UNIVERSITY OF COPENHAGEN

XMP: Exam
- Theory

Jens Fredskov (chw752)

January 22, 2014

Problem 1

a)

We first consider our given processes to eliminate the general choice in *PROD*.

$$PROD = (\mathsf{in}?x \to \mathsf{chk}!(x-4) \to PROD) \;\square\; (\mathsf{rej}?y \to \mathsf{chk}!(y-2) \to PROD)$$
$$= (\mathsf{in}?x \to \mathsf{chk}!(x-4) \to PROD) \mid (\mathsf{rej}?y \to \mathsf{chk}!(y-2) \to PROD) \qquad [3.3.1\ \mathrm{L5}]$$

$$QUAL = \mathsf{chk}?z \to ((\mathsf{out}!z \to QUAL) \lhd ok(z) \rhd (\mathsf{rej}!z \to QUAL))$$

We now define an auxiliary process $TMP0_x$

$$TMP0_x = (\mathsf{chk}!x \to PROD) \parallel QUAL$$
$$= \mathsf{chk}.x \to (PROD \parallel ((c!z \to QUAL) \lhd ok(z) \rhd (\mathsf{rej}!z \to QUAL))) \qquad [2.3.1\ \mathrm{L4A}]$$
$$= \mathsf{chk}.x \to ((PROD \parallel (c!z \to QUAL)) \lhd ok(z) \rhd (PROD \parallel (\mathsf{rej}!z \to QUAL))) \qquad [\mathrm{LCD}]$$

With the sets $A = \{|\mathsf{in}|, |\mathsf{rej}|\}, B = \{\mathsf{out}.x\}, C = \{|\mathsf{in}|, \mathsf{out}.x\}$, the left hand side of the *if-then*-clause becomes

$$LHS = PROD \parallel (\mathsf{out}!x \to QUAL)$$
$$= (\mathsf{in}?z \to ((\mathsf{chk}!(z-4) \to PROD) \parallel (\mathsf{out}!x \to QUAL))$$
$$\mid \mathsf{out}!x \to (PROD \parallel QUAL)) \qquad [2.3.1\ \mathrm{L7}]$$
$$= (\mathsf{in}?z \to \mathsf{out}!x \to ((\mathsf{chk}!(z-4) \to PROD) \parallel QUAL)$$
$$\mid \mathsf{out}!x \to \mathsf{in}?z \to ((\mathsf{chk}!(z-4) \to PROD) \parallel QUAL)) \qquad [2.3.1\ \mathrm{L5A/B}]$$
$$= (\mathsf{in}?z \to \mathsf{out}!x \to TMP0_{z-4} \mid \mathsf{out}!x \to \mathsf{in}?z \to TMP0_{z-4})$$

With the sets $A = \{|\mathsf{in}|, |\mathsf{rej}|\}, B = \{\mathsf{rej}.x\}, C = \{|\mathsf{in}|, \mathsf{rej}.x\}$, the right hand side becomes

$$RHS = PROD \parallel (\mathsf{rej}!x \to QUAL)$$
$$= (\mathsf{in}?z \to ((\mathsf{chk}!(z-4)PROD) \parallel (\mathsf{rej}!x \to QUAL))$$
$$\mid \mathsf{rej}.x \to ((\mathsf{chk}!(x-2) \to PROD) \parallel QUAL)) \qquad [2.3.1\ \mathrm{L7}]$$
$$= (\mathsf{in}?z \to STOP \mid \mathsf{rej}.x \to TMP0_{x-2})$$

We then put the side together again

$$TMP0_x = \mathsf{chk}.x \to ((\mathsf{in}?z \to \mathsf{out}!x \to TMP0_{z-4} \mid \mathsf{out}!x \to \mathsf{in}?z \to TMP0_{z-4})$$
$$\lhd ok(x) \rhd (\mathsf{in}?z \to STOP \mid \mathsf{rej}.x \to TMP0_{x-2}))$$

We now define a new process which is the same as before, but with *CR* concealed.

$$TMP1_x = TMP0_x \setminus CR$$
$$= (\mathsf{chk}.x \to ((\mathsf{in}?z \to \mathsf{out}!x \to TMP0_{z-4} \mid \mathsf{out}!x \to \mathsf{in}?z \to TMP0_{z-4})$$
$$\lhd ok(x) \rhd (\mathsf{in}?z \to STOP \mid \mathsf{rej}.x \to TMP0_{x-2}))) \setminus CR$$
$$= ((\mathsf{in}?z \to \mathsf{out}!x \to TMP0_{z-4} \mid \mathsf{out}!x \to \mathsf{in}?z \to TMP0_{z-4})$$
$$\lhd ok(x) \rhd (\mathsf{in}?z \to STOP \mid \mathsf{rej}.x \to TMP0_{x-2})) \setminus CR \qquad [3.5.1\ \mathrm{L5}]$$
$$= (((\mathsf{in}?z \to \mathsf{out}!x \to TMP0_{z-4} \mid \mathsf{out}!x \to \mathsf{in}?z \to TMP0_{z-4})$$
$$\setminus CR) \lhd ok(x) \rhd ((\mathsf{in}?z \to STOP \mid \mathsf{rej}.x \to TMP0_{x-2}) \setminus CR)) \qquad [\mathrm{LCD}]$$

