Timed Process Algebra

Part I

Jos Baeten

Division of Computer Science
Eindhoven University of Technology

Process algebra

Has ordering of actions.

But we want quantative analysis.

Time domain

Discrete vs. dense.

Linear vs. branching.

Least element.

Here,

- natural numbers,
- non-negative reals.

Timing syntax

Time-stamped vs. two-phase.

Axioms in two-phase.

But operational semantics and predicates for dense time work better time-stamped.

Timing semantics

Absolute vs. relative ($\rightarrow$ parametric).

Time-determinism: $\sigma.a + \sigma.\sigma.\sigma.b$.

- No delay of 2 possible: strong time-determinism,
- Choice by delay of 1: time-nondeterminism,
- Choice by delay of 2, not by delay of 1: weak time-determinism.

Timing semantics (cont.)

Duration of actions.

Urgent actions vs. multi-actions.

The book

PAwT, with Kees Middelburg, Springer 2002.

Add skip / explicit termination:

- Action prefix
- Focus on relative, dense

MPA$_{drt}$

- $\underline{\underline{\delta}}$ undelayable deadlock
- $\underline{\underline{\epsilon}}$ undelayable termination
- $\underline{\underline{a}}.$ current time slice action prefix
- $\sigma.$ next time slice prefix (‘tick’)
- $+$ alternative composition (choice)

Axioms

$$x + y = y + x$$

$$(x + y) + z = x + (y + z)$$

$$x + x = x$$

$$x + \underline{\underline{\delta}} = x$$

$$\sigma.x + \sigma.y = \sigma.(x + y)$$

Operational rules

$$
{}^{1}\frac{}{\underline{\underline{\epsilon}}\downarrow}
\qquad
{}^{2}\frac{}{\underline{\underline{a}}.x \xrightarrow{a} x}
\qquad
{}^{3}\frac{}{\sigma.x \overset{1}{\longmapsto} x}
$$

$$
{}^{4}\frac{x\downarrow}{x+y\downarrow}
\qquad
{}^{5}\frac{y\downarrow}{x+y\downarrow}
\qquad
{}^{6}\frac{x \xrightarrow{a} x'}{x+y \xrightarrow{a} x'}
\qquad
{}^{7}\frac{y \xrightarrow{a} y'}{x+y \xrightarrow{a} y'}
$$

$$
{}^{8}\frac{x \overset{1}{\longmapsto} x',\ y \overset{1}{\longmapsto} y'}{x+y \overset{1}{\longmapsto} x'+y'}
\qquad
{}^{9}\frac{x \overset{1}{\longmapsto} x',\ y \overset{1}{\not\longmapsto}}{x+y \overset{1}{\longmapsto} x'}
\qquad
{}^{10}\frac{x \overset{1}{\not\longmapsto},\ y \overset{1}{\longmapsto} y'}{x+y \overset{1}{\longmapsto} y'}
$$

$$a \in \mathsf{A}$$

Bisimulation

A bisimulation R is a symmetric, binary relation on processes:

$\forall p, q : pRq \Rightarrow$

- $p \downarrow \quad \Rightarrow \quad q \downarrow$
- $p \xrightarrow{a} p' \quad \Rightarrow \quad \exists q' : q \xrightarrow{a} q' \;\&\; p' R q'$
- $p \overset{1}{\longmapsto} p' \quad \Rightarrow \quad \exists q' : q \overset{1}{\longmapsto} q' \;\&\; p' R q'$

$p \leftrightarrows q \quad \Leftrightarrow \quad \exists$ bisimulation R

Theorems

Bisimilarity is a congruence relation for the operators.

MPA_{drt} is a sound and complete axiomatization of the model of bisimulation equivalence classes of closed terms.

Generalized time step

Generalize σ to σ^n.

