

## Linear usages in signal based communication

---

Roberto Amadio

Mehdi Dogguy

Université Paris Diderot  
Laboratoire Preuves, Programmes et Systèmes



---

December 13, 2007

# Plan

- 1  $S\pi$ -calculus
- 2 Typing system 1
- 3 Extended system
- 4 Future work

# 1 - $S\pi$ -calculus

- is a **synchronous