Then twice with sets $B = \{|\mathsf{in}|, \mathsf{out}.x\}, C = \{|\mathsf{chk}|, |\mathsf{rej}|\}$

$$
\begin{aligned}
&= ((\mathsf{in}?z \to \mathsf{out}!x \to \mathit{TMP1}_{z-4} \mid \mathsf{out}!x \to \mathsf{in}?z \to \mathit{TMP1}_{z-4}) \\
&\quad \lhd ok(x) \rhd ((\mathsf{in}?z \to \mathit{STOP} \mid \mathsf{rej}.x \to \mathit{TMP0}_{x-2}) \setminus \mathit{CR})) && [3.5.1\ L8]
\end{aligned}
$$

Finally with sets $B = \{|\mathsf{in}|, \mathsf{rej}.x\}, C = \{|\mathsf{chk}|, |\mathsf{rej}|\}$

$$
\begin{aligned}
&= ((\mathsf{in}?z \to \mathsf{out}!x \to \mathit{TMP1}_{z-4} \mid \mathsf{out}!x \to \mathsf{in}?z \to \mathit{TMP1}_{z-4}) \\
&\quad \lhd ok(x) \rhd (\mathit{TMP1}_{x-2} \sqcap (\mathit{TMP1}_{x-2} \,\Box\, (\mathsf{in}?z \to \mathit{STOP})))) && [3.5.1\ L10]
\end{aligned}
$$

To eliminate the general choice after hiding CR we define a new process

$$
\begin{aligned}
\mathit{TMP2}_x &= \mathit{TMP1}_x \,\Box\, (\mathsf{in}?z \to \mathit{STOP}) \\
&= ((\mathsf{in}?z \to \mathsf{out}!x \to \mathit{TMP1}_{z-4} \mid \mathsf{out}!x \to \mathsf{in}?z \to \mathit{TMP1}_{z-4}) \\
&\quad \lhd ok(x) \rhd (\mathit{TMP1}_{x-2} \sqcap (\mathit{TMP1}_{x-2} \,\Box\, (\mathsf{in}?z \to \mathit{STOP})))) \,\Box\, (\mathsf{in}?z \to \mathit{STOP}) \\
&= (((\mathsf{in}?z \to \mathsf{out}!x \to \mathit{TMP1}_{z-4} \mid \mathsf{out}!x \to \mathsf{in}?z \to \mathit{TMP1}_{z-4}) \,\Box\, (\mathsf{in}?z \to \mathit{STOP})) \\
&\quad \lhd ok(x) \rhd ((\mathit{TMP1}_{x-2} \sqcap (\mathit{TMP1}_{x-2} \,\Box\, (\mathsf{in}?z \to \mathit{STOP}))) \,\Box\, (\mathsf{in}?z \to \mathit{STOP}))) && [LCD]
\end{aligned}
$$

The left hand side of the *if-then*-clause then becomes

$$
\begin{aligned}
\mathit{LHS} &= (\mathsf{in}?z \to \mathsf{out}!x \to \mathit{TMP1}_{z-4} \mid \mathsf{out}!x \to \mathsf{in}?z \to \mathit{TMP1}_{z-4}) \,\Box\, (\mathsf{in}?z \to \mathit{STOP}) \\
&= (\mathsf{in}?z \to ((\mathsf{out}!x \to \mathit{TMP1}_{z-4}) \sqcap \mathit{STOP}) \mid \mathsf{out}!x \to \mathsf{in}?z \to \mathit{TMP1}_{z-4}) && [3.3.1\ L5]
\end{aligned}
$$

And the right hand side becomes

$$
\begin{aligned}
\mathit{RHS} &= (\mathit{TMP1}_{x-2} \sqcap (\mathit{TMP1}_{x-2} \,\Box\, (\mathsf{in}?z \to \mathit{STOP}))) \,\Box\, (\mathsf{in}?z \to \mathit{STOP}) \\
&= (\mathit{TMP1}_{x-2} \,\Box\, (\mathsf{in}?z \to \mathit{STOP})) \sqcap ((\mathit{TMP1}_{x-2} \,\Box\, (\mathsf{in}?z \to \mathit{STOP})) \,\Box\, (\mathsf{in}?z \to \mathit{STOP})) && [3.3.1\ L6] \\
&= \mathit{TMP2}_{x-2} \sqcap \mathit{TMP2}_{x-2} && [3.3.1\ L1\text{-}L3] \\
&= \mathit{TMP2}_{x-2} && [3.2.1\ L1]
\end{aligned}
$$