$\sigma^0.x = x$

$\sigma^{n+1}.x = \sigma.\sigma^n.x$

Generalized time step as basic operator

$$x + y = y + x$$

$$(x + y) + z = x + (y + z)$$

$$x + x = x$$

$$x + \underline{\underline{\delta}} = x$$

$$\sigma^0 . x = x$$

$$\sigma^n . \sigma^m . x = \sigma^{n+m} . x$$

$$\sigma^n . x + \sigma^n . y = \sigma^n . (x + y)$$

Operational rules

Either $\xrightarrow{a}$, $\downarrow$, $\xmapsto{1}$ as before,

or $\xrightarrow{a}$, $\downarrow$, $\xmapsto{n}$.

Both do not generalize to dense time easily.

Axiomatically, generalization is straightforward.

MPT_{srt}

- $\tilde{\tilde{\delta}}$ undelayable deadlock
- $\tilde{\tilde{\epsilon}}$ undelayable termination
- $\tilde{\tilde{a}}.$ urgent action prefix
- $\sigma^t.$ relative time delay prefix
- $+$ alternative composition (choice)

Axioms

$$x + y = y + x$$

$$(x + y) + z = x + (y + z)$$

$$x + x = x$$

$$x + \tilde{\tilde{\delta}} = x$$

$$\sigma^0 . x = x$$

$$\sigma^t . \sigma^s . x = \sigma^{t+s} . x$$

$$\sigma^t . x + \sigma^t . y = \sigma^t . (x + y)$$

Operational rules

Can use $\xrightarrow{a}$, $\downarrow$, $\overset{t}{\longmapsto}$.

But will have problems further on. Therefore go to time-stamped:

- $x \xrightarrow{a}_t x'$: after delay of t, x can execute a and thereby evolve to x'
- $x \downarrow_t$: x can terminate after delay of t
- $\Delta_t(x)$: x can delay for time t

Operational rules

$$ {}^{1}\frac{}{\tilde{\tilde{\epsilon}} \downarrow_0} \qquad {}^{2}\frac{}{\tilde{\tilde{a}}.x \xrightarrow{a}_0 x} \qquad {}^{3}\frac{}{\Delta_0(x)} $$

$$ {}^{4}\frac{x \downarrow_t}{x + y \downarrow_t} \qquad {}^{5}\frac{y \downarrow_t}{x + y \downarrow_t} \qquad {}^{6}\frac{x \xrightarrow{a}_t x'}{x + y \xrightarrow{a}_t x'} \qquad {}^{7}\frac{y \xrightarrow{a}_t y'}{x + y \xrightarrow{a}_t y'} $$

$$ {}^{8}\frac{\Delta_t(x)}{\Delta_t(x + y)} \qquad {}^{9}\frac{\Delta_t(y)}{\Delta_t(x + y)} $$

$$ {}^{10}\frac{x \downarrow_t}{\sigma^s.x \downarrow_{t+s}} \qquad {}^{11}\frac{x \xrightarrow{a}_t x'}{\sigma^s.x \xrightarrow{a}_{t+s} x'} \qquad {}^{12}\frac{t \leq s}{\Delta_t(\sigma^s.x)} \qquad {}^{13}\frac{\Delta_t(x)}{\Delta_{t+s}(\sigma^s.x)} $$

$$ a \in \mathsf{A}, t, s \geq 0 $$

Theorems

Congruence, soundness, completeness.

Uses elimination to **basic terms**:

- $\forall t \geq 0 \quad \sigma^t.\tilde{\tilde{\epsilon}} \in \mathcal{B}$
- $\forall t \geq 0 \quad \sigma^t.\tilde{\tilde{\delta}} \in \mathcal{B}$
- $\forall a \in \mathsf{A}, t \geq 0, x \in \mathcal{B} \quad \sigma^t.\tilde{\tilde{a}}.x \in \mathcal{B}$
- $\forall x, y \in \mathcal{B} \quad x + y \in \mathcal{B}$

