SESSION TYPES

Processes, types and properties

SESSION TYPES

- Motivation
- Session Calculus
  - Syntax
  - Semantics
- Session Types
  - Syntax
  - Typing rules
- Properties

SESSION TYPE: MOTIVATION

Can we have types that describe the communication, not the computation?

How to formally describe/specify and practically describe/implement communications?

SESSION TYPES: OVERVIEW

- Since their appearance, session types have developed into a significant theme in programming languages.
- Computing has moved from the era of data processing to the era of communication.
- **Data types** codify the structure of **data** and make it available to programming tools.
- **Session types** codify the structure of **communication** and make it available to programming tools.

PROCESSES

or how to formally implement protocols

WHAT CAN GO WRONG?

A protocol in session calculus

An ATM agent offers two services: funds balance or deposit.

- If balance is chosen, then it shows a balance of the account, and recurs to the menu with the same amount.
- If deposit is chosen, then it receives a deposited amount z, and returns to the menu with the new state as their sum $y + z$.

The following is an implementation of the ATM (first try):

$$\mathbf{ATM}(a, y) \stackrel{\mathsf{df}}{=} a \rhd [\ \begin{array}{ll} \mathsf{balance}: & \overline{a}\langle y\rangle.\mathbf{ATM}\langle a, y\rangle \ [] \\ \mathsf{deposit}: & a(z).\overline{a}\langle y+z\rangle.\mathbf{ATM}\langle a, y+z\rangle \ ] \end{array}$$

The following is an implementation of the customer:

$$\mathbf{Customer}(a, y) \stackrel{\mathsf{df}}{=} a \lhd \mathsf{deposit}.\overline{a}\langle y\rangle.a(x).P$$

The interaction between these three parties is incorrect:

$$\mathbf{ATM}\langle a, 0\rangle \mid \mathbf{Customer}\langle a, 100\rangle \mid \mathbf{Customer}\langle a, 100\rangle$$

SESSION CHANNELS

Let's try again:

The following is the implementation of the ATM (second try):

$$\begin{array}{lll} \mathbf{ATM}(a,y) & \stackrel{\text{df}}{=} & a(z).\ \mathbf{ATM_1}\langle a,y,z\rangle \\ \mathbf{ATM_1}(a,y,s) & \stackrel{\text{df}}{=} & s \triangleright [\ \ \mathsf{balance} : \overline{s}\langle y\rangle.\mathbf{ATM_1}\langle a,y,s\rangle\ \ [\!] \\ & & \qquad \mathsf{deposit} : s(z).\overline{s}\langle \mathrm{y}{+}z\rangle.\mathbf{ATM_1}\langle a, y+z, s\rangle\ ] \end{array}$$

An example of the customer is:

$$\mathbf{Customer}(a,y) \stackrel{\text{df}}{=} (\nu\, s)\overline{a}\langle s\rangle.s \triangleleft \mathsf{deposit}.\overline{s}\langle y\rangle.s(x).P$$

You can check the interaction between the three parties is safe:

$$\mathbf{ATM}\langle a,0\rangle \mid \mathbf{Customer}\langle a,100\rangle \mid \mathbf{Customer}\langle a,100\rangle$$

Here a is called *shared* name and it allows interference of interactions.
s is called *session* name and is used for structured interactions.

SESSION CALCULUS: SYNTAX

$P ::=$		processes
	$\overline{u}(s).P$	session request
	$u(s).P$	session accept
	$\overline{s}\langle\widetilde{e}\rangle.P$	message send
	$s(\widetilde{x}).P$	message received
	$\overline{s}(s').P$	channel send
	$s(s').P$	channel received
	$s \triangleright \{l_1 : P_1 \,[\!]\, \cdots \,[\!]\, l_n : P_n\}$	branching
	$s \triangleleft l.P$	selection
	$\mathbf{0}$	nil process
	$P \mid Q$	parallel composition of P and Q
	$(\nu\, s)P, (\nu\, a)P$	fresh name generation
	$\mathtt{def}\ D\ \mathtt{in}\ P$	recursion definition
	$X\langle\widetilde{es}\rangle$	recursion call
	$\mathtt{if}\ e\ \mathtt{then}\ P\ \mathtt{else}\ Q$	conditional
$u ::=$	a, b, x	shared name and variable
$e ::=$	$v, e\ \mathtt{or}\ e, e\ \mathtt{and}\ e, \mathtt{not}\ e$	expressions
$v ::=$	$\mathtt{true}, \mathtt{false}, a$	values
$D ::=$	$X_1(\widetilde{x_1}\widetilde{s_1}) = P_1,\ \ldots,\ X_n(\widetilde{x_n}\widetilde{s_n}) = P_n$	declaration for recursion

SESSION CALCULUS: SYNTAX

P ::=		processes
	$\overline{u}(s).P$	session request
	$u(s).P$	session accept
	$\overline{s}\langle\widetilde{e}\rangle.P$	message send
	$s(\widetilde{x}).P$	message received
	$\overline{s}(s').P$	channel send
	$s(s').P$	channel received
	$s \triangleright \{l_1 : P_1 \, [\!] \cdots [\!] \, l_n : P_n\}$	branching
	$s \triangleleft l.P$	selection
	$\mathbf{0}$	nil process
	$P \mid Q$	parallel composition of P and Q
	$(\nu\, s)P, (\nu\, a)P$	fresh name generation
	$\texttt{def}\ D\ \texttt{in}\ P$	recursion definition
	$X\langle\widetilde{es}\rangle$	recursion call
	$\texttt{if}\ e\ \texttt{then}\ P\ \texttt{else}\ Q$	conditional
u ::=	a, b, x	shared name and variable
e ::=	$v, e\ \texttt{or}\ e, e\ \texttt{and}\ e, \texttt{not}\ e$	expressions
v ::=	$\texttt{true}, \texttt{false}, a$	values
D ::=	$X_1(\widetilde{x_1}\widetilde{s_1}) = P_1, \ \ldots, \ X_n(\widetilde{x_n}\widetilde{s_n}) = P_n$	declaration for recursion

Distinguish **session** and **shared** channels!

SESSION CALCULUS: SYNTAX

$P ::=$	processes
$\quad \overline{u}(s).P$	session request
$\quad u(s).P$	session accept
$\quad \overline{s}\langle\widetilde{e}\rangle.P$	message send
$\quad s(\widetilde{x}).P$	message received
$\quad \overline{s}(s').P$	channel send
$\quad s(s').P$	channel received
$\quad s \triangleright \{l_1 : P_1 \,[\!]\, \cdots \,[\!]\, l_n : P_n\}$	branching
$\quad s \triangleleft l.P$	selection
$\quad \mathbf{0}$	nil process
$\quad P \mid Q$	parallel composition of P and Q
$\quad (\nu\, s)P, (\nu\, a)P$	fresh name generation
$\quad \mathtt{def}\ D\ \mathtt{in}\ P$	recursion definition
$\quad X\langle\widetilde{es}\rangle$	recursion call
$\quad \mathtt{if}\ e\ \mathtt{then}\ P\ \mathtt{else}\ Q$	conditional
$u ::= a, b, x$	shared name and variable
$e ::= v, e\ \mathtt{or}\ e, e\ \mathtt{and}\ e, \mathtt{not}\ e$	expressions
$v ::= \mathtt{true}, \mathtt{false}, a$	values
$D ::= X_1(\widetilde{x_1}\widetilde{s_1}) = P_1, \ \ldots, \ X_n(\widetilde{x_n}\widetilde{s_n}) = P_n$	declaration for recursion

session initiation

interact in a session

other constructs

SESSION CALCULUS: SYNTAX

$P ::=$	pr
$\overline{u}(s).P$	se
$u(s).P$	se
$\overline{s}\langle\widetilde{e}\rangle.P$	m
$s(\widetilde{x}).P$	message received
$\overline{s}(s').P$	cl
$s(s').P$	cl
$s \triangleright \{l_1 : P_1 \,[\!]\, \cdots \,[\!]\, l_n : P_n\}$	b
$s \triangleleft l.P$	se
$\mathbf{0}$	nil process
$P \mid Q$	parallel composition of P and Q
$(\nu\, s)P, (\nu\, a)P$	fresh name generation
$\mathtt{def}\ D\ \mathtt{in}\ P$	recursion definition
$X\langle\widetilde{e}\widetilde{s}\rangle$	recursion call
$\mathtt{if}\ e\ \mathtt{then}\ P\ \mathtt{else}\ Q$	conditional
$u ::= a, b, x$	shared name and variable
$e ::= v, e\ \mathtt{or}\ e, e\ \mathtt{and}\ e, \mathtt{not}\ e$	expressions
$v ::= \mathtt{true}, \mathtt{false}, a$	values
$D ::= X_1(\widetilde{x_1}\widetilde{s_1}) = P_1,\ \ldots,\ X_n(\widetilde{x_n}\widetilde{s_n}) = P_n$	declaration for recursion

shared channels a, a', …

used to initiate a sessions

n

session channels: s, s', …

used for session communication

ion

other constructs

SESSION CALCULUS: SYNTAX

$P ::=$		processes
	$\overline{u}(s).P$	session request
	$u(s).P$	session accept
	$\overline{s}\langle\widetilde{e}\rangle.P$	message send
	$s(\widetilde{x}).P$	message received
	$\overline{s}(s').P$	channel send
	$s(s').P$	channel received
	$s \triangleright \{l_1 : P_1 \mathbin{[\!]} \cdots \mathbin{[\!]} l_n : P_n\}$	branching
	$s \triangleleft l.P$	selection
	$\mathbf{0}$	nil process
	$P \mid Q$	parallel composition of P and Q
	$(\nu\, s)P, (\nu\, a)P$	fresh name generation
	$\mathtt{def}\ D\ \mathtt{in}\ P$	recursion definition
	$X\langle\widetilde{es}$	
	$\mathtt{if}\ e\ \mathtt{t}$	
$u ::=$	a, b, x	shared name and variable
$e ::=$	$v, e\ \mathtt{or}\ e, e\ \mathtt{and}\ e, \mathtt{not}\ e$	expressions
$v ::=$	$\mathtt{true}, \mathtt{false}, a$	values
$D ::=$	$X_1(\widetilde{x_1}\widetilde{s_1}) = P_1,\ \ldots,\ X_n(\widetilde{x_n}\widetilde{s_n}) = P_n$	declaration for recursion

As before the system is a parallel composition of processes

SESSION CALCULUS: SEMANTICS

(Link) $\overline{a}(s).P_1 \mid a(s).P_2 \longrightarrow (\nu\, s)(P_1 \mid P_2)$

(Com) $\overline{s}\langle\widetilde{e}\rangle.P_1 \mid s(\widetilde{x}).P_2 \longrightarrow P_1 \mid P_2\{\widetilde{v}/\widetilde{x}\}\quad (e_i \downarrow v_i)$

(Label) $s \lhd l.P \mid s \rhd \{l_1 : P_1 \,[\!]\, \cdots \,[\!]\, l_n : P_n\} \longrightarrow P \mid P_i\quad (1 \le i \le n)$

(Pass) $\overline{s}(s').P_1 \mid s(s').P_2 \longrightarrow P_1 \mid P_2$

(Def) $\mathtt{def}\ D\ \mathtt{in}\ (X\langle\widetilde{e}\widetilde{s}\rangle \mid Q) \longrightarrow\ \mathtt{def}\ D\ \mathtt{in}\ (P\{\widetilde{v}/\widetilde{x}\} \mid Q)\quad (e_i \downarrow v_i, X(\widetilde{x}\widetilde{s}) = P \in D)$

(IF1) $\mathtt{if}\ e\ \mathtt{then}\ P_1\ \mathtt{else}\ P_2 \longrightarrow\ P_1\quad (e \downarrow \mathtt{true})$

(IF2) $\mathtt{if}\ e\ \mathtt{then}\ P_1\ \mathtt{else}\ P_2 \longrightarrow\ P_2\quad (e \downarrow \mathtt{false})$

EXAMPLE: A VARIABLE AGENT

A variable agent stores a value and offers the following operations:
1) read returns the stored value and recurs to the same variable;
2) write receives a different value and returns to the variable with the new state;

A Variable Agent:

$$\mathbf{Var}(a, x) \stackrel{\mathrm{df}}{=} ?$$

Reader Process:

$$\mathbf{Reader}(a) \stackrel{\mathrm{df}}{=} ?$$

EXAMPLE: A VARIABLE AGENT

A variable agent stores a value and offers the following operations:
1) read returns the stored value and recurs to the same variable;
2) write receives a different value and returns to the variable with the new state;

A Variable Agent:

$$\mathbf{Var}(a,x) \stackrel{\mathsf{df}}{=} a(s).s \triangleright [\mathsf{read} : \overline{s}\langle x\rangle.\mathbf{Var}\langle a,x\rangle \;[\!]\; \mathsf{write} : s(y).\mathbf{Var}\langle a,y\rangle]$$

Reader Process:

$$\mathbf{Reader}(a) \stackrel{\mathsf{df}}{=} \overline{a}(s).s \triangleleft \mathsf{read}.s(y).\mathbf{0}$$

Write Process:

$$\mathbf{Writer}(a,x) \stackrel{\mathsf{df}}{=} \overline{a}(s).s \triangleleft \mathsf{write}.\overline{s}\langle x\rangle.\mathbf{0}$$

Updating a value:

$$\mathbf{Var}\langle a,0\rangle \mid \mathbf{Writer}\langle a,5\rangle \longrightarrow \mathbf{Var}\langle a,5\rangle$$

SEMANTICS BY EXAMPLE: PROCESS DEFINITION RULE

$$(\text{Def}) \quad \texttt{def } D \texttt{ in } (X\langle\widetilde{es}\rangle \mid Q) \longrightarrow \texttt{def } D \texttt{ in } (P\{\widetilde{v}/\widetilde{x}\} \mid Q) \quad (e_i \downarrow v_i, X(\widetilde{xs}) = P \in D)$$

$$\mathbf{Var}(a,x) \stackrel{\mathsf{df}}{=} a(s).s \triangleright [\mathsf{read} : \overline{s}\langle x\rangle.\mathbf{Var}\langle a,x\rangle \; [\!] \; \mathsf{write} : s(y).\mathbf{Var}\langle a,y\rangle]$$

$$\mathbf{Writer}(a,x) \stackrel{\mathsf{df}}{=} \overline{a}(s).s \triangleleft \mathsf{write}.\overline{s}\langle x\rangle.\mathbf{0}$$

`in`

$$\mathbf{Var}\langle a, 0\rangle \mid \mathbf{Writer}\langle a, 5\rangle$$

$$\longrightarrow$$

SEMANTICS BY EXAMPLE: PROCESS DEFINITION RULE

$$\text{(Def)} \quad \texttt{def } D \texttt{ in } (X\langle\widetilde{es}\rangle \,|\, Q) \longrightarrow \texttt{def } D \texttt{ in } (P\{\tilde{v}/\tilde{x}\} \,|\, Q) \quad (e_i \downarrow v_i,\, X(\widetilde{xs}) = P \in D)$$

$$\mathbf{Var}(a,x) \stackrel{\text{df}}{=} a(s).s \triangleright [\mathsf{read} : \overline{s}\langle x\rangle.\mathbf{Var}\langle a,x\rangle \;[\!]\; \mathsf{write} : s(y).\mathbf{Var}\langle a,y\rangle]$$

$$\mathbf{Writer}(a,x) \stackrel{\text{df}}{=} \overline{a}(s).s \triangleleft \mathsf{write}.\overline{s}\langle x\rangle.\mathbf{0}$$

`in`

$$\mathbf{Var}\langle a,0\rangle \,|\, \mathbf{Writer}\langle a,5\rangle$$

$$\longrightarrow a(s).s \triangleright [\mathsf{read} : \overline{s}\langle x\rangle.\mathbf{Var}\langle a,x\rangle \;[\!]\; \mathsf{write} : s(y).\mathbf{Var}\langle a,y\rangle]\{0/x\}$$
$$|\, \overline{a}(s).s \triangleleft \mathsf{write}.\overline{s}\langle x\rangle.\mathbf{0}\{5/x\}$$

$$\text{(Def)}\quad \mathtt{def}\ D\ \mathtt{in}\ (X\langle\widetilde{es}\rangle \mid Q) \longrightarrow \mathtt{def}\ D\ \mathtt{in}\ (P\{\tilde{v}/\tilde{x}\} \mid Q)\quad (e_i \downarrow v_i,\ X(\widetilde{x}\widetilde{s}) = P \in D)$$

$$\begin{aligned}
&\mathbf{Var}\langle a, 0\rangle \mid \mathbf{Writer}\langle a, 5\rangle \\
&\longrightarrow a(s).s \triangleright [\mathsf{read} : \overline{s}\langle x\rangle.\mathbf{Var}\langle a, x\rangle \mathbin{[\!]} \mathsf{write} : s(y).\mathbf{Var}\langle a, y\rangle]\{0/x\} \\
&\quad \mid \overline{a}(s).s \triangleleft \mathsf{write}.\overline{s}\langle x\rangle.\mathbf{0}\{5/x\}
\end{aligned}$$

$$\begin{aligned}
&\equiv a(s).s \triangleright [\mathsf{read} : \overline{s}\langle 0\rangle.\mathbf{Var}\langle a, 0\rangle \mathbin{[\!]} \mathsf{write} : s(y).\mathbf{Var}\langle a, y\rangle] \\
&\quad \mid \overline{a}(s).s \triangleleft \mathsf{write}.\overline{s}\langle 5\rangle.\mathbf{0} \\
&\longrightarrow (\nu\, s)(s \triangleright [\mathsf{read} : \overline{s}\langle 0\rangle.\mathbf{Var}\langle a, 0\rangle \mathbin{[\!]} \mathsf{write} : s(y).\mathbf{Var}\langle a, y\rangle] \\
&\quad \mid\ s \triangleleft \mathsf{write}.\overline{s}\langle 5\rangle.\mathbf{0}) \\
&\longrightarrow (\nu\, s)(s(y).\mathbf{Var}\langle a, y\rangle \mid \overline{s}\langle 5\rangle.\mathbf{0}) \\
&\longrightarrow (\nu\, s)(\mathbf{Var}\langle a, y\rangle\{5/y\} \mid \mathbf{0}) \\
&\equiv \mathbf{Var}\langle 5\rangle
\end{aligned}$$

$$(\text{Def})\quad \texttt{def}\ D\ \texttt{in}\ (X\langle\widetilde{es}\rangle \mid Q) \longrightarrow \texttt{def}\ D\ \texttt{in}\ (P\{\tilde{v}/\tilde{x}\} \mid Q)\quad (e_i \downarrow v_i, X(\widetilde{xs}) = P \in D)$$

$$\begin{aligned}
&\mathbf{Var}\langle a, 0\rangle \mid \mathbf{Writer}\langle a, 5\rangle \\
&\longrightarrow a(s).s \triangleright [\mathsf{read} : \overline{s}\langle x\rangle.\mathbf{Var}\langle a, x\rangle \mathbin{[\!]} \mathsf{write} : s(y).\mathbf{Var}\langle a, y\rangle]\{0/x\} \\
&\quad \mid \overline{a}(s).s \triangleleft \mathsf{write}.\overline{s}\langle x\rangle.\mathbf{0}\{5/x\}
\end{aligned}$$

$$\begin{aligned}
&= a(s).s \triangleright [\mathsf{read} : \overline{s}\langle 0\rangle.\mathbf{Var}\langle a, 0\rangle \mathbin{[\!]} \mathsf{write} : s(y).\mathbf{Var}\langle a, y\rangle] \\
&\quad \mid \overline{a}(s).s \triangleleft \mathsf{write}.\overline{s}\langle 5\rangle.\mathbf{0}
\end{aligned}$$

$$\begin{aligned}
&\longrightarrow (\nu\, s)(s \triangleright [\mathsf{read} : \overline{s}\langle 0\rangle.\mathbf{Var}\langle a, 0\rangle \mathbin{[\!]} \mathsf{write} : s(y).\mathbf{Var}\langle a, y\rangle] \\
&\quad \mid s \triangleleft \mathsf{write}.\overline{s}\langle 5\rangle.\mathbf{0})
\end{aligned}$$

$$\longrightarrow (\nu\, s)(s(y).\mathbf{Var}\langle a, y\rangle \mid \overline{s}\langle 5\rangle.\mathbf{0})$$

$$\begin{aligned}
&\longrightarrow (\nu\, s)(\mathbf{Var}\langle a, y\rangle\{5/y\} \mid \mathbf{0}) \\
&\equiv \mathbf{Var}\langle 5\rangle
\end{aligned}$$