We then put the side together again

$$
\begin{aligned}
\mathit{TMP2}_x &= ((\mathsf{in}?z \to ((\mathsf{out}!x \to \mathit{TMP1}_{z-4}) \sqcap \mathit{STOP}) \mid \mathsf{out}!x \to \mathsf{in}?z \to \mathit{TMP1}_{z-4}) \\
&\quad \lhd ok(x) \rhd \mathit{TMP2}_{x-2})
\end{aligned}
$$

Finally we can use this in the definition of the earlier process

$$
\begin{aligned}
\mathit{TMP1}_x &= ((\mathsf{in}?z \to \mathsf{out}!x \to \mathit{TMP1}_{z-4} \mid \mathsf{out}!x \to \mathsf{in}?z \to \mathit{TMP1}_{z-4}) \\
&\quad \lhd ok(x) \rhd (\mathit{TMP1}_{x-2} \sqcap \mathit{TMP2}_{x-2}))
\end{aligned}
$$

We are now ready to find an equivalent definition for *MILL* using the subprocesses we have created.

$$
\begin{aligned}
\mathit{MILL} &= (\mathit{PROD} \parallel \mathit{QUAL}) \setminus \mathit{CR} \\
&= (\mathsf{in}?x \to ((\mathsf{chk}!(x-4) \to \mathit{PROD}) \parallel \mathit{QUAL})) \setminus \mathit{CR} && [2.3.1\ L5A] \\
&= \mathsf{in}?x \to (\mathit{TMP0}_{x-4} \setminus \mathit{CR}) && [3.5.1\ L5] \\
&= \mathsf{in}?x \to \mathit{TMP1}_{x-4}
\end{aligned}
$$

b)

One trace which could make *MILL* deadlock is $s = \langle \mathsf{in}.106, \mathsf{in}.106\rangle$. We first show that $s \in traces(MILL)$.

$$
\begin{aligned}
traces(MILL) &= traces(\mathsf{in}?106 \rightarrow TMP1_{102}) \\
&= \{t \mid t = \langle\rangle \vee \underline{(t_0 = \mathsf{in}.x \wedge t' \in traces(TMP1_{102}))}\} && [1.8.1\ \text{L2}] \\
traces(TMP1_{102}) &= traces(TMP1_{100} \sqcap TMP2_{100}) \\
&= traces(TMP1_{100}) \cup \underline{traces(TMP2_{100})} && [3.2.3\ \text{L1}] \\
traces(TMP2_{100}) &= traces(\mathsf{in}?106 \rightarrow ((\mathsf{out}!100 \rightarrow TMP1_{102}) \sqcap STOP) \\
&\qquad\qquad \mid \mathsf{out}!100 \rightarrow \mathsf{in}?106 \rightarrow TMP1_{102})) \\
&= \{t \mid t = \langle\rangle \vee \underline{(t_0 = \mathsf{in}.106 \wedge t' = traces((\mathsf{out}!100 \rightarrow TMP1_{102}) \sqcap STOP))} \\
&\qquad\qquad \vee (t_0 = \mathsf{out}.100 \wedge t' = traces(\mathsf{in}?106 \rightarrow TMP1_{102}))\} && [1.8.1\ \text{L3}] \\
traces((\mathsf{out}!100 \rightarrow TMP1_{102}) \sqcap STOP) &= traces((\mathsf{out}!100 \rightarrow TMP1_{102})) \cup \underline{traces(STOP)} && [3.2.3\ \text{L1}] \\
traces(STOP) &= \{\underline{\langle\rangle}\} && [1.8.1\ \text{L1}]
\end{aligned}
$$

Notice that we here have substituted the variables with actual values, as this means we easily can choose the correct side of the *if-else*-clause. We now show that $MILL\ /\ s = STOP$.