Sequential composition

$$(x + y) \cdot z = x \cdot z + y \cdot z$$

$$(x \cdot y) \cdot z = x \cdot (y \cdot z)$$

$$\tilde{\tilde{a}}.x \cdot y = \tilde{\tilde{a}}.(x \cdot y)$$

$$\sigma^t.x \cdot y = \sigma^t.(x \cdot y)$$

$$\tilde{\tilde{\delta}} \cdot x = \tilde{\tilde{\delta}}$$

$$\tilde{\tilde{\epsilon}} \cdot x = x$$

$$x \cdot \tilde{\tilde{\epsilon}} = x$$

Operational rules

$$
{}_{1}\frac{x \downarrow_t,\ y \downarrow_s}{x \cdot y \downarrow_{t+s}}
\qquad
{}_{2}\frac{x \xrightarrow{a}_t x'}{x \cdot y \xrightarrow{a}_t x' \cdot y}
\qquad
{}_{3}\frac{x \downarrow_t,\ y \xrightarrow{a}_s y'}{x \cdot y \xrightarrow{a}_{t+s} y'}
$$

$$
{}_{4}\frac{\Delta_t(x)}{\Delta_t(x \cdot y)}
\qquad
{}_{5}\frac{x \downarrow_t,\ \Delta_s(y)}{\Delta_{t+s}(x \cdot y)}
$$

$$a \in \mathsf{A}, t, s \geq 0$$

The problem with two-phase semantics

Sequential composition in two phase, discrete time:

$$\frac{x \stackrel{1}{\longmapsto} x',\ (x \not\downarrow \vee y \not\stackrel{1}{\longmapsto})}{x \cdot y \stackrel{1}{\longmapsto} x' \cdot y}\,1 \qquad \frac{x \not\stackrel{1}{\longmapsto},\ x\downarrow,\ y \stackrel{1}{\longmapsto} y'}{x \cdot y \stackrel{1}{\longmapsto} y'}\,2 \qquad \frac{x \stackrel{1}{\longmapsto} x',\ x\downarrow,\ y \stackrel{1}{\longmapsto} y'}{x \cdot y \stackrel{1}{\longmapsto} x' \cdot y + y'}\,3$$

$$a \in \mathsf{A}$$

A delay t of $x \cdot y$ may be written in several ways as $r + s$, and

$x \stackrel{r}{\longmapsto} x_r,\ x_r\downarrow,\ y \stackrel{s}{\longmapsto} y_s$ leads to $x \cdot y \stackrel{t}{\longmapsto} y_s + \ldots$.

Theorems

Congruence,

soundness,

completeness,

elimination to basic terms,

conservative extension.

Parallel composition

$x \parallel y$ has interleaving and synchronisation.

Interleaving: x or y can execute an action now, urgent and independent. Left merge.

Synchronisation (communication merge):

- x and y can execute a communication action synchronously, by each executing their half (send and receive give communication);
- x and y can terminate now, synchronously;
- x and y can delay together.

AXIOMS

$x \parallel y = x \mathbin{\lfloor\!\lfloor} y + y \mathbin{\lfloor\!\lfloor} x + x \mid y$	$x \mid y = y \mid x$
	$\tilde{\tilde{\delta}} \mid x = \tilde{\tilde{\delta}}$
$\tilde{\tilde{\delta}} \mathbin{\lfloor\!\lfloor} x = \tilde{\tilde{\delta}}$	$\tilde{\tilde{\epsilon}} \mid \tilde{\tilde{\epsilon}} = \tilde{\tilde{\epsilon}}$
$\tilde{\tilde{\epsilon}} \mathbin{\lfloor\!\lfloor} x = \tilde{\tilde{\delta}}$	$\tilde{\tilde{a}}.x \mid \tilde{\tilde{\epsilon}} = \tilde{\tilde{\delta}}$
$\tilde{\tilde{a}}.x \mathbin{\lfloor\!\lfloor} y = \tilde{\tilde{a}}.(x \parallel y)$	$\tilde{\tilde{a}}.x \mid \tilde{\tilde{b}}.y = \tilde{\tilde{c}}.(x \parallel y)$ if $\gamma(a,b) = c$
$\sigma^u.x \mathbin{\lfloor\!\lfloor} y = \tilde{\tilde{\delta}}$	$\tilde{\tilde{a}}.x \mid \tilde{\tilde{b}}.y = \tilde{\tilde{\delta}}$ otherwise
$(x + y) \mathbin{\lfloor\!\lfloor} z = x \mathbin{\lfloor\!\lfloor} z + y \mathbin{\lfloor\!\lfloor} z$	$\tilde{\tilde{a}}.x \mid \sigma^u.y = \tilde{\tilde{\delta}}$
	$\sigma^u.x \mid \tilde{\tilde{\epsilon}} = \sigma^u.x$
	$\sigma^u.x \mid \sigma^u.y = \sigma^u.(x \parallel y)$
	$(x + y) \mid z = x \mid z + y \mid z$