SEMANTICS BY EXAMPLE

$$\mathbf{Var}\langle a, 0\rangle \mid \mathbf{Writer}\langle a, 5\rangle$$

$$\longrightarrow a(s).s \triangleright [\mathsf{read} : \overline{s}\langle x\rangle.\mathbf{Var}\langle a, x\rangle \mathbin{[\!]} \mathsf{write} : s(y).\mathbf{Var}\langle a, y\rangle]\{^0/_x\}$$

$$\quad \mid \overline{a}(s).s \triangleleft \mathsf{write}.\overline{s}\langle x\rangle.\mathbf{0}\{^5/_x\}$$

$$= a(s).s \triangleright [\mathsf{read} : \overline{s}\langle 0\rangle.\mathbf{Var}\langle a, 0\rangle \mathbin{[\!]} \mathsf{write} : s(y).\mathbf{Var}\langle a, y\rangle]$$

$$\quad \mid \overline{a}(s).s \triangleleft \mathsf{write}.\overline{s}\langle 5\rangle.\mathbf{0}$$

$$\longrightarrow (\nu\, s)(s \triangleright [\mathsf{read} : \overline{s}\langle 0\rangle.\mathbf{Var}\langle a, 0\rangle \mathbin{[\!]} \mathsf{write} : s(y).\mathbf{Var}\langle a, y\rangle]$$

$$\quad \mid s \triangleleft \mathsf{write}.\overline{s}\langle 5\rangle.\mathbf{0})$$

$$\longrightarrow (\nu\, s)(s(y).\mathbf{Var}\langle a, y\rangle \mid \overline{s}\langle 5\rangle.\mathbf{0})$$

$$\longrightarrow (\nu\, s)(\mathbf{Var}\langle a, y\rangle\{^5/_y\} \mid \mathbf{0})$$

$$\equiv \mathbf{Var}\langle 5\rangle$$

$$(\mathsf{Link}) \quad \overline{a}(s).P_1 \mid a(s).P_2 \longrightarrow (\nu\, s)(P_1 \mid P_2)$$

$$\begin{aligned}
&\mathrm{Var}\langle a, 0\rangle \mid \mathrm{Writer}\langle a, 5\rangle \\
&\longrightarrow a(s).s \triangleright [\mathsf{read} : \overline{s}\langle x\rangle.\mathrm{Var}\langle a, x\rangle \mathbin{[\!]} \mathsf{write} : s(y).\mathrm{Var}\langle a, y\rangle]\{^0/_x\} \\
&\quad \mid \overline{a}(s).s \triangleleft \mathsf{write}.\overline{s}\langle x\rangle.\mathbf{0}\{^5/_x\}
\end{aligned}$$

$$\begin{aligned}
&= a(s).s \triangleright [\mathsf{read} : \overline{s}\langle 0\rangle.\mathbf{Var}\langle a, 0\rangle \mathbin{[\!]} \mathsf{write} : s(y).\mathbf{Var}\langle a, y\rangle] \\
&\quad \mid \overline{a}(s).s \triangleleft \mathsf{write}.\overline{s}\langle 5\rangle.\mathbf{0}
\end{aligned}$$

$$\begin{aligned}
&\longrightarrow (\nu\, s)(s \triangleright [\mathsf{read} : \overline{s}\langle 0\rangle.\mathrm{Var}\langle a, 0\rangle \mathbin{[\!]} \mathsf{write} : s(y).\mathrm{Var}\langle a, y\rangle] \\
&\quad \mid \; s \triangleleft \mathsf{write}.\overline{s}\langle 5\rangle.\mathbf{0}) \\
&\longrightarrow (\nu\, s)(s(y).\mathrm{Var}\langle a, y\rangle \mid \overline{s}\langle 5\rangle.\mathbf{0}) \\
&\longrightarrow (\nu\, s)(\mathrm{Var}\langle a, y\rangle\{^5/_y\} \mid \mathbf{0}) \\
&\equiv \mathrm{Var}\langle 5\rangle
\end{aligned}$$

$$\text{(Link)}\quad \overline{a}(s).P_1 \mid a(s).P_2 \longrightarrow (\nu\, s)(P_1 \mid P_2)$$

$$\begin{aligned}
&\mathrm{Var}\langle a, 0\rangle \mid \mathrm{Writer}\langle a, 5\rangle \\
&\longrightarrow a(s).s \rhd [\mathsf{read} : \overline{s}\langle x\rangle.\mathrm{Var}\langle a, x\rangle \;[\!]\; \mathsf{write} : s(y).\mathrm{Var}\langle a, y\rangle]\{^0/_x\} \\
&\quad \mid \overline{a}(s).s \lhd \mathsf{write}.\overline{s}\langle x\rangle.\mathbf{0}\{^5/_x\}
\end{aligned}$$

$$\begin{aligned}
&= a(s).s \rhd [\mathsf{read} : \overline{s}\langle 0\rangle.\mathbf{Var}\langle a, 0\rangle \;[\!]\; \mathsf{write} : s(y).\mathbf{Var}\langle a, y\rangle] \\
&\quad \mid \overline{a}(s).s \lhd \mathsf{write}.\overline{s}\langle 5\rangle.\mathbf{0}
\end{aligned}$$

$$\begin{aligned}
&\longrightarrow (\nu\, s)(s \rhd [\mathsf{read} : \overline{s}\langle 0\rangle.\mathrm{Var}\langle a, 0\rangle \;[\!]\; \mathsf{write} : s(y).\mathrm{Var}\langle a, y\rangle] \\
&\quad \mid s \lhd \mathsf{write}.\overline{s}\langle 5\rangle.\mathbf{0}) \\
&\longrightarrow (\nu\, s)(s(y).\mathrm{Var}\langle a, y\rangle \mid \overline{s}\langle 5\rangle.\mathbf{0}) \\
&\longrightarrow (\nu\, s)(\mathrm{Var}\langle a, y\rangle\{^5/_y\} \mid \mathbf{0}) \\
&\equiv \mathrm{Var}\langle 5\rangle
\end{aligned}$$

(Link) $\overline{a}(s).P_1 \mid a(s).P_2 \longrightarrow (\nu\, s)(P_1 \mid P_2)$

$$\mathbf{Var}\langle a, 0\rangle \mid \mathbf{Writer}\langle a, 5\rangle$$
$$\longrightarrow a(s).s \rhd [\mathsf{read} : \overline{s}\langle x\rangle.\mathbf{Var}\langle a, x\rangle \,[\!]\, \mathsf{write} : s(y).\mathbf{Var}\langle a, y\rangle]\{^0/_x\}$$
$$\mid \overline{a}(s).s \lhd \mathsf{write}.\overline{s}\langle x\rangle.\mathbf{0}\{^5/_x\}$$

$$\equiv a(s).s \rhd [\mathsf{read} : \overline{s}\langle 0\rangle.\mathbf{Var}\langle a, 0\rangle \,[\!]\, \mathsf{write} : s(y).\mathbf{Var}\langle a, y\rangle]$$
$$\mid \overline{a}(s).s \lhd \mathsf{write}.\overline{s}\langle 5\rangle.\mathbf{0}$$

$$\longrightarrow (\nu\, s)(s \rhd [\mathsf{read} : \overline{s}\langle 0\rangle.\mathbf{Var}\langle a, 0\rangle \,[\!]\, \mathsf{write} : s(y).\mathbf{Var}\langle a, y\rangle]$$
$$\mid s \lhd \mathsf{write}.\overline{s}\langle 5\rangle.\mathbf{0})$$

$$\longrightarrow (\nu\, s)(s(y).\mathbf{Var}\langle a, y\rangle \mid \overline{s}\langle 5\rangle.\mathbf{0})$$

$$\longrightarrow (\nu\, s)(\mathbf{Var}\langle a, y\rangle\{^5/_y\} \mid \mathbf{0})$$
$$\equiv \mathbf{Var}\langle 5\rangle$$

SEMANTICS BY EXAMPLE

$$\mathbf{Var}\langle a, 0\rangle \mid \mathbf{Writer}\langle a, 5\rangle$$
$$\longrightarrow a(s).s \triangleright [\mathsf{read} : \overline{s}\langle x\rangle.\mathbf{Var}\langle a, x\rangle \mathbin{[\!]} \mathsf{write} : s(y).\mathbf{Var}\langle a, y\rangle]\{^0/_x\}$$
$$\quad \mid \overline{a}(s).s \triangleleft \mathsf{write}.\overline{s}\langle x\rangle.\mathbf{0}\{^5/_x\}$$

$$\equiv a(s).s \triangleright [\mathsf{read} : \overline{s}\langle 0\rangle.\mathbf{Var}\langle a, 0\rangle \mathbin{[\!]} \mathsf{write} : s(y).\mathbf{Var}\langle a, y\rangle]$$
$$\quad \mid \overline{a}(s).s \triangleleft \mathsf{write}.\overline{s}\langle 5\rangle.\mathbf{0}$$

$$\longrightarrow (\nu\, s)(s \triangleright [\mathsf{read} : \overline{s}\langle 0\rangle.\mathbf{Var}\langle a, 0\rangle \mathbin{[\!]} \mathsf{write} : s(y).\mathbf{Var}\langle a, y\rangle]$$
$$\quad \mid s \triangleleft \mathsf{write}.\overline{s}\langle 5\rangle.\mathbf{0})$$

$$\longrightarrow (\nu\, s)(s(y).\mathbf{Var}\langle a, y\rangle \mid \overline{s}\langle 5\rangle.\mathbf{0})$$

$$\longrightarrow (\nu\, s)(\mathbf{Var}\langle a, y\rangle\{^5/_y\} \mid \mathbf{0})$$
$$\equiv \mathbf{Var}\langle 5\rangle$$

(Label) $s \triangleleft l.P \mid s \triangleright \{l_1 : P_1 \,[\!]\, \cdots \,[\!]\, l_n : P_n\} \longrightarrow P \mid P_i \quad (1 \leq i \leq n)$

$$\mathbf{Var}\langle a, 0\rangle \mid \mathbf{Writer}\langle a, 5\rangle$$
$$\longrightarrow a(s).s \triangleright [\mathsf{read} : \overline{s}\langle x\rangle.\mathbf{Var}\langle a, x\rangle \,[\!]\, \mathsf{write} : s(y).\mathbf{Var}\langle a, y\rangle]\{^0/_x\}$$
$$\mid \overline{a}(s).s \triangleleft \mathsf{write}.\overline{s}\langle x\rangle.\mathbf{0}\{^5/_x\}$$

$$\equiv a(s).s \triangleright [\mathsf{read} : \overline{s}\langle 0\rangle.\mathbf{Var}\langle a, 0\rangle \,[\!]\, \mathsf{write} : s(y).\mathbf{Var}\langle a, y\rangle]$$
$$\mid \overline{a}(s).s \triangleleft \mathsf{write}.\overline{s}\langle 5\rangle.\mathbf{0}$$

$$\longrightarrow (\nu\, s)(s \triangleright [\mathsf{read} : \overline{s}\langle 0\rangle.\mathbf{Var}\langle a, 0\rangle \,[\!]\, \mathsf{write} : s(y).\mathbf{Var}\langle a, y\rangle]$$
$$\mid s \triangleleft \mathsf{write}.\overline{s}\langle 5\rangle.\mathbf{0})$$

$$\longrightarrow (\nu\, s)(s(y).\mathbf{Var}\langle a, y\rangle \mid \overline{s}\langle 5\rangle.\mathbf{0})$$

$$\longrightarrow (\nu\, s)(\mathbf{Var}\langle a, y\rangle\{^5/_y\} \mid \mathbf{0})$$
$$\equiv \mathbf{Var}\langle 5\rangle$$

(Label) $s \triangleleft l.P \mid s \triangleright \{l_1 : P_1 \talloblong \cdots \talloblong l_n : P_n\} \longrightarrow P \mid P_i \quad (1 \le i \le n)$

$$\mathbf{Var}\langle a, 0\rangle \mid \mathbf{Writer}\langle a, 5\rangle$$
$$\longrightarrow a(s).s \triangleright [\mathsf{read} : \overline{s}\langle x\rangle.\mathbf{Var}\langle a, x\rangle \talloblong \mathsf{write} : s(y).\mathbf{Var}\langle a, y\rangle]\{^0/_x\}$$
$$\mid \overline{a}(s).s \triangleleft \mathsf{write}.\overline{s}\langle x\rangle.\mathbf{0}\{^5/_x\}$$

$$\equiv a(s).s \triangleright [\mathsf{read} : \overline{s}\langle 0\rangle.\mathbf{Var}\langle a, 0\rangle \talloblong \mathsf{write} : s(y).\mathbf{Var}\langle a, y\rangle]$$
$$\mid \overline{a}(s).s \triangleleft \mathsf{write}.\overline{s}\langle 5\rangle.\mathbf{0}$$

$$\longrightarrow (\nu\, s)(s \triangleright [\mathsf{read} : \overline{s}\langle 0\rangle.\mathbf{Var}\langle a, 0\rangle \talloblong \mathsf{write} : s(y).\mathbf{Var}\langle a, y\rangle]$$
$$\mid s \triangleleft \mathsf{write}.\overline{s}\langle 5\rangle.\mathbf{0})$$

$$\longrightarrow (\nu\, s)(s(y).\mathbf{Var}\langle a, y\rangle \mid \overline{s}\langle 5\rangle.\mathbf{0})$$

$$\longrightarrow (\nu\, s)(\mathbf{Var}\langle a, y\rangle\{^5/_y\} \mid \mathbf{0})$$
$$\equiv \mathbf{Var}\langle 5\rangle$$

SEMANTICS BY EXAMPLE

$$\mathbf{Var}\langle a, 0\rangle \mid \mathbf{Writer}\langle a, 5\rangle$$
$$\longrightarrow a(s).s \triangleright [\mathsf{read} : \overline{s}\langle x\rangle.\mathbf{Var}\langle a, x\rangle \mathbin{[\!]} \mathsf{write} : s(y).\mathbf{Var}\langle a, y\rangle]\{^{0}/_{x}\}$$
$$\mid \overline{a}(s).s \triangleleft \mathsf{write}.\overline{s}\langle x\rangle.\mathbf{0}\{^{5}/_{x}\}$$

$$\equiv a(s).s \triangleright [\mathsf{read} : \overline{s}\langle 0\rangle.\mathbf{Var}\langle a, 0\rangle \mathbin{[\!]} \mathsf{write} : s(y).\mathbf{Var}\langle a, y\rangle]$$
$$\mid \overline{a}(s).s \triangleleft \mathsf{write}.\overline{s}\langle 5\rangle.\mathbf{0}$$

$$\longrightarrow (\nu\, s)(s \triangleright [\mathsf{read} : \overline{s}\langle 0\rangle.\mathbf{Var}\langle a, 0\rangle \mathbin{[\!]} \mathsf{write} : s(y).\mathbf{Var}\langle a, y\rangle]$$
$$\mid s \triangleleft \mathsf{write}.\overline{s}\langle 5\rangle.\mathbf{0})$$

$$\longrightarrow (\nu\, s)(s(y).\mathbf{Var}\langle a, y\rangle \mid \overline{s}\langle 5\rangle.\mathbf{0})$$

$$\longrightarrow (\nu\, s)(\mathbf{Var}\langle a, y\rangle\{^{5}/_{y}\} \mid \mathbf{0})$$
$$\equiv \mathbf{Var}\langle 5\rangle$$

$$(\mathsf{Com}) \quad \overline{s}\langle\widetilde{e}\rangle.P_1 \mid s(\widetilde{x}).P_2 \longrightarrow P_1 \mid P_2\{\widetilde{v}/\widetilde{x}\} \quad (e_i \downarrow v_i)$$

$$\begin{aligned}
&\mathrm{Var}\langle a, 0\rangle \mid \mathrm{Writer}\langle a, 5\rangle \\
&\longrightarrow a(s).s \triangleright [\mathsf{read} : \overline{s}\langle x\rangle.\mathrm{Var}\langle a, x\rangle \;[]\; \mathsf{write} : s(y).\mathrm{Var}\langle a, y\rangle]\{0/x\} \\
&\quad \mid \overline{a}(s).s \triangleleft \mathsf{write}.\overline{s}\langle x\rangle.\mathbf{0}\{5/x\} \\
&\equiv a(s).s \triangleright [\mathsf{read} : \overline{s}\langle 0\rangle.\mathrm{Var}\langle a, 0\rangle \;[]\; \mathsf{write} : s(y).\mathrm{Var}\langle a, y\rangle] \\
&\quad \mid \overline{a}(s).s \triangleleft \mathsf{write}.\overline{s}\langle 5\rangle.\mathbf{0} \\
&\longrightarrow (\nu\, s)(s \triangleright [\mathsf{read} : \overline{s}\langle 0\rangle.\mathrm{Var}\langle a, 0\rangle \;[]\; \mathsf{write} : s(y).\mathrm{Var}\langle a, y\rangle] \\
&\quad \mid \; s \triangleleft \mathsf{write}.\overline{s}\langle 5\rangle.\mathbf{0}) \\
&\longrightarrow (\nu\, s)(s(y).\mathbf{Var}\langle a, y\rangle \mid \overline{s}\langle 5\rangle.\mathbf{0}) \\
&\longrightarrow (\nu\, s)(\mathrm{Var}\langle a, y\rangle\{5/y\} \mid \mathbf{0}) \\
&\equiv \mathrm{Var}\langle 5\rangle
\end{aligned}$$

(Com) $\overline{s}\langle\widetilde{e}\rangle.P_1 \mid s(\widetilde{x}).P_2 \longrightarrow P_1 \mid P_2\{\widetilde{v}/\widetilde{x}\} \quad (e_i \downarrow v_i)$

$$\mathbf{Var}\langle a, 0\rangle \mid \mathbf{Writer}\langle a, 5\rangle$$

$$\longrightarrow a(s).s \triangleright [\mathsf{read} : \overline{s}\langle x\rangle.\mathbf{Var}\langle a, x\rangle \; [\!] \; \mathsf{write} : s(y).\mathbf{Var}\langle a, y\rangle]\{0/x\}$$
$$\mid \overline{a}(s).s \triangleleft \mathsf{write}.\overline{s}\langle x\rangle.\mathbf{0}\{5/x\}$$

$$\equiv a(s).s \triangleright [\mathsf{read} : \overline{s}\langle 0\rangle.\mathbf{Var}\langle a, 0\rangle \; [\!] \; \mathsf{write} : s(y).\mathbf{Var}\langle a, y\rangle]$$
$$\mid \overline{a}(s).s \triangleleft \mathsf{write}.\overline{s}\langle 5\rangle.\mathbf{0}$$

$$\longrightarrow (\nu\, s)(s \triangleright [\mathsf{read} : \overline{s}\langle 0\rangle.\mathbf{Var}\langle a, 0\rangle \; [\!] \; \mathsf{write} : s(y).\mathbf{Var}\langle a, y\rangle]$$
$$\mid s \triangleleft \mathsf{write}.\overline{s}\langle 5\rangle.\mathbf{0})$$

$$\longrightarrow (\nu\, s)(s(y).\mathbf{Var}\langle a, y\rangle \mid \overline{s}\langle 5\rangle.\mathbf{0})$$

$$\longrightarrow (\nu\, s)(\mathbf{Var}\langle a, y\rangle\{5/y\} \mid \mathbf{0})$$

$$\equiv \mathbf{Var}\langle 5\rangle$$

SEMANTICS BY EXAMPLE

$$\mathbf{Var}\langle a, 0\rangle \mid \mathbf{Writer}\langle a, 5\rangle$$
$$\longrightarrow a(s).s \triangleright [\mathsf{read} : \overline{s}\langle x\rangle.\mathbf{Var}\langle a, x\rangle \mathbin{[\!]} \mathsf{write} : s(y).\mathbf{Var}\langle a, y\rangle]\{^0/_x\}$$
$$\mid \overline{a}(s).s \triangleleft \mathsf{write}.\overline{s}\langle x\rangle.\mathbf{0}\{^5/_x\}$$

$$\equiv a(s).s \triangleright [\mathsf{read} : \overline{s}\langle 0\rangle.\mathbf{Var}\langle a, 0\rangle \mathbin{[\!]} \mathsf{write} : s(y).\mathbf{Var}\langle a, y\rangle]$$
$$\mid \overline{a}(s).s \triangleleft \mathsf{write}.\overline{s}\langle 5\rangle.\mathbf{0}$$

$$\longrightarrow (\nu\, s)(s \triangleright [\mathsf{read} : \overline{s}\langle 0\rangle.\mathbf{Var}\langle a, 0\rangle \mathbin{[\!]} \mathsf{write} : s(y).\mathbf{Var}\langle a, y\rangle]$$
$$\mid\ s \triangleleft \mathsf{write}.\overline{s}\langle 5\rangle.\mathbf{0})$$

$$\longrightarrow (\nu\, s)(s(y).\mathbf{Var}\langle a, y\rangle \mid \overline{s}\langle 5\rangle.\mathbf{0})$$

$$\longrightarrow (\nu\, s)(\mathbf{Var}\langle a, y\rangle\{^5/_y\} \mid \mathbf{0})$$
$$\equiv \mathbf{Var}\langle 5\rangle$$

SEMANTICS BY EXAMPLE

$$\begin{aligned}
&\mathbf{Var}\langle a, 0\rangle \mid \mathbf{Writer}\langle a, 5\rangle \\
&\longrightarrow a(s).s \triangleright [\mathsf{read} : \overline{s}\langle x\rangle.\mathbf{Var}\langle a, x\rangle \mathbin{[\!]} \mathsf{write} : s(y).\mathbf{Var}\langle a, y\rangle]\{^0/_x\} \\
&\quad \mid \overline{a}(s).s \triangleleft \mathsf{write}.\overline{s}\langle x\rangle.\mathbf{0}\{^5/_x\} \\
&\equiv a(s).s \triangleright [\mathsf{read} : \overline{s}\langle 0\rangle.\mathbf{Var}\langle a, 0\rangle \mathbin{[\!]} \mathsf{write} : s(y).\mathbf{Var}\langle a, y\rangle] \\
&\quad \mid \overline{a}(s).s \triangleleft \mathsf{write}.\overline{s}\langle 5\rangle.\mathbf{0} \\
&\longrightarrow (\nu\, s)(s \triangleright [\mathsf{read} : \overline{s}\langle 0\rangle.\mathbf{Var}\langle a, 0\rangle \mathbin{[\!]} \mathsf{write} : s(y).\mathbf{Var}\langle a, y\rangle] \\
&\quad \mid \ s \triangleleft \mathsf{write}.\overline{s}\langle 5\rangle.\mathbf{0}) \\
&\longrightarrow (\nu\, s)(s(y).\mathbf{Var}\langle a, y\rangle \mid \overline{s}\langle 5\rangle.\mathbf{0}) \\
&\longrightarrow (\nu\, s)(\mathbf{Var}\langle a, y\rangle\{^5/_y\} \mid \mathbf{0}) \\
&\equiv \mathbf{Var}\langle 5\rangle
\end{aligned}$$

