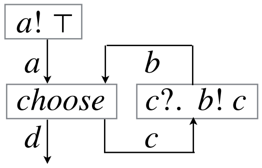
525 (choose) The following picture shows a network of communicating processes.



The formal description of this network is

$$\textbf{new}\ a, b, c?\ bin\cdot\ a!\ \top\ \|\ choose\ \|\ (c?.\ b!\ c)$$

Formally define $choose$, add transit time, and state the output message and time if

(a) $choose$ either reads from a and then outputs $\top$ on c and d, or reads from b and then outputs $\bot$ on c and d. The choice is made freely.

(b) as in part (a), $choose$ either reads from a and then outputs $\top$ on c and d, or reads from b and then outputs $\bot$ on c and d. But this time the choice is not made freely; $choose$ reads from the channel whose input is available first (if there's a tie, then take either one).

After trying the question, scroll down to the solution.

(a) *choose* either reads from a and then outputs $\top$ on c and d, or reads from b and then outputs $\bot$ on c and d. The choice is made freely.

§ We define *choose* as follows:

$$choose \;=\; (a?.\ (c!\top \,\|\, d!\top)) \vee (b?.\ (c!\bot \,\|\, d!\bot))$$

Now we calculate.

$$\textbf{new}\ a, b, c?\ bin\cdot\ a!\top \;\|\; choose \;\|\; (c?.\ b!\,c)$$

$= \exists \mathcal{M}a, \mathcal{T}a, \mathbf{r}a, \mathbf{r}a', \mathbf{w}a, \mathbf{w}a', \mathcal{M}b, \mathcal{T}b, \mathbf{r}b, \mathbf{r}b', \mathbf{w}b, \mathbf{w}b', \mathcal{M}c, \mathcal{T}c, \mathbf{r}c, \mathbf{r}c', \mathbf{w}c, \mathbf{w}c' \cdot$
$\quad \mathbf{r}a{=}\mathbf{w}a{=}\mathbf{r}b{=}\mathbf{w}b{=}\mathbf{r}c{=}\mathbf{w}c{=}0$
$\quad \wedge\ (\mathcal{M}a_{\mathbf{w}a}{=}\top \wedge \mathcal{T}a_{\mathbf{w}a}{=}t \wedge (\mathbf{w}a{:=}\ \mathbf{w}a{+}1)$
$\quad \|\qquad (\ \ t{:=}\ t\uparrow(\mathcal{T}a_{\mathbf{r}a} + 1).\ \ \mathbf{r}a{:=}\ \mathbf{r}a{+}1.$
$\qquad\qquad (\ \mathcal{M}c_{\mathbf{w}c}{=}\top \wedge \mathcal{T}c_{\mathbf{w}c}{=}t \wedge (\mathbf{w}c{:=}\ \mathbf{w}c{+}1)$
$\qquad\qquad \|\ \mathcal{M}d_{\mathbf{w}d}{=}\top \wedge \mathcal{T}d_{\mathbf{w}d}{=}t \wedge (\mathbf{w}d{:=}\ \mathbf{w}d{+}1)\ )\ )$
$\qquad \vee\ \ (\ \ t{:=}\ t\uparrow(\mathcal{T}b_{\mathbf{r}b} + 1).\ \ \mathbf{r}b{:=}\ \mathbf{r}b{+}1.$
$\qquad\qquad (\ \mathcal{M}c_{\mathbf{w}c}{=}\bot \wedge \mathcal{T}c_{\mathbf{w}c}{=}t \wedge (\mathbf{w}c{:=}\ \mathbf{w}c{+}1)$
$\qquad\qquad \|\ \mathcal{M}d_{\mathbf{w}d}{=}\bot \wedge \mathcal{T}d_{\mathbf{w}d}{=}t \wedge (\mathbf{w}d{:=}\ \mathbf{w}d{+}1)\ )\ )$
$\quad \|\ \ (\ \ \ t{:=}\ t\uparrow(\mathcal{T}c_{\mathbf{r}c} + 1).\ \ \mathbf{r}c{:=}\ \mathbf{r}c{+}1.$
$\qquad\quad \mathcal{M}b_{\mathbf{w}b}{=}\mathcal{M}c_{\mathbf{r}c-1} \wedge \mathcal{T}b_{\mathbf{w}b}{=}t \wedge (\mathbf{w}b{:=}\ \mathbf{w}b{+}1)\ )\ )$