$$a, b \in \mathsf{A},\ u > 0$$

Operational rules

$$1\;\frac{x \downarrow_t,\; y \downarrow_s}{x \parallel y \downarrow_{max(t,s)}} \qquad 2\;\frac{x \xrightarrow{a}_t x',\; \Delta_t(y)}{x \parallel y \xrightarrow{a}_t x' \parallel (t \gg y)} \qquad 3\;\frac{x \xrightarrow{a}_t x',\; y \downarrow_s,\; s \leq t}{x \parallel y \xrightarrow{a}_t x'}$$

$$4\;\frac{x \xrightarrow{a}_t x',\; y \xrightarrow{b}_t y',\; \gamma(a,b) = c}{x \parallel y \xrightarrow{c}_t x' \parallel y'} \qquad 5\;\frac{\Delta_t(y),\; y \xrightarrow{a}_t y'}{x \parallel y \xrightarrow{a}_t (t \gg x') \parallel y} \qquad 6\;\frac{y \downarrow_s,\; s \leq t,\; y \xrightarrow{a}_t y'}{x \parallel y \xrightarrow{a}_t y'}$$

$$7\;\frac{\Delta_t(x),\; \Delta_t(y)}{\Delta_t(x \parallel y)} \qquad 8\;\frac{\Delta_t(x),\; y \downarrow_s, s \leq t}{\Delta_t(x \parallel y)} \qquad 9\;\frac{x \downarrow_s,\; s \leq t,\; \Delta_t(y)}{\Delta_t(x \parallel y)}$$

$$a, b, c \in \mathsf{A}, t, s \geq 0$$

Additional axioms

$\partial_H(\tilde{\tilde{\delta}}) = \tilde{\tilde{\delta}}$		$0 \gg x = x$
$\partial_H(\tilde{\tilde{\epsilon}}) = \tilde{\tilde{\epsilon}}$		$t \gg \tilde{\tilde{\delta}} = \tilde{\tilde{\delta}}$
$\partial_H(\tilde{\tilde{a}}.x) = \tilde{\tilde{a}}.\partial_H(x)$	if $a \notin H$	$u \gg \tilde{\tilde{\epsilon}} = \tilde{\tilde{\delta}}$
$\partial_H(\tilde{\tilde{a}}.x) = \tilde{\tilde{\delta}}$	if $a \in H$	$u \gg \tilde{\tilde{a}}.x = \tilde{\tilde{\delta}}$
$\partial_H(\sigma^t.x) = \sigma^t.\partial_H(x)$		$t \gg \sigma^{t+s}.x = \sigma^s.x$
$\partial_H(x+y) = \partial_H(x) + \partial_H(y)$		$(t+s) \gg \sigma^s.x = t \gg x$
		$t \gg (x+y) = t \gg x + t \gg y$

$$a \in \mathsf{A}, H \subseteq \mathsf{A},\ t, s \geq 0, u > 0$$

Additional rules

Straightforward.

Congruence, soundness, completeness, elimination, conservativity.

Axioms of standard concurrency

$$x \parallel \tilde{\tilde{\epsilon}} = x$$

$$(x \parallel y) \parallel z = x \parallel (y \parallel z)$$

$$(x \mid y) \mid z = x \mid (y \mid z)$$

$$(x \mathbin{\lfloor\!\lfloor} y) \mathbin{\lfloor\!\lfloor} z = x \mathbin{\lfloor\!\lfloor} (y \parallel z)$$

$$(x \mid y) \mathbin{\lfloor\!\lfloor} z = x \mid (y \mathbin{\lfloor\!\lfloor} z)$$

Expansion theorem.

Handshaking communication

Only binary communication.