EXERCISE

$$\mathbf{Var}\langle a, 5\rangle \mid \mathbf{Reader}\langle a\rangle \longrightarrow ?$$

EXERCISE

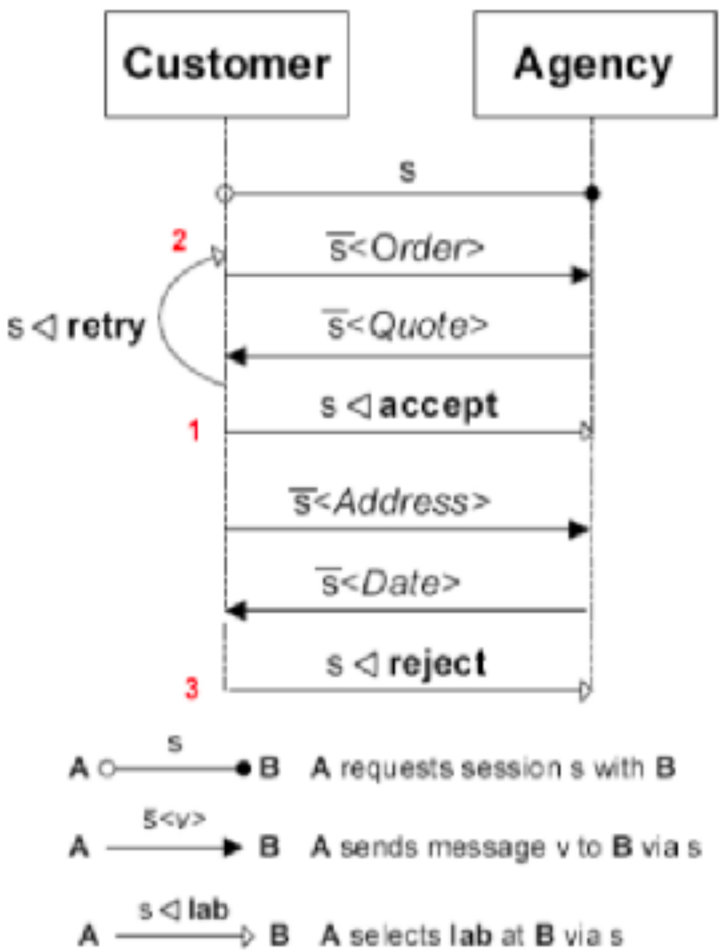
$$\mathbf{Var}(a,x) \stackrel{\mathsf{df}}{=} a(s).s \triangleright [\mathsf{read} : \overline{s}\langle x\rangle.\mathbf{Var}\langle a,x\rangle \mathbin{[\!]} \mathsf{write} : s(y).\mathbf{Var}\langle a,y\rangle]$$

$$\mathbf{Reader}(a) \stackrel{\mathsf{df}}{=} \overline{a}(s).s \triangleleft \mathsf{read}.s(y).\mathbf{0}$$

`in`

$$\mathbf{Var}\langle a,5\rangle \mid \mathbf{Reader}\langle a\rangle \longrightarrow ?$$

EXAMPLE: TRAVEL AGENCY

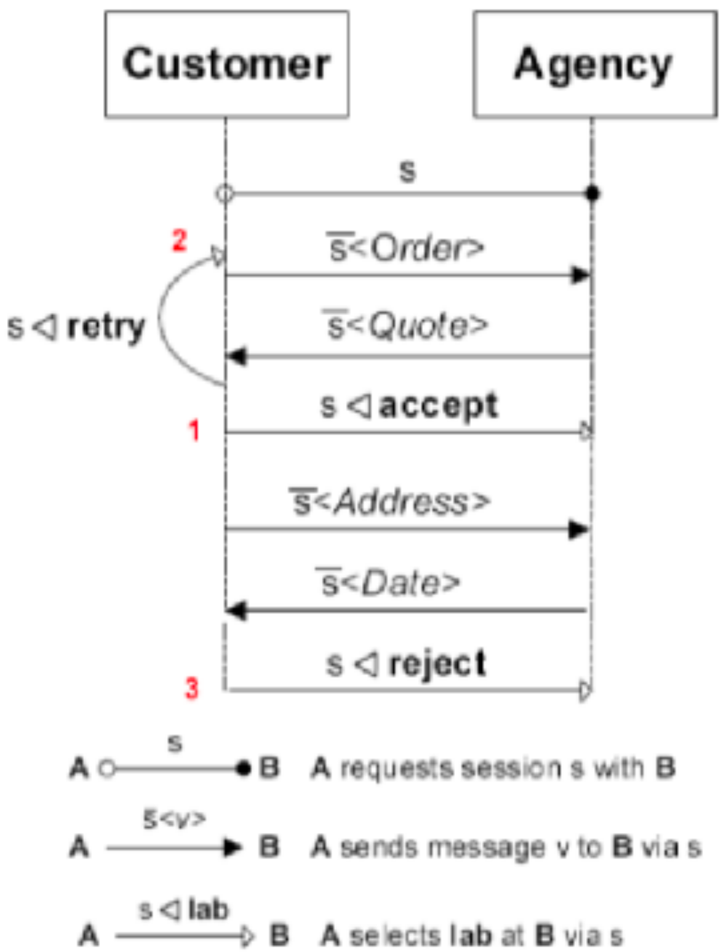


Web Service Protocol (Usecase from WS-CDL)

Two parties are involved: a client (Customer) and a travel agency (Agency).

1. Customer begins an *order session* s with Agency, then requests and receives the price for the desired journey.
2. Customer either accepts (label accept) an offer from Agency or decides that none of the received quotes are satisfactory.
3. if the offer is accepted, the Customer sends a delivery address and the Agency Service replies with the dispatch date for the purchased tickets. The transaction is now complete.
4. Customer retries (label retry) transactions with new journeys some number of times if Agency gave are reasonable quote.
5. Customer rejects (label reject) the transaction if no quotes were suitable after some retries and the session terminates.

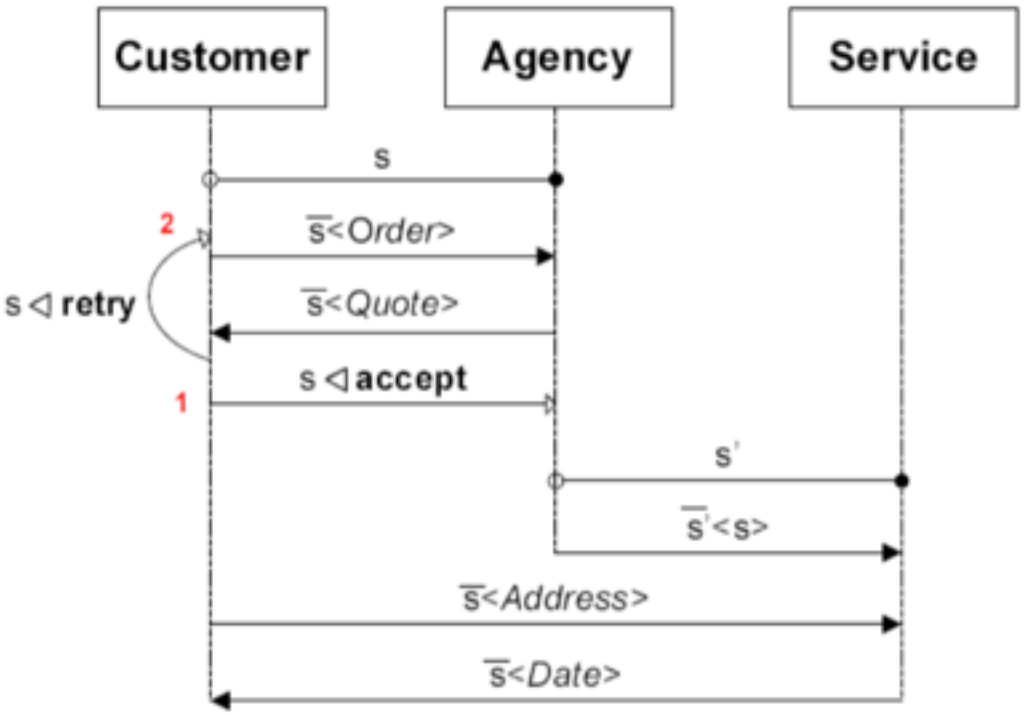
EXAMPLE: TRAVEL AGENCY

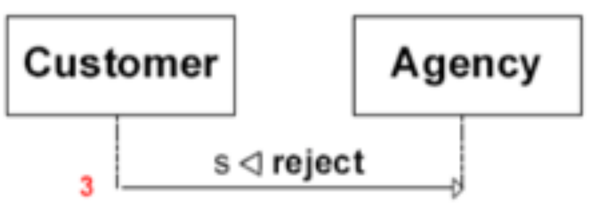


$$
\begin{array}{l}
\mathbf{Agency}(a,b) \stackrel{\text{df}}{=} a(s).\mathbf{Agency_1}\langle a,b,s\rangle \\
\mathbf{Agency_1}(a,b,s) \stackrel{\text{df}}{=} \quad s(x).\overline{s}\langle \mathtt{price}(x)\rangle \\
\qquad s \rhd [\;\; \mathsf{accept} : s(x).\overline{s}\langle \mathtt{date}\rangle.\mathbf{Agency}\langle a,b\rangle \;\; [\!] \\
\qquad\qquad \mathsf{retry} : \mathbf{Agency_1}\langle a,b,s\rangle \;\; [\!] \\
\qquad\qquad \mathsf{reject} : \mathbf{Agency}\langle a,b\rangle]
\end{array}
$$

$$
\begin{array}{ll}
\multicolumn{2}{l}{\mathbf{Customer}(a, place, i, n) \stackrel{\text{df}}{=} \overline{a}(s).\mathbf{Customer_1}\langle s, place, i, n\rangle} \\
\mathbf{Customer_1}(s, place, i, n) \stackrel{\text{df}}{=} & \overline{s}\langle place(i)\rangle.s(price). \\
\mathtt{if\ is\_acceptable}(price)\ \mathtt{then} & s \lhd \mathsf{accept}.\overline{s}\langle \mathtt{address}\rangle.s(date) \\
\mathtt{elseif}\ i \leq n\ \mathtt{then} & s \lhd \mathsf{retry}.\mathbf{Customer_1}\langle s, place, i+1, n\rangle \\
\mathtt{else} & s \lhd \mathsf{reject}
\end{array}
$$

DELEGATION

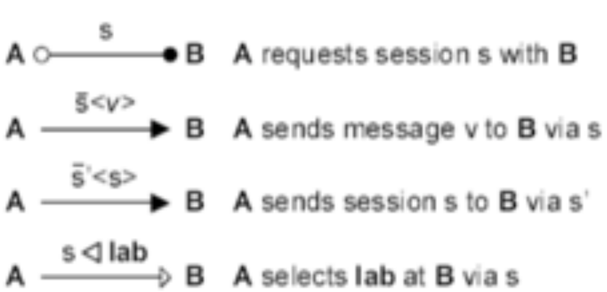




Web Service Protocol (Usecase from WS-CDL)

Adding a third party - A Travel Service. Customer and Service are initially unknown to each other but later communicate directly through the use of *name passing*.

1. Customer begins an *order session* s with Agency, then requests and receives the price for the desired journey.
2. Customer either accepts an offer from Agency or decides that none of the received quotes are satisfactory.
3. new: If an offer is accepted, Agency opens the session s' with Service and *delegates* to Service, through s', the interactions with Customer remaining for s.
4. new: Customer then sends a delivery address (unaware that he/she is now talking to Service), and Service replies with the dispatch date for the purchased tickets. The transaction is now complete.
5. Customer retries transactions with new journeys some number of times if Agency gave are reasonable quote.
6. Customer rejects the transaction if no quotes were suitable after some retries and the session terminates.



DELEGATION

- The original idea of delegation in object-based concurrent programming allows an object to delegate the processing of a request to another object. Its basic purpose is distributing of processing, while maintaining the transparency of name space for clients of that service.

- Can we delegate the processing of a request in the current session calculus ?

- Refresh on the syntax:

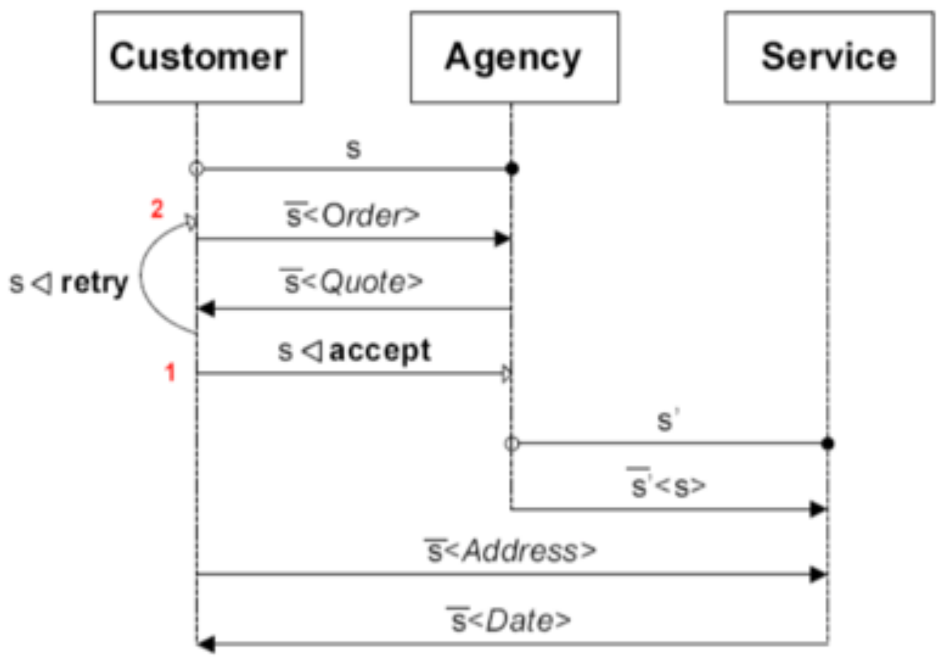
$\overline{s}(s').P$	channel send
$s(s').P$	channel received

  Therefore, we can pass session channels: $\overline{s}(s')$

- Delegation: the ability to pass session channels

- Let's see how we can improve the Web Service Agency example using delegation.

DELEGATION



$$\mathbf{Agency}(a,b) \stackrel{\mathsf{df}}{=} a(y).\mathbf{Agency}_1\langle a,b,s\rangle$$

$$\mathbf{Agency}_1(a,b,s) \stackrel{\mathsf{df}}{=}$$

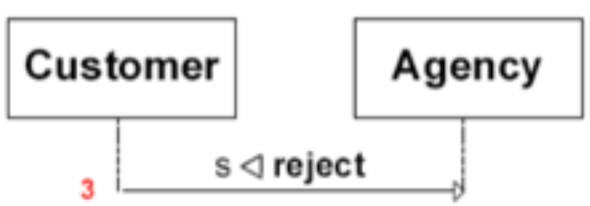
$$s(x).\overline{s}\langle \mathtt{price(x)}\rangle.s \rhd [\ \ \mathsf{accept}: \ \overline{b}(s').\overline{s'}(s).\mathbf{Agency}\langle a,b\rangle \ [] \ \ \mathsf{retry}: \ \mathbf{Agency}_1\langle a,b,s\rangle \ [] \ \ \mathsf{reject}: \ \mathbf{Agency}\langle a,b\rangle]$$

The message $\overline{s'}\langle s\rangle$ delegates the interaction with Customer to Service.

Service is defined as:

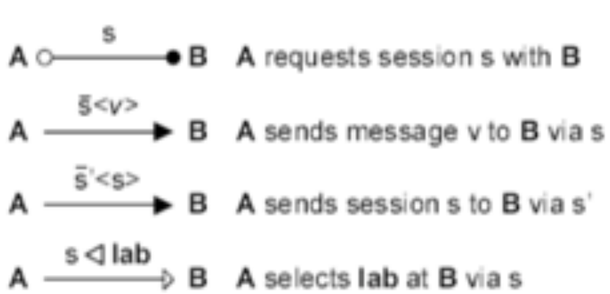
$$\mathbf{Service}(b) \stackrel{\mathsf{df}}{=} b(s').s'(s).s(address)\overline{s}\langle\mathtt{Date}\rangle.\mathbf{0}$$

where $s'(s)$ receives the session with Customer and continue the interaction with Customer as if it were Agency, e.g receives the delivery address and sends the delivery date.



$$\mathbf{Customer}(a, place, n) \stackrel{\mathsf{df}}{=} \overline{a}(s).\mathbf{Customer}_1\langle s, place, 1, n\rangle$$

$$\begin{array}{ll}
\mathbf{Customer}_1(s, place, i, n) \stackrel{\mathsf{df}}{=} & \overline{s}\langle place(i)\rangle.s(price). \\
\mathtt{if\ is_acceptable}(price)\ \mathtt{then} & s \lhd \mathsf{accept}.\overline{s}\langle\mathtt{Address}\rangle.s(date).\mathbf{0} \\
\mathtt{elseif}\ i \leq n\ \mathtt{then} & s \lhd \mathsf{retry}.\mathbf{Customer}_1\langle s, place, i+1, n\rangle \\
\mathtt{else} & s \lhd \mathsf{reject}.\mathbf{0}
\end{array}$$



ERRORS

We wish to avoid the following runtime errors in various protocols.

- Base Type Error $\overline{s}\langle \texttt{Apple} \rangle.P_1 \mid s(x).\overline{s'}\langle 1 + x \rangle$
- Arity Mismatch $\overline{s}\langle 1 \rangle.P_1 \mid s(x, y).\overline{s'}\langle x + y \rangle$
- Label Undefined $s \triangleright \{\mathsf{repeat} : P_1 \;[\!]\; \mathsf{reject} : P_2\} \mid s \triangleleft \mathsf{apple}$
- Race during Session Interaction

  Bad $s(x).P_1 \mid \overline{s}\langle v \rangle.P_2 \mid \overline{s}\langle w \rangle.P_3$

  Good $s(x).P_1 \mid \overline{s}\langle v \rangle.P_2 \mid s'(x).Q_1 \mid \overline{s'}\langle w \rangle.Q_2$
- Communication Mismatch

  Bad $s(x).\overline{s}\langle w \rangle.\mathbf{0} \mid s(y).\overline{s}\langle v \rangle.\mathbf{0}$ Good $s(x).\overline{s}\langle w \rangle.\mathbf{0} \mid \overline{s}\langle v \rangle.s(y).\mathbf{0}$

Can we *statically* ensure no such errors occur during communications programming!