Except for time, all processes in concurrent compositions change different variables, so $\|$ is easily replaced by conjunction. Also, make all substitutions indicated by assignments.

$= \exists \mathcal{M}a, \mathcal{T}a, \mathbf{r}a, \mathbf{r}a', \mathbf{w}a, \mathbf{w}a', \mathcal{M}b, \mathcal{T}b, \mathbf{r}b, \mathbf{r}b', \mathbf{w}b, \mathbf{w}b', \mathcal{M}c, \mathcal{T}c, \mathbf{r}c, \mathbf{r}c', \mathbf{w}c, \mathbf{w}c' \cdot$
$\quad \mathbf{r}a{=}\mathbf{w}a{=}\mathbf{r}b{=}\mathbf{w}b{=}\mathbf{r}c{=}\mathbf{w}c{=}0$
$\quad \wedge\ \exists ta, tc, tb\cdot$
$\qquad ta{=}\mathcal{T}a_0{=}t \wedge \mathcal{M}a_0{=}\top \wedge \mathbf{w}a'{=}1$
$\qquad \wedge\ (\ \ \ tc = \mathcal{T}c_0 = \mathcal{T}d_{\mathbf{w}d} = \mathcal{T}a_0 + 1 \wedge \mathbf{r}a'{=}\mathbf{w}c'{=}1 \wedge \mathcal{M}c_0{=}\mathcal{M}d_{\mathbf{w}d}{=}\top \wedge \mathbf{w}d'{=}\mathbf{w}d{+}1$
$\qquad\quad \vee\ \ tc = \mathcal{T}c_0 = \mathcal{T}d_{\mathbf{w}d} = \mathcal{T}b_0{+}1 \wedge \mathbf{r}b'{=}\mathbf{w}c'{=}1 \wedge \mathcal{M}c_0{=}\mathcal{M}d_{\mathbf{w}d}{=}\bot \wedge \mathbf{w}d'{=}\mathbf{w}d{+}1)$
$\qquad \wedge\ \ tb = \mathcal{T}b_0 = \mathcal{T}c_0 + 1\ \ \wedge\ \ \mathbf{r}c'{=}\mathbf{w}b'{=}1\ \ \wedge\ \ \mathcal{M}b_0{=}\mathcal{M}c_0$
$\qquad \wedge\ \ t' = ta\uparrow tc\uparrow tb$ — use One-Point laws to eliminate most quantifiers

$= \exists tc, tb\cdot$
$\qquad (\ \ \ tc = \mathcal{T}d_{\mathbf{w}d} = t{+}1\ \ \wedge\ \ \mathcal{M}d_{\mathbf{w}d}{=}\top\ \ \wedge\ \ \mathbf{w}d'{=}\mathbf{w}d{+}1$
$\qquad \vee\ \ tc = \mathcal{T}d_{\mathbf{w}d} = tb + 1\ \ \wedge\ \ \mathcal{M}d_{\mathbf{w}d}{=}\bot\ \ \wedge\ \ \mathbf{w}d'{=}\mathbf{w}d{+}1\ )$
$\quad \wedge\ \ tb = tc{+}1$
$\quad \wedge\ \ t' = t\uparrow tc\uparrow tb$ — move the conjunctions into the disjunction

