{\tau} (a.B|0)\backslash\{b\}$

3. Solve the CCS exercises in PseuCo.com

4. Consider the following definitions of a researcher that takes coffee and publishes articles.

$$\begin{aligned} CM &:= coin?.coffee!.CM \\ CS &:= pub!.coin!.coffee?.CS \\ Uni &:= (CM|CS)\backslash\{coin, coffee\} \end{aligned}$$

   Use the CSS inference rules to obtain the reachable fragment of $[\![Uni]\!]_\Gamma$. Test in pseuco.com. Compare with $Spec := pub!.Spec$.

5. Let $A := (a.A)\backslash\{b\}$ show that the reachable fragment of $[\![A]\!]_\Gamma$ is infinite.

6. The following definitions try to solve the mutual exclusion with a semaphore.
   (a)

$$\begin{aligned} Mutex_1 &:= (User|Sem)\backslash\{p, v\} \\ User &:= \overline{p}.enter.exit.\overline{v}.User \\ Sem &:= p.v.Sem \end{aligned}$$

   Use the CCS inference rules to obtain the reachable fragment of $[\![Mutex_1]\!]_\Gamma$. Test in pseuco.com.

   (b) Let

$$Mutex_2 := ((User|Sem)|User)\backslash\{p, v\}$$

   Use the CCS inference rules to obtain the reachable fragment of $[\![Mutex_2]\!]_\Gamma$. Test in pseuco.com. Would it be different if one uses $User := \overline{p}.enter.\overline{v}.exit.User$?

   (c) Let

$$\begin{aligned} FMutex &:= ((User|Sem)|FUser)\backslash\{p, v\} \\ FUser &:= \overline{p}.enter.(exit.\overline{v}.FUser + exit.\overline{v}.0) \end{aligned}$$

   Use the CCS inference rules to obtain the reachable fragment of $[\![FMutex]\!]_\Gamma$. Test in pseuco.com. Do you think that $Mutex_2$ and $FMutex$ has the same behaviour?

7. Model a level crossing with three components: the train, the barrier, and a controller. Consider that:

   - The train's actions characterize its location in relation to the crossing: it approaches (approach), enters (in), and exits (exit).
   - The barrier can be down or up.
   - The train and the barrier do not communicate directly; only the controller communicates with them. The controller must ensure that when the train enters, the barrier is down.

   (a) Describe each component in CCS: train, barrier, and controller.

   (b) Write the system process that corresponds to the parallel execution of the various components, indicating the actions in which they must synchronize.

   (c) Use the CSS inference rules (mentioning the name of the rule used in each step) to obtain the reachable fragment of that process.

8. Consider:

$$
\begin{aligned}
C_0 &:= inc.C_1 \\
C_n &:= inc.C_{n+1} + dec.C_{n-1}, \text{ for } n \geq 1
\end{aligned}
$$

   e $SC := inc.(SC|dec.0)$.

   (a) Determine the inicial fragment $[\![C_0]\!]_\Gamma$. Argue that the LTS cannot be state-finite.

   (b) Determine the accessible fragment of LTS (and initial) $[\![SC]\!]_\Gamma$

   (c) Determine if the two LTS are isomorphic. Is there any similarity between them?