SESSION TYPES

or how to formally specify protocols

“well-typed programs cannot go wrong”

A Theory of Type Polymorphism in Programming (Milner 1978)

Properties of Session Types

1. Communication Error-Freedom
   No communication mismatch

2. Session Fidelity
   The communication sequence in a session follows the scenario declared in the types.

3. Progress
   No deadlock/ Stuck in a session

"well-typed **channels** cannot go wrong"

Properties of Session Types

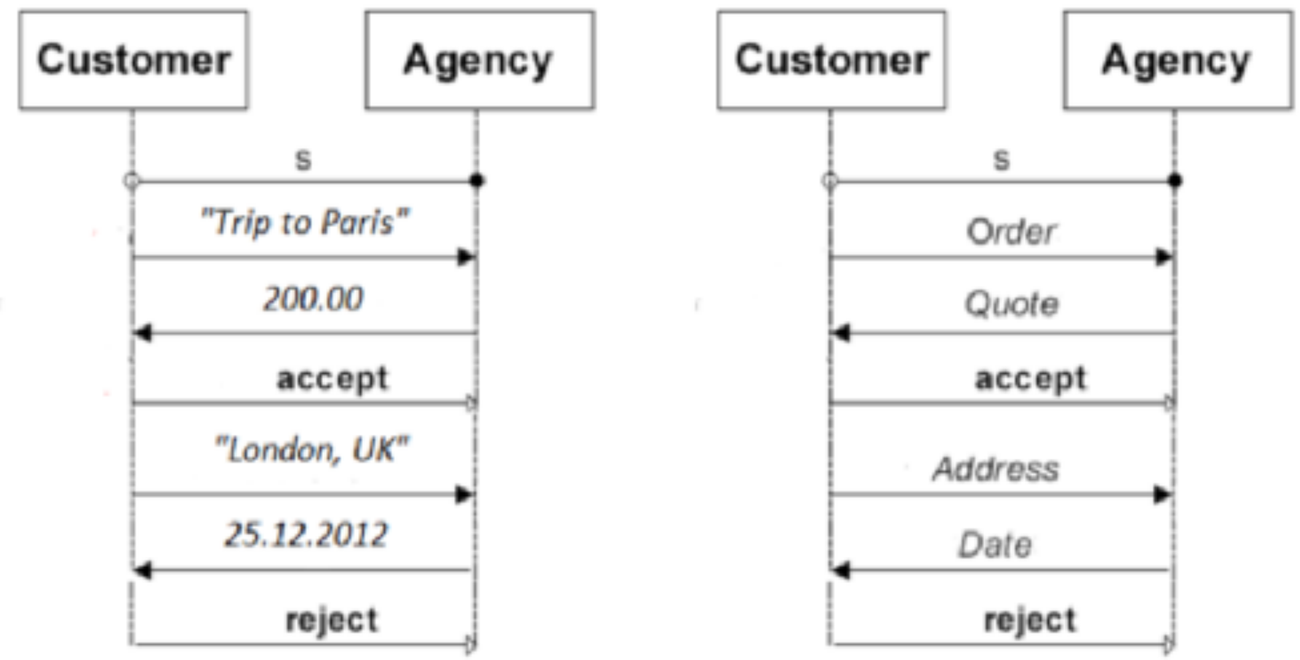
1. Communication Error-Freedom
   No communication mismatch

2. Session Fidelity
   The communication sequence in a session follows the scenario declared in the types.

3. Progress
   No deadlock/ Stuck in a session

"well-typed **channels** are free from communication errors"

SESSION TYPES ON ONE SLIDE



The type for the Customer is:

$![\mathtt{string}];\ ?[\mathtt{double}];\ \oplus\{\mathsf{accept}: ![\mathtt{string}];\ ?[\mathtt{nat},\mathtt{nat},\mathtt{nat}];\ \mathtt{end},\ \mathsf{reject}: \mathtt{end}\}$

Its dual type is:

$?[\mathtt{string}];\ ![\mathtt{double}];\ \&\{\mathsf{accept}: ?[\mathtt{string}];\ ![\mathtt{nat},\mathtt{nat},\mathtt{nat}];\ \mathtt{end},\ \mathsf{reject}: \mathtt{end}\}$

SESSION TYPES:SYNTAX

S ::=	Sort
$\mathbf{bool} \mid \mathbf{nat} \mid \mathbf{string} \ldots$	
T ::=	Type
$![\widetilde{S}];\, T$	sending a value of type $\widetilde{S}$
$![T];\, T'$	sending a type T (delegation)
$\&\{l_1 : T_1, \ldots, l_n : T_n\}$	branching behaviour (external choice)
$?[\widetilde{S}];\, T$	receiving a value of type $\widetilde{S}$
$?[T];\, T'$	receiving a type T (delegation)
$\oplus\{l_1 : T_1, \ldots, l_n : T_n\}$	selection (internal choice)
t	
$\mu t.T$	recursive behaviour
$\mathbf{end}$	end of a session

SESSION TYPES: SYNTAX EXPLAINED

The type $![\widetilde{S}];T$ represents the behaviour of first sending values of type $\widetilde{S}$ and then continues as specified by the type T;
$![T];T'$ represents similar behaviour, which starts with sending a channel(delegation) instead.

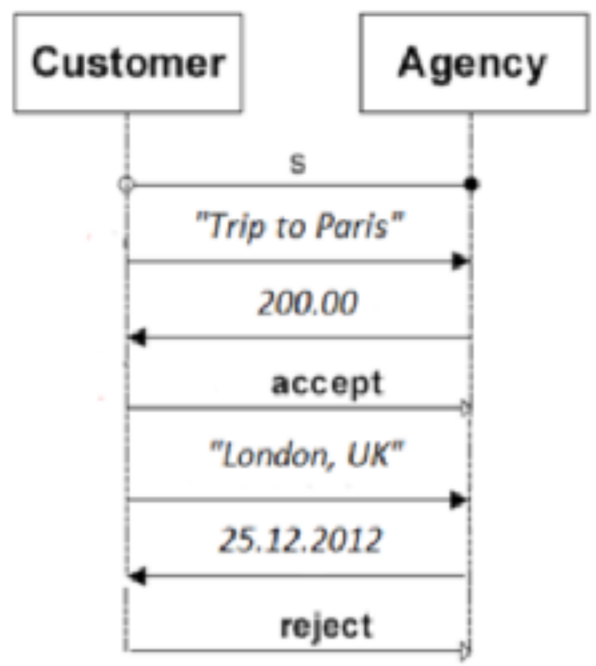
The $?[\widetilde{S}];T$ and $?[T];T'$ are the dual of $![\widetilde{S}];\overline{T}$ and $![T];\overline{T'}$ respectively, receiving values, instead of sending.

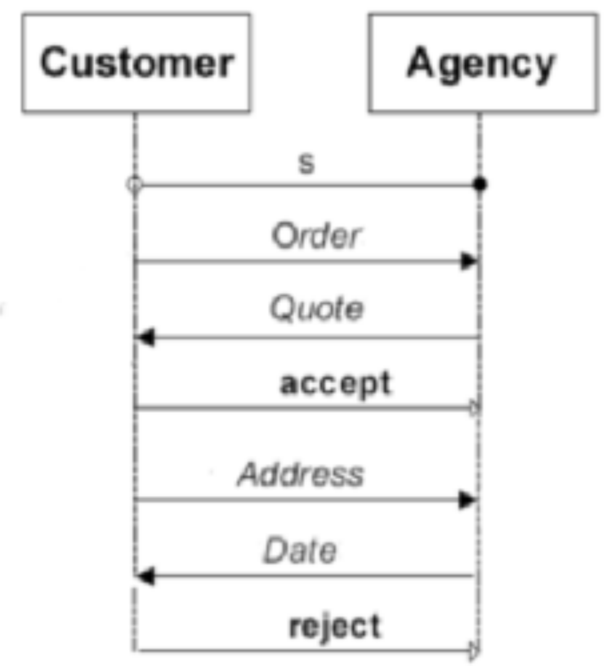
$\mu t.T$ represents recursive behaviour - start doing T, when t is encountered recur to T again.

$\&\{l_1 : T_1, \ldots, l_n : T_n\}$ shows the branching behaviour: it waits with n options, and behaves as type T_i if i-th action is selected (external choice).

$\oplus\{l_1 : T_1, \ldots, l_n : T_n\}$ then represents the behaviour which would select one of l_i and then behaves as T_i, according to the selected l_i (internal choice).

FROM SESSION CALCULI TO SESSION TYPES



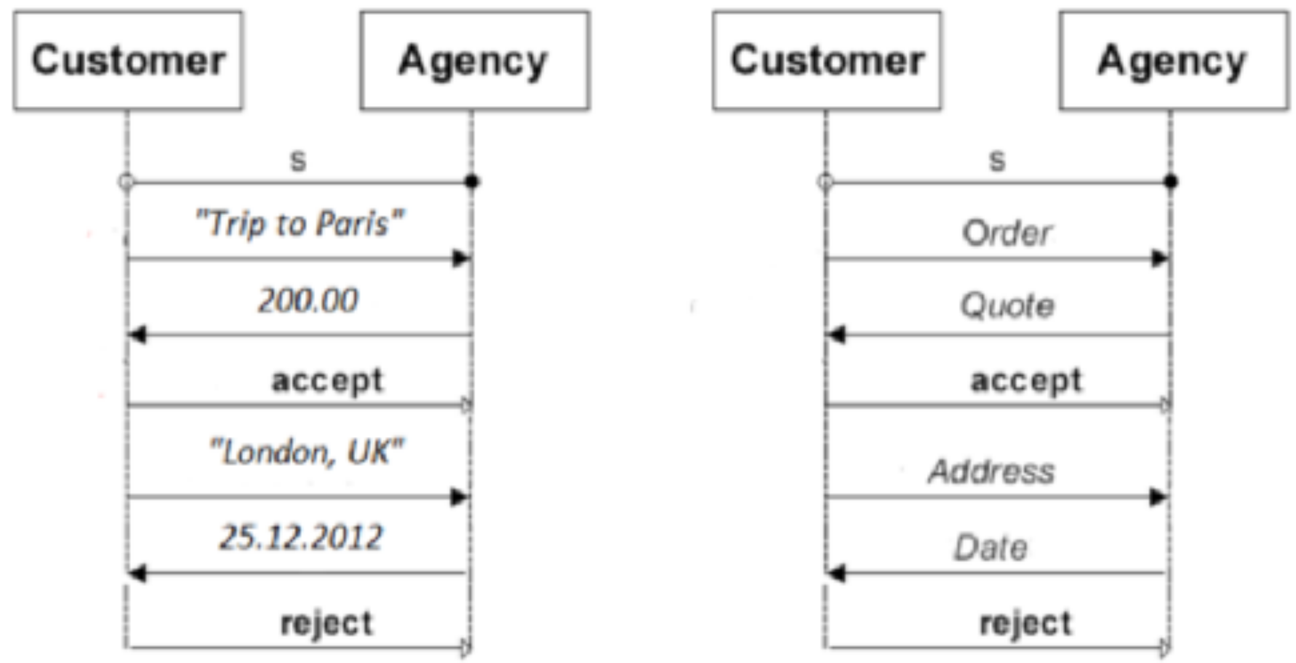


$\texttt{Session Calculus} : \overline{s}\langle place(\texttt{i})\rangle.s(price).$ $\texttt{if is\_acceptable}(price)$ $\texttt{then}\ s \lhd \mathsf{accept}.\overline{s}\langle\texttt{address}\rangle.s(date)$ $\texttt{else}\ s \lhd \mathsf{reject}$

$\texttt{Session Types} : ![\texttt{string}]; ?[\texttt{double}]; \oplus\{$ $\mathsf{accept} : ![\texttt{string}]; ?[\texttt{nat,nat,nat}]; \texttt{end},$ $\mathsf{reject} : \texttt{end}\}$

Session Types are **types** for **session** channels

FROM SESSION CALCULI TO SESSION TYPES

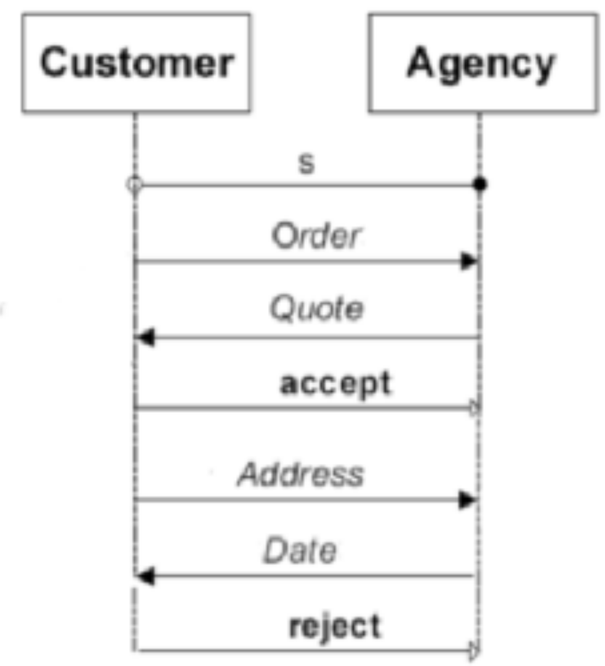


Session Calculus : $\overline{s}\langle place(\mathtt{i})\rangle.s(price).$ $\quad$ if is_acceptable($price$)
then $s \lhd \mathsf{accept}.\overline{s}\langle\mathtt{address}\rangle.s(date)$
else $s \lhd \mathsf{reject}$

Session Types : $![\mathtt{string}]; ?[\mathtt{double}]; \oplus\{ \quad \mathsf{accept} : ![\mathtt{string}]; ?[\mathtt{nat,nat,nat}]; \mathtt{end},$
$\mathsf{reject} : \mathtt{end}\}$

Session Types are **types** for **session** channels

FROM SESSION CALCULI TO SESSION TYPES



$$P : \overline{s}\langle place(\mathtt{i})\rangle.s(price).\quad \begin{array}{l}\mathtt{if\ is\_acceptable}(price)\\ \mathtt{then}\ s \lhd \mathsf{accept}.\overline{s}\langle \mathtt{address}\rangle.s(date)\\ \mathtt{else}\ s \lhd \mathsf{reject}\end{array}$$

$$T : {!}[\mathtt{string}]; {?}[\mathtt{double}]; \oplus\{\begin{array}{l}\mathsf{accept} : {!}[\mathtt{string}]; {?}[\mathtt{nat,nat,nat}]; \mathtt{end},\\ \mathsf{reject} : \mathtt{end}\}\end{array}$$

***P* is typable:** $\Gamma \vdash P \rhd \Delta \cdot s : T$

DUALITY

$\bar{a}(s).P$

where

s in P has a type T

$\Gamma \vdash P \triangleright \Delta \cdot s : T$

$a(s).Q$

where

s in Q has a type $\overline{T}$

$\Gamma \vdash Q \triangleright \Delta' \cdot s : \overline{T}$

$P \mid Q$ is typable

DUAL TYPES

For a type T, its dual or co type, written $\overline{T}$, is defined by exchanging ? and !, and & and $\oplus$. The inductive definition is given below:

$$
\begin{array}{rclcrclcrcl}
\overline{?[\widetilde{S}];T} & = & ![\widetilde{S}];\overline{T} & \quad & \overline{\oplus\{l_i : T_i\}_{i\in I}} & = & \&\{l_i : \overline{T_i}\}_{i\in I} & \quad & \overline{?[T];T'} & = & ![T];\overline{T'} \\
\overline{![\widetilde{S}];T} & = & ?[\widetilde{S}];\overline{T} & & \overline{\&\{l_i : T_i\}_{i\in I}} & = & \oplus\{l_i : \overline{T_i}\}_{i\in I} & & \overline{![T];T'} & = & ?[T];\overline{T'} \\
\overline{\mathbf{end}} & = & \mathbf{end} & & \overline{\mu t.T} & = & \mu t.\overline{T} & & \overline{t} & = & t
\end{array}
$$

Duality is essential for checking type compatability. Compatible types mean that each common channel s is associated with complementary behaviour, thus ensuring the interactions on s to run without errors.

SESSION TYPES:SYNTAX

S ::=	$\langle T, \overline{T} \rangle$ |	Sort
	$\mathtt{bool} \mid \mathtt{nat} \mid \mathtt{string} \ldots$	
T ::=		Type
	$![\widetilde{S}];\, T$	sending a value of type $\widetilde{S}$
	$![T];\, T'$	sending a type T (delegation)
	$\&\{l_1 : T_1, \ldots, l_n : T_n\}$	branching behaviour (external choice)
	$?[\widetilde{S}];\, T$	receiving a value of type $\widetilde{S}$
	$?[T];\, T'$	receiving a type T (delegation)
	$\oplus\{l_1 : T_1, \ldots, l_n : T_n\}$	selection (internal choice)
	t	
	$\mu t.T$	recursive behaviour
	$\mathtt{end}$	end of a session

EXERCISE

Give the dual type for the following types:

1. $![\mathtt{string}]; ?[\mathtt{int}]$

2. $![\mathtt{string}]; ![T']; \mathtt{end}$

3. $\&\{\mathtt{read}: ?[\mathtt{nat}]; \mathtt{end}, \mathtt{write}: ![\mathtt{nat}]; \mathtt{end}\}$

4. $\mu t. \oplus \{\mathtt{read}: ?[\mathtt{nat}]; t, \mathtt{write}: ![\mathtt{nat}]; \mathtt{end}\}$

EXERCISE

Give the dual type for the following types:

1. $![\texttt{string}]; ?[\texttt{int}]$

   $?[\texttt{string}]; ![\texttt{int}]$

2. $![\texttt{string}]; ![T']; \mathsf{end}$

   $?[\texttt{string}]; ?[T']; \mathsf{end}$

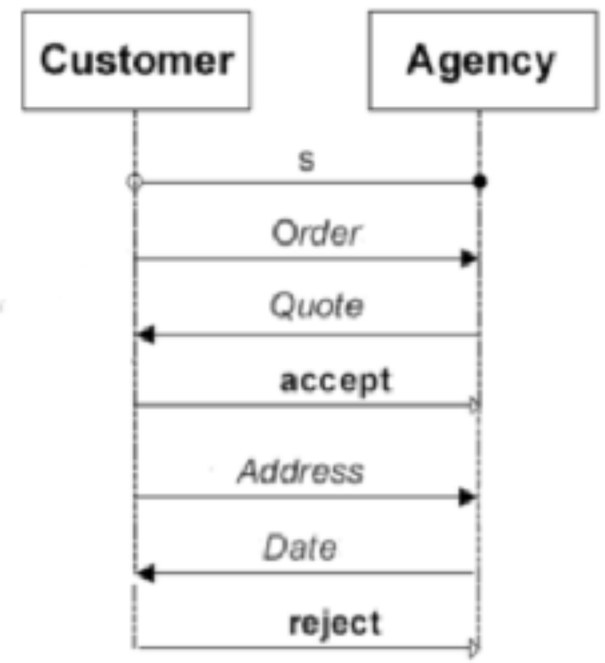
3. $\&\{\texttt{read}: ?[\mathrm{nat}]; \mathsf{end}, \texttt{write}: ![\mathrm{nat}]; \mathsf{end}\}$

   $\oplus\{\texttt{read}: ![\mathrm{nat}]; \mathsf{end}, \texttt{write}: ?[\mathrm{nat}]; \mathsf{end}\}$

4. $\mu t. \oplus \{\texttt{read}: ?[\mathrm{nat}]; t, \texttt{write}: ![\mathrm{nat}]; \mathsf{end}\}$

   $\mu t. \& \{\texttt{read}: ![\mathrm{nat}]; t, \texttt{write}: ?[\mathrm{nat}]; \mathsf{end}\}$

FROM SESSION CALCULI TO SESSION TYPES



$$P : \overline{s}\langle place(\mathtt{i})\rangle.s(price).\ \mathtt{if}\ \mathtt{is\_acceptable}(price)\ \mathtt{then}\ s \lhd \mathsf{accept}.\overline{s}\langle \mathtt{address}\rangle.s(date)\ \mathtt{else}\ s \lhd \mathsf{reject}$$

$$T : ![\mathtt{string}];?[\mathtt{double}];\oplus\{\mathsf{accept} : ![\mathtt{string}];?[\mathtt{nat,nat,nat}];\mathtt{end},\ \mathsf{reject} : \mathtt{end}\}$$

***P* is typable:** $\Gamma \vdash P \rhd \Delta \cdot s : T$

TYPING JUDGEMENT

$$\boxed{\Theta; \Gamma \vdash P \triangleright \Delta}$$

process environment

Θ is a mapping from process variables to sorts and types.
Example: $X : ST$

typing environment

Γ is a mapping from share channels and variables to sorts.
Example: $a : \langle T, \overline{T} \rangle$, $x : S$

session environment
mapping from session channels to session types.
Example: $s : T$

The typing judgement reads as:

Under the environment Θ and Γ , the process P has typing Δ

TYPING JUDGEMENT

$$\Theta ; \Gamma \vdash P \triangleright \Delta$$

process environment

Θ is a mapping from process variables to sorts and types.
Example: $X : ST$

typing environment

Γ is a mapping from share channels and variables to sorts.
Example: $a : \langle T, \overline{T} \rangle$, $x : S$

session environment
mapping from session channels to session types.
Example: $s : T$

$$\frac{\Gamma \vdash e \triangleright \texttt{bool} \quad \Theta ; \Gamma \vdash P \triangleright \Delta \quad \Theta ; \Gamma \vdash Q \triangleright \Delta}{\Theta ; \Gamma \vdash \texttt{if } e \texttt{ then } P \texttt{ else } Q \triangleright \Delta} \text{[If]}$$

TYPING JUDGEMENT

$$\Theta; \Gamma \vdash P \triangleright \Delta$$

process environment

Θ is a mapping from process variables to sorts and types.
Example: $X : ST$

typing environment

Γ is a mapping from share channels and variables to sorts.
Example: $a : \langle T, \overline{T} \rangle$, $x : S$

session environment
mapping from session channels to session types.
Example: $s : T$

$$\frac{\Gamma \vdash a \triangleright \langle T, \overline{T} \rangle \qquad \Theta; \Gamma \vdash P \triangleright \Delta \cdot s : T}{\Theta; \Gamma \vdash a(s).P \triangleright \Delta} \text{[Acc]}$$

$$\frac{\Gamma \vdash a \triangleright \langle T, \overline{T} \rangle \qquad \Theta; \Gamma \vdash P \triangleright \Delta \cdot s : \overline{T}}{\Theta; \Gamma \vdash \overline{a}(s).P \triangleright \Delta} \text{[Req]}$$

TYPING JUDGEMENT

$$\Theta; \Gamma \vdash P \triangleright \Delta$$

process environment

Θ is a mapping from process variables to sorts and types.
Example: $X : ST$

typing environment

Γ is a mapping from share channels and variables to sorts.
Example: $a : \langle T, \overline{T} \rangle$, $x : S$

session environment
mapping from session channels to session types.
Example: $s : T$

$$\frac{\Gamma \vdash \tilde{e} \triangleright \tilde{S} \qquad \Theta; \Gamma \vdash P \triangleright \Delta \cdot s : T}{\Theta; \Gamma \vdash \overline{s}\langle \tilde{e} \rangle.P \triangleright \Delta \cdot s :![\tilde{S}]; T} \text{[Send]}$$

$$\frac{\Theta; \Gamma \cdot \tilde{x} : \tilde{S} \vdash P \triangleright \Delta \cdot s : T}{\Theta; \Gamma \vdash s(\tilde{x}).P \triangleright \Delta \cdot s :?[\tilde{S}]; T} \text{[Recv]}$$