$= \exists tc, tb\cdot$
$\qquad tc = \mathcal{T}d_{\mathbf{w}d} = t{+}1\ \ \wedge\ \ \mathcal{M}d_{\mathbf{w}d}{=}\top\ \ \wedge\ \ \mathbf{w}d'{=}\mathbf{w}d{+}1\ \ \wedge\ \ tb = tc{+}1\ \ \wedge\ \ t' = t\uparrow tc\uparrow tb$
$\quad \vee\ \ tc = \mathcal{T}d_{\mathbf{w}d} = tb{+}1\ \ \wedge\ \ \mathcal{M}d_{\mathbf{w}d}{=}\top\ \ \wedge\ \ \mathbf{w}d'{=}\mathbf{w}d{+}1\ \ \wedge\ \ tb = tc{+}1\ \ \wedge\ \ t' = t\uparrow tc\uparrow tb$

now we can eliminate tc and tb in each disjunct separately

$= \qquad \mathcal{T}d_{\mathbf{w}d} = t{+}1\ \ \wedge\ \ \mathcal{M}d_{\mathbf{w}d}{=}\top\ \ \wedge\ \ \mathbf{w}d'{=}\mathbf{w}d{+}1\ \ \wedge\ \ t' = t{+}2$
$\quad \vee\ \ \mathcal{T}d_{\mathbf{w}d} = \infty\ \ \wedge\ \ \mathcal{M}d_{\mathbf{w}d}{=}\bot\ \ \wedge\ \ \mathbf{w}d'{=}\mathbf{w}d{+}1\ \ \wedge\ \ t'{=}\infty$

$= (t{:=}\ t{+}1.\ \ d!\top.\ \ t{:=}\ t{+}1) \vee (t{:=}\ \infty.\ \ d!\bot)$

Either a $\top$ is output after time 1 or nothing ever happens. There is probably a better way to do this question by using laws of programs and not translating to ordinary logic.

(b) as in part (a), *choose* either reads from a and then outputs $\top$ on c and d, or reads from b and then outputs $\bot$ on c and d. But this time the choice is not made freely; *choose* reads from the channel whose input is available first (if there's a tie, then take either one).

§ There is a slight ambiguity in the question. It says “the channel whose input is available first”. Does this mean the channel whose input arrived first? Or, if two inputs have already arrived, no matter which arrived first, they are both available now (at the same time)? I'll do it both ways. Formalizing makes the meaning clear.

Suppose “available first” means “arrived first”. We define *choose* as follows:

$$\begin{array}{lll} choose & = & \mathcal{T}a_{ra}\le\mathcal{T}b_{rb} \wedge (a?.\ \ (c!\ \top \parallel d!\ \top)) \\ & & \vee\ \ \mathcal{T}b_{rb}\le\mathcal{T}a_{ra} \wedge (b?.\ \ (c!\ \bot \parallel d!\ \bot)) \end{array}$$

Now we calculate.

$$\mathbf{new}\ a, b, c?\ bin\cdot\ \ a!\ \top\ \parallel\ choose\ \parallel\ (c?.\ \ b!\ c)$$