$$
\begin{aligned}
MILL\ /\ s &= (\mathsf{in}?x \rightarrow TMP1_{x-4})\ /\ \langle \mathsf{in}.106, \mathsf{in}.106\rangle \\
&= TMP1_{102}\ /\ \langle \mathsf{in}.106\rangle && [1.8.3\ \text{L3A}] \\
&= (TMP1_{100}\ /\ \langle \mathsf{in}.106\rangle) \sqcap (TMP2_{100}\ /\ \langle \mathsf{in}.106\rangle) && [3.2.3\ \text{L2}] \\
&= ((\mathsf{out}!100 \rightarrow TMP1_{102})\ /\ \langle\rangle) \sqcap (TMP2_{100}\ /\ \langle \mathsf{in}.106\rangle) && [1.8.3\ \text{L3}] \\
&= (\mathsf{out}!100 \rightarrow TMP1_{102}) \sqcap ((\mathsf{out}!100 \rightarrow TMP1_{102}) \sqcap STOP)\ /\ \langle\rangle) && [1.8.3\ \text{L3A}] \\
&= (\mathsf{out}!100 \rightarrow TMP1_{102}) \sqcap STOP && [3.2.1\ \text{L1}]
\end{aligned}
$$

Now depending on which side of the internal choice *MILL* chooses we get that $MILL\ /\ s = STOP$.

To see that we indeed have a deadlock we consider the state of *PROD* and *QUAL*. After the first action *QUAL* will be trying to send rej as the log was too long. However as only *PROD* has the in-channel it is allowed to choose both of its branches, and thus it might choose the first as this does not require participation from the environment (*QUAL*). In this situtation *PROD* will be waiting on chk which *QUAL* is unwilling to do at the moment, and *QUAL* will be waiting on rej which *PROD* is unwilling to do. Thus we clearly have a deadlock.

c)

If we consider the trace from the previous question we could simply add one more action such that $t = \langle \mathsf{in}.106, \mathsf{in}.106, \mathsf{out}.100\rangle$. We have already calculated $MILL\ /\ s$. Thus we can just calculate $refusals(MILL\ /\ s)$. We first use [3.4.1 L4]. Then since *STOP* refuses the entire alphabet, per [3.4.1 L1], we have that $\{\mathsf{out}.100\} \in refusals(MILL\ /\ s)$. We however also have that $\langle \mathsf{out}.100\rangle \in traces(MILL\ /\ s)$, as per [3.2.3 L2] and [1.8.3 L3A]. This proves that that $MILL\ /\ s$, and therefore also *MILL*, may behave nondeterministically.

Problem 2

a)

We first show that $PROD' \sqsubseteq_T PROD$. We consider the composition of traces from $PROD$, which has two different branches one can go down. Using [1.8.1 L2-L3], we can easily see that the first branch has the traces

$$\{\langle\rangle, \langle \mathsf{in}.x\rangle, \langle \mathsf{in}.x, \mathsf{chk}.(x-4)\rangle\}$$

In the same way the second branch has the traces

$$\{\langle\rangle, \langle \mathsf{rej}.y\rangle, \langle \mathsf{rej}.y, \mathsf{chk}.(y-2)\rangle\}$$

After the longest trace in both of the branches go back to $PROD$ and can then again do either of the branches. Thus if we define the sets

$$\begin{aligned} S &= \{\langle \mathsf{in}.x, \mathsf{chk}.(x-4)\rangle, \langle \mathsf{rej}.y, \mathsf{chk}.(y-2)\rangle\} \\ T &= \{\langle\rangle, \langle \mathsf{in}.x\rangle, \langle \mathsf{rej}.y\rangle\} \end{aligned}$$

We can describe all traces of $PROD$ as

$$traces(PROD) = \{(s^n) ^\frown t \mid n \in \mathbb{Z}_{\geqslant 0} \wedge s \in S \wedge t \in T\}$$

We now need to show that $PROD'$ can do all of these traces too. We first rewrite the definition of $PROD'$ by expanding $PRODN_x$ once and eliminating general choice

$$\begin{aligned} PROD' &= (\mathsf{in}?x \to PRODN_x) \,\square\, (\mathsf{rej}?y \to \mathsf{chk}!(y-2) \to PROD') \\ &= (\mathsf{in}?x \to ((\mathsf{chk}!(x-4) \to PROD') \,\square\, (\mathsf{rej}?y \to \mathsf{chk}!(y-2) \to PRODN_x))) \\ &\quad \square\, (\mathsf{rej}?y \to \mathsf{chk}!(y-2) \to PROD') \\ &= (\mathsf{in}?x \to ((\mathsf{chk}!(x-4) \to PROD') \mid (\mathsf{rej}?y \to \mathsf{chk}!(y-2) \to PRODN_x))) \\ &\quad \mid (\mathsf{rej}?y \to \mathsf{chk}!(y-2) \to PROD') && \text{[3.3.1 L5]} \end{aligned}$$

If we ignore the inner branch which leads to $PRODN_x$ this definition has the exact same branches as $PROD$. Thus we can go down either of these two branches to generate the same traces as in $PROD$.

