Software Modeling and Verification
Lehrstuhl für Informatik 2
RWTH Aachen University

Prof. Dr. Ir. J.-P. Katoen
Priv.-Doz. Dr. T. Noll
T. Han, M. Neuhäußer

Modeling Concurrent and Probabilistic Systems

Winter Term 07/08

– Solution 2 –

Exercise 1 **(3 points)**

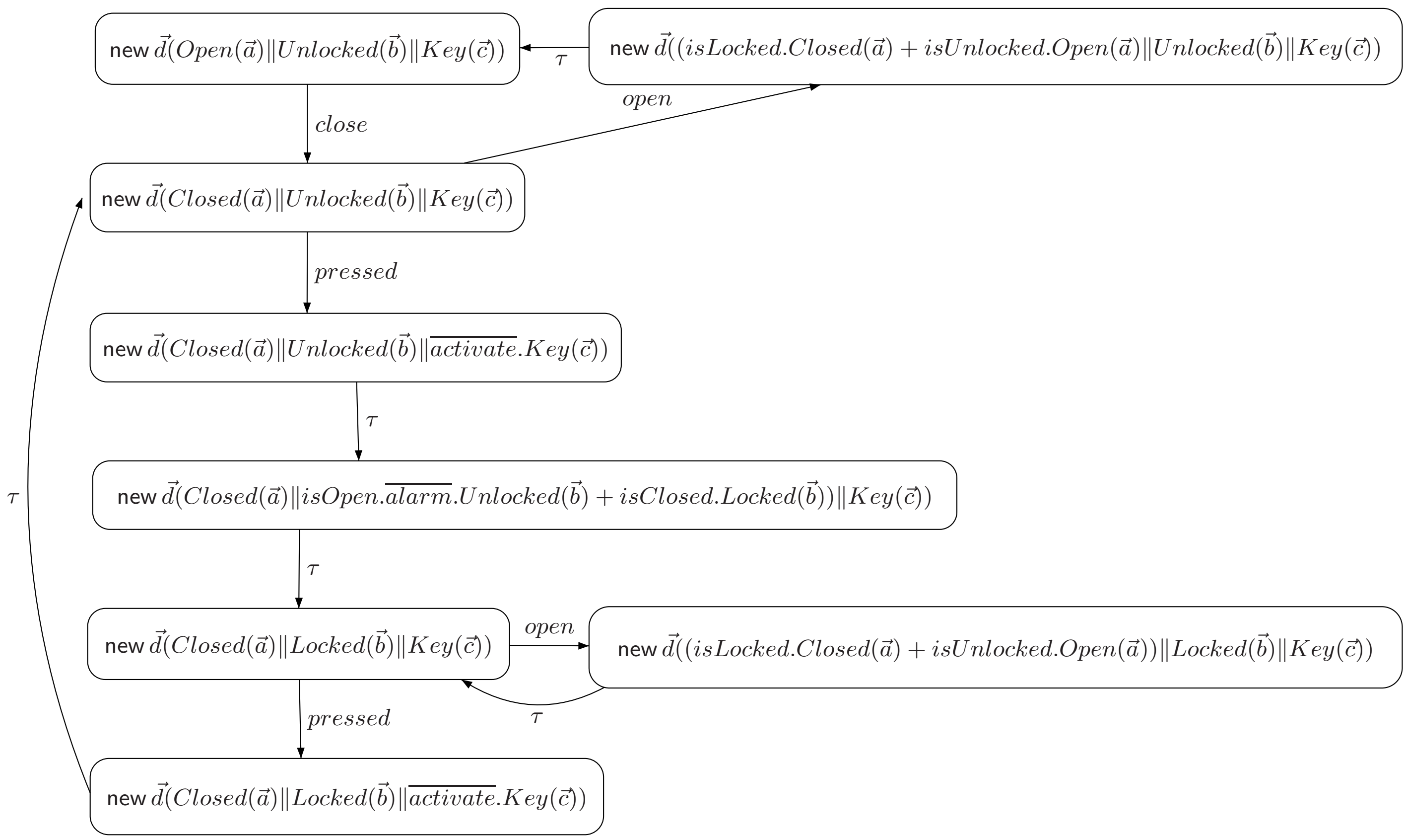
$$\begin{aligned}
Door(\vec{a}) &= Open(\vec{a})\\
Open(\vec{a}) &= \overline{isOpen}.Open(\vec{a}) + close.Closed(\vec{a})\\
Closed(\vec{a}) &= \overline{isClosed}.Closed(\vec{a}) + open.(isLocked.Closed(\vec{a}) + isUnlocked.Open(\vec{a}))
\end{aligned}$$