TYPING JUDGEMENT

$$\Theta; \Gamma \vdash P \triangleright \Delta$$

process environment

Θ is a mapping from process variables to sorts and types.
Example: $X : ST$

typing environment

Γ is a mapping from share channels and variables to sorts.
Example: $a : \langle T, \overline{T} \rangle$, $x : S$

session environment
mapping from session channels to session types.
Example: $s : T$

$$\frac{\Theta; \Gamma \vdash P_1 \triangleright \Delta \cdot s : T_1 \quad \ldots \quad \Theta; \Gamma \vdash P_n \triangleright \Delta \cdot s : T_n}{\Theta; \Gamma \vdash s \triangleright \{l_1 : P_1 \mathbin{[\!]} \cdots \mathbin{[\!]} l_n : P_n\} \triangleright \Delta \cdot s : \&\{l_1 : T_1, \ldots, l_n : T_n\}} \text{[Br]}$$

$$\frac{\Theta; \Gamma \vdash P \triangleright \Delta \cdot s : T_j \quad (1 \leq j \leq n)}{\Theta; \Gamma \vdash s \triangleleft l_j.P \triangleright \Delta \cdot s : \oplus\{l_1 : T_1, \ldots, l_n : T_n\}} \text{[Sel]}$$

TYPING JUDGEMENT

$$\Theta; \Gamma \vdash P \triangleright \Delta$$

process environment

Θ is a mapping from process variables to sorts and types.
Example: $X : ST$

typing environment

Γ is a mapping from share channels and variables to sorts.
Example: $a : \langle T, \overline{T} \rangle$, $x : S$

session environment
mapping from session channels to session types.
Example: $s : T$

$$\frac{\Theta \cdot X : ST; \Gamma \cdot x : S \vdash P \triangleright s : T \qquad \Theta \cdot X : ST; \Gamma \vdash Q \triangleright \Delta}{\Theta; \Gamma \vdash \mathtt{def}\ X(x, s) = P \ \mathsf{in}\ Q \triangleright \Delta}\ [\mathsf{Def}]$$

$$\frac{\Delta \text{ contains only } \mathtt{end} \qquad \Gamma \vdash e \triangleright S}{\Theta \cdot X : ST; \Gamma \vdash X\langle e, s \rangle \triangleright \Delta \cdot s : T}\ [\mathsf{Var}]$$

TYPING DERIVATION

Consider the variable example from the beginning of the lecture.

$$\mathbf{Var}(a,x) \stackrel{\text{df}}{=} a(s).s \triangleright [\mathsf{read} : \overline{s}\langle x\rangle.\mathbf{Var}\langle a,x\rangle \mathbin{[\!]} \mathsf{write} : s(y).\mathbf{Var}\langle a,y\rangle]$$

$$\mathbf{Reader}(a) = \overline{a}(s).s \triangleleft \mathsf{read}.s(y).\mathbf{0}$$

$$\mathbf{Writer}(a,x) = \overline{a}(s).s \triangleleft \mathsf{write}.\overline{s}\langle x\rangle.\mathbf{0}$$

where $a : \langle T, \overline{T}\rangle$ and

$$T = \&\{\mathtt{read}: ![\mathtt{nat}]; \mathbf{end}, \mathtt{write}: ?[\mathtt{nat}]; \mathbf{end}\}$$

$$\overline{T} = \oplus\{\mathtt{read}: ?[\mathtt{nat}]; \mathbf{end}, \mathtt{write}: ![\mathtt{nat}]; \mathbf{end}\}$$

TYPING DERIVATION

$$\mathbf{Writer}(a, x) \stackrel{\mathsf{df}}{=} \overline{a}(s).s \triangleleft \mathsf{write}.\overline{s}\langle x\rangle.\mathbf{0}$$

$s : \overline{T} = \oplus\{\mathtt{read}:?[\mathtt{nat}];\,\mathtt{end}, \mathtt{write}:![\mathtt{nat}];\,\mathtt{end}\}$

$a : \langle T, \overline{T}\rangle$

Typing derivation for the **Writer** process:

$$\frac{}{\Theta; \Gamma \vdash \overline{a}(s).s \triangleleft \mathsf{write}.\overline{s}\langle 5\rangle.\mathbf{0} \triangleright \Delta}\;[\mathrm{Req}]$$

$$\frac{\Gamma \vdash a \triangleright \langle T, \overline{T} \rangle \qquad \Theta; \Gamma \vdash P \triangleright \Delta \cdot s : \overline{T}}{\Theta; \Gamma \vdash \overline{a}(s).P \triangleright \Delta} \text{[Req]}$$

$$\mathbf{Writer}(a, x) \stackrel{\text{df}}{=} \overline{a}(s).s \triangleleft \mathsf{write}.\overline{s}\langle x \rangle.\mathbf{0}$$

$s : \overline{T} = \oplus\{\mathtt{read}: ?[\mathtt{nat}]; \mathtt{end}, \mathtt{write}: ![\mathtt{nat}]; \mathtt{end}\}$

$a : \langle T, \overline{T} \rangle$

Typing derivation for the **Writer** process:

$$\frac{}{\Theta; \Gamma \vdash \overline{a}(s).s \triangleleft \mathsf{write}.\overline{s}\langle 5 \rangle.\mathbf{0} \triangleright \Delta} \text{[Req]}$$

$$\frac{\boxed{\Gamma \vdash a \triangleright \langle T, \overline{T} \rangle} \qquad \Theta; \Gamma \vdash P \triangleright \Delta \cdot s : \overline{T}}{\Theta; \Gamma \vdash \overline{a}(s).P \triangleright \Delta} \text{[Req]}$$

$$\mathbf{Writer}(a, x) \stackrel{\text{df}}{=} \overline{a}(s).s \triangleleft \mathsf{write}.\overline{s}\langle x \rangle.\mathbf{0}$$

$s : \overline{T} = \oplus\{\mathtt{read} : ?[\mathtt{nat}]; \mathtt{end}, \mathtt{write} : ![\mathtt{nat}]; \mathtt{end}\}$

$a : \langle T, \overline{T} \rangle$

Typing derivation for the **Writer** process:

$$\frac{\boxed{\Gamma \vdash a \triangleright \langle T, \overline{T} \rangle}}{\Theta; \Gamma \vdash \overline{a}(s).s \triangleleft \mathsf{write}.\overline{s}\langle 5 \rangle.\mathbf{0} \triangleright \Delta} \text{[Req]}$$

$$\frac{\Gamma \vdash a \triangleright \langle T, \overline{T} \rangle \qquad \boxed{\Theta; \Gamma \vdash P \triangleright \Delta \cdot s : \overline{T}}}{\Theta; \Gamma \vdash \overline{a}(s).P \triangleright \Delta} \text{[Req]}$$

$$\mathbf{Writer}(a, x) \stackrel{\text{df}}{=} \overline{a}(s).s \lhd \mathsf{write}.\overline{s}\langle x \rangle.\mathbf{0}$$

$s : \overline{T} = \oplus\{\mathtt{read}\!:\!?[\mathtt{nat}];\mathtt{end}, \mathtt{write}\!:\![\mathtt{nat}];\mathtt{end}\}$

$a : \langle T, \overline{T} \rangle$

Typing derivation for the **Writer** process:

$$\frac{\Gamma \vdash a \triangleright \langle T, \overline{T} \rangle \qquad \boxed{\Theta; \Gamma \vdash s \lhd \mathsf{write}.\overline{s}\langle 5 \rangle.\mathbf{0} \triangleright s : \oplus\{\mathtt{read}\!:\!?[\mathtt{nat}];\mathtt{end}, \mathtt{write}\!:\![\mathtt{nat}];\mathtt{end}\}}}{\Theta; \Gamma \vdash \overline{a}(s).s \lhd \mathsf{write}.\overline{s}\langle 5 \rangle.\mathbf{0} \triangleright \Delta} \text{[Req]}$$

$$\mathbf{Writer}(a, x) \stackrel{\text{df}}{=} \overline{a}(s).s \triangleleft \text{write}.\overline{s}\langle x\rangle.\mathbf{0}$$

$$s : \overline{T} = \oplus\{\texttt{read}:?[\texttt{nat}];\texttt{end}, \texttt{write}:![\texttt{nat}];\texttt{end}\}$$

$$a : \langle T, \overline{T}\rangle$$

Typing derivation for the **Writer** process:

$$\dfrac{\Gamma \vdash a \triangleright \langle T, \overline{T}\rangle \qquad \dfrac{}{\Theta;\Gamma \vdash s \triangleleft \text{write}.\overline{s}\langle 5\rangle.\mathbf{0} \triangleright s : \oplus\{\texttt{read}:?[\texttt{nat}];\texttt{end}, \texttt{write}:![\texttt{nat}];\texttt{end}\}}\,[\text{Sel}]}{\Theta;\Gamma \vdash \overline{a}(s).s \triangleleft \text{write}.\overline{s}\langle 5\rangle.\mathbf{0} \triangleright \Delta}\,[\text{Req}]$$

$$\frac{\boxed{\Theta; \Gamma \vdash P \triangleright \Delta \cdot s : T_j} \quad (1 \leq j \leq n)}{\Theta; \Gamma \vdash s \triangleleft l_j.P \triangleright \Delta \cdot s : \oplus\{l_1 : T_1, \ldots, l_n : T_n\}} \text{[Sel]}$$

$$\mathbf{Writer}(a, x) \stackrel{\text{df}}{=} \overline{a}(s).s \triangleleft \mathsf{write}.\overline{s}\langle x \rangle.\mathbf{0}$$

$$s : \overline{T} = \oplus\{\mathtt{read}:?[\mathtt{nat}];\mathtt{end}, \mathtt{write}:![\mathtt{nat}];\mathtt{end}\}$$

$$a : \langle T, \overline{T} \rangle$$

Typing derivation for the **Writer** process:

$$\frac{\Gamma \vdash a \triangleright \langle T, \overline{T} \rangle \quad \dfrac{\boxed{\Theta; \Gamma \vdash \overline{s}\langle 5 \rangle.\mathbf{0} \triangleright s :![\mathtt{nat}];\mathtt{end}}}{\Theta; \Gamma \vdash s \triangleleft \mathsf{write}.\overline{s}\langle 5 \rangle.\mathbf{0} \triangleright s : \oplus\{\mathtt{read}:?[\mathtt{nat}];\mathtt{end}, \mathtt{write}:![\mathtt{nat}];\mathtt{end}\}} \text{[Sel]}}{\Theta; \Gamma \vdash \overline{a}(s).s \triangleleft \mathsf{write}.\overline{s}\langle 5 \rangle.\mathbf{0} \triangleright \Delta} \text{[Req]}$$

$$\frac{\Gamma \vdash \tilde{e} \triangleright \tilde{S} \qquad \Theta; \Gamma \vdash P \triangleright \Delta \cdot s : T}{\Theta; \Gamma \vdash \overline{s}\langle \tilde{e} \rangle . P \triangleright \Delta \cdot s : ![\tilde{S}]; T} \text{[Send]}$$

$$\mathbf{Writer}(a, x) \stackrel{\text{df}}{=} \overline{a}(s).s \triangleleft \text{write}.\overline{s}\langle x \rangle.\mathbf{0}$$

$$s : \overline{T} = \oplus\{\texttt{read} : ?[\texttt{nat}]; \texttt{end}, \texttt{write} : ![\texttt{nat}]; \texttt{end}\}$$

$$a : \langle T, \overline{T} \rangle$$

Typing derivation for the **Writer** process:

$$\dfrac{\Gamma \vdash a \triangleright \langle T, \overline{T} \rangle \qquad \dfrac{\dfrac{}{\Theta; \Gamma \vdash \overline{s}\langle 5 \rangle.\mathbf{0} \triangleright s : ![\texttt{nat}]; \texttt{end}} \text{[Send]}}{\Theta; \Gamma \vdash s \triangleleft \text{write}.\overline{s}\langle 5 \rangle.\mathbf{0} \triangleright s : \oplus\{\texttt{read} : ?[\texttt{nat}]; \texttt{end}, \texttt{write} : ![\texttt{nat}]; \texttt{end}\}} \text{[Sel]}}{\Theta; \Gamma \vdash \overline{a}(s).s \triangleleft \text{write}.\overline{s}\langle 5 \rangle.\mathbf{0} \triangleright \Delta} \text{[Req]}$$

$$\dfrac{\boxed{\Gamma \vdash \tilde{e} \triangleright \tilde{S}} \qquad \Theta;\Gamma \vdash P \triangleright \Delta \cdot s : T}{\Theta;\Gamma \vdash \overline{s}\langle \tilde{e} \rangle.P \triangleright \Delta \cdot s :![\tilde{S}];\, T}\ [\text{Send}]$$

$$\mathbf{Writer}(a, x) \stackrel{\text{df}}{=} \overline{a}(s).s \triangleleft \mathsf{write}.\overline{s}\langle x \rangle.\mathbf{0}$$

$$s : \overline{T} = \oplus\{\mathtt{read}:?[\mathtt{nat}];\, \mathtt{end}, \mathtt{write}:![\mathtt{nat}];\, \mathtt{end}\}$$

$$a : \langle T, \overline{T} \rangle$$

Typing derivation for the $\mathbf{Writer}$ process:

$$\dfrac{\Gamma \vdash a \triangleright \langle T, \overline{T} \rangle \qquad \dfrac{\dfrac{\boxed{\Gamma \vdash 5 \triangleright \mathtt{nat}}}{\Theta;\Gamma \vdash \overline{s}\langle 5 \rangle.\mathbf{0} \triangleright s :![\mathtt{nat}];\, \mathtt{end}}\ [\text{Send}]}{\Theta;\Gamma \vdash s \triangleleft \mathsf{write}.\overline{s}\langle 5 \rangle.\mathbf{0} \triangleright s : \oplus\{\mathtt{read}:?[\mathtt{nat}];\, \mathtt{end}, \mathtt{write}:![\mathtt{nat}];\, \mathtt{end}\}}\ [\text{Sel}]}{\Theta;\Gamma \vdash \overline{a}(s).s \triangleleft \mathsf{write}.\overline{s}\langle 5 \rangle.\mathbf{0} \triangleright \Delta}\ [\text{Req}]$$

$$\dfrac{\Gamma \vdash \tilde{e} \rhd \tilde{S} \qquad \boxed{\Theta;\Gamma \vdash P \rhd \Delta \cdot s : T}}{\Theta;\Gamma \vdash \overline{s}\langle\tilde{e}\rangle.P \rhd \Delta \cdot s :![\tilde{S}];\, T}\ \text{[Send]}$$

$$\mathbf{Writer}(a, x) \stackrel{\text{df}}{=} \overline{a}(s).s \lhd \mathsf{write}.\overline{s}\langle x\rangle.\mathbf{0}$$

$$s : \overline{T} = \oplus\{\mathtt{read}:?[\mathtt{nat}];\,\mathtt{end},\, \mathtt{write}:![\mathtt{nat}];\,\mathtt{end}\}$$

$$a : \langle T, \overline{T}\rangle$$

Typing derivation for the **Writer** process:

$$\dfrac{\Gamma \vdash a \rhd \langle T, \overline{T}\rangle \qquad \dfrac{\dfrac{\Gamma \vdash 5 \rhd \mathtt{nat} \qquad \boxed{\Theta;\Gamma \vdash \mathbf{0} \rhd s : \mathtt{end}}}{\Theta;\Gamma \vdash \overline{s}\langle 5\rangle.\mathbf{0} \rhd s :![\mathtt{nat}];\,\mathtt{end}}\ \text{[Send]}}{\Theta;\Gamma \vdash s \lhd \mathsf{write}.\overline{s}\langle 5\rangle.\mathbf{0} \rhd s : \oplus\{\mathtt{read}:?[\mathtt{nat}];\,\mathtt{end},\, \mathtt{write}:![\mathtt{nat}];\,\mathtt{end}\}}\ \text{[Sel]}}{\Theta;\Gamma \vdash \overline{a}(s).s \lhd \mathsf{write}.\overline{s}\langle 5\rangle.\mathbf{0} \rhd \Delta}\ \text{[Req]}$$

$$\mathbf{Writer}(a,x) \stackrel{\mathsf{df}}{=} \overline{a}(s).s \lhd \mathsf{write}.\overline{s}\langle x\rangle.\mathbf{0}$$

$s : \overline{T} = \oplus\{\mathtt{read}:?[\mathtt{nat}];\mathtt{end}, \mathtt{write}:![\mathtt{nat}];\mathtt{end}\}$

$a : \langle T, \overline{T}\rangle$

Typing derivation for the **Writer** process:

$$\dfrac{\Gamma \vdash a \rhd \langle T,\overline{T}\rangle \qquad \dfrac{\dfrac{\Gamma \vdash 5 \rhd \mathtt{nat} \qquad \Theta;\Gamma \vdash \mathbf{0} \rhd s : \mathtt{end}}{\Theta;\Gamma \vdash \overline{s}\langle 5\rangle.\mathbf{0} \rhd s :![\mathtt{nat}];\mathtt{end}}\,[\mathsf{Send}]}{\Theta;\Gamma \vdash s \lhd \mathsf{write}.\overline{s}\langle 5\rangle.\mathbf{0} \rhd s : \oplus\{\mathtt{read}:?[\mathtt{nat}];\mathtt{end}, \mathtt{write}:![\mathtt{nat}];\mathtt{end}\}}\,[\mathsf{Sel}]}{\Theta;\Gamma \vdash \overline{a}(s).s \lhd \mathsf{write}.\overline{s}\langle 5\rangle.\mathbf{0} \rhd \Delta}\,[\mathsf{Req}]$$

TYPING DERIVATION

$$\mathbf{Writer}(a, x) \stackrel{\mathsf{df}}{=} \overline{a}(s).s \lhd \mathsf{write}.\overline{s}\langle x\rangle.\mathbf{0}$$

$$s : \overline{T} = \oplus\{\mathtt{read}:?[\mathtt{nat}];\, \mathtt{end}, \mathtt{write}:![\mathtt{nat}];\, \mathtt{end}\}$$

$$a : \langle T, \overline{T}\rangle$$

Typing derivation for the $\mathbf{Writer}$ process:

$$\dfrac{\Gamma \vdash a \rhd \langle T, \overline{T}\rangle \quad \dfrac{\dfrac{\dfrac{}{\Gamma \vdash 5 \rhd \mathtt{nat}}\,[\mathsf{Nat}] \quad \dfrac{}{\Theta;\Gamma \vdash \mathbf{0} \rhd s : \mathtt{end}}\,[\mathsf{Inact}]}{\Theta;\Gamma \vdash \overline{s}\langle 5\rangle.\mathbf{0} \rhd s :![\mathtt{nat}];\, \mathtt{end}}\,[\mathsf{Send}]}{\Theta;\Gamma \vdash s \lhd \mathsf{write}.\overline{s}\langle 5\rangle.\mathbf{0} \rhd s : \oplus\{\mathtt{read}:?[\mathtt{nat}];\, \mathtt{end}, \mathtt{write}:![\mathtt{nat}];\, \mathtt{end}\}}\,[\mathsf{Sel}]}{\Theta;\Gamma \vdash \overline{a}(s).s \lhd \mathsf{write}.\overline{s}\langle 5\rangle.\mathbf{0} \rhd \Delta}\,[\mathsf{Req}]$$

Exercise: Give the derivation tree for the Reader

$$\frac{?}{\Theta; \Gamma \vdash s \triangleleft \mathsf{read}.s(y).0 \triangleright s : \oplus\{\mathtt{read}:?[\mathtt{nat}]; \mathtt{end}, \mathtt{write}:![\mathtt{nat}]; \mathtt{end}\}} \; [?]$$

TYPING THE VARIABLE PROCESS

$$\mathbf{Var}(a,x) \stackrel{\text{df}}{=} a(s).s \triangleright [\text{read} : \overline{s}\langle x\rangle.\mathbf{Var}\langle a,x\rangle \;[\!]\; \text{write} : s(y).\mathbf{Var}\langle a,y\rangle]$$

$s : T = \&\{\mathtt{read}{:}![\mathtt{nat}];\mathtt{end}, \mathtt{write}{:}?[\mathtt{nat}];\mathtt{end}\}$

$a : \langle T, \overline{T}\rangle$

$x : \mathtt{nat}$

Typing derivation for the $\mathbf{Var}$ process:

$$\frac{}{\Theta;\Gamma \vdash a(s).s \triangleright [\text{read} : \overline{s}\langle x\rangle.\mathbf{Var}\langle a,x\rangle \;[\!]\; \text{write} : s(y).\mathbf{Var}\langle a,y\rangle] \triangleright \Delta}\;[\text{Acc}]$$

where

$$T = \&\{\mathtt{read}{:}![\mathtt{nat}];\mathtt{end}, \mathtt{write}{:}?[\mathtt{nat}];\mathtt{end}\}$$

and

$$\Theta = \Theta' \cdot \mathbf{Var} : [\langle T, \overline{T}\rangle; \mathtt{nat}]$$

Δ is empty

$$\frac{\boxed{\Gamma \vdash a \triangleright \langle T, \overline{T} \rangle} \qquad \Theta; \Gamma \vdash P \triangleright \Delta \cdot s : T}{\Theta; \Gamma \vdash a(s).P \triangleright \Delta}\text{[Acc]}$$

$$\mathbf{Var}(a, x) \stackrel{\text{df}}{=} a(s).s \triangleright [\text{read} : \overline{s}\langle x \rangle.\mathbf{Var}\langle a, x \rangle \ [\!]\ \text{write} : s(y).\mathbf{Var}\langle a, y \rangle]$$