We then show that $PROD \not\sqsubseteq_T PROD'$. This is done by exhibiting a trace $s \in traces(PROD') \wedge s \notin traces(PROD)$. Consider e.g. $s = \langle \mathsf{in}.x, \mathsf{rej}.y\rangle$. We first try to produce the trace in $PROD$

$$\begin{aligned} traces(PROD) &= traces((\mathsf{in}?x \to \mathsf{chk}!(x-4) \to PROD) \\ &\qquad \mid (\mathsf{rej}?y \to \mathsf{chk}!(y-2) \to PROD)) \\ &= \{t \mid t = \langle\rangle \vee \underline{(t_0 = \mathsf{in}.x \wedge t' \in traces(\mathsf{chk}!(x-4) \to PROD))} \\ &\qquad \vee (t_0 = \mathsf{rej}.y \wedge t' \in traces(\mathsf{chk}!(y-2) \to PROD))\} && \text{[1.8.1 L3]} \end{aligned}$$

$$traces(\mathsf{chk}!(x-4) \to PROD) = \{t \mid t = \langle\rangle \vee (t_0 = \mathsf{chk}.(x-4) \wedge t' \in traces(PROD))\} \qquad \text{[1.8.1 L2]}$$

As can be seen $s \notin traces(PROD)$. We now try to produce the trace in $PROD'$

$$\begin{aligned}
traces(PROD') &= traces(\mathsf{in}?x \to PRODN_x \mid \mathsf{rej}?y \to \mathsf{chk}!(y-2) \to PROD') \\
&= \{t \mid t = \langle\rangle \vee \underline{(t_0 = \mathsf{in}.x \wedge t' \in traces(PRODN_x))} \\
&\qquad \vee (t_0 = \mathsf{rej}.y \wedge t' \in traces(\mathsf{chk}!(y-2) \to PROD'))\} && [1.8.1\ L3] \\
traces(PRODN_x) &= traces(\mathsf{chk}!(x-4) \to PROD' \mid \mathsf{rej}?y \to \mathsf{chk}!(y-2) \to PRODN_x) \\
&= \{t \mid t = \langle\rangle \vee (t_0 = \mathsf{chk}.(x-4) \wedge t' \in traces(PROD')) \\
&\qquad \vee \underline{(t_0 = \mathsf{rej}.y \wedge t' \in traces(\mathsf{chk}!(y-2) \to PRODN_x))}\} && [1.8.1\ L3] \\
traces(\mathsf{chk}!(y-2) \to PRODN_x) &= \{t \mid \underline{t = \langle\rangle} \vee (t_0 = \mathsf{chk}.(y-2) \wedge t' \in traces(PRODN_x))\} && [1.8.1\ L2]
\end{aligned}$$

As seen $s \in traces(PROD')$, proving that $PROD \not\sqsubseteq_T PROD'$.

b)

The problem in 1b was that $PROD$ was only able to do chk after it had done in, and since meant that we risked getting in a situation where a log is fails the tests, and thus $QUAL$ wishes to send it back on rej. But instead of doing rej $PROD$ could also in again, meaning it would then be unwilling to do rej and instead wanting to do chk, which $QUAL$ would be unwilling to do.

Thus the situtation we need to check in

c)

Problem 3

a)

We first show that $PROD'' \sqsubseteq_T PROD'$. We consider the state of $PROD'$ after some trace. Before reaching a process definition $PROD'$ can do the trace $s = \langle \mathsf{in}.x\rangle$ or $t = \langle \mathsf{rej}.y, \mathsf{chk}.(y-2)\rangle$. after the traces we have (using [3.3.1 L5] to eliminate external choice and then [1.8.3 L3] to determine the process after the trace)

$$PROD' \ / \ s = PRODN_x$$
$$PROD' \ / \ t = PROD'$$

We thus see, that if we do that latter trace in $PROD''$, and $PROD'' \ / \ t = PROD''$ Then $PROD''$ can do every possible trace that $PROD'$ can do when going down its latter branch (as the both reach their own definition after the trace). It is easily seen using [3.6.2 L3] and [1.8.3 L3A] that the trace t results in $PRODR$ being advanced up to its recursive definition and thus $PROD'' \ / \ t = PROD''$.

We now consider the first branch of $PROD'$. After doing the trace s we reach $PRODN_x$, whereas we in $PROD''$, using [3.6.2 L3] and [1.8.3 L3A], have that

$$PROD'' \ / \ s = (\mathsf{chk}!(x-4) \to PRODI) \ ||| \ PRODR = PRODN''_x$$

For ease of use we have renamed the process after s. Both processes can thus do the trace of the first branch, after which they reach a new process definition. We therefore now consider whether we can do the same traces in $PRODN''_x$ as can be done in $PRODN_x$.

