

INTERACTION

var $a: T \cdot S = \exists a, a': T \cdot S$ boundary variable

ivar $x: T \cdot S = \exists x: time \rightarrow T \cdot S$ interactive variable

The value of variable x at time t is $x t$

$$\begin{aligned}
 P \parallel Q &= \exists t_P, t_Q \cdot && \text{(substitute } t_P \text{ for } t' \text{ in } P) \\
 &\wedge \text{(substitute } t_Q \text{ for } t' \text{ in } Q) \\
 &\wedge t' = \max t_P t_Q \\
 &\wedge (\forall t'' \cdot t_P \leq t'' \leq t' \Rightarrow x t'' = x(t_P)) && * \\
 &\wedge (\forall t'' \cdot t_Q \leq t'' \leq t' \Rightarrow y t'' = y(t_Q)) && **
 \end{aligned}$$

* and similarly for each interactive variable of P

** and similarly for each interactive variable of Q

Suppose in some process the boundary variables are a and b and the interactive variables are x and y .

$$ok = a'=a \wedge b'=b \wedge t'=t$$

$$a:=e = a'=e \wedge b'=b \wedge t'=t$$

$$\begin{aligned} x:=e &= a'=a \wedge b'=b \wedge x'=e \wedge (\forall t''. t \leq t'' \leq t' \Rightarrow y''=y) \\ &\wedge t' = t + (\text{the time required to evaluate and store } e) \\ &= a'=a \wedge b'=b \wedge xt'=e \wedge (\forall t''. t \leq t'' \leq t' \Rightarrow yt''=yt) \\ &\wedge t' = t + (\text{the time required to evaluate and store } e) \end{aligned}$$

$P. Q$

$$\begin{aligned} = \exists a'', b'', t''. & \quad (\text{substitute } a'', b'', t'' \text{ for } a', b', t' \text{ in } P) \\ & \wedge (\text{substitute } a'', b'', t'' \text{ for } a, b, t \text{ in } Q) \end{aligned}$$

Most laws still work

but not the Substitution Law

Example in boundary variables a , b ,

interactive variables x , y , and time variable t

$$(x:= 2. \ x:= x+y. \ x:= x+y) \parallel (y:= 3. \ y:= x+y)$$

x is a variable in the left process

and y is a variable in the right process.

Let's put a in the left process and b in the right process.

$$= (a'=a \wedge xt'=2 \wedge t'=t+1. \ a'=a \wedge xt'=xt+yt \wedge t'=t+1.$$

$$a'=a \wedge xt'=xt+yt \wedge t'=t+1)$$

$$\parallel (b'=b \wedge yt'=3 \wedge t'=t+1. \ b'=b \wedge yt'=xt+yt \wedge t'=t+1)$$

$$= (a'=a \wedge x(t+1)=2 \wedge x(t+2)=x(t+1)+y(t+1)$$

$$\wedge x(t+3)=x(t+2)+y(t+2) \wedge t'=t+3)$$

$$\parallel (b'=b \wedge y(t+1)=3 \wedge y(t+2)=x(t+1)+y(t+1) \wedge t'=t+2)$$

$$= x(t+1)=2 \wedge x(t+2)=x(t+1)+y(t+1) \wedge x(t+3)=x(t+2)+y(t+2)$$

$$\wedge y(t+1)=3 \wedge y(t+2)=x(t+1)+y(t+1) \wedge y(t+3)=y(t+2)$$

$$\wedge a'=a \wedge b'=b \wedge t'=t+3$$

$$= x(t+1)=2 \wedge x(t+2)=5 \wedge x(t+3)=10$$

$$\wedge y(t+1)=3 \wedge y(t+2)=y(t+3)=5$$

$$\wedge a'=a \wedge b'=b \wedge t'=t+3$$

Thermostat for a gas burner

thermometer || *control* || *thermostat* || *burner*

inputs to the thermostat:

- real *temperature* , which comes from the thermometer and indicates the actual temperature.
- real *desired* , which comes from the control and indicates the desired temperature.
- boolean *flame* , which comes from a flame sensor in the burner and indicates whether there is a flame.

outputs of the thermostat:

- boolean *gas* ; assigning it \top turns the gas on and \perp turns the gas off.
- boolean *spark* ; assigning it \top causes sparks for the purpose of igniting the gas.