$$\begin{aligned}
Locker(\vec{b}) &= Unlocked(\vec{b})\\
Unlocked(\vec{b}) &= \overline{isUnlocked}.Unlocked(\vec{b}) + activate.(isOpen.\overline{alarm}.Unlocked(\vec{b}) + isClosed.Locked(\vec{b}))\\
Locked(\vec{b}) &= \overline{isLocked}.Locked(\vec{b}) + activate.Unlocked(\vec{b})
\end{aligned}$$

$$Key(\vec{c}) = pressed.\overline{activate}.Key(\vec{c})$$

$$System(\vec{e}) = \mathsf{new}\, activate, isOpen, isClosed, isUnlocked, isLocked\,(Door(\vec{a})\|Locker(\vec{b})\|Key(\vec{c}))$$

Here, we do not outline the entire labelled transition system but only a subset that shows the essential idea of the above process definition. To shorten notation, let $\vec{d} = (activate, isOpen, isClosed, isUnlocked, isLocked)$.



Exercise 2 (1 + 3 points)

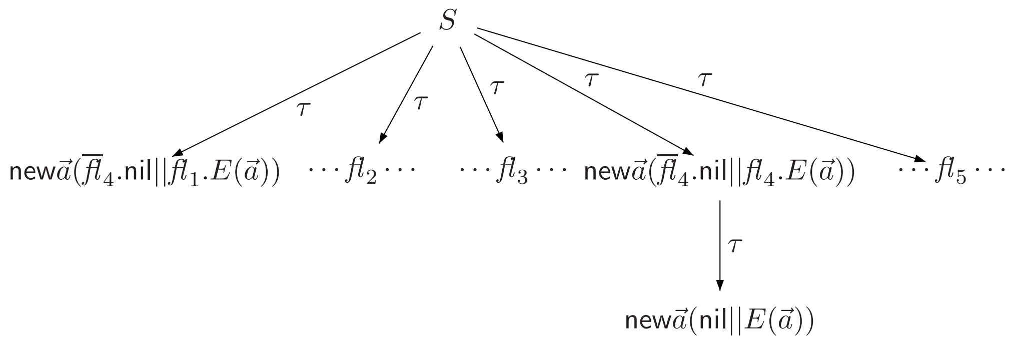
To shorten notation, let $\vec{a} = (req, fl_1, \ldots, fl_5)$.

a) Let $E(\vec{a}) = req.fl_1.E(\vec{a}) + \ldots + req.fl_5.E(\vec{a})$ and $E' = req.(fl_1.E'(\vec{a}) + \ldots + fl_5.E'(\vec{a}))$.
$\Rightarrow Tr(E) = [req.(fl_1 + \ldots + fl_5)]^*.(req + \varepsilon) = Tr(E')$
$\Rightarrow E$ and E' are trace equivalent.

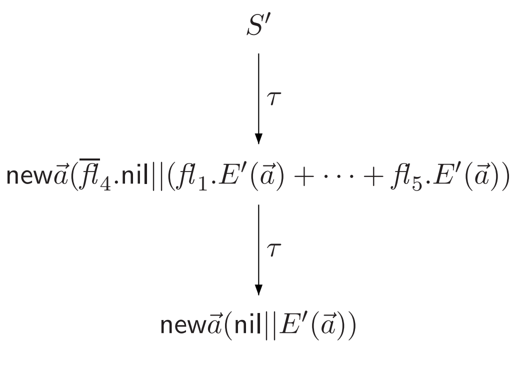
b) To test the elevator specification against a user who wants to go to the fourth floor, we compose the elevator specification and the user in parallel:

$$U(\vec{a}) = \overline{req}.\overline{fl_4}.\mathsf{nil}$$
$$S = \mathsf{new}\, req, fl_1, \ldots, fl_5(U \| E)$$
$$S' = \mathsf{new}\, req, fl_1, \ldots, fl_5(U \| E')$$

In its LTS, process S has five different outgoing τ-transitions:



Formally, $S \xrightarrow{\tau} \mathsf{new}\,\vec{a}[(\overline{fl_4}.\mathsf{nil})\|(fl_i.E(\vec{a})]$ for $1 \leq i \leq 5$. Four of these transitions (those where $i \neq 4$) exhibit τ deadlocks. The LTS of process S' has only one outgoing transition, which does not cause deadlocks. The choice is delayed such that the corresponding communication can take place:



Formally, $S' \xrightarrow{\tau} \mathsf{new}\,\vec{a}[\overline{fl_4}.\mathsf{nil}\|fl_1.E'(\vec{a}) + \ldots + fl_5.E'(\vec{a})] \xrightarrow{\tau} \mathsf{new}\,\vec{a}(\mathsf{nil}||E')$

Thus S' guarantees that the user will reach the fourth floor, whereas S does not.