As with $PROD'$ we consider the traces of $PRODN_x$. Before reaching a process definition $PRODN_x$ can do the trace $p = \langle \mathsf{chk}.(x-4) \rangle$ or t. Again we have after each trace that (using [3.3.1 L5] to eliminate external choice and then [1.8.3 L3] to determine the process after the trace)

$$PRODN_x \ / \ p = PROD'$$
$$PRODN_x \ / \ t = PRODN_x$$

Like before we then consider $PRODN''_x \ / \ t$. As before and citing the same laws t advances $PRODR$ up to its recursive definition, and thus

$$PRODN''_x \ / \ t = PRODN''_x$$

As before $PRODN''_x$ can thus produce the same traces as the second branch of $PRODN_x$ because they both reach their own definition.

We then consider $PRODN''_x \ / \ p$. Again using using [3.6.2 L3] and [1.8.3 L3A], we have that

$$PRODN''_x \ / \ p = PROD''$$

Thus we are now back at the original definition for both processes. This means that $PRODN''_x$ can do the traces of $PRODN_x$ if $PROD''$ can do the traces of $PROD'$, and that $PROD''$ can do the traces of $PROD'$ if $PRODN''_x$ can do the traces of $PRODN_x$. Since we know that second branches can be done, we have recursively covered all branches of $PROD'$, meaning that $PROD'' \sqsubseteq_T PROD'$.

We now show that $PROD' \not\sqsubseteq_T PROD''$. This is done by exhibiting a trace $s \in traces(PROD'') \land s \notin traces(PROD')$. Consider e.g. $s = \langle \mathsf{rej}.y, \mathsf{in}.x \rangle$. We first try to produce the trace in $PROD'$

$$\begin{aligned} traces(PROD') &= traces(\mathsf{in}?x \to PRODN_x \mid \mathsf{rej}?y \to \mathsf{chk}!(y-2) \to PROD') \\ &= \{t \mid t = \langle\rangle \lor (t_0 = \mathsf{in}.x \land t' \in traces(PRODN_x)) \\ &\qquad \lor \underline{(t_0 = \mathsf{rej}.y \land t' \in traces(\mathsf{chk}!(y-2) \to PROD'))}\} \quad [1.8.1\ L3] \end{aligned}$$

$$traces(\mathsf{chk}!(y-2) \to PROD') = \{t \mid t = \langle\rangle \lor (t_0 = \mathsf{chk}.(y-2) \land t' \in traces(PROD'))\} \quad [1.8.1\ L2]$$

As can be seen we have no way to produce s and so $s \notin traces(PROD')$. We now try to produce the trace in $PROD''$.

$$\begin{aligned} traces(PROD'') &= traces(PRODI \mid\mid\mid PRODR) \\ &= \{s \mid \exists\, t : traces(PRODI) \bullet \exists\, u : traces(PRODR) \bullet s \ interleaves \ (t, u)\} \end{aligned}$$

If we now define the traces $t = \langle \mathsf{in}.x \rangle$ and $u = \langle \mathsf{rej}.y \rangle$, we need only show that $t \in \mathit{traces}(\mathit{PRODI})$ and $u \in \mathit{traces}(\mathit{PRODR})$, as one interleaving of these are s

$$
\begin{aligned}
\mathit{traces}(\mathit{PRODR}) &= \mathit{traces}(\mathsf{in}?x \to \mathsf{chk}!(x-4) \to \mathit{PRODI}) \\
&= \{t \mid t = \langle\rangle \vee \underline{(t_0 = \mathsf{in}.x \wedge t' \in \mathit{traces}(\mathsf{chk}!(x-4) \to \mathit{PRODI}))}\} && [1.8.1\ \text{L2}] \\
\mathit{traces}(\mathsf{chk}!(x-4) \to \mathit{PRODI}) &= \{t \mid \underline{t = \langle\rangle} \vee (t_0 = \mathsf{chk}.(x-4) \wedge t' \in \mathit{traces}(\mathit{PRODI}))\} && [1.8.1\ \text{L2}]
\end{aligned}
$$

$$
\begin{aligned}
\mathit{traces}(\mathit{PRODR}) &= \mathit{traces}(\mathsf{rej}?y \to \mathsf{chk}!(y-2) \to \mathit{PRODR}) \\
&= \{t \mid t = \langle\rangle \vee \underline{(t_0 = \mathsf{rej}.y \wedge t' \in \mathit{traces}(\mathsf{chk}!(y-2) \to \mathit{PRODR}))}\} && [1.8.1\ \text{L2}] \\
\mathit{traces}(\mathsf{chk}!(y-2) \to \mathit{PRODR}) &= \{t \mid \underline{t = \langle\rangle} \vee (t_0 = \mathsf{chk}.(y-2) \wedge t' \in \mathit{traces}(\mathit{PRODR}))\} && [1.8.1\ \text{L2}]
\end{aligned}
$$