Heat is wanted when the actual temperature falls ε below the desired temperature, and not wanted when the actual temperature rises ε above the desired temperature, where ε is small enough to be unnoticeable, but large enough to prevent rapid oscillation. To obtain heat, the spark should be applied to the gas for at least 1 second to give it a chance to ignite and to allow the flame to become stable. But a safety regulation states that the gas must not remain on and unlit for more than 3 seconds. Another regulation says that when the gas is shut off, it must not be turned on again for at least 20 seconds to allow any accumulated gas to clear. And finally, the gas burner must respond to its inputs within 1 second.

$$\textit{thermostat} = (\textit{gas} := \perp \parallel \textit{spark} := \perp). \textit{GasOff}$$

$$\begin{aligned} \textit{GasOff} = & \textbf{if } \textit{temperature} < \textit{desired} - \varepsilon \\ & \textbf{then } ((\textit{gas} := \top \parallel \textit{spark} := \top \parallel t+1 \leq t' \leq t+3). \\ & \quad \textit{spark} := \perp. \textit{GasOn}) \\ & \textbf{else } (((\textbf{frame } \textit{gas}, \textit{spark} \cdot \textit{ok}) \parallel t < t' \leq t+1). \textit{GasOff}) \end{aligned}$$

$$\begin{aligned} \textit{GasOn} = & \textbf{if } \textit{temperature} < \textit{desired} + \varepsilon \wedge \textit{flame} \\ & \textbf{then } (((\textbf{frame } \textit{gas}, \textit{spark} \cdot \textit{ok}) \parallel t < t' \leq t+1). \textit{GasOn}) \\ & \textbf{else } ((\textit{gas} := \perp \parallel (\textbf{frame } \textit{spark} \cdot \textit{ok}) \parallel t+20 \leq t' \leq t+21). \\ & \quad \textit{GasOff}) \end{aligned}$$

COMMUNICATION

Channel c is described by

| | | |
|----------------|-----------------|---------------------------|
| message script | \mathcal{M}_c | list constant |
| time script | \mathcal{T}_c | list constant |
| read cursor | r_c | extended natural variable |
| write cursor | w_c | extended natural variable |

$$\begin{array}{rcl}
 \mathcal{M} & = & [6 ; 4 ; 7 ; 1 ; 0 ; 3 ; 8 ; 9 ; 2 ; 5 ; \dots] \\
 \mathcal{T} & = & [3 ; 5 ; 5 ; 20 ; 25 ; 28 ; 31 ; 31 ; 45 ; 48 ; \dots] \\
 & & \quad \uparrow \quad \quad \uparrow \\
 & & \quad r \quad \quad w
 \end{array}$$

INPUT AND OUTPUT

$$c! e = \mathcal{M}w = e \wedge \mathcal{T}w = t \wedge (w := w+1)$$

$$c! = \mathcal{T}w = t \wedge (w := w+1)$$

$$c? = r := r+1$$

$$c = \mathcal{M}(r-1)$$

$$\sqrt{c} = \mathcal{T}r \leq t$$

if \sqrt{key}

then ($key?$.

if $key = `y$

then *screen!* "If you wish."

else *screen!* "Not if you don't want.")

else *screen!* "Well?"

Repeatedly input numbers from channel c , and output their doubles on channel d .

$$S = \forall n: \text{nat}. \mathcal{M}_d(w_{d+n}) = 2 \times \mathcal{M}_c(r_{c+n})$$

$$S \Leftarrow c?. d! 2 \times c. S$$

Proof:

$$\begin{aligned} & c?. d! 2 \times c. S \\ = & r_c := r_c + 1. \mathcal{M}_d w_d = 2 \times \mathcal{M}_c(r_c - 1) \wedge (w_d := w_d + 1). S \\ = & \mathcal{M}_d w_d = 2 \times \mathcal{M}_c r_c \wedge \forall n: \text{nat}. \mathcal{M}_d(w_{d+1+n}) = 2 \times \mathcal{M}_c(r_{c+1+n}) \\ = & \forall n: \text{nat}. \mathcal{M}_d(w_{d+n}) = 2 \times \mathcal{M}_c(r_{c+n}) \\ = & S \end{aligned}$$