$$\begin{array}{ll}
= & \exists \mathcal{M}a, \mathcal{T}a, \boldsymbol{r}a, \boldsymbol{r}a', \boldsymbol{w}a, \boldsymbol{w}a', \mathcal{M}b, \mathcal{T}b, \boldsymbol{r}b, \boldsymbol{r}b', \boldsymbol{w}b, \boldsymbol{w}b', \mathcal{M}c, \mathcal{T}c, \boldsymbol{r}c, \boldsymbol{r}c', \boldsymbol{w}c, \boldsymbol{w}c'\cdot \\
& \quad \boldsymbol{r}a=\boldsymbol{w}a=\boldsymbol{r}b=\boldsymbol{w}b=\boldsymbol{r}c=\boldsymbol{w}c=0 \\
& \wedge\ (\quad \mathcal{M}a_{\boldsymbol{w}a}=\top \wedge \mathcal{T}a_{\boldsymbol{w}a}=t \wedge (\boldsymbol{w}a{:=}\ \boldsymbol{w}a+1) \\
* & \quad \parallel\ \ (\ \ (\qquad \mathcal{T}a_{\boldsymbol{r}a}\le\mathcal{T}b_{\boldsymbol{r}b} \\
& \qquad\qquad \wedge\ \ (t{:=}\ t\uparrow(\mathcal{T}a_{\boldsymbol{r}a}+1).\ \ \boldsymbol{r}a{:=}\ \boldsymbol{r}a+1. \\
& \qquad\qquad\quad (\ \mathcal{M}c_{\boldsymbol{w}c}=\top \wedge \mathcal{T}c_{\boldsymbol{w}c}=t \wedge (\boldsymbol{w}c{:=}\ \boldsymbol{w}c+1) \\
& \qquad\qquad\quad \parallel \mathcal{M}d_{\boldsymbol{w}d}=\top \wedge \mathcal{T}d_{\boldsymbol{w}d}=t \wedge (\boldsymbol{w}d{:=}\ \boldsymbol{w}d+1)\ )\ )\ ) \\
* & \qquad\quad \vee\ \ (\qquad \mathcal{T}b_{\boldsymbol{r}b}\le\mathcal{T}a_{\boldsymbol{r}a} \\
& \qquad\qquad \wedge\ \ (t{:=}\ t\uparrow(\mathcal{T}b_{\boldsymbol{r}b}+1).\ \ \boldsymbol{r}b{:=}\ \boldsymbol{r}b+1. \\
& \qquad\qquad\quad (\ \mathcal{M}c_{\boldsymbol{w}c}=\bot \wedge \mathcal{T}c_{\boldsymbol{w}c}=t \wedge (\boldsymbol{w}c{:=}\ \boldsymbol{w}c+1) \\
& \qquad\qquad\quad \parallel \mathcal{M}d_{\boldsymbol{w}d}=\bot \wedge \mathcal{T}d_{\boldsymbol{w}d}=t \wedge (\boldsymbol{w}d{:=}\ \boldsymbol{w}d+1)\ )\ )\ )\ ) \\
& \quad \parallel\ \ (\ \ t{:=}\ t\uparrow(\mathcal{T}c_{\boldsymbol{r}c}+1).\ \ \boldsymbol{r}c{:=}\ \boldsymbol{r}c+1. \\
& \qquad\quad \mathcal{M}b_{\boldsymbol{w}b}=\mathcal{M}c_{\boldsymbol{r}c-1} \wedge \mathcal{T}b_{\boldsymbol{w}b}=t \wedge (\boldsymbol{w}b{:=}\ \boldsymbol{w}b+1)\ )\ )
\end{array}$$

Except for time, all processes in concurrent compositions change different variables, so $\parallel$ is easily replaced by conjunction. Also, make all substitutions indicated by assignments.