Thus we have shown that $s \in \mathit{traces}(\mathit{PROD}'')$ and therefore that $\mathit{PROD}' \not\sqsubseteq_T \mathit{PROD}''$.

b)

We consider a trace where we read a value which is rejected first time around thrice (the third does not have to reject). This could e.g. be

$$t = \langle \mathsf{in}.108, \mathsf{in}.108, \mathsf{in}.108 \rangle$$

To show that this can create a deadlock we consider the state of QUAL and PROD'' after the trace. For PROD'' the trace we use will have rej and chk injected and the trace then becomes

$$t_1 = \langle \mathsf{in}.108, \mathsf{chk}.104, \mathsf{rej}.104, \mathsf{in}.108, \mathsf{chk}.104, \mathsf{in}.108 \rangle$$

For QUAL we remove in and inject rej and chk, giving us the following trace

$$t_2 = \langle \mathsf{chk}.104, \mathsf{rej}.104, \mathsf{chk}.104 \rangle$$

We now run the two processes in parallel to show that we reach a point after t where MILL'' will be in a state of deadlock.

$$
\begin{aligned}
\mathit{PROD}' \ / \ t_1 &= ((\mathsf{in}?x \to \mathsf{chk}!(x-4) \to \mathit{PRODI}) \\
&\quad ||| \ (\mathsf{rej}?y \to \mathsf{chk}!(y-2) \to \mathit{PRODR})) \ / \ t_1 \\
\mathit{QUAL} \ / \ t_2 &= (\mathsf{chk}?z \to ((\mathsf{out}!z \to \mathit{QUAL}) \\
&\quad \lhd \mathit{ok}(z) \rhd (\mathsf{rej}!z \to \mathit{QUAL}))) \ / \ t_2
\end{aligned}
$$

PROD' first read from in

$$
\begin{aligned}
\mathit{PROD}' \ / \ t_1 &= ((\mathsf{chk}!(104) \to \mathit{PRODI}) \\
&\quad ||| \ (\mathsf{rej}?y \to \mathsf{chk}!(y-2) \to \mathit{PRODR})) \ / \ t_1' && [3.6.2\ \text{L3}] + [1.8.3\ \text{L3A}] \\
\mathit{QUAL} \ / \ t_2 &= (\mathsf{chk}?z \to ((\mathsf{out}!z \to \mathit{QUAL}) \\
&\quad \lhd \mathit{ok}(z) \rhd (\mathsf{rej}!z \to \mathit{QUAL}))) \ / \ t_2
\end{aligned}
$$

Both are now willing to do chk

$$\begin{aligned} PROD' \ / \ t_1 &= (PRODI \\ &\quad ||| \ (\mathsf{rej}?y \to \mathsf{chk}!(y-2) \to PRODR)) \ / \ t_1'' && [3.6.2\ L3] + [1.8.3\ L3A] \\ QUAL \ / \ t_2 &= ((\mathsf{out}!104 \to QUAL) \\ &\quad \lhd ok(104) \rhd (\mathsf{rej}!104 \to QUAL)) \ / \ t_2' && [1.8.3\ L3A] \end{aligned}$$

$ok(104)$ is false and, both are now willing to do rej

$$\begin{aligned} PROD' \ / \ t_1 &= ((\mathsf{in}?x \to \mathsf{chk}!(x-4) \to PRODI) \\ &\quad ||| \ (\mathsf{chk}!(102) \to PRODR)) \ / \ \langle \mathsf{in}.108, \mathsf{chk}.104, \mathsf{in}.108 \rangle && [3.6.2\ L3] + [1.8.3\ L3A] \\ QUAL \ / \ t_2 &= (QUAL) \ / \ \langle \mathsf{chk}.104 \rangle && [5.5.1\ L8] + [1.8.3\ L3A] \end{aligned}$$

$PROD'$ now does another read from in

$$\begin{aligned} PROD' \ / \ t_1 &= ((\mathsf{chk}!(104) \to PRODI) \\ &\quad ||| \ (\mathsf{chk}!(102) \to PRODR)) \ / \ \langle \mathsf{chk}.104, \mathsf{in}.108 \rangle && [3.6.2\ L3] + [1.8.3\ L3A] \\ QUAL \ / \ t_2 &= (\mathsf{chk}?z \to ((\mathsf{out}!z \to QUAL) \\ &\quad \lhd ok(z) \rhd (\mathsf{rej}!z \to QUAL))) \ / \ \langle \mathsf{chk}.104 \rangle \end{aligned}$$

