UNIVERSITY OF COPENHAGEN

XMP: Exam
- Theory

Jens Fredskov (chw752)

January 16, 2014

Problem 1

a)

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

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

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

We now define an auxiliary process $TMP0_x$

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

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

$$
\begin{aligned}
LHS &= PROD \parallel (\mathsf{out}!x \to QUAL) \\
&= (\mathsf{in}?z \to ((\mathsf{chk}!(z-4) \to PROD) \parallel (\mathsf{out}!x \to QUAL)) \\
&\quad \mid \mathsf{out}!x \to (PROD \parallel QUAL)) && [2.3.1\ \text{L7}] \\
&= (\mathsf{in}?z \to \mathsf{out}!x \to ((\mathsf{chk}!(z-4) \to PROD) \parallel QUAL) \\
&\quad \mid \mathsf{out}!x \to \mathsf{in}?z \to ((\mathsf{chk}!(z-4) \to PROD) \parallel QUAL)) && [2.3.1\ \text{L5A/B}] \\
&= (\mathsf{in}?z \to \mathsf{out}!x \to TMP0_{z-4} \mid \mathsf{out}!x \to \mathsf{in}?z \to TMP0_{z-4})
\end{aligned}
$$

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

$$
\begin{aligned}
RHS &= PROD \parallel (\mathsf{rej}!x \to QUAL) \\
&= (\mathsf{in}?z \to ((\mathsf{chk}!(z-4) PROD) \parallel (\mathsf{rej}!x \to QUAL)) \\
&\quad \mid \mathsf{rej}.x \to ((\mathsf{chk}!(x-2) \to PROD) \parallel QUAL)) && [2.3.1\ \text{L7}] \\
&= (\mathsf{in}?z \to STOP \mid \mathsf{rej}.x \to TMP0_{x-2})
\end{aligned}
$$

We then put the side together again

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

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

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

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

$$\begin{aligned}
&= ((\mathsf{in}?z \to \mathsf{out}!x \to TMP1_{z-4} \mid \mathsf{out}!x \to \mathsf{in}?z \to TMP1_{z-4}) \\
&\quad \lhd ok(x) \rhd ((\mathsf{in}?z \to STOP \mid \mathsf{rej}.x \to TMP0_{x-2}) \setminus CR)) && [3.5.1\ L8]
\end{aligned}$$

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

$$\begin{aligned}
&= ((\mathsf{in}?z \to \mathsf{out}!x \to TMP1_{z-4} \mid \mathsf{out}!x \to \mathsf{in}?z \to TMP1_{z-4}) \\
&\quad \lhd ok(x) \rhd (TMP1_{x-2} \sqcap (TMP1_{x-2} \Box (\mathsf{in}?z \to STOP)))) && [3.5.1\ L10]
\end{aligned}$$

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

$$\begin{aligned}
TMP2_x &= TMP1_x \Box (\mathsf{in}?z \to STOP) \\
&= ((\mathsf{in}?z \to \mathsf{out}!x \to TMP1_{z-4} \mid \mathsf{out}!x \to \mathsf{in}?z \to TMP1_{z-4}) \\
&\quad \lhd ok(x) \rhd (TMP1_{x-2} \sqcap (TMP1_{x-2} \Box (\mathsf{in}?z \to STOP)))) \Box (\mathsf{in}?z \to STOP) \\
&= (((\mathsf{in}?z \to \mathsf{out}!x \to TMP1_{z-4} \mid \mathsf{out}!x \to \mathsf{in}?z \to TMP1_{z-4}) \Box (\mathsf{in}?z \to STOP)) \\
&\quad \lhd ok(x) \rhd ((TMP1_{x-2} \sqcap (TMP1_{x-2} \Box (\mathsf{in}?z \to STOP))) \Box (\mathsf{in}?z \to STOP))) && [LCD]
\end{aligned}$$

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

$$\begin{aligned}
LHS &= (\mathsf{in}?z \to \mathsf{out}!x \to TMP1_{z-4} \mid \mathsf{out}!x \to \mathsf{in}?z \to TMP1_{z-4}) \Box (\mathsf{in}?z \to STOP) \\
&= (\mathsf{in}?z \to ((\mathsf{out}!x \to TMP1_{z-4}) \sqcap STOP) \mid \mathsf{out}!x \to \mathsf{in}?z \to TMP1_{z-4}) && [3.3.1\ L5]
\end{aligned}$$

And the right hand side becomes

$$\begin{aligned}
RHS &= (TMP1_{x-2} \sqcap (TMP1_{x-2} \Box (\mathsf{in}?z \to STOP))) \Box (\mathsf{in}?z \to STOP) \\
&= (TMP1_{x-2} \Box (\mathsf{in}?z \to STOP)) \sqcap ((TMP1_{x-2} \Box (\mathsf{in}?z \to STOP)) \Box (\mathsf{in}?z \to STOP)) && [3.3.1\ L6] \\
&= TMP2_{x-2} \sqcap TMP2_{x-2} && [3.3.1\ L1\text{-}L3] \\
&= TMP2_{x-2} && [3.2.1\ L1]
\end{aligned}$$

We then put the side together again

$$\begin{aligned}
TMP2_x &= ((\mathsf{in}?z \to ((\mathsf{out}!x \to TMP1_{z-4}) \sqcap STOP) \mid \mathsf{out}!x \to \mathsf{in}?z \to TMP1_{z-4}) \\
&\quad \lhd ok(x) \rhd TMP2_{x-2})
\end{aligned}$$