Communication Timing

real time: need to know implementation

transit time: input and output take time 0

communication transit takes time 1

input $c?$ becomes $t := \max t (\mathcal{T}r_c + 1). c?$

check \sqrt{c} becomes $\mathcal{T}r_c + 1 \leq t$



$W = t := \max t (\mathcal{T}r + 1). c?$

$=$ wait for input on channel c and then read it

$W \Leftarrow$ **if** \sqrt{c} **then** $r := r+1$ **else** $(t := t+1. W)$

Proof:

if \sqrt{c} **then** $r := r+1$ **else** $(t := t+1. W)$

$=$ **if** $\mathcal{T}r+1 \leq t$ **then** $r := r+1$

else $(t := t+1. t := \max t (\mathcal{T}r + 1). r := r+1)$

$=$ **if** $\mathcal{T}r+1 \leq t$ **then** $(t := t. r := r+1)$

else $(t := \max (t+1) (\mathcal{T}r + 1). r := r+1)$

$=$ **if** $\mathcal{T}r+1 \leq t$ **then** $(t := \max t (\mathcal{T}r + 1). r := r+1)$

else $(t := \max t (\mathcal{T}r + 1). r := r+1)$

$= W$

RECURSIVE COMMUNICATION

$$dbl = c?. d! 2 \times c. t := t + 1. dbl$$

The weakest solution:

$$\forall n: nat. \mathcal{M}_d(w_{d+n}) = 2 \times \mathcal{M}_c(r_{c+n}) \wedge \mathcal{T}_d(w_{d+n}) = t+n$$

A strongest implementable solution:

$$\begin{aligned} & (\forall n: nat. \mathcal{M}_d(w_{d+n}) = 2 \times \mathcal{M}_c(r_{c+n}) \wedge \mathcal{T}_d(w_{d+n}) = t+n) \\ & \wedge r'_c = w'_d = t' = \infty \wedge w'_c = w_c \wedge r'_d = r_d \end{aligned}$$

The strongest solution:

⊥

$$\forall n: nat. \mathcal{M}_d(w_{d+n}) = 2 \times \mathcal{M}_c(r_{c+n}) \wedge \mathcal{T}_d(w_{d+n}) = t+n$$

$$\Leftarrow dbl$$

$$dbl \Leftarrow c?. d! 2 \times c. t := t + 1. dbl$$

Recursive Construction

$$dbl_0 = \top$$

$$dbl_1 = c?. d! 2 \times c. t := t+1. dbl_0$$

$$= r_c := r_c + 1.$$

$$\mathcal{M}_d w_d = 2 \times \mathcal{M}_c(r_c - 1) \wedge \mathcal{I}_d w_d = t \wedge (w_d := w_d + 1).$$

$$t := t+1. \top$$

$$= \mathcal{M}_d w_d = 2 \times \mathcal{M}_c r_c \wedge \mathcal{I}_d w_d = t$$

$$dbl_2 = c?. d! 2 \times c. t := t+1. dbl_1$$

$$= r_c := r_c + 1.$$

$$\mathcal{M}_d w_d = 2 \times \mathcal{M}_c(r_c - 1) \wedge \mathcal{I}_d w_d = t \wedge (w_d := w_d + 1).$$

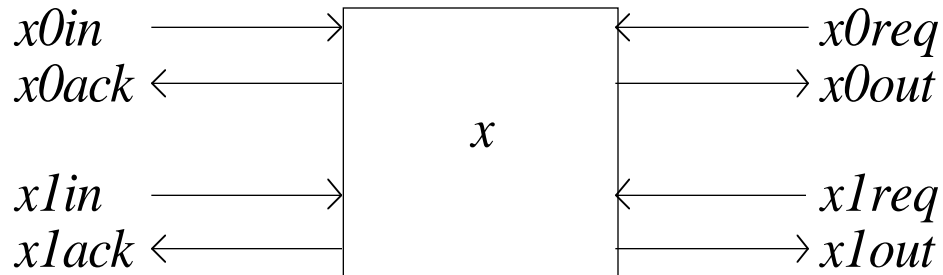
$$t := t+1. \mathcal{M}_d w_d = 2 \times \mathcal{M}_c r_c \wedge \mathcal{I}_d w_d = t$$

$$= \mathcal{M}_d w_d = 2 \times \mathcal{M}_c r_c \wedge \mathcal{I}_d w_d = t$$

$$\wedge \mathcal{M}_d(w_d + 1) = 2 \times \mathcal{M}_c(r_c + 1) \wedge \mathcal{I}_d(w_d + 1) = t + 1$$

$$dbl_\infty = \forall n: nat. \mathcal{M}_d(w_d + n) = 2 \times \mathcal{M}_c(r_c + n) \wedge \mathcal{I}_d(w_d + n) = t + n$$