$$\begin{array}{ll}
= & \exists \mathcal{M}a, \mathcal{T}a, \boldsymbol{r}a, \boldsymbol{r}a', \boldsymbol{w}a, \boldsymbol{w}a', \mathcal{M}b, \mathcal{T}b, \boldsymbol{r}b, \boldsymbol{r}b', \boldsymbol{w}b, \boldsymbol{w}b', \mathcal{M}c, \mathcal{T}c, \boldsymbol{r}c, \boldsymbol{r}c', \boldsymbol{w}c, \boldsymbol{w}c'\cdot \\
& \quad \boldsymbol{r}a=\boldsymbol{w}a=\boldsymbol{r}b=\boldsymbol{w}b=\boldsymbol{r}c=\boldsymbol{w}c=0 \\
& \wedge\ \ \exists ta, tc, tb\cdot \\
& \qquad ta=\mathcal{T}a_0=t \wedge \mathcal{M}a_0=\top \wedge \boldsymbol{w}a'=1 \\
* & \quad \wedge\ \ (\qquad \mathcal{T}a_0\le\mathcal{T}b_0 \\
& \qquad\quad \wedge\ \ tc=\mathcal{T}c_0=\mathcal{T}d_{\boldsymbol{w}d}=\mathcal{T}a_0+1 \wedge \boldsymbol{r}a'=\boldsymbol{w}c'=1 \wedge \mathcal{M}c_0=\mathcal{M}d_{\boldsymbol{w}d}=\top \wedge \boldsymbol{w}d'=\boldsymbol{w}d+1 \\
* & \qquad \vee\qquad \mathcal{T}b_0\le\mathcal{T}a_0 \\
& \qquad\quad \wedge\ \ tc=\mathcal{T}c_0=\mathcal{T}d_{\boldsymbol{w}d}=\mathcal{T}b_0+1 \wedge \boldsymbol{r}b'=\boldsymbol{w}c'=1 \wedge \mathcal{M}c_0=\mathcal{M}d_{\boldsymbol{w}d}=\bot \wedge \boldsymbol{w}d'=\boldsymbol{w}d+1) \\
& \quad \wedge\ \ tb=\mathcal{T}b_0=\mathcal{T}c_0+1\ \ \wedge\ \ \boldsymbol{r}c'=\boldsymbol{w}b'=1\ \ \wedge\ \ \mathcal{M}b_0=\mathcal{M}c_0 \\
& \quad \wedge\ \ t'=ta\uparrow tc\uparrow tb
\end{array}$$

use the One-Point laws to eliminate most quantifiers

$$\begin{array}{ll}
= & \exists tc, tb\cdot \\
& \quad (\quad t\le tb\ \ \wedge\ \ tc=\mathcal{T}d_{\boldsymbol{w}d}=t+1\ \ \wedge\ \ \mathcal{M}d_{\boldsymbol{w}d}=\top\ \ \wedge\ \ \boldsymbol{w}d'=\boldsymbol{w}d+1 \\
& \quad \vee\ \ tb\le t\ \ \wedge\ \ tc=\mathcal{T}d_{\boldsymbol{w}d}=tb+1\ \ \wedge\ \ \mathcal{M}d_{\boldsymbol{w}d}=\bot\ \ \wedge\ \ \boldsymbol{w}d'=\boldsymbol{w}d+1\ ) \\
& \wedge\ \ tb=tc+1 \\
& \wedge\ \ t'=t\uparrow tc\uparrow tb
\end{array}$$

move the conjunctions into the disjunction

$$\begin{array}{ll}
= & \exists tc, tc\cdot \\
& \quad tc=\mathcal{T}d_{\boldsymbol{w}d}=t+1\ \ \wedge\ \ \mathcal{M}d_{\boldsymbol{w}d}=\top\ \ \wedge\ \ \boldsymbol{w}d'=\boldsymbol{w}d+1\ \ \wedge tb=tc+1 \wedge t'=t\uparrow tc\uparrow tb \\
& \vee\ \ tb\le t\ \ \wedge\ \ tc=\mathcal{T}d_{\boldsymbol{w}d}=tb+1\ \ \wedge\ \ \mathcal{M}d_{\boldsymbol{w}d}=\bot \wedge \boldsymbol{w}d'=\boldsymbol{w}d+1 \wedge tb=tc+1 \wedge t'=t\uparrow tc\uparrow tb
\end{array}$$

now we can eliminate tc and tb in each disjunct separately

$$\begin{array}{ll}
= & \qquad \mathcal{T}d_{\boldsymbol{w}d}=t+1\ \ \wedge\ \ \mathcal{M}d_{\boldsymbol{w}d}=\top\ \ \wedge\ \ \boldsymbol{w}d'=\boldsymbol{w}d+1\ \ \wedge\ \ t'=t+2 \\
& \vee\ \ \infty\le t\ \ \wedge\ \ \mathcal{T}d_{\boldsymbol{w}d}=\infty\ \ \wedge\ \ \mathcal{M}d_{\boldsymbol{w}d}=\bot\ \ \wedge\ \ \boldsymbol{w}d'=\boldsymbol{w}d+1\ \ \wedge\ \ t'=\infty \\
= & (t{:=}\ t+1.\ \ d!\ \top.\ \ t{:=}\ t+1)\ \vee\ \ t=t'=\infty \wedge (d!\ \bot)
\end{array}$$