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

$$\begin{aligned}
TMP1_x &= ((\mathsf{in}?z \to \mathsf{out}!x \to TMP1_{z-4} \mid \mathsf{out}!x \to \mathsf{in}?z \to TMP1_{z-4}) \\
&\quad \lhd ok(x) \rhd (TMP1_{x-2} \sqcap TMP2_{x-2}))
\end{aligned}$$

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

$$\begin{aligned}
MILL &= (PROD \parallel QUAL) \setminus CR \\
&= (\mathsf{in}?x \to ((\mathsf{chk}!(x-4) \to PROD) \parallel QUAL)) \setminus CR && [2.3.1\ L5A] \\
&= \mathsf{in}?x \to (TMP0_{x-4} \setminus CR) && [3.5.1\ L5] \\
&= \mathsf{in}?x \to TMP1_{x-4}
\end{aligned}$$

b)

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

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

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 $MILL \ / \ s = STOP$.

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

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 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 we 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

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

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

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

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 $s \in traces(PROD') \land s \notin traces(PROD)$. Consider e.g. $s = \langle \mathsf{in}.x, \mathsf{rej}.y\rangle$. We first try to produce the trace in $PROD$

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

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

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)

c)

Problem 3

a)

b)

We consider a trace where we read the 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 have be in a state of deadlock.

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

PROD′ first read from in

$$PROD' \;/\; t_1 = ((\mathsf{chk!}(104) \rightarrow PRODI) \\ \quad ||| \; (\mathsf{rej?}y \rightarrow \mathsf{chk!}(y-2) \rightarrow PRODR)) \;/\; t_1' \qquad [3.6.2\ \mathrm{L3}] + [1.8.3\ \mathrm{L3A}]$$

$$QUAL \;/\; t_2 = (\mathsf{chk?}z \rightarrow ((\mathsf{out!}z \rightarrow QUAL) \\ \quad \lhd\, ok(z) \,\rhd\, (\mathsf{rej!}z \rightarrow QUAL))) \;/\; t_2$$

Both are now willing to do chk

$$PROD' \;/\; t_1 = (PRODI \\ \quad ||| \; (\mathsf{rej?}y \rightarrow \mathsf{chk!}(y-2) \rightarrow PRODR)) \;/\; t_1'' \qquad [3.6.2\ \mathrm{L3}] + [1.8.3\ \mathrm{L3A}]$$

$$QUAL \;/\; t_2 = ((\mathsf{out!}104 \rightarrow QUAL) \\ \quad \lhd\, ok(104) \,\rhd\, (\mathsf{rej!}104 \rightarrow QUAL)) \;/\; t_2' \qquad [1.8.3\ \mathrm{L3A}]$$

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

$$PROD' \;/\; t_1 = ((\mathsf{in?}x \rightarrow \mathsf{chk!}(x-4) \rightarrow PRODI) \\ \quad ||| \; (\mathsf{chk!}(102) \rightarrow PRODR)) \;/\; \langle \mathsf{in.108}, \mathsf{chk.104}, \mathsf{in.108} \rangle \qquad [3.6.2\ \mathrm{L3}] + [1.8.3\ \mathrm{L3A}]$$

$$QUAL \;/\; t_2 = (QUAL) \;/\; \langle \mathsf{chk.104} \rangle \qquad [5.5.1\ \mathrm{L8}] + [1.8.3\ \mathrm{L3A}]$$

PROD′ now does another read from in

$$PROD' \;/\; t_1 = ((\mathsf{chk!}(104) \rightarrow PRODI) \\ \quad ||| \; (\mathsf{chk!}(102) \rightarrow PRODR)) \;/\; \langle \mathsf{chk.104}, \mathsf{in.108} \rangle \qquad [3.6.2\ \mathrm{L3}] + [1.8.3\ \mathrm{L3A}]$$

$$QUAL \;/\; t_2 = (\mathsf{chk?}z \rightarrow ((\mathsf{out!}z \rightarrow QUAL) \\ \quad \lhd\, ok(z) \,\rhd\, (\mathsf{rej!}z \rightarrow QUAL))) \;/\; \langle \mathsf{chk.104} \rangle$$

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

$$PROD' \;/\; t_1 = (PRODI \\ \quad ||| \; (\mathsf{chk!}(102) \rightarrow PRODR)) \;/\; \langle \mathsf{in.108} \rangle \qquad [3.6.2\ \mathrm{L3}] + [1.8.3\ \mathrm{L3A}]$$

$$QUAL \;/\; t_2 = ((\mathsf{out!}104 \rightarrow QUAL) \\ \quad \lhd\, ok(104) \,\rhd\, (\mathsf{rej!}104 \rightarrow QUAL)) \qquad [1.8.3\ \mathrm{L3A}]$$

Finally *PROD′* reads another value from in and the system then looks as following

$$PROD' \;/\; t_1 = (\mathsf{chk!}(104) \rightarrow PRODI) \;|||\; (\mathsf{chk!}(102) \rightarrow PRODR) \qquad [3.6.2\ \mathrm{L3}] + [1.8.3\ \mathrm{L3A}]$$

$$QUAL \;/\; t_2 = \mathsf{rej!}104 \rightarrow QUAL \qquad [5.5.1\ \mathrm{L8}]$$

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)