Processes send, receive and communicate data at **ports**.

Internal and external ports.

$s_i(d)$: the action of sending datum d at port i;

$r_i(d)$: the action of receiving datum d at port i;

$c_i(d)$: the action of communicating datum d at port i.

$\gamma(s_i(d), r_i(d)) = \gamma(r_i(d), s_i(d)) = c_i(d)$

Delayable actions

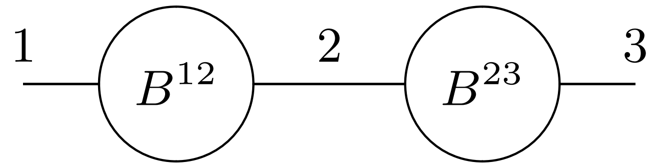
$a.x = \underline{\underline{a}}.x + \sigma.a.x$

$a.x \xrightarrow{a}_t x \qquad \Delta_t(a.x)$

Embedding of time-free theory.

Arbitrary delay prefix.

Communicating buffers



Time-free process algebra:

$$B^{ij} = \sum_{d \in D} r_i(d).s_j(d).B^{ij}$$

Using time-free calculations:

$$\tau_I \circ \partial_H (B^{12} \parallel B^{23}) = B_2^{13}$$

No delay buffers

At most one input per time slice, output with no delay.

$$C^{ij} = \sum_{d \in D} r_i(d).\underline{\underline{s_j(d)}}.\sigma.C^{ij}$$

or equivalently

$$C^{ij} = \sum_{d \in D} \underline{\underline{r_i(d)}}.\underline{\underline{s_j(d)}}.\sigma.C^{ij} + \sigma.C^{ij}$$

Time-free projection

$$\pi_{\mathsf{tf}}(C^{ij}) = B^{ij}$$

CALCULATION

$$\partial_H(C^{12} \parallel C^{23}) = \sum_{d \in D} r_1(d).\partial_H(\underline{\underline{s_2(d)}}.\sigma.C^{12} \parallel C^{23}) =$$

$$= r_1(d).\underline{\underline{c_2(d)}}.\partial_H(\sigma.C^{12} \parallel \underline{\underline{s_3(d)}}.\sigma.C^{23}) =$$

$$= r_1(d).\underline{\underline{c_2(d)}}.\underline{\underline{s_3(d)}}.\partial_H(\sigma.C^{12} \parallel \sigma.C^{23}) =$$

$$= r_1(d).\underline{\underline{c_2(d)}}.\underline{\underline{s_3(d)}}.\sigma.\partial_H(C^{12} \parallel C^{23}).$$

Abstraction from internal communications:

$$\tau_I \circ \partial_H(C^{12} \parallel C^{23}) = r_1(d).\underline{\underline{s_3(d)}}.\sigma.\tau_I \circ \partial_H(C^{12} \parallel C^{23}) = C^{13}$$

Unit delay buffers

At most one input per time slice, output with unit delay.

$$D^{ij} = \sum_{d \in D} r_i(d).\sigma.\underline{\underline{s_j(d)}}.D^{ij}$$

Time-free projection

$$\pi_{\mathsf{tf}}(D^{ij}) = B^{ij}$$

Calculation

$$X = \partial_H(D^{12} \parallel D^{23}) = \sum_{d \in D} r_1(d).\sigma\partial_H(\underline{\underline{s_2(d)}}.D^{12} \parallel D^{23}) =$$

$$= r_1(d).\sigma.\underline{\underline{c_2(d)}}.\partial_H(D^{12} \parallel \sigma.\underline{\underline{s_3(d)}}.D^{23}) = r_1(d).\sigma.\underline{\underline{c_2(d)}}.X_d$$

$$X_d = \partial_H(D^{12} \parallel \sigma.\underline{\underline{s_3(d)}}.D^{23}) =$$

$$= \sum_{e \in D} \underline{\underline{r_1(e)}}.\partial_H(\sigma.\underline{\underline{s_2(e)}}.D^{12} \parallel \sigma.\underline{\underline{s_3(d)}}.D^{23}) + \sigma.\partial_H(D^{12} \parallel \underline{\underline{s_3(d)}}.D^{23}) =$$