$s : T = \&\{\mathtt{read}{:}![\mathtt{nat}]; \mathtt{end}, \mathtt{write}{:}?[\mathtt{nat}]; \mathtt{end}\}$

$a : \langle T, \overline{T} \rangle$

$x : \mathtt{nat}$

Typing derivation for the **Var** process:

$$\frac{\boxed{\Gamma \vdash a \triangleright \langle T, \overline{T} \rangle}}{\Theta; \Gamma \vdash a(s).s \triangleright [\text{read} : \overline{s}\langle x \rangle.\mathbf{Var}\langle a, x \rangle \ [\!]\ \text{write} : s(y).\mathbf{Var}\langle a, y \rangle] \triangleright \Delta}\text{[Acc]}$$

where

$$T = \&\{\mathtt{read}{:}![\mathtt{nat}]; \mathtt{end}, \mathtt{write}{:}?[\mathtt{nat}]; \mathtt{end}\}$$

and

$$\Theta = \Theta' \cdot \mathbf{Var} : [\langle T, \overline{T} \rangle; \mathtt{nat}]$$

Δ is empty

$$\frac{\Gamma \vdash a \triangleright \langle T, \overline{T}\rangle \qquad \boxed{\Theta; \Gamma \vdash P \triangleright \Delta \cdot s : T}}{\Theta; \Gamma \vdash a(s).P \triangleright \Delta}\text{[Acc]}$$

$$\mathbf{Var}(a, x) \stackrel{\text{df}}{=} a(s).s \triangleright [\text{read} : \overline{s}\langle x\rangle.\mathbf{Var}\langle a, x\rangle \;[\!]\; \text{write} : s(y).\mathbf{Var}\langle a, y\rangle]$$

$s : T = \&\{\mathtt{read}{:}![\mathtt{nat}]; \mathtt{end}, \mathtt{write}{:}?[\mathtt{nat}]; \mathtt{end}\}$

$a : \langle T, \overline{T}\rangle$

$x : \mathtt{nat}$

Typing derivation for the $\mathbf{Var}$ process:

$$\frac{\Gamma \vdash a \triangleright \langle T, \overline{T}\rangle \qquad \boxed{\Theta; \Gamma \vdash s \triangleright [\text{read} : \overline{s}\langle x\rangle.\mathbf{Var}\langle a, x\rangle \;[\!]\; \text{write} : s(y).\mathbf{Var}\langle a, y\rangle] \triangleright s : T}}{\Theta; \Gamma \vdash a(s).s \triangleright [\text{read} : \overline{s}\langle x\rangle.\mathbf{Var}\langle a, x\rangle \;[\!]\; \text{write} : s(y).\mathbf{Var}\langle a, y\rangle] \triangleright \Delta}\text{[Acc]}$$

where

$$T = \&\{\mathtt{read}{:}![\mathtt{nat}]; \mathtt{end}, \mathtt{write}{:}?[\mathtt{nat}]; \mathtt{end}\}$$

and

$$\Theta = \Theta' \cdot \mathbf{Var} : [\langle T, \overline{T}\rangle; \mathtt{nat}]$$

Δ is empty

$$\frac{\Theta;\Gamma \vdash P_1 \triangleright \Delta \cdot s : T_1 \quad \ldots \quad \Theta;\Gamma \vdash P_n \triangleright \Delta \cdot s : T_n}{\Theta;\Gamma \vdash s \triangleright \{l_1 : P_1 \,[\!]\, \cdots \,[\!]\, l_n : P_n\} \triangleright \Delta \cdot s : \oplus\{l_1 : T_1, \ldots, l_n : T_n\}}\text{[Br]}$$

$$\mathbf{Var}(a,x) \stackrel{\mathsf{df}}{=} a(s).s \triangleright [\mathsf{read} : \overline{s}\langle x\rangle.\mathbf{Var}\langle a,x\rangle \,[\!]\, \mathsf{write} : s(y).\mathbf{Var}\langle a,y\rangle]$$

$s : T = \&\{\mathtt{read}:![\mathtt{nat}];\mathtt{end}, \mathtt{write}:?[\mathtt{nat}];\mathtt{end}\}$

$a : \langle T, \overline{T}\rangle$

$x : \mathtt{nat}$

Typing derivation for the $\mathbf{Var}$ process:

$$\frac{\dfrac{}{\Gamma \vdash a \triangleright \langle T, \overline{T}\rangle \qquad \Theta;\Gamma \vdash s \triangleright [\mathsf{read} : \overline{s}\langle x\rangle.\mathbf{Var}\langle a,x\rangle \,[\!]\, \mathsf{write} : s(y).\mathbf{Var}\langle a,y\rangle] \triangleright s : T}\text{[Bra]}}{\Theta;\Gamma \vdash a(s).s \triangleright [\mathsf{read} : \overline{s}\langle x\rangle.\mathbf{Var}\langle a,x\rangle \,[\!]\, \mathsf{write} : s(y).\mathbf{Var}\langle a,y\rangle] \triangleright \Delta}\text{[Acc]}$$

where

$$T = \&\{\mathtt{read}:![\mathtt{nat}];\mathtt{end}, \mathtt{write}:?[\mathtt{nat}];\mathtt{end}\}$$

and

$$\Theta = \Theta' \cdot \mathbf{Var} : [\langle T, \overline{T}\rangle; \mathtt{nat}]$$

Δ is empty

$$\frac{\boxed{\Theta; \Gamma \vdash P_1 \triangleright \Delta \cdot s : T_1} \quad \ldots \quad \Theta; \Gamma \vdash P_n \triangleright \Delta \cdot s : T_n}{\Theta; \Gamma \vdash s \triangleright \{l_1 : P_1 \parallel \cdots \parallel l_n : P_n\} \triangleright \Delta \cdot s : \oplus\{l_1 : T_1, \ldots, l_n : T_n\}} \text{[Br]}$$

$\mathbf{Var}(a, x) \stackrel{\mathsf{df}}{=} a(s).s \triangleright [\mathsf{read} : \overline{s}\langle x \rangle.\mathbf{Var}\langle a, x \rangle \parallel \mathsf{write} : s(y).\mathbf{Var}\langle a, y \rangle]$

$s : T = \&\{\mathtt{read}{:}![\mathbf{nat}]; \mathbf{end}, \mathtt{write}{:}?[\mathbf{nat}]; \mathbf{end}\}$

$a : \langle T, \overline{T} \rangle$

$x : \mathbf{nat}$

Typing derivation for the $\mathbf{Var}$ process:

$$\frac{\Gamma \vdash a \triangleright \langle T, \overline{T} \rangle \qquad \dfrac{\boxed{\Theta; \Gamma \vdash s(y).\mathbf{Var}\langle a, y \rangle \triangleright s : ?[\mathbf{nat}]; \mathbf{end}}}{\Theta; \Gamma \vdash s \triangleright [\mathsf{read} : \overline{s}\langle x \rangle.\mathbf{Var}\langle a, x \rangle \parallel \mathsf{write} : s(y).\mathbf{Var}\langle a, y \rangle] \triangleright s : T} \text{[Bra]}}{\Theta; \Gamma \vdash a(s).s \triangleright [\mathsf{read} : \overline{s}\langle x \rangle.\mathbf{Var}\langle a, x \rangle \parallel \mathsf{write} : s(y).\mathbf{Var}\langle a, y \rangle] \triangleright \Delta} \text{[Acc]}$$

where

$$T = \&\{\mathtt{read}{:}![\mathbf{nat}]; \mathbf{end}, \mathtt{write}{:}?[\mathbf{nat}]; \mathbf{end}\}$$

and

$$\Theta = \Theta' \cdot \mathbf{Var} : [\langle T, \overline{T} \rangle; \mathbf{nat}]$$

Δ is empty

$$\frac{\Theta;\Gamma \vdash P_1 \triangleright \Delta \cdot s : T_1 \quad \ldots \quad \boxed{\Theta;\Gamma \vdash P_n \triangleright \Delta \cdot s : T_n}}{\Theta;\Gamma \vdash s \triangleright \{l_1 : P_1 \mathbin{[\!]} \cdots \mathbin{[\!]} l_n : P_n\} \triangleright \Delta \cdot s : \oplus\{l_1 : T_1, \ldots, l_n : T_n\}} \text{[Br]}$$

$$\mathbf{Var}(a,x) \stackrel{\text{df}}{=} a(s).s \triangleright [\text{read} : \overline{s}\langle x\rangle.\mathbf{Var}\langle a,x\rangle \mathbin{[\!]} \text{write} : s(y).\mathbf{Var}\langle a,y\rangle]$$

$s : T = \&\{\mathtt{read}{:}![\mathrm{nat}];\mathrm{end}, \mathtt{write}{:}?[\mathrm{nat}];\mathrm{end}\}$

$a : \langle T, \overline{T}\rangle$

$x : \mathrm{nat}$

Typing derivation for the **Var** process:

$$\cfrac{\Gamma \vdash a \triangleright \langle T,\overline{T}\rangle \qquad \cfrac{\Theta;\Gamma \vdash s(y).\mathbf{Var}\langle a,y\rangle \triangleright s:?[\mathrm{nat}];\mathrm{end} \qquad \boxed{\Theta;\Gamma \vdash \overline{s}\langle x\rangle.\mathbf{Var}\langle a,x\rangle \triangleright s:![\mathrm{nat}];\mathrm{end}}}{\Theta;\Gamma \vdash s \triangleright [\text{read} : \overline{s}\langle x\rangle.\mathbf{Var}\langle a,x\rangle \mathbin{[\!]} \text{write} : s(y).\mathbf{Var}\langle a,y\rangle] \triangleright s : T} \text{[Bra]}}{\Theta;\Gamma \vdash a(s).s \triangleright [\text{read} : \overline{s}\langle x\rangle.\mathbf{Var}\langle a,x\rangle \mathbin{[\!]} \text{write} : s(y).\mathbf{Var}\langle a,y\rangle] \triangleright \Delta} \text{[Acc]}$$

where

$$T = \&\{\mathtt{read}{:}![\mathrm{nat}];\mathrm{end}, \mathtt{write}{:}?[\mathrm{nat}];\mathrm{end}\}$$

and

$$\Theta = \Theta' \cdot \mathbf{Var} : [\langle T,\overline{T}\rangle;\mathrm{nat}]$$

Δ is empty

$$\mathbf{Var}(a,x) \stackrel{\text{df}}{=} a(s).s \rhd [\mathsf{read} : \overline{s}\langle x\rangle.\mathbf{Var}\langle a,x\rangle \mathbin{[\!]} \mathsf{write} : s(y).\mathbf{Var}\langle a,y\rangle]$$

$s : T = \&\{\mathtt{read}{:}\,![\mathtt{nat}];\mathtt{end}, \mathtt{write}{:}\,?[\mathtt{nat}];\mathtt{end}\}$

$a : \langle T, \overline{T}\rangle$

$x : \mathtt{nat}$

Typing derivation for the $\mathbf{Var}$ process:

$$\dfrac{\Gamma \vdash a \rhd \langle T,\overline{T}\rangle \qquad \dfrac{\dfrac{}{\Theta;\Gamma \vdash s(y).\mathbf{Var}\langle a,y\rangle \rhd s : ?[\mathtt{nat}];\mathtt{end}}\,[\mathrm{Recv}] \qquad \Theta;\Gamma \vdash \overline{s}\langle x\rangle.\mathbf{Var}\langle a,x\rangle \rhd s : ![\mathtt{nat}];\mathtt{end}}{\Theta;\Gamma \vdash s \rhd [\mathsf{read} : \overline{s}\langle x\rangle.\mathbf{Var}\langle a,x\rangle \mathbin{[\!]} \mathsf{write} : s(y).\mathbf{Var}\langle a,y\rangle] \rhd s : T}\,[\mathrm{Bra}]}{\Theta;\Gamma \vdash a(s).s \rhd [\mathsf{read} : \overline{s}\langle x\rangle.\mathbf{Var}\langle a,x\rangle \mathbin{[\!]} \mathsf{write} : s(y).\mathbf{Var}\langle a,y\rangle] \rhd \Delta}\,[\mathrm{Acc}]$$

where

$$T = \&\{\mathtt{read}{:}\,![\mathtt{nat}];\mathtt{end}, \mathtt{write}{:}\,?[\mathtt{nat}];\mathtt{end}\}$$

and

$$\Theta = \Theta' \cdot \mathbf{Var} : [\langle T,\overline{T}\rangle;\mathtt{nat}]$$

Δ is empty

$$\frac{\Theta; \Gamma \cdot \tilde{x} : \tilde{S} \vdash P \triangleright \Delta \cdot s : T}{\Theta; \Gamma \vdash s(\tilde{x}).P \triangleright \Delta \cdot s : ?[\tilde{S}]; T} \text{[Recv]}$$

$$\mathbf{Var}(a, x) \stackrel{\text{df}}{=} a(s).s \triangleright [\mathsf{read} : \overline{s}\langle x \rangle.\mathbf{Var}\langle a, x\rangle \; [\!] \; \mathsf{write} : s(y).\mathbf{Var}\langle a, y\rangle]$$

$s : T = \&\{\texttt{read}: ![\mathbf{nat}]; \mathbf{end}, \texttt{write}: ?[\mathbf{nat}]; \mathbf{end}\}$

$a : \langle T, \overline{T} \rangle$

$x : \mathbf{nat}$

Typing derivation for the $\mathbf{Var}$ process:

$$\dfrac{\Gamma \vdash a \triangleright \langle T, \overline{T}\rangle \qquad \dfrac{\dfrac{}{\Theta; \Gamma \vdash s(y).\mathbf{Var}\langle a, y\rangle \triangleright s : ?[\mathbf{nat}]; \mathbf{end}} \text{[Recv]} \qquad \Theta; \Gamma \vdash \overline{s}\langle x\rangle.\mathbf{Var}\langle a, x\rangle \triangleright s : ![\mathbf{nat}]; \mathbf{end}}{\Theta; \Gamma \vdash s \triangleright [\mathsf{read} : \overline{s}\langle x\rangle.\mathbf{Var}\langle a, x\rangle \; [\!] \; \mathsf{write} : s(y).\mathbf{Var}\langle a, y\rangle] \triangleright s : T} \text{[Bra]}}{\Theta; \Gamma \vdash a(s).s \triangleright [\mathsf{read} : \overline{s}\langle x\rangle.\mathbf{Var}\langle a, x\rangle \; [\!] \; \mathsf{write} : s(y).\mathbf{Var}\langle a, y\rangle] \triangleright \Delta} \text{[Acc]}$$

where

$$T = \&\{\texttt{read}: ![\mathbf{nat}]; \mathbf{end}, \texttt{write}: ?[\mathbf{nat}]; \mathbf{end}\}$$

and

$$\Theta = \Theta' \cdot \mathbf{Var} : [\langle T, \overline{T}\rangle; \mathbf{nat}]$$

Δ is empty

$$\frac{\Delta \text{ contains only end} \qquad \boxed{\Gamma \vdash e \triangleright S}}{\Theta \cdot X : S\,T; \Gamma \vdash X\langle e, s\rangle \triangleright \Delta \cdot s : T}\,[\text{Var}]$$

$\mathbf{Var}(a, x) \stackrel{\text{df}}{=} a(s).s \triangleright [\text{read} : \overline{s}\langle x\rangle.\mathbf{Var}\langle a, x\rangle \; [\!] \; \text{write} : s(y).\mathbf{Var}\langle a, y\rangle]$

$s : T = \&\{\mathtt{read}{:}![\mathbf{nat}]; \mathbf{end}, \mathtt{write}{:}?[\mathbf{nat}]; \mathbf{end}\}$

$a : \langle T, \overline{T}\rangle$

$x : \mathbf{nat}$

Typing derivation for the $\mathbf{Var}$ process:

$$\dfrac{\Gamma \vdash a \triangleright \langle T, \overline{T}\rangle \qquad \dfrac{\dfrac{\dfrac{\Gamma \vdash a \triangleright \langle T, \overline{T}\rangle \qquad \Gamma \cdot y : \mathbf{nat} \vdash y \triangleright \mathbf{nat}}{\Theta; \Gamma \cdot y : \mathbf{nat} \vdash \mathbf{Var}\langle a, y\rangle \triangleright s : \mathbf{end}}\,[\text{Var}]}{\Theta; \Gamma \vdash s(y).\mathbf{Var}\langle a, y\rangle \triangleright s : ?[\mathbf{nat}]; \mathbf{end}}\,[\text{Recv}] \qquad \Theta; \Gamma \vdash \overline{s}\langle x\rangle.\mathbf{Var}\langle a, x\rangle \triangleright s : ![\mathbf{nat}]; \mathbf{end}}{\Theta; \Gamma \vdash s \triangleright [\text{read} : \overline{s}\langle x\rangle.\mathbf{Var}\langle a, x\rangle \; [\!] \; \text{write} : s(y).\mathbf{Var}\langle a, y\rangle] \triangleright s : T}\,[\text{Bra}]}{\Theta; \Gamma \vdash a(s).s \triangleright [\text{read} : \overline{s}\langle x\rangle.\mathbf{Var}\langle a, x\rangle \; [\!] \; \text{write} : s(y).\mathbf{Var}\langle a, y\rangle] \triangleright \Delta}\,[\text{Acc}]$$

where

$$T = \&\{\mathtt{read}{:}![\mathbf{nat}]; \mathbf{end}, \mathtt{write}{:}?[\mathbf{nat}]; \mathbf{end}\}$$

and

$$\Theta = \Theta' \cdot \mathbf{Var} : [\langle T, \overline{T}\rangle; \mathbf{nat}]$$

Δ is empty

$$\frac{\boxed{\Gamma \vdash \tilde{e} \triangleright \tilde{S}} \qquad \Theta; \Gamma \vdash P \triangleright \Delta \cdot s : T}{\Theta; \Gamma \vdash \overline{s}\langle \tilde{e} \rangle.P \triangleright \Delta \cdot s : ![\tilde{S}]; T} \text{[Send]}$$

$$\mathbf{Var}(a, x) \stackrel{\text{df}}{=} a(s).s \triangleright [\mathsf{read} : \overline{s}\langle x \rangle.\mathbf{Var}\langle a, x \rangle \mathbin{[\!]} \mathsf{write} : s(y).\mathbf{Var}\langle a, y \rangle]$$

$s : T = \&\{\mathtt{read}: ![\mathtt{nat}]; \mathtt{end}, \mathtt{write}: ?[\mathtt{nat}]; \mathtt{end}\}$

$a : \langle T, \overline{T} \rangle$

$x : \mathtt{nat}$

Typing derivation for the $\mathbf{Var}$ process:

$$\dfrac{\dfrac{\dfrac{\dfrac{\Gamma \vdash a \triangleright \langle T, \overline{T} \rangle \qquad \Gamma \cdot y : \mathtt{nat} \vdash y \triangleright \mathtt{nat}}{\Theta; \Gamma \cdot y : \mathtt{nat} \vdash \mathbf{Var}\langle a, y \rangle \triangleright s : \mathtt{end}} \text{[Var]}}{\Theta; \Gamma \vdash s(y).\mathbf{Var}\langle a, y \rangle \triangleright s : ?[\mathtt{nat}]; \mathtt{end}} \text{[Recv]} \qquad \dfrac{\dfrac{}{\Gamma \vdash x \triangleright \mathtt{nat}}}{\Theta; \Gamma \vdash \overline{s}\langle x \rangle.\mathbf{Var}\langle a, x \rangle \triangleright s : ![\mathtt{nat}]; \mathtt{end}} \text{[Send]}}{\Gamma \vdash a \triangleright \langle T, \overline{T} \rangle \qquad \Theta; \Gamma \vdash s \triangleright [\mathsf{read} : \overline{s}\langle x \rangle.\mathbf{Var}\langle a, x \rangle \mathbin{[\!]} \mathsf{write} : s(y).\mathbf{Var}\langle a, y \rangle] \triangleright s : T} \text{[Bra]}}{\Theta; \Gamma \vdash a(s).s \triangleright [\mathsf{read} : \overline{s}\langle x \rangle.\mathbf{Var}\langle a, x \rangle \mathbin{[\!]} \mathsf{write} : s(y).\mathbf{Var}\langle a, y \rangle] \triangleright \Delta} \text{[Acc]}$$

where

$$T = \&\{\mathtt{read}: ![\mathtt{nat}]; \mathtt{end}, \mathtt{write}: ?[\mathtt{nat}]; \mathtt{end}\}$$

and

$$\Theta = \Theta' \cdot \mathbf{Var} : [\langle T, \overline{T} \rangle; \mathtt{nat}]$$

Δ is empty

$$\frac{\Gamma \vdash \tilde{e} \triangleright \tilde{S} \qquad \boxed{\Theta; \Gamma \vdash P \triangleright \Delta \cdot s : T}}{\Theta; \Gamma \vdash \overline{s}\langle \tilde{e} \rangle . P \triangleright \Delta \cdot s :! [\tilde{S}]; T} \text{[Send]}$$

$$\mathbf{Var}(a, x) \stackrel{\text{df}}{=} a(s).s \triangleright [\text{read} : \overline{s}\langle x \rangle . \mathbf{Var}\langle a, x \rangle \;[\!]\; \text{write} : s(y).\mathbf{Var}\langle a, y \rangle]$$