If the computation starts before time ∞ the output is definitely $\top$ after time 1. Again, there is probably a better way to do this question by using laws of programs and not translating to ordinary logic.

Suppose “available first” means “if it's available now, it doesn't matter when it arrived”. We define *choose* as follows:

$$\begin{array}{lll} choose & = & (\surd a \ \lor\ \mathcal{T}a_{ra}{\le}\mathcal{T}b_{rb}) \land (a?.\ \ (c!\ \top \parallel d!\ \top)) \\ & & \lor\ (\surd b \ \lor\ \mathcal{T}b_{rb}{\le}\mathcal{T}a_{ra}) \land (b?.\ \ (c!\ \bot \parallel d!\ \bot)) \end{array}$$

Now we calculate.

$$\mathbf{new}\ a, b, c?\ bin\cdot\ \ a!\ \top\ \parallel\ choose\ \parallel\ (c?.\ \ b!\ c)$$

$$\begin{array}{ll}
= & \exists \mathcal{M}a, \mathcal{T}a, \boldsymbol{ra}, \boldsymbol{ra}', \boldsymbol{wa}, \boldsymbol{wa}', \mathcal{M}b, \mathcal{T}b, \boldsymbol{rb}, \boldsymbol{rb}', \boldsymbol{wb}, \boldsymbol{wb}', \mathcal{M}c, \mathcal{T}c, \boldsymbol{rc}, \boldsymbol{rc}', \boldsymbol{wc}, \boldsymbol{wc}'\cdot \\
& \quad \boldsymbol{ra}{=}\boldsymbol{wa}{=}\boldsymbol{rb}{=}\boldsymbol{wb}{=}\boldsymbol{rc}{=}\boldsymbol{wc}{=}0 \\
& \land\ (\quad \mathcal{M}a_{\boldsymbol{wa}}{=}\top \land \mathcal{T}a_{\boldsymbol{wa}}{=}t \land (\boldsymbol{wa}{:=}\ \boldsymbol{wa}{+}1) \\
& \quad \parallel\ (\ \ (\qquad (\mathcal{T}a_{\boldsymbol{ra}}{\le}t \lor \mathcal{T}a_{\boldsymbol{ra}}{\le}\mathcal{T}b_{\boldsymbol{rb}}) \\
& \qquad\qquad\quad \land\ \ (t{:=}\ t{\uparrow}(\mathcal{T}a_{\boldsymbol{ra}} + 1).\ \ \boldsymbol{ra}{:=}\ \boldsymbol{ra}{+}1. \\
& \qquad\qquad\qquad (\ \mathcal{M}c_{\boldsymbol{wc}}{=}\top \land \mathcal{T}c_{\boldsymbol{wc}}{=}t \land (\boldsymbol{wc}{:=}\ \boldsymbol{wc}{+}1) \\
& \qquad\qquad\qquad \parallel \mathcal{M}d_{\boldsymbol{wd}}{=}\top \land \mathcal{T}d_{\boldsymbol{wd}}{=}t \land (\boldsymbol{wd}{:=}\ \boldsymbol{wd}{+}1)\ )\ )\ ) \\
& \qquad\quad \lor\ \ (\qquad (\mathcal{T}b_{\boldsymbol{rb}}{\le}t \lor \mathcal{T}b_{\boldsymbol{rb}}{\le}\mathcal{T}a_{\boldsymbol{ra}}) \\
& \qquad\qquad\quad \land\ \ (t{:=}\ t{\uparrow}(\mathcal{T}b_{\boldsymbol{rb}} + 1).\ \ \boldsymbol{rb}{:=}\ \boldsymbol{rb}{+}1. \\
& \qquad\qquad\qquad (\ \mathcal{M}c_{\boldsymbol{wc}}{=}\bot \land \mathcal{T}c_{\boldsymbol{wc}}{=}t \land (\boldsymbol{wc}{:=}\ \boldsymbol{wc}{+}1) \\
& \qquad\qquad\qquad \parallel \mathcal{M}d_{\boldsymbol{wd}}{=}\bot \land \mathcal{T}d_{\boldsymbol{wd}}{=}t \land (\boldsymbol{wd}{:=}\ \boldsymbol{wd}{+}1)\ )\ )\ )\ ) \\
& \quad \parallel\ (\ \ t{:=}\ t{\uparrow}(\mathcal{T}c_{\boldsymbol{rc}} + 1).\ \ \boldsymbol{rc}{:=}\ \boldsymbol{rc}{+}1. \\
& \qquad\quad \mathcal{M}b_{\boldsymbol{wb}}{=}\mathcal{M}c_{\boldsymbol{rc}-1} \land \mathcal{T}b_{\boldsymbol{wb}}{=}t \land (\boldsymbol{wb}{:=}\ \boldsymbol{wb}{+}1)\ )\ )
\end{array}$$