$$= \sum_{e \in D} \underline{\underline{r_1(e)}}.\sigma.\partial_H(\underline{\underline{s_2(e)}}.D^{12} \parallel \underline{\underline{s_3(d)}}.D^{23}) +$$

$$+\sigma.(\sum_{e \in D} \underline{\underline{r_1(e)}}.\partial_H(\sigma.\underline{\underline{s_2(e)}}.D^{12} \parallel \underline{\underline{s_3(d)}}.D^{23}) + \underline{\underline{s_3(d)}}.\partial_H(D^{12} \parallel D^{23})) =$$

$$
\begin{aligned}
&= \sum_{e\in D} \underline{\underline{r_i(e)}}.\sigma.\underline{\underline{s_3(d)}}.\partial_H(\underline{\underline{s_2(e)}}.D^{12} \parallel D^{23})+ \\
&\qquad +\sigma.(\sum_{e\in D} \underline{\underline{r_1(e)}}.\underline{\underline{s_3(d)}}.\partial_H(\sigma.\underline{\underline{s_2(e)}}.D^{12} \parallel D^{23}) + \underline{\underline{s_3(d)}}.X) = \\
&= \sum_{e\in D} \underline{\underline{r_1(e)}}.\sigma.\underline{\underline{s_3(d)}}.\underline{\underline{c_2(e)}}.\partial_H(D^{12} \parallel \sigma.\underline{\underline{s_3(e)}}.D^{23})+ \\
&\qquad +\sigma.(\sum_{e\in D} \underline{\underline{r_1(e)}}.\underline{\underline{s_3(d)}}.\sigma.\partial_H(\underline{\underline{s_2(e)}}.D^{12} \parallel D^{23}) + \underline{\underline{s_3(d)}}.X) = \\
&= \sum_{e\in D} \underline{\underline{r_1(e)}}.\sigma.\underline{\underline{s_3(d)}}.\underline{\underline{c_2(e)}}.X_e+ \\
&\qquad +\sigma.(\sum_{e\in D} \underline{\underline{r_1(e)}}.\underline{\underline{s_3(d)}}.\sigma.\underline{\underline{c_2(e)}}.\partial_H(D^{12} \parallel \sigma.\underline{\underline{s_3(e)}}.D^{23}) + \underline{\underline{s_3(d)}}.X) = \\
&= \sum_{e\in D} \underline{\underline{r_1(e)}}.\sigma.\underline{\underline{s_3(d)}}.\underline{\underline{c_2(e)}}.X_e + \sigma.(\sum_{e\in D} \underline{\underline{r_1(e)}}.\underline{\underline{s_3(d)}}.\sigma.\underline{\underline{c_2(e)}}.X_e + \underline{\underline{s_3(d)}}.X).
\end{aligned}
$$

Calculation (cont.)

After abstraction:

$$\tau_I(X) = \sum_{d \in D} r_1(d).\sigma.\tau_I(X_d)$$

$$\tau_I(X_d) = \sum_{e \in D} \underline{\underline{r_1(e)}}.\sigma.\underline{\underline{s_3(d)}}.\tau_I(X_e) +$$

$$+\sigma.(\sum_{e \in D} \underline{\underline{r_1(e)}}.\underline{\underline{s_3(d)}}.\sigma.\tau_I(X_e) + \underline{\underline{s_3(d)}}.\tau_I(X).$$

A two-place buffer with delay of two time units.

In accordance with time-free theory.

See picture.