$s : T = \&\{\mathtt{read}:![\mathtt{nat}]; \mathtt{end}, \mathtt{write}:?[\mathtt{nat}]; \mathtt{end}\}$

$a : \langle T, \overline{T} \rangle$

$x : \mathtt{nat}$

Typing derivation for the $\mathbf{Var}$ process:

$$\dfrac{\Gamma \vdash a \triangleright \langle T, \overline{T} \rangle \qquad \dfrac{\dfrac{\dfrac{\Gamma \vdash a \triangleright \langle T, \overline{T} \rangle \qquad \Gamma \cdot y : \mathtt{nat} \vdash y \triangleright \mathtt{nat}}{\Theta; \Gamma \cdot y : \mathtt{nat} \vdash \mathbf{Var}\langle a, y \rangle \triangleright s : \mathtt{end}} \text{[Var]}}{\Theta; \Gamma \vdash s(y).\mathbf{Var}\langle a, y \rangle \triangleright s :?[\mathtt{nat}]; \mathtt{end}} \text{[Recv]} \qquad \dfrac{\dfrac{}{\Gamma \vdash x \triangleright \mathtt{nat}} \qquad \Theta; \Gamma \vdash \mathbf{Var}\langle a, x \rangle \triangleright s : \mathtt{end}}{\Theta; \Gamma \vdash \overline{s}\langle x \rangle . \mathbf{Var}\langle a, x \rangle \triangleright s :![\mathtt{nat}]; \mathtt{end}} \text{[Send]}}{\Theta; \Gamma \vdash s \triangleright [\text{read} : \overline{s}\langle x \rangle . \mathbf{Var}\langle a, x \rangle \;[\!]\; \text{write} : s(y).\mathbf{Var}\langle a, y \rangle] \triangleright s : T} \text{[Bra]}}{\Theta; \Gamma \vdash a(s).s \triangleright [\text{read} : \overline{s}\langle x \rangle . \mathbf{Var}\langle a, x \rangle \;[\!]\; \text{write} : s(y).\mathbf{Var}\langle a, y \rangle] \triangleright \Delta} \text{[Acc]}$$

where

$$T = \&\{\mathtt{read}:![\mathtt{nat}]; \mathtt{end}, \mathtt{write}:?[\mathtt{nat}]; \mathtt{end}\}$$

and

$$\Theta = \Theta' \cdot \mathbf{Var} : [\langle T, \overline{T} \rangle; \mathtt{nat}]$$

Δ is empty

$$\dfrac{\Delta \text{ contains only } \mathsf{end} \qquad \boxed{\Gamma \vdash e \triangleright S}}{\Theta \cdot X : S\,T; \Gamma \vdash X\langle e, s\rangle \triangleright \Delta \cdot s : T}\,[\mathsf{Var}]$$

$$\mathbf{Var}(a,x) \stackrel{\mathsf{df}}{=} a(s).s \triangleright [\mathsf{read} : \overline{s}\langle x\rangle.\mathbf{Var}\langle a,x\rangle \ \| \ \mathsf{write} : s(y).\mathbf{Var}\langle a,y\rangle]$$

$s : T = \&\{\mathtt{read}\!:\!![\mathtt{nat}];\mathsf{end}, \mathtt{write}\!:\!?[\mathtt{nat}];\mathsf{end}\}$

$a : \langle T, \overline{T}\rangle$

$x : \mathtt{nat}$

Typing derivation for the $\mathbf{Var}$ process:

$$\dfrac{\Gamma \vdash a \triangleright \langle T,\overline{T}\rangle \qquad \dfrac{\dfrac{\dfrac{\Gamma \vdash a \triangleright \langle T,\overline{T}\rangle \qquad \Gamma\cdot y : \mathtt{nat} \vdash y \triangleright \mathtt{nat}}{\Theta;\Gamma\cdot y:\mathtt{nat} \vdash \mathbf{Var}\langle a,y\rangle \triangleright s : \mathsf{end}}\,[\mathsf{Var}]}{\Theta;\Gamma \vdash s(y).\mathbf{Var}\langle a,y\rangle \triangleright s : ?[\mathtt{nat}];\mathsf{end}}\,[\mathsf{Recv}] \qquad \dfrac{\dfrac{}{\Gamma \vdash x \triangleright \mathtt{nat}} \qquad \dfrac{\dfrac{}{\Gamma \vdash a \triangleright \langle T,\overline{T}\rangle} \qquad \dfrac{}{\Gamma \vdash x \triangleright \mathtt{nat}}}{\Theta;\Gamma \vdash \mathbf{Var}\langle a,x\rangle \triangleright s : \mathsf{end}}\,[\mathsf{Var}]}{\Theta;\Gamma \vdash \overline{s}\langle x\rangle.\mathbf{Var}\langle a,x\rangle \triangleright s : ![\mathtt{nat}];\mathsf{end}}\,[\mathsf{Send}]}{\Theta;\Gamma \vdash s \triangleright [\mathsf{read} : \overline{s}\langle x\rangle.\mathbf{Var}\langle a,x\rangle \ \| \ \mathsf{write} : s(y).\mathbf{Var}\langle a,y\rangle] \triangleright s : T}\,[\mathsf{Bra}]}{\Theta;\Gamma \vdash a(s).s \triangleright [\mathsf{read} : \overline{s}\langle x\rangle.\mathbf{Var}\langle a,x\rangle \ \| \ \mathsf{write} : s(y).\mathbf{Var}\langle a,y\rangle] \triangleright \Delta}\,[\mathsf{Acc}]$$

where

$$T = \&\{\mathtt{read}\!:\!![\mathtt{nat}];\mathsf{end}, \mathtt{write}\!:\!?[\mathtt{nat}];\mathsf{end}\}$$

and

$$\Theta = \Theta' \cdot \mathbf{Var} : [\langle T, \overline{T}\rangle; \mathtt{nat}]$$

Δ is empty

$$\mathbf{Var}(a,x) \stackrel{\mathsf{df}}{=} a(s).s \triangleright [\mathsf{read} : \overline{s}\langle x\rangle.\mathbf{Var}\langle a,x\rangle \ [\!]\ \mathsf{write} : s(y).\mathbf{Var}\langle a,y\rangle]$$

$s : T = \&\{\mathtt{read}\colon ![\mathbf{nat}];\mathsf{end}, \mathtt{write}\colon ?[\mathbf{nat}];\mathsf{end}\}$

$a : \langle T, \overline{T}\rangle$

$x : \mathbf{nat}$

Typing derivation for the $\mathbf{Var}$ process:

$$\dfrac{\Gamma \vdash a \triangleright \langle T,\overline{T}\rangle \qquad \dfrac{\dfrac{\dfrac{\Gamma \vdash a \triangleright \langle T,\overline{T}\rangle \quad \Gamma\cdot y : \mathbf{nat} \vdash y \triangleright \mathbf{nat}}{\Theta;\Gamma\cdot y:\mathbf{nat} \vdash \mathbf{Var}\langle a,y\rangle \triangleright s : \mathsf{end}}\,[\mathsf{Var}]}{\Theta;\Gamma \vdash s(y).\mathbf{Var}\langle a,y\rangle \triangleright s : ?[\mathbf{nat}];\mathsf{end}}\,[\mathsf{Recv}] \qquad \dfrac{\dfrac{}{\Gamma \vdash x \triangleright \mathbf{nat}} \quad \dfrac{\dfrac{}{\Gamma \vdash a \triangleright \langle T,\overline{T}\rangle} \quad \dfrac{}{\Gamma \vdash x \triangleright \mathbf{nat}}}{\Theta;\Gamma \vdash \mathbf{Var}\langle a,x\rangle \triangleright s : \mathsf{end}}\,[\mathsf{Var}]}{\Theta;\Gamma \vdash \overline{s}\langle x\rangle.\mathbf{Var}\langle a,x\rangle \triangleright s : ![\mathbf{nat}];\mathsf{end}}\,[\mathsf{Send}]}{\Theta;\Gamma \vdash s \triangleright [\mathsf{read} : \overline{s}\langle x\rangle.\mathbf{Var}\langle a,x\rangle \ [\!]\ \mathsf{write} : s(y).\mathbf{Var}\langle a,y\rangle] \triangleright s : T}\,[\mathsf{Bra}]}{\Theta;\Gamma \vdash a(s).s \triangleright [\mathsf{read} : \overline{s}\langle x\rangle.\mathbf{Var}\langle a,x\rangle \ [\!]\ \mathsf{write} : s(y).\mathbf{Var}\langle a,y\rangle] \triangleright \Delta}\,[\mathsf{Acc}]$$

where

$$T = \&\{\mathtt{read}\colon ![\mathbf{nat}];\mathsf{end}, \mathtt{write}\colon ?[\mathbf{nat}];\mathsf{end}\}$$

and

$$\Theta = \Theta' \cdot \mathbf{Var} : [\langle T,\overline{T}\rangle ; \mathbf{nat}]$$

Δ is empty

DERIVATION TREE

$$\dfrac{\Gamma \vdash a \rhd \langle T, \overline{T}\rangle \qquad \Theta;\Gamma \vdash s \rhd [\mathsf{read} : \overline{s}\langle x\rangle.\mathbf{Var}\langle a,x\rangle \,[\!]\, \mathsf{write} : s(y).\mathbf{Var}\langle a,y\rangle] \rhd s : T}{\Theta;\Gamma \vdash a(s).s \rhd [\mathsf{read} : \overline{s}\langle x\rangle.\mathbf{Var}\langle a,x\rangle \,[\!]\, \mathsf{write} : s(y).\mathbf{Var}\langle a,y\rangle] \rhd \Delta}\,[\mathrm{Acc}]$$

where the second premise is derived by

$$\dfrac{\dfrac{\dfrac{\Gamma \vdash a \rhd \langle T,\overline{T}\rangle \qquad \Gamma\cdot y : \mathtt{nat} \vdash y \rhd \mathtt{nat}}{\Theta;\Gamma\cdot y:\mathtt{nat} \vdash \mathbf{Var}\langle a,y\rangle \rhd s : \mathtt{end}}\,[\mathrm{Var}]}{\Theta;\Gamma \vdash s(y).\mathbf{Var}\langle a,y\rangle \rhd s : ?[\mathtt{nat}];\mathtt{end}}\,[\mathrm{Recv}] \qquad \dfrac{\dfrac{}{\Gamma \vdash x \rhd \mathtt{nat}} \qquad \dfrac{\Gamma \vdash a \rhd \langle T,\overline{T}\rangle \qquad \Gamma \vdash x \rhd \mathtt{nat}}{\Theta;\Gamma \vdash \mathbf{Var}\langle a,x\rangle \rhd s : \mathtt{end}}\,[\mathrm{Var}]}{\Theta;\Gamma \vdash \overline{s}\langle x\rangle.\mathbf{Var}\langle a,x\rangle \rhd s : ![\mathtt{nat}];\mathtt{end}}\,[\mathrm{Send}]}{\Theta;\Gamma \vdash s \rhd [\mathsf{read} : \overline{s}\langle x\rangle.\mathbf{Var}\langle a,x\rangle \,[\!]\, \mathsf{write} : s(y).\mathbf{Var}\langle a,y\rangle] \rhd s : T}\,[\mathrm{Bra}]$$

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma \vdash 5 \rhd \mathtt{nat}}\,[\mathrm{Nat}] \qquad \dfrac{}{\Theta;\Gamma \vdash \mathbf{0} \rhd s : \mathtt{end}}\,[\mathrm{Inact}]}{\Theta;\Gamma \vdash \overline{s}\langle 5\rangle.\mathbf{0} \rhd s : ![\mathtt{nat}];\mathtt{end}}\,[\mathrm{Send}]}{\Theta;\Gamma \vdash s \lhd \mathsf{write}.\overline{s}\langle 5\rangle.\mathbf{0} \rhd s : \oplus\{\mathtt{read}:?[\mathtt{nat}];\mathtt{end}, \mathtt{write}:![\mathtt{nat}];\mathtt{end}\}}\,[\mathrm{Sel}]}{\Theta;\Gamma\cdot a : \langle T,\overline{T}\rangle \vdash \overline{a}(s).s \lhd \mathsf{write}.\overline{s}\langle 5\rangle.\mathbf{0} \rhd \Delta}\,[\mathrm{Req}]$$

s has a type T

s has a type $\overline{T}$

$$a(s).s \rhd [\mathsf{read} : \overline{s}\langle x\rangle.\mathbf{Var}\langle a,x\rangle \,[\!]\, \mathsf{write} : s(y).\mathbf{Var}\langle a,y\rangle] \quad \Big| \quad \overline{a}(s).s \lhd \mathsf{write}.\overline{s}\langle x\rangle.\mathbf{0}$$

$$\mathbf{Var}\langle a, 0\rangle \mid \mathbf{Writer}\langle a, 5\rangle$$

a has a type $\langle T, \overline{T}\rangle$

EXERCISE

So far we have shown that:
(1) variable process is typable with $a : \langle T, \overline{T} \rangle$ and $s : T$
(2) writer process is typable with $a : \langle T, \overline{T} \rangle$ and $s : \overline{T}$

The final step is to show that our whole program is typable:
$\mathbf{Var}(a, x) \stackrel{\mathsf{df}}{=} P, \mathbf{Writer}(a, x) \stackrel{\mathsf{df}}{=} Q$ in $(\mathbf{Var}\langle a, 0 \rangle \,|\, \mathbf{Writer}\langle a, 5 \rangle)$

Using the rule [Def] and the rule [Var] try to complete the derivation tree on your own.

EXERCISE: IS IT WELL-TYPED?

$$s(x : \texttt{Bool}).\mathbf{0} \mid \overline{s}\langle \texttt{true} \rangle.\mathbf{0}$$

$$\overline{s}\langle x \rangle.\mathbf{0} \mid \overline{s}\langle \texttt{true} \rangle.\mathbf{0}$$

$$\overline{s}\langle \texttt{false} \rangle.\mathbf{0} \mid s(x : \texttt{Bool}).\mathbf{0} \mid s(y : \texttt{Bool})$$

$$\overline{s}\langle 42 \rangle.s'(x : \texttt{Int}).\mathbf{0} \mid \overline{s}\langle 11 \rangle.s(y : \texttt{Int})$$

$$s(x : \texttt{Int}).\overline{s'}\langle 42 \rangle.\mathbf{0} \mid \overline{s}\langle 11 \rangle.s'(y : \texttt{Int}).\mathbf{0}$$

$$s \triangleleft \mathsf{ack}.P \mid s \triangleright \{\mathsf{req}_1 : P_1 \mathbin{[\!]} \cdots \mathbin{[\!]} \mathsf{req}_n : P_n\}$$

EXERCISE: IS IT WELL-TYPED? ❌/✅

$s(x : \mathtt{Bool}).\mathbf{0} \mid \overline{s}\langle \mathtt{true} \rangle.\mathbf{0}$ ✅

$\overline{s}\langle x \rangle.\mathbf{0} \mid \overline{s}\langle \mathtt{true} \rangle.\mathbf{0}$ ❌

$\overline{s}\langle \mathtt{false} \rangle.\mathbf{0} \mid s(x : \mathtt{Bool}).\mathbf{0} \mid s(y : \mathtt{Bool})$ ❌

$\overline{s}\langle 42 \rangle.s'(x : \mathtt{Int}).\mathbf{0} \mid \overline{s}\langle 11 \rangle.s(y : \mathtt{Int})$ ✅

$s(x : \mathtt{Int}).\overline{s'}\langle 42 \rangle.\mathbf{0} \mid \overline{s}\langle 11 \rangle.s'(y : \mathtt{Int}).\mathbf{0}$ ✅

$s \lhd \mathsf{ack}.P \mid s \rhd \{\mathsf{req}_1 : P_1 \ [\!]\ \cdots \ [\!]\ \mathsf{req}_n : P_n\}$ ❌

TYPING RECURSIVE PROCESSES

Consider the following counter process X that counts from 0 to infinity.
$\mathtt{def}\ X(x,s) = \overline{s}\langle x\rangle.X\langle x+1,s\rangle$ in $X\langle 0,s\rangle$

The type for X is $[\mathbf{nat}];T$ where $T = \mu t.![\mathbf{nat}];t$
The type for s is T

The derivation tree is given below:

$$\dfrac{\dfrac{\dfrac{}{\Gamma' \vdash x \triangleright \mathbf{nat}}\,[\text{nat}] \qquad \dfrac{\dfrac{\dfrac{}{\Gamma' \vdash x \triangleright \mathbf{nat}}\,[\text{nat}]}{\Gamma' \vdash x+1 \triangleright \mathbf{nat}}\,[\text{sum}]}{\Theta';\Gamma' \vdash X\langle x+1,s\rangle \triangleright s : T}\,[\text{Var}]}{\Theta';\Gamma' \vdash \overline{s}\langle x\rangle.X\langle x+1,s\rangle \triangleright s :![\mathbf{nat}];T}\,[\text{Send}] \qquad \dfrac{\dfrac{}{\Gamma \vdash 0 \triangleright \mathbf{nat}}}{\Theta';\Gamma \vdash X\langle 0,s\rangle \triangleright s : T}\,[\text{Var}]}{\Theta;\Gamma \vdash \mathtt{def}\ X(x,s) = \overline{s}\langle x\rangle.X\langle x+1,s\rangle \text{ in } X\langle 0,s\rangle \triangleright s : T}\,[\text{Def}]$$

where $\Theta' = \Theta \cdot X : \mathbf{nat};T$ and $\Gamma' = \Gamma \cdot x : \mathbf{nat}$

$$\dfrac{\Theta \cdot X : ST;\Gamma \cdot x : S \vdash P \triangleright s : T \qquad \Theta \cdot X : ST;\Gamma \vdash Q \triangleright \Delta}{\Theta;\Gamma \vdash \mathtt{def}\ X(x,s) = P \text{ in } Q \triangleright \Delta}\,[\text{Def}]$$

$$\dfrac{\Delta \text{ contains only } \mathtt{end} \qquad \Gamma \vdash e \triangleright S}{\Theta \cdot X : ST;\Gamma \vdash X\langle e,s\rangle \triangleright \Delta \cdot s : T}\,[\text{Var}]$$

$$\frac{\Theta \cdot X : S\,T; \Gamma \cdot x : S \vdash P \triangleright s : T \qquad \Theta \cdot X : S\,T; \Gamma \vdash Q \triangleright \Delta}{\Theta; \Gamma \vdash \mathtt{def}\ X(x, s) = P \text{ in } Q \triangleright \Delta}\text{[Def]}$$

Consider the following counter process X that counts from 0 to infinity.
$\mathtt{def}\ X(x, s) = \overline{s}\langle x\rangle.X\langle x + 1, s\rangle$ in $X\langle 0, s\rangle$

The type for X is $[\mathtt{nat}]; T$ where $T = \mu t.![\mathtt{nat}]; t$
The type for s is T

$$\frac{}{\Theta; \Gamma \vdash \mathtt{def}\ X(x, s) = \overline{s}\langle x\rangle.X\langle x + 1, s\rangle \text{ in } X\langle 0, s\rangle \triangleright s : T}\text{[Def]}$$

where $\Theta' = \Theta \cdot X : \mathtt{nat}; T$ and $\Gamma' = \Gamma \cdot x : \mathtt{nat}$

$$\frac{\Theta \cdot X : ST; \Gamma \cdot x : S \vdash P \triangleright s : T \qquad \Theta \cdot X : ST; \Gamma \vdash Q \triangleright \Delta}{\Theta; \Gamma \vdash \mathbf{def}\ X(x, s) = P \text{ in } Q \triangleright \Delta}\ [\text{Def}]$$

Consider the following counter process X that counts from 0 to infinity.
$\mathbf{def}\ X(x, s) = \overline{s}\langle x\rangle.X\langle x+1, s\rangle$ in $X\langle 0, s\rangle$

The type for X is $[\mathtt{nat}]; T$ where $T = \mu t.![\mathtt{nat}]; t$
The type for s is T

$$\frac{\dfrac{}{\Theta'; \Gamma' \vdash \overline{s}\langle x\rangle.X\langle x+1, s\rangle \triangleright s :![\mathtt{nat}]; T}}{\Theta; \Gamma \vdash \mathbf{def}\ X(x, s) = \overline{s}\langle x\rangle.X\langle x+1, s\rangle \text{ in } X\langle 0, s\rangle \triangleright s : T}\ [\text{Def}]$$

where $\Theta' = \Theta \cdot X : \mathtt{nat}; T$ and $\Gamma' = \Gamma \cdot x : \mathtt{nat}$

$$\dfrac{\Theta \cdot X : ST; \Gamma \cdot x : S \vdash P \triangleright s : T \qquad \boxed{\Theta \cdot X : ST; \Gamma \vdash Q \triangleright \Delta}}{\Theta; \Gamma \vdash \mathtt{def}\ \ X(x, s) = P \text{ in } Q \triangleright \Delta}\,[\text{Def}]$$

Consider the following counter process X that counts from 0 to infinity.
$\mathtt{def}\ \ X(x, s) = \overline{s}\langle x \rangle . X\langle x + 1, s\rangle$ in $X\langle 0, s\rangle$