This is the turning point. Instead of handling the rejected log first we let *PRODI* do chk together with *QUAL*

$$\begin{aligned} PROD' \ / \ t_1 &= (PRODI \\ &\quad ||| \ (\mathsf{chk}!(102) \to PRODR)) \ / \ \langle \mathsf{in}.108 \rangle && [3.6.2\ L3] + [1.8.3\ L3A] \\ QUAL \ / \ t_2 &= ((\mathsf{out}!104 \to QUAL) \\ &\quad \lhd ok(104) \rhd (\mathsf{rej}!104 \to QUAL)) && [1.8.3\ L3A] \end{aligned}$$

Finally $PROD'$ reads another value from in and the system then looks as following

$$\begin{aligned} PROD' \ / \ t_1 &= (\mathsf{chk}!(104) \to PRODI) \ ||| \ (\mathsf{chk}!(102) \to PRODR) && [3.6.2\ L3] + [1.8.3\ L3A] \\ QUAL \ / \ t_2 &= \mathsf{rej}!104 \to QUAL && [5.5.1\ L8] \end{aligned}$$

Clearly we have now reached a deadlock as both of *PRODI* and *PRODR* wants to send on chk, which *QUAL* is unwilling to receive, and *QUAL* wants to send on rej which both *PRODI* and *PRODR* is unwilling to receive.

c)

To show this we first rewrite $PROD''$

$$\begin{aligned} PROD'' &= PRODI \ ||| \ PRODR \\ &= (\mathsf{in}?x \to ((\mathsf{chk}!(x-4) \to PRODI) \ ||| \ PRODR)) \\ &\quad \Box \ (\mathsf{rej}?y \to (PRODI \ ||| \ (\mathsf{chk}!(y-2) \to PRODR))) && [3.6.1\ L6] \end{aligned}$$

We now define a new process $PRODA_x$

$$\begin{aligned} PRODA_x &= (\mathsf{chk}!(x-4) \to PRODI) \mid\mid\mid PRODR \\ &= (\mathsf{chk}!(x-4) \to (PRODI \mid\mid\mid PRODR)) \\ &\quad \Box\ (\mathsf{rej}?y \to ((\mathsf{chk}!(x-4) \to PRODI) \mid\mid\mid (\mathsf{chk}!(y-2) \to PRODR))) && [3.6.1\ L6] \\ &= (\mathsf{chk}!(x-4) \to PROD'') \\ &\quad \Box\ (\mathsf{rej}?y \to ((\mathsf{chk}!(x-4) \to PRODI) \mid\mid\mid (\mathsf{chk}!(y-2) \to PRODR))) \end{aligned}$$

We now define a new process $PRODB_y$

$$\begin{aligned} PRODB_y &= PRODI \mid\mid\mid (\mathsf{chk}!(y-2) \to PRODR) \\ &= (\mathsf{in}?x \to ((\mathsf{chk}!(x-4) \to PRODI) \mid\mid\mid (\mathsf{chk}!(y-2) \to PRODR))) \\ &\quad \Box\ (\mathsf{chk}!(y-2) \to (PRODI \mid\mid\mid PRODR)) && [3.6.1\ L6] \\ &= (\mathsf{in}?x \to ((\mathsf{chk}!(x-4) \to PRODI) \mid\mid\mid (\mathsf{chk}!(y-2) \to PRODR))) \\ &\quad \Box\ (\mathsf{chk}!(y-2) \to PROD'') \end{aligned}$$

We now define a new process $PRODC_{x,y}$

$$\begin{aligned} PRODC_{x,y} &= (\mathsf{chk}!(x-4) \to PRODI) \mid\mid\mid (\mathsf{chk}!(y-2) \to PRODR) \\ &= (\mathsf{chk}!(x-4) \to (PRODI \mid\mid\mid (\mathsf{chk}!(y-2) \to PRODR))) \\ &\quad \Box\ (\mathsf{chk}!(y-2) \to ((\mathsf{chk}!(x-4) \to PRODI) \mid\mid\mid PRODR)) && [3.6.1\ L6] \\ &= (\mathsf{chk}!(x-4) \to PRODB_y \ \Box\ (\mathsf{chk}!(y-2) \to PRODA_x) \end{aligned}$$

Using the definition of each process we then have

$$\begin{aligned} PROD'' &= (\mathsf{in}?x \to PRODA_x \mid \mathsf{rej}?y \to PRODB_y) && [3.3.1\ L5] \\ PRODA_x &= (\mathsf{chk}!(x-4) \to PROD'' \mid \mathsf{rej}?y \to PRODC_{x,y}) && [3.3.1\ L5] \\ PRODB_y &= (\mathsf{in}?x \to PRODC_{x,y} \mid \mathsf{chk}!(y-2) \to PROD'') && [3.3.1\ L5] \\ PRODC_{x,y} &= (\mathsf{chk}!(x-4) \to PRODB_y \ \Box\ (\mathsf{chk}!(y-2) \to PRODA_x) \end{aligned}$$