Input-enabled buffer

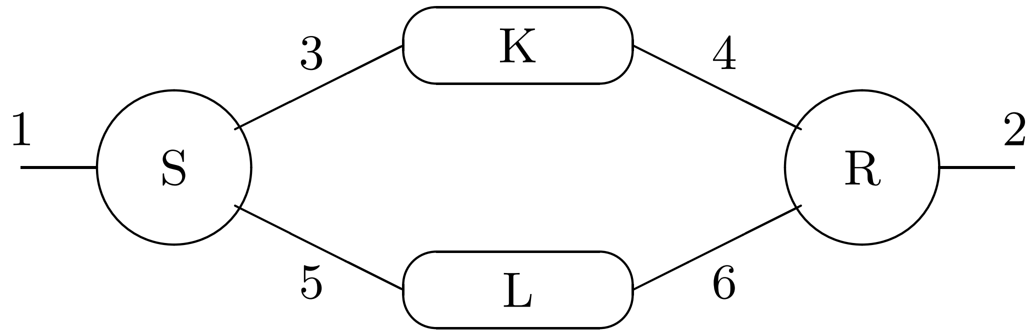
$$E^{ij} = \sum_{d \in D} r_i(d).\sigma.\left(\underline{\underline{s_j(d)}}.\underline{\underline{\epsilon}} \parallel E^{ij}\right)$$

Again, delay of 1 between input and output.

But input always possible, at most one per time slice.

Possible to have three elements inside in $\partial_H(E^{12} \parallel E^{23})$.

PAR PROTOCOL



$$S = S_0$$

$$S_b = \sum_{d \in D} r_1(d).\sigma^{t_1}.SF_{d,b}$$

$$SF_{d,b} = \underline{\underline{s_3(d,b)}}.\left(\sum_{k<t_1'} \sigma^k.\underline{\underline{r_5(ack)}}.S_{1-b} + \sigma^{t_1'}.SF_{d,b}\right)$$

SPECIFICATION

$$R = R_0$$

$$R_b = \sum_{d \in D} r_4(d,b).\sigma^{t_2}.\underline{\underline{s_2(d)}}.\sigma^{t_2'}.\underline{\underline{s_6(ack)}}.R_{1-b} + \sum_{d \in D} r_4(d,1-b).\sigma^{t_2'}.\underline{\underline{s_6(ack)}}.R_b$$

$$K = \sum_{f \in F} r_3(f).\left(\sigma^{t_3}.\underline{\underline{s_4(f)}}.K + \sum_{k \leq t_3} \sigma^k.\underline{\underline{error}}.K\right)$$

$$L = r_6(ack).\left(\sigma^{t_4}.\underline{\underline{s_5(ack)}}.L + \sum_{k \leq t_4} \sigma^k.\underline{\underline{error}}.L\right)$$

$$\partial_H(S \parallel K \parallel L \parallel R)$$

Expansion

$$S''_{d,b,t} = \sum_{k<t} \sigma^k.\underline{\underline{r_5(ack)}}.S_{1-b} + \sigma^t.\underline{\underline{s_3(d,b)}}.S''_{d,b}$$

$$\partial_H(S_b \parallel K \parallel L \parallel R_b) = \sum_{d \in D} r_1(d).\partial_H(S'_{d,b} \parallel K \parallel L \parallel R_b)$$

$$\partial_H(S'_{d,b} \parallel K \parallel L \parallel R_b) = \sigma^{t_1}.\underline{\underline{c_3(d,b)}}.\partial_H(S''_{d,b} \parallel K'_{d,b} \parallel L \parallel R_b)$$

$$\begin{aligned}\partial_H(S''_{d,b} \parallel K'_{d,b} \parallel L \parallel R_b) = \sigma^{t_3}.\underline{\underline{c_4(d,b)}}.\partial_H(S''_{d,b,t'_1-t_3} \parallel K \parallel L \parallel R'_{d,b})+ \\ + \sum_{k \leq t_3} \sigma^k.\underline{\underline{error}}.\partial_H(S''_{d,b,t'_1-k} \parallel K \parallel L \parallel R_b)\end{aligned}$$

$$\partial_H(S''_{d,b,t} \parallel K \parallel L \parallel R'_{d,b}) = \sigma^{t_2}.\underline{\underline{s_2(d)}}.\partial_H(S''_{d,b,t-t_2} \parallel K \parallel L \parallel R''_{1-b})$$

$$\partial_H(S''_{d,b,t} \parallel K \parallel L \parallel R''_{1-b}) = \sigma^{t'_2}.\underline{\underline{c_6(ack)}}.\partial_H(S''_{d,b,t-t'_2} \parallel K \parallel L' \parallel R_{1-b})$$