Monitor



Let $m = MIN[\mathcal{T}_{x0in}r_{x0in}; \mathcal{T}_{x1in}r_{x1in}; \mathcal{T}_{x0req}r_{x0req}; \mathcal{T}_{x1req}r_{x1req}]$

$monitor =$

$$\begin{aligned}
 & (\sqrt{x0in} \vee \mathcal{T}_{x0in}r_{x0in} = m) \wedge (x0in?. \ x:=x0in. \ x0ack!) \\
 \vee & (\sqrt{x1in} \vee \mathcal{T}_{x1in}r_{x1in} = m) \wedge (x1in?. \ x:=x1in. \ x1ack!) \\
 \vee & (\sqrt{x0req} \vee \mathcal{T}_{x0req}r_{x0req} = m) \wedge (x0req?. \ x0out! \ x) \\
 \vee & (\sqrt{x1req} \vee \mathcal{T}_{x1req}r_{x1req} = m) \wedge (x1req?. \ x1out! \ x).
 \end{aligned}$$

$monitor$

$monitor \Leftarrow$ **if** $\sqrt{x0in}$ **then** $(x0in?. \ x:=x0in. \ x0ack!)$ **else** *ok*.
if $\sqrt{x1in}$ **then** $(x1in?. \ x:=x1in. \ x1ack!)$ **else** *ok*.
if $\sqrt{x0req}$ **then** $(x0req?. \ x0out! \ x)$ **else** *ok*.
if $\sqrt{x1req}$ **then** $(x1req?. \ x1out! \ x)$ **else** *ok*.
 $t:=t+1. \ monitor$

COMMUNICATING PROCESSES

$$\begin{aligned}
 & c! 2 \parallel (c?. x:= c) \\
 = & \mathcal{M}w = 2 \wedge (w:= w+1) \parallel (r:= r+1. x:= \mathcal{M}(r-1)) \\
 = & \mathcal{M}w = 2 \wedge w' = w+1 \wedge r' = r+1 \wedge x' = \mathcal{M}r
 \end{aligned}$$

$$c! 1. (c! 2 \parallel (c?. x:= c)). c?$$

Channel Declaration

$$\begin{aligned}
 & \mathbf{chan} \ c: T \cdot P \\
 = & \exists \mathcal{M}_c: [\infty^*T]. \exists \mathcal{I}_c: [\infty^*xnat]. \mathbf{var} \ r_c, w_c: xnat := 0. P
 \end{aligned}$$

ignoring time

chan $c: int \cdot c! 2 \parallel (c?. x:= c)$

= $\exists \mathcal{M}: [\infty^*int] \cdot \exists \mathcal{T}: [\infty^*xnat] \cdot \mathbf{var} \ r, w: xnat := 0 \cdot$

$x' = \mathcal{M}r \wedge \mathcal{M}w = 2 \wedge r' = r+1 \wedge w' = w+1$

\wedge (other variables unchanged)

= $\exists \mathcal{M}: [\infty^*int] \cdot \exists \mathcal{T}: [\infty^*xnat] \cdot \mathbf{var} \ r, w: xnat \cdot$

$x' = \mathcal{M}0 \wedge \mathcal{M}0 = 2 \wedge r'=1 \wedge w'=1$

\wedge (other variables unchanged)

= $x'=2 \wedge$ (other variables unchanged)

= $x:= 2$

including time

chan $c: int \cdot c! 2 \parallel (t:= \max t (\mathcal{T}r + 1). c?. x:= c)$

= $x'=2 \wedge t' = t+1 \wedge$ (other variables unchanged)

Deadlock

chan $c: int$. $t := \max t (\mathcal{T}r + 1)$. $c?$. $c! 5$

= $\exists \mathcal{M}: [\infty^*int]$. $\exists \mathcal{T}: [\infty^*xnat]$. **var** $r, w: xnat := 0$.