Except for time, all processes in concurrent compositions change different variables, so $\parallel$ is easily replaced by conjunction. Also, make all substitutions indicated by assignments.

$$\begin{array}{ll}
= & \exists \mathcal{M}a, \mathcal{T}a, \boldsymbol{ra}, \boldsymbol{ra}', \boldsymbol{wa}, \boldsymbol{wa}', \mathcal{M}b, \mathcal{T}b, \boldsymbol{rb}, \boldsymbol{rb}', \boldsymbol{wb}, \boldsymbol{wb}', \mathcal{M}c, \mathcal{T}c, \boldsymbol{rc}, \boldsymbol{rc}', \boldsymbol{wc}, \boldsymbol{wc}'\cdot \\
& \quad \boldsymbol{ra}{=}\boldsymbol{wa}{=}\boldsymbol{rb}{=}\boldsymbol{wb}{=}\boldsymbol{rc}{=}\boldsymbol{wc}{=}0 \\
& \land\ \exists ta, tc, tb\cdot \\
& \qquad ta{=}\mathcal{T}a_0{=}t \land \mathcal{M}a_0{=}\top \land \boldsymbol{wa}'{=}1 \\
& \quad \land\ \ (\qquad (\mathcal{T}a_0{\le}t \lor \mathcal{T}a_0{\le}\mathcal{T}b_0) \\
& \qquad\quad \land\ \ tc{=}\mathcal{T}c_0{=}\mathcal{T}d_{\boldsymbol{wd}}{=}\mathcal{T}a_0{+}1 \land \boldsymbol{ra}'{=}\boldsymbol{wc}'{=}1 \land \mathcal{M}c_0{=}\mathcal{M}d_{\boldsymbol{wd}}{=}\top \land \boldsymbol{wd}'{=}\boldsymbol{wd}{+}1 \\
& \qquad \lor \qquad (\mathcal{T}b_0{\le}t \lor \mathcal{T}b_0{\le}\mathcal{T}a_0) \\
& \qquad\quad \land\ \ tc{=}\mathcal{T}c_0{=}\mathcal{T}d_{\boldsymbol{wd}}{=}\mathcal{T}b_0{+}1 \land \boldsymbol{rb}'{=}\boldsymbol{wc}'{=}1 \land \mathcal{M}c_0{=}\mathcal{M}d_{\boldsymbol{wd}}{=}\bot \land \boldsymbol{wd}'{=}\boldsymbol{wd}{+}1) \\
& \quad \land\ \ tb = \mathcal{T}b_0 = \mathcal{T}c_0 + 1\ \ \land\ \ \boldsymbol{rc}'{=}\boldsymbol{wb}'{=}1\ \ \land\ \ \mathcal{M}b_0{=}\mathcal{M}c_0 \\
& \quad \land\ \ t' = ta{\uparrow}tc{\uparrow}tb
\end{array}$$