Expansion (cont.)

$$\partial_H(S''_{d,b,t} \parallel K \parallel L' \parallel R_{1-b}) = \sigma^{t_4}.\underline{\underline{c_5(ack)}}.\partial_H(S_{1-b} \parallel K \parallel L \parallel R_{1-b}) + \\ + \sum_{k \le t_4} \sigma^k.\underline{\underline{error}}.\partial_H(S''_{d,b,t-k} \parallel K \parallel L \parallel R_{1-b})$$

$$\partial_H(S''_{d,b,t} \parallel K \parallel L \parallel R_b) = \sigma^t.\underline{\underline{c_3(d,b)}}.\partial_H(S''_{d,b} \parallel K'_{d,b} \parallel L \parallel R_b)$$

$$\partial_H(S''_{d,b,t} \parallel K \parallel L \parallel R_{1-b}) = \sigma^t.\underline{\underline{c_3(d,b)}}.\partial_H(S''_{d,b} \parallel K'_{d,b} \parallel L \parallel R_{1-b})$$

$$\partial_H(S''_{d,b} \parallel K'_{d,b} \parallel L \parallel R_{1-b}) = \sigma^{t_3}.\underline{\underline{c_4(d,b)}}.\partial_H(S''_{d,b,t'_1-t_3} \parallel K \parallel L \parallel R''_{1-b}) + \\ + \sum_{k \le t_3} \sigma^k.\underline{\underline{error}}.\partial_H(S''_{d,b,t'_1-k} \parallel K \parallel L \parallel R_{1-b})$$

Conclusion: system without deadlocks if $t'_1 > t_2 + t'_2 + t_3 + t_4$.

Abstraction

Abstract from all actions except inputs (r_1) and outputs (s_2).

Functional correctness: first abstract from timing, then use time-free theory, yields a one-place buffer. Still, timing was necessary to reach this result!

Can also do abstraction in the timed theory, using branching laws. Obtain the following recursive specification for $\tau_I \circ \partial_H(S_b \parallel K \parallel L \parallel R_b)$:

Additional τ law

$$X = \sum_{d \in D} r_1(d).\sigma^{t_1}.Y_d$$

$$Y_d = \sigma^{t_3}.\underline{\underline{\tau}}.\sigma^{t_2}.\underline{\underline{s_2(d)}}.\sigma^{t_2'}.Z + \sum_{k \leq t_3} \sigma^k.\underline{\underline{\tau}}.\sigma^{t_1'-k}.Y_d,$$

$$Z = \sigma^{t_4}.\underline{\underline{\tau}}.X + \sum_{k \leq t_4} \sigma^k.\underline{\underline{\tau}}.\sigma^{t_1'-(t_2+t_2'+t_3+k)}.U$$

$$U = \sigma^{t_3}.\underline{\underline{\tau}}.\sigma^{t_2'}.V'' + \sum_{k \leq t_3} \sigma^k.\underline{\underline{\tau}}.\sigma^{t_1'-k}.U$$

$$V = \sigma^{t_4}.\underline{\underline{\tau}}.X + \sum_{k \leq t_4} \sigma^k.\underline{\underline{\tau}}.\sigma^{t_1'-(t_2'+t_3+k)}.U$$

Timing of each silent action is preserved, so timing of all choices is preserved.

Leads to formulate extra τ-law: $\underline{\underline{\tau}}.\sigma.x = \sigma.\underline{\underline{\tau}}.x.$

Formulation of semantical equivalence difficult.

Final result, using the extra law

Functional **and** timing behaviour of the PAR protocol is captured in the following.

$$A = \sum_{d \in D} r_1(d).\sigma^{t_1+t_2+t_3}.B_d$$

$$B_d = \underline{\underline{s_2(d)}}.\sigma^{t'_2}.C + \sigma^{t'_1}.B_d$$

$$C = \sigma^{t_4}.(A + \sigma^{t'_1-t_2-t_4}.D)$$

$$D = \sigma^{t_4}.(A + \sigma^{t'_1-t_4}.D)$$