$t := \max t (\mathcal{T}r + 1)$. $r := r + 1$.

$\mathcal{M}w = 5 \wedge \mathcal{T}w = t \wedge (w := w + 1)$

= $\exists \mathcal{M}: [\infty^*int]$. $\exists \mathcal{T}: [\infty^*xnat]$. $\exists r, r', w, w': xnat$.

$r := 0$. $w := 0$. $t := \max t (\mathcal{T}r + 1)$. $r := r + 1$.

$\mathcal{M}w = 5 \wedge \mathcal{T}w = t \wedge r' = r \wedge w' = w + 1 \wedge t' = t$

= $\exists \mathcal{M}: [\infty^*int]$. $\exists \mathcal{T}: [\infty^*xnat]$. $\exists r, r', w, w': xnat$.

$\mathcal{M}0 = 5 \wedge \mathcal{T}0 = \max t (\mathcal{T}0 + 1) \wedge r' = 1 \wedge w' = 1$

$\wedge t' = \max t (\mathcal{T}0 + 1)$

= $t' = \infty$

chan $c, d: \text{int} \cdot (c?. d! 6) \parallel (d?. c! 7)$

chan $c, d: \text{int} \cdot$

$(t := \max t (\mathcal{I}_c r_c + 1). c?. d! 6)$

$\parallel (t := \max t (\mathcal{I}_d r_d + 1). d?. c! 7)$

$= \exists \mathcal{M}_c, \mathcal{M}_d: [\infty^* \text{int}] \cdot \exists \mathcal{I}_c, \mathcal{I}_d: [\infty^* \text{xnat}] \cdot$

$\exists r_c, r'_c, w_c, w'_c, r_d, r'_d, w_d, w'_d: \text{xnat} \cdot$

$\mathcal{M}_d 0 = 6 \wedge \mathcal{I}_d 0 = \max t (\mathcal{I}_c 0 + 1) \wedge \mathcal{M}_c 0 = 7$

$\wedge \mathcal{I}_c 0 = \max t (\mathcal{I}_d 0 + 1) \wedge r'_c = w'_c = r'_d = w'_d = 1$

$\wedge t' = \max t (\mathcal{I}_c 0 + 1) (\mathcal{I}_d 0 + 1)$

$= t' = \infty$

POWER SERIES MULTIPLICATION

Input on channel a : $a_0 a_1 a_2 \dots$

Input on channel b : $b_0 b_1 b_2 \dots$

Output on channel c : $c_0 c_1 c_2 \dots$

$$A = a_0 + a_1x + a_2x^2 + \dots \qquad B = b_0 + b_1x + b_2x^2 + \dots$$

$$A_1 = a_1 + a_2x + a_3x^2 + \dots \qquad B_1 = b_1 + b_2x + b_3x^2 + \dots$$

$$A_2 = a_2 + a_3x + a_4x^2 + \dots \qquad B_2 = b_2 + b_3x + b_4x^2 + \dots$$

$$\begin{aligned} C &= c_0 + c_1x + c_2x^2 + \\ &= A \times B \\ &= a_0 \times b_0 + (a_0 \times b_1 + a_1 \times b_0)x + (a_0 \times b_2 + A_1 \times B_1 + A_2 \times b_0)x^2 \end{aligned}$$

$(\lambda!c: \text{rat} \cdot C = A \times B) \ c \ \Leftarrow$

$(a? \parallel b?). \ c! \ a \times b.$

var $a_0: \text{rat} := a. \ \mathbf{var} \ b_0: \text{rat} := b. \ \mathbf{chan} \ d: \text{rat}.$

$(\lambda!c: \text{rat} \cdot C = A \times B) \ d$

$\parallel ((a? \parallel b?). \ c! \ a_0 \times b + a \times b_0. \ C = a_0 \times B + D + A \times b_0)$

$C = a_0 \times B + D + A \times b_0 \ \Leftarrow$

$(a? \parallel b? \parallel d?). \ c! \ a_0 \times b + d + a \times b_0. \ C = a_0 \times B + D + A \times b_0$