Exercise 1*

Which of the following expressions are correctly built CCS expressions? Why?
(Assume that A, B are process constants and a, b are channel names.)

- $a.b.A + B$
- $(a.Nil + \overline{a}.A) \smallsetminus \{a, b\}$
- $(a.Nil \,|\, \overline{a}.A) \smallsetminus \{a, \tau\}$
- $a.B + [a/b]$
- $\tau.\tau.B + Nil$
- $(a.B + b.B)[a/b, b/a]$
- $(a.B + \tau.B)[a/\tau, b/a]$
- $(a.B + \tau.B)[\tau/a]$
- $(a.b.A + \overline{a}.Nil) \,|\, B$
- $(a.b.A + \overline{a}.Nil).B$
- $(a.b.A + \overline{a}.Nil) + B$
- $(Nil \,|\, Nil) + Nil$

Exercise 2*

By using SOS rules for CCS prove the existence of the following transitions (assume that $A \stackrel{\text{def}}{=} b.a.B$):

- $(A \,|\, \overline{b}.Nil) \smallsetminus \{b\} \xrightarrow{\tau} (a.B \,|\, Nil) \smallsetminus \{b\}$
- $(A \,|\, \overline{b}.a.B) + (\overline{b}.A)[a/b] \xrightarrow{\overline{b}} (A \,|\, a.B)$
- $(A \,|\, \overline{b}.a.B) + (\overline{b}.A)[a/b] \xrightarrow{\overline{a}} A[a/b]$

Exercise 3*

Consider the following CCS defining equations:

$$CM \stackrel{\text{def}}{=} coin.\overline{coffee}.CM$$
$$CS \stackrel{\text{def}}{=} \overline{pub}.\overline{coin}.coffee.CS$$
$$Uni \stackrel{\text{def}}{=} (CM \mid CS) \smallsetminus \{coin, coffee\}$$

Use the rules of the SOS semantics for CCS to derive the labelled transition system for the process Uni defined above. The proofs can be omitted and a drawing of the resulting LTS is enough.

Exercise 4

Draw (part of) the labelled transition system for the process constant A defined by

$$A \stackrel{\text{def}}{=} (a.A) \smallsetminus \{b\}.$$

The resulting LTS should have infinitely many reachable states. Can you think of a CCS term that generates a finite LTS and intuitively has the same behaviour as A?

Exercise 5 (optional)

1. Draw the transition graph for the process name Mutex_1 whose behaviour is given by the following defining equations.

$$\begin{aligned} \text{Mutex}_1 &\stackrel{\text{def}}{=} (\text{User} \mid \text{Sem}) \setminus \{p, v\} \\ \text{User} &\stackrel{\text{def}}{=} \bar{p}.\text{enter}.\text{exit}.\bar{v}.\text{User} \\ \text{Sem} &\stackrel{\text{def}}{=} p.v.\text{Sem} \end{aligned}$$

2. Draw the transition graph for the process name Mutex_2 whose behaviour is given by the defining equation

$$\text{Mutex}_2 \stackrel{\text{def}}{=} ((\text{User}|\text{Sem})|\text{User}) \setminus \{p, v\}$$

   where User and Sem are defined as before. Would the behaviour of the process change if User was defined as

$$\text{User} \stackrel{\text{def}}{=} \bar{p}.\text{enter}.\bar{v}.\text{exit}.\text{User} \ \ ?$$

3. Draw the transition graph for the process name FMutex whose behaviour is given by the defining equation

$$\text{FMutex} \stackrel{\text{def}}{=} ((\text{User} \mid \text{Sem}) \mid \text{FUser}) \setminus \{p, v\}$$

   where User and Sem are defined as before, and the behaviour of FUser is given by the defining equation

$$\text{FUser} \stackrel{\text{def}}{=} \bar{p}.\text{enter}.(\text{exit}.\bar{v}.\text{FUser} + \text{exit}.\bar{v}.Nil)$$

   Do you think that Mutex_2 and FMutex are offering the same behaviour? Can you argue informally for your answer?