use the One-Point laws to eliminate most quantifiers

$$\begin{array}{ll}
= & \exists tc, tb\cdot \\
& \quad (\quad (t{\le}tb \lor t{\le}tb)\ \ \land\ \ tc = \mathcal{T}d_{\boldsymbol{wd}} = t{+}1\ \ \land\ \ \mathcal{M}d_{\boldsymbol{wd}}{=}\top\ \ \land\ \ \boldsymbol{wd}'{=}\boldsymbol{wd}{+}1 \\
& \quad \lor\ \ (tb{\le}t \lor tb{\le}t)\ \ \land\ \ tc = \mathcal{T}d_{\boldsymbol{wd}} = tb{+}1\ \ \land\ \ \mathcal{M}d_{\boldsymbol{wd}}{=}\bot\ \ \land\ \ \boldsymbol{wd}'{=}\boldsymbol{wd}{+}1\ ) \\
& \land\ \ tb = tc{+}1 \\
& \land\ \ t' = t{\uparrow}tc{\uparrow}tb
\end{array}$$

simplify the two minor disjunctions and move the conjunctions into the major disjunction

$$\begin{array}{ll}
= & \exists tc, tb\cdot \\
& \quad tc{=}\mathcal{T}d_{\boldsymbol{wd}}{=}t{+}1\ \ \land\ \ \mathcal{M}d_{\boldsymbol{wd}}{=}\top\ \ \land\ \ \boldsymbol{wd}' = \boldsymbol{wd}{+}1\ \ \land tb = tc{+}1 \land t' = t{\uparrow}tc{\uparrow}tb \\
& \lor\ \ tb{\le}t\ \ \land\ \ tc{=}\mathcal{T}d_{\boldsymbol{wd}}{=}tb{+}1\ \ \land\ \ \mathcal{M}d_{\boldsymbol{wd}}{=}\bot \land \boldsymbol{wd}'{=}\boldsymbol{wd}{+}1 \land tb = tc{+}1 \land t' = t{\uparrow}tc{\uparrow}tb
\end{array}$$

now we can eliminate tc and tb in each disjunct separately

$$\begin{array}{ll}
= & \quad \mathcal{T}d_{\boldsymbol{wd}} = t{+}1\ \ \land\ \ \mathcal{M}d_{\boldsymbol{wd}}{=}\top\ \ \land\ \ \boldsymbol{wd}' = \boldsymbol{wd}{+}1\ \ \land\ \ t' = t{+}2 \\
& \lor\ \ \infty{\le}t\ \ \land\ \ \mathcal{T}d_{\boldsymbol{wd}}{=}\infty\ \ \land\ \ \mathcal{M}d_{\boldsymbol{wd}}{=}\bot\ \ \land\ \ \boldsymbol{wd}'{=}\boldsymbol{wd}{+}1\ \ \land\ \ t'{=}\infty \\
= & (t{:=}\ t{+}1.\ \ d!\ \top.\ \ t{:=}\ t{+}1)\ \lor\ t{=}t'{=}\infty \land (d!\ \bot)
\end{array}$$

If the computation starts before time ∞ the output is definitely $\top$ after time 1. This is the same as before, so the ambiguity didn't matter. This is good, because our programming constructs do not require us to keep track of the time messages arrive. Again, there is probably a better way to do this question by using laws of programs and not translating to ordinary logic.