Extreme Multiprogramming 2013/14 Assignment 2 (theory part)

Reasoning about communication networks

Let add and fetch be CSP channels, both with alphabet $\mathbf{Z}$ (the integers). Consider the following process, with add as an input channel and fetch as an output channel:

$$STORE_x \triangleq (\mathsf{add}?y \to STORE_{x+y}) \,\square\, (\mathsf{fetch}!x \to STORE_{x-1})$$

Let $A = \{\mathsf{add}.n \mid n \in \mathbf{Z}\}$, i.e., the events arising from communication on add. Then we can construct a two-process network,

$$NET = (STORE_{10} \parallel (\mathsf{add}!3 \to STOP_A)) \setminus A$$

Both ends of channel add are evidently used in the concurrent composition, so as usual, we conceal this internal channel from the environment. Thus, NET has only a single output channel, fetch.

1. Use the CSP algebraic laws to find an equivalent definition of NET that does not use concealment ("$\setminus$"), general choice ("$\square$"), or concurrent composition ("$\parallel$") anywhere. It is ok to introduce named auxilary process definitions; you do not have to use μ-notation to express recursion.
2. Let $s = \langle \mathsf{fetch}.10, \mathsf{fetch}.12, \mathsf{fetch}.11 \rangle$.
   (a) Use the laws to show that $s \in traces(NET)$.
   (b) Use the laws to compute $NET \,/\, s$ as a process expression not involving residuation ("/").
3. Compare the refusal sets of NET and $NET \,/\, s$:
   (a) Use the laws to show that $\{\mathsf{fetch}.10, \mathsf{fetch}.42\} \in refusals(NET)$, but $\{\mathsf{fetch}.13\} \notin refusals(NET)$.
   (b) Use the laws to show that $\{\mathsf{fetch}.13, \mathsf{fetch}.42\} \in refusals(NET \,/\, s)$, but $\{\mathsf{fetch}.10\} \notin refusals(NET \,/\, s)$.

In all cases, explicitly identify the laws you used, including the section number. When using laws about parameterized choice (such as 1.8.1 L4, 2.3.1 L7, or 3.5.1 L10[1]) to reason about choices with communication guards, state explicitly how the choice sets are instantiated in each case (i.e., the exact event sets being offered, not just the channel names).

To keep the notation compact, for a channel c, you may use the abbreviation $\{\!|c|\!\} = \{c.m \mid m \in \alpha c\}$; for example, we have $A = \{\!|\mathsf{add}|\!\}$. Also, if you are not typesetting your answers (and possibly even if you are), please write all process parameters between parentheses: render $STORE_{x+y}$ as `STORE(x+y)`, not `STORE_x+y`, or similar.

Hint: Do not be deceived by the apparent simplicity of the assignment; it requires a thorough understanding of the CSP laws, and skill in applying them systematically. If you find that you cannot complete part 2 or 3, it may be because you made an error in an earlier part; go back and re-check your work.

[1] Remember the misprint in this law, as noted in Lecture 8, Slide 11.