# XMP 2013/14 Assignment 1 (theory part)

## Reasoning about deterministic processes

Consider the following process definitions:

$$\begin{aligned}
\alpha VM &= \{\mathsf{coin}, \mathsf{pepsi}, \mathsf{sprite}\} \\
VM &\triangleq \mathsf{coin} \rightarrow (\mathsf{pepsi} \rightarrow VM \mid \mathsf{coin} \rightarrow \mathsf{sprite} \rightarrow VM) \\
\alpha RICH &= \{\mathsf{coin}, \mathsf{think}\} \\
RICH &\triangleq \mathsf{coin} \rightarrow \mathsf{think} \rightarrow RICH \\
\alpha THIRSTY1 &= \{\mathsf{sprite}, \mathsf{pepsi}\} \\
THIRSTY1 &\triangleq \mathsf{sprite} \rightarrow THIRSTY1 \\
\alpha THIRSTY2 &= \{\mathsf{sprite}, \mathsf{pepsi}\} \\
THIRSTY2 &\triangleq \mathsf{pepsi} \rightarrow THIRSTY2
\end{aligned}$$

**Note:** there is not necessarily any intuitive sense to these definitions; just reason about the processes exactly as defined. Pay particular attention to the alphabets.

1. Show that the vending machine defined by $VM$ never accepts more than two coins over the value of the products it has dispensed. More precisely, show that

   $$VM \ \mathbf{sat}\ (tr \downarrow \mathsf{coin}) - (tr \downarrow \mathsf{pepsi}) - 2 \cdot (tr \downarrow \mathsf{sprite}) \leq 2\,.$$

2. For each of the following processes, use the CSP algebra laws to derive a provably equivalent process definition that doesn't use concurrent composition ("$\|$") anywhere. In each case, start by stating the alphabet of the resulting process.

   a. $Sa = VM \parallel RICH$.
   b. $Sb = Sa \parallel THIRSTY1$, building on the result from (a).
   c. $Sc = RICH \parallel THIRSTY1$.
      *Hint:* Also define $Sc' = (\mathsf{think} \rightarrow RICH) \parallel THIRSTY1$.
   d. $Sd = VM \parallel Sc$, building on the result from (c). Compare with the result from (b).
   e. $Se = Sa \parallel THIRSTY2$, building on the result from (a).
   f. $Sf = VM \parallel f(Sc)$, building on the result from (c), and where the renaming function $f : \alpha Sc \rightarrow \alpha Sc$ is given by $f(\mathsf{coin}) = \mathsf{coin}, f(\mathsf{think}) = \mathsf{think}, f(\mathsf{pepsi}) = \mathsf{sprite}$, and $f(\mathsf{sprite}) = \mathsf{pepsi}$. Compare with the result from (e).

3. A deterministic process $P$ can deadlock if, for some trace $s \in traces(P)$, we have $P \,/\, s = STOP_{\alpha P}$.

   a. Argue that none of the processes from 1(a–d) can deadlock.
   b. Show that the process from 1(e) can deadlock, by exhibiting a trace $s$ after which $Se$ is deadlocked. (That is, show formally that $s \in traces(Se)$ and that $Se \,/\, s = STOP$.)
   c. Show that your $s$ from 2(b) is also a valid trace for the original, explicitly concurrent process $Sx = (VM \parallel RICH) \parallel THIRSTY2$, i.e., $s \in traces(Sx)$, *without* eliminating the concurrent composition first, i.e., without using the laws in 2.3.1.
   d Compute $Sy = Sx \,/\, s$, again without eliminating the concurrent compositions. Then show, for each event $x \in \{\mathsf{coin}, \mathsf{pepsi}, \mathsf{sprite}, \mathsf{think}\}$, that $\langle x \rangle \notin traces(Sy)$.

Show your derivations. Make it clear exactly which law you use where (include the section numbers before the law numbers, e.g., "by 2.3.1 L3A"), and be careful about parentheses and indentation. (Proofread your solutions!) When using 2.3.1 L7, say explicitly how the sets $A$, $B$, and $C$ are instantiated in the law, even if the choices are expressed as explicit enumerations.

You may abbreviate the event names to just their initial letters ($\{\mathsf{c}, \mathsf{p}, \mathsf{s}, \mathsf{t}\}$), and the process names to $V$, $R$, $T1$, and $T2$. If you use any non-obvious ASCII substitutes for symbols (e.g., "$" for "↓"), be sure to explain then at the top of your document.