The type for X is $[\mathtt{nat}]; T$ where $T = \mu t.![\mathtt{nat}]; t$
The type for s is T

$$\dfrac{\dfrac{}{\Theta'; \Gamma' \vdash \overline{s}\langle x \rangle . X\langle x + 1, s\rangle \triangleright s :![\mathtt{nat}]; T} \qquad \dfrac{}{\Theta'; \Gamma \vdash X\langle 0, s\rangle \triangleright s : T}}{\Theta; \Gamma \vdash \mathtt{def}\ \ X(x, s) = \overline{s}\langle x \rangle . X\langle x + 1, s\rangle \text{ in } X\langle 0, s\rangle \triangleright s : T}\,[\text{Def}]$$

where $\Theta' = \Theta \cdot X : \mathtt{nat}; T$ and $\Gamma' = \Gamma \cdot x : \mathtt{nat}$

$$\frac{\Gamma \vdash \tilde{e} \triangleright \tilde{S} \qquad \Theta; \Gamma \vdash P \triangleright \Delta \cdot s : T}{\Theta; \Gamma \vdash \overline{s}\langle \tilde{e} \rangle . P \triangleright \Delta \cdot s :! [\tilde{S}]; T} \text{[Send]}$$

Consider the following counter process X that counts from 0 to infinity.
$\mathtt{def}\ X(x, s) = \overline{s}\langle x \rangle . X\langle x + 1, s \rangle$ in $X\langle 0, s \rangle$

The type for X is $[\mathtt{nat}]; T$ where $T = \mu t.![\mathtt{nat}]; t$
The type for s is T

$$\dfrac{\dfrac{}{\Theta'; \Gamma' \vdash \overline{s}\langle x \rangle . X\langle x + 1, s \rangle \triangleright s :! [\mathtt{nat}]; T} \text{[Send]} \qquad \dfrac{}{\Theta'; \Gamma \vdash X\langle 0, s \rangle \triangleright s : T}}{\Theta; \Gamma \vdash \mathtt{def}\ X(x, s) = \overline{s}\langle x \rangle . X\langle x + 1, s \rangle \text{ in } X\langle 0, s \rangle \triangleright s : T} \text{[Def]}$$

where $\Theta' = \Theta \cdot X : \mathtt{nat}; T$ and $\Gamma' = \Gamma \cdot x : \mathtt{nat}$

$$\frac{\Gamma \vdash \tilde{e} \triangleright \tilde{S} \qquad \Theta; \Gamma \vdash P \triangleright \Delta \cdot s : T}{\Theta; \Gamma \vdash \overline{s}\langle \tilde{e} \rangle.P \triangleright \Delta \cdot s :![\tilde{S}]; T} \text{[Send]}$$

Consider the following counter process X that counts from 0 to infinity.
$\mathtt{def}\ X(x, s) = \overline{s}\langle x \rangle.X\langle x + 1, s \rangle$ in $X\langle 0, s \rangle$

The type for X is $[\mathtt{nat}]; T$ where $T = \mu t.![\mathtt{nat}]; t$
The type for s is T

$$\cfrac{\cfrac{\cfrac{}{\Gamma' \vdash x \triangleright \mathtt{nat}} \qquad \cfrac{}{\Theta'; \Gamma' \vdash X\langle x + 1, s \rangle \triangleright s : T}}{\Theta'; \Gamma' \vdash \overline{s}\langle x \rangle.X\langle x + 1, s \rangle \triangleright s :![\mathtt{nat}]; T} \text{[Send]} \qquad \cfrac{}{\Theta'; \Gamma \vdash X\langle 0, s \rangle \triangleright s : T}}{\Theta; \Gamma \vdash \mathtt{def}\ X(x, s) = \overline{s}\langle x \rangle.X\langle x + 1, s \rangle \text{ in } X\langle 0, s \rangle \triangleright s : T} \text{[Def]}$$

where $\Theta' = \Theta \cdot X : \mathtt{nat}; T$ and $\Gamma' = \Gamma \cdot x : \mathtt{nat}$

Consider the following counter process X that counts from 0 to infinity.

$\mathtt{def}\ X(x,s) = \overline{s}\langle x\rangle.X\langle x+1,s\rangle$ in $X\langle 0,s\rangle$

The type for X is $[\mathtt{nat}];T$ where $T = \mu t.![\mathtt{nat}];t$

The type for s is T

$$\dfrac{\dfrac{\dfrac{}{\Gamma' \vdash x \triangleright \mathtt{nat}}\,[\mathsf{nat}] \quad \dfrac{}{\Theta';\Gamma' \vdash X\langle x+1,s\rangle \triangleright s : T}}{\Theta';\Gamma' \vdash \overline{s}\langle x\rangle.X\langle x+1,s\rangle \triangleright s :![\mathtt{nat}];T}\,[\mathsf{Send}] \quad \dfrac{}{\Theta';\Gamma \vdash X\langle 0,s\rangle \triangleright s : T}}{\Theta;\Gamma \vdash \mathtt{def}\ X(x,s) = \overline{s}\langle x\rangle.X\langle x+1,s\rangle \text{ in } X\langle 0,s\rangle \triangleright s : T}\,[\mathsf{Def}]$$

where $\Theta' = \Theta \cdot X : \mathtt{nat};T$ and $\Gamma' = \Gamma \cdot x : \mathtt{nat}$

$$\frac{\Delta \text{ contains only } \mathbf{end} \qquad \Gamma \vdash e \triangleright S}{\Theta \cdot X : S\,T; \Gamma \vdash X\langle e, s\rangle \triangleright \Delta \cdot s : T}\ [\mathrm{Var}]$$

Consider the following counter process X that counts from 0 to infinity.
$\mathbf{def}\ X(x,s) = \overline{s}\langle x\rangle.X\langle x+1, s\rangle$ in $X\langle 0, s\rangle$

The type for X is $[\mathbf{nat}]; T$ where $T = \mu t.![\mathbf{nat}]; t$
The type for s is T

$$\cfrac{\cfrac{\cfrac{}{\Gamma' \vdash x \triangleright \mathbf{nat}}\ [\mathrm{nat}] \qquad \cfrac{}{\Theta'; \Gamma' \vdash X\langle x+1, s\rangle \triangleright s : T}\ [\mathrm{Var}]}{\Theta'; \Gamma' \vdash \overline{s}\langle x\rangle.X\langle x+1, s\rangle \triangleright s :![\mathbf{nat}]; T}\ [\mathrm{Send}] \qquad \cfrac{}{\Theta'; \Gamma \vdash X\langle 0, s\rangle \triangleright s : T}}{\Theta; \Gamma \vdash \mathbf{def}\ X(x,s) = \overline{s}\langle x\rangle.X\langle x+1, s\rangle \text{ in } X\langle 0, s\rangle \triangleright s : T}\ [\mathrm{Def}]$$

where $\Theta' = \Theta \cdot X : \mathbf{nat}; T$ and $\Gamma' = \Gamma \cdot x : \mathbf{nat}$

$$\frac{\Delta \text{ contains only } \mathtt{end} \qquad \Gamma \vdash e \rhd S}{\Theta \cdot X : S\,T; \Gamma \vdash X\langle e, s\rangle \rhd \Delta \cdot s : T}\ [\text{Var}]$$

Consider the following counter process X that counts from 0 to infinity.
$\mathtt{def}\ X(x, s) = \overline{s}\langle x\rangle.X\langle x+1, s\rangle$ in $X\langle 0, s\rangle$

The type for X is $[\mathtt{nat}]; T$ where $T = \mu t.![\mathtt{nat}]; t$
The type for s is T

$$\dfrac{\dfrac{\dfrac{}{\Gamma' \vdash x \rhd \mathtt{nat}}\ [\text{nat}] \qquad \dfrac{\dfrac{}{\Gamma' \vdash x+1 \rhd \mathtt{nat}}\ [\text{sum}]}{\Theta'; \Gamma' \vdash X\langle x+1, s\rangle \rhd s : T}\ [\text{Var}]}{\Theta'; \Gamma' \vdash \overline{s}\langle x\rangle.X\langle x+1, s\rangle \rhd s :![\mathtt{nat}]; T}\ [\text{Send}] \qquad \dfrac{}{\Theta'; \Gamma \vdash X\langle 0, s\rangle \rhd s : T}}{\Theta; \Gamma \vdash \mathtt{def}\ X(x, s) = \overline{s}\langle x\rangle.X\langle x+1, s\rangle \text{ in } X\langle 0, s\rangle \rhd s : T}\ [\text{Def}]$$

where $\Theta' = \Theta \cdot X : \mathtt{nat}; T$ and $\Gamma' = \Gamma \cdot x : \mathtt{nat}$

Consider the following counter process X that counts from 0 to infinity.
$\mathtt{def}\ X(x, s) = \overline{s}\langle x\rangle.X\langle x+1, s\rangle$ in $X\langle 0, s\rangle$

The type for X is $[\mathbf{nat}]; T$ where $T = \mu t.![\mathbf{nat}]; t$
The type for s is T

$$
\dfrac{
\dfrac{
\dfrac{}{\Gamma' \vdash x \triangleright \mathbf{nat}}\,[\text{nat}]
\qquad
\dfrac{
\dfrac{
\dfrac{}{\Gamma' \vdash x \triangleright \mathbf{nat}}\,[\text{nat}]
}{\Gamma' \vdash x + 1 \triangleright \mathbf{nat}}\,[\text{sum}]
}{\Theta'; \Gamma' \vdash X\langle x+1, s\rangle \triangleright s : T}\,[\text{Var}]
}{\Theta'; \Gamma' \vdash \overline{s}\langle x\rangle.X\langle x+1, s\rangle \triangleright s :![\mathbf{nat}]; T}\,[\text{Send}]
\qquad
\dfrac{}{\Theta'; \Gamma \vdash X\langle 0, s\rangle \triangleright s : T}
}{\Theta; \Gamma \vdash \mathtt{def}\ X(x, s) = \overline{s}\langle x\rangle.X\langle x+1, s\rangle \text{ in } X\langle 0, s\rangle \triangleright s : T}\,[\text{Def}]
$$

where $\Theta' = \Theta \cdot X : \mathbf{nat}; T$ and $\Gamma' = \Gamma \cdot x : \mathbf{nat}$

Consider the following counter process X that counts from 0 to infinity.
$\mathtt{def}\ X(x, s) = \overline{s}\langle x\rangle.X\langle x + 1, s\rangle$ in $X\langle 0, s\rangle$

The type for X is $[\mathtt{nat}]; T$ where $T = \mu t.![\mathtt{nat}]; t$
The type for s is T

$$\dfrac{\dfrac{}{\Gamma' \vdash x \triangleright \mathtt{nat}}\ \text{[nat]} \qquad \dfrac{\dfrac{\dfrac{}{\Gamma' \vdash x \triangleright \mathtt{nat}}\ \text{[nat]}}{\Gamma' \vdash x + 1 \triangleright \mathtt{nat}}\ \text{[sum]}}{\Theta'; \Gamma' \vdash X\langle x + 1, s\rangle \triangleright s : T}\ \text{[Var]}}{\Theta'; \Gamma' \vdash \overline{s}\langle x\rangle.X\langle x + 1, s\rangle \triangleright s :![\mathtt{nat}]; T}\ \text{[Send]} \qquad \dfrac{}{\Theta'; \Gamma \vdash X\langle 0, s\rangle \triangleright s : T}\ \text{[Var]}$$
$$\overline{\Theta; \Gamma \vdash \mathtt{def}\ X(x, s) = \overline{s}\langle x\rangle.X\langle x + 1, s\rangle \text{ in } X\langle 0, s\rangle \triangleright s : T}\ \text{[Def]}$$

where $\Theta' = \Theta \cdot X : \mathtt{nat}; T$ and $\Gamma' = \Gamma \cdot x : \mathtt{nat}$

$$\frac{\Delta \text{ contains only } \mathtt{end} \qquad \Gamma \vdash e \triangleright S}{\Theta \cdot X : S\,T; \Gamma \vdash X\langle e, s\rangle \triangleright \Delta \cdot s : T}\,[\mathrm{Var}]$$

Consider the following counter process X that counts from 0 to infinity.
$\mathtt{def}\ X(x,s) = \overline{s}\langle x\rangle.X\langle x+1, s\rangle$ in $X\langle 0, s\rangle$

The type for X is $[\mathtt{nat}]; T$ where $T = \mu t.![\mathtt{nat}]; t$
The type for s is T

$$\dfrac{\dfrac{\dfrac{}{\Gamma' \vdash x \triangleright \mathtt{nat}}\,[\mathrm{nat}] \qquad \dfrac{\dfrac{}{\Gamma' \vdash x + 1 \triangleright \mathtt{nat}}\,[\mathrm{sum}]}{\Theta'; \Gamma' \vdash X\langle x+1, s\rangle \triangleright s : T}\,[\mathrm{Var}]}{\Theta'; \Gamma' \vdash \overline{s}\langle x\rangle.X\langle x+1, s\rangle \triangleright s : ![\mathtt{nat}]; T}\,[\mathrm{Send}] \qquad \dfrac{}{\Theta'; \Gamma \vdash X\langle 0, s\rangle \triangleright s : T}\,[\mathrm{Var}]}{\Theta; \Gamma \vdash \mathtt{def}\ X(x,s) = \overline{s}\langle x\rangle.X\langle x+1, s\rangle \text{ in } X\langle 0, s\rangle \triangleright s : T}\,[\mathrm{Def}]$$

where $\Theta' = \Theta \cdot X : \mathtt{nat}; T$ and $\Gamma' = \Gamma \cdot x : \mathtt{nat}$

$$\frac{\Delta \text{ contains only } \mathtt{end} \qquad \Gamma \vdash e \triangleright S}{\Theta \cdot X : ST; \Gamma \vdash X\langle e, s\rangle \triangleright \Delta \cdot s : T}\,[\mathrm{Var}]$$

Consider the following counter process X that counts from 0 to infinity.
$\mathtt{def}\ X(x,s) = \overline{s}\langle x\rangle.X\langle x+1,s\rangle$ in $X\langle 0,s\rangle$

The type for X is $[\mathtt{nat}]; T$ where $T = \mu t.![\mathtt{nat}]; t$
The type for s is T

$$\dfrac{\dfrac{\dfrac{}{\Gamma' \vdash x \triangleright \mathtt{nat}}\,[\mathrm{nat}] \quad \dfrac{\dfrac{}{\Gamma' \vdash x+1 \triangleright \mathtt{nat}}\,[\mathrm{sum}]}{\Theta';\Gamma' \vdash X\langle x+1,s\rangle \triangleright s : T}\,[\mathrm{Var}]}{\Theta';\Gamma' \vdash \overline{s}\langle x\rangle.X\langle x+1,s\rangle \triangleright s :![\mathtt{nat}]; T}\,[\mathrm{Send}] \quad \dfrac{\dfrac{}{\Gamma \vdash 0 \triangleright \mathtt{nat}}}{\Theta';\Gamma \vdash X\langle 0,s\rangle \triangleright s : T}\,[\mathrm{Var}]}{\Theta;\Gamma \vdash \mathtt{def}\ X(x,s) = \overline{s}\langle x\rangle.X\langle x+1,s\rangle \text{ in } X\langle 0,s\rangle \triangleright s : T}\,[\mathrm{Def}]$$

where $\Theta' = \Theta \cdot X : \mathtt{nat}; T$ and $\Gamma' = \Gamma \cdot x : \mathtt{nat}$

EXERCISE: TYPING RECURSIVE PROCESSES

Consider now the reader process:

Consider the variable example from the beginning of the lecture.

$$\mathbf{Var}(a, x) = a(s).\mathbf{Var_1}\langle x, s\rangle$$

$$\mathbf{Var_1}(x, s) = s \triangleright [\mathsf{read} : \overline{s}\langle x\rangle.\mathbf{Var_1}\langle x, s\rangle \; [] \; \mathsf{write} : s(y).\mathbf{Var_1}\langle y, s\rangle \; [] \; \mathsf{stop} : \mathbf{Var}\langle a, x\rangle]$$

Types:

$$T = \mu t.\&\{\mathtt{read}\!:\!\,![\mathtt{nat}]; t, \mathtt{write}\!:\!?[\mathtt{nat}]; t, \mathtt{stop}\!:\!\mathtt{end}\}$$

$$\overline{T} = \mu t. \oplus \{\mathbf{read}\!:\!?[\mathbf{nat}]; t, \mathbf{write}\!:!\,[\mathbf{nat}]; t, \mathbf{stop}\!:\!\mathbf{end}\}$$

Exercise: Show the type derivation for $\mathbf{Var_1} : \mathbf{nat}; T$ and $s : T$

TYPING DELEGATION

Sending a session channel

$$\frac{\Theta; \Gamma \vdash P \triangleright \Delta \cdot s : T'}{\Theta; \Gamma \vdash \overline{s}(s').P \triangleright \Delta \cdot s :![T]; T' \cdot s' : T} \text{[Thr]}$$

Receiving a session channel

$$\frac{\Theta; \Gamma \vdash P \triangleright \Delta \cdot s : T' \cdot s' : T}{\Theta; \Gamma \vdash s(s').P \triangleright \Delta \cdot s :?[T]; T'} \text{[Cat]}$$

SESSION TYPES

properties
or how to verify protocols

ERRORS: REVISITED

We wish to avoid the following runtime errors in various protocols.

- Base Type Error $\overline{s}\langle \texttt{Apple} \rangle.P_1 \mid s(x).\overline{s'}\langle 1 + x \rangle$
- Arity Mismatch $\overline{s}\langle 1 \rangle.P_1 \mid s(x, y).\overline{s'}\langle x + y \rangle$
- Label Undefined $s \triangleright \{\mathsf{repeat} : P_1 \ [\!] \ \mathsf{reject} : P_2\} \mid s \triangleleft \mathsf{apple}$
- Race during Session Interaction

  Bad $s(x).P_1 \mid \overline{s}\langle v \rangle.P_2 \mid \overline{s}\langle w \rangle.P_3$

  Good $s(x).P_1 \mid \overline{s}\langle v \rangle.P_2 \mid s'(x).Q_1 \mid \overline{s'}\langle w \rangle.Q_2$
- Communication Mismatch

  Bad $s(x).\overline{s}\langle w \rangle.\mathbf{0} \mid s(y).\overline{s}\langle v \rangle.\mathbf{0}$ Good $s(x).\overline{s}\langle w \rangle.\mathbf{0} \mid \overline{s}\langle v \rangle.s(y).\mathbf{0}$

But the following should be OK since a is a shared channel.

$$!a(x).P \mid \overline{a}(s_1).P_1 \mid \cdots \mid \overline{a}(s_n).P_n$$

Can we *statically* ensure no such errors occur during communications programming!

PROPERTIES

- Communication safety

  No communication mismatch

- Session Fidelity

  The communication sequence in a session follows the scenarios declared by the types

- Progress

  No deadlock/stuck in a session

SUBJECT REDUCTION

$$\Theta;\Gamma \vdash P \triangleright \Delta \qquad\qquad \Theta;\Gamma \vdash P \triangleright \Delta$$

$$\downarrow^{*} \qquad\qquad\qquad\qquad \equiv$$

$$\Theta;\Gamma \vdash P' \triangleright \Delta \qquad\qquad \Theta;\Gamma \vdash P' \triangleright \Delta$$

Subject reduction

Subject congruence

Yields: **Communication Safety** and **Session Fidelity**

PROPERTIEAS

- Subject Reduction
  - session fidelity
  - communication safety
    - error-freedom: no communication mismatch on session channels
    - linearity: session channels are used linearly (exactly once)
- Type Safety
  - a typable program never reduces to an error

SUMMARY

- Session calculus: a conservative extension of pi-calculus
- Session Types: types for session channels
- Key features:
  - Duality: the relationship between the types of opposite endpoints of a session channel
  - Linearity: each channel endpoint occurs exactly once in a collection of a parallel processes
  - A session is a structure sequence of interactions
- Properties: communication safety, session fidelity, progress

READING LIST

- Honda, “Types for Dyadic Interaction”, CONCUR 1993.
- Takeuchi, Honda, Kubo: An Interaction-based Language and its Typing System. PARLE 1994: 398-413
- Honda, Vasconcelos, Kubo:
  Language Primitives and Type Discipline for Structured Communication- Based Programming. ESOP 1998: 122-138
- Yoshida, Vasconcelos: Language Primitives and Type Discipline for Structured Communication-Based Programming Revisited: Two Systems for Higher-Order Session. Communication. Electr. Notes Theor. Comput. Sci. 171(4): 73-93 (2007)

EXERCISE: IS IT WELL-TYPED?

$s(x : \texttt{Bool}).\mathbf{0} \mid \overline{s}\langle \texttt{true}\rangle.\mathbf{0}$	✓
$\overline{s}\langle x\rangle.\mathbf{0} \mid \overline{s}\langle \texttt{true}\rangle.\mathbf{0}$	✗
$\overline{s}\langle \texttt{false}\rangle.\mathbf{0} \mid s(x : \texttt{Bool}).\mathbf{0} \mid s(y : \texttt{Bool})$	✗
$\overline{s}\langle 42\rangle.s'(x : \texttt{Int}).\mathbf{0} \mid \overline{s}\langle 11\rangle.s(y : \texttt{Int})$	✓
$s(x : \texttt{Int}).\overline{s'}\langle 42\rangle.\mathbf{0} \mid \overline{s}\langle 11\rangle.s'(y : \texttt{Int}).\mathbf{0}$	✓
$s \triangleleft \mathsf{ack}.P \mid s \triangleright \{\mathsf{req}_1 : P_1 \;[]\; \cdots \;[]\; \mathsf{req}_n : P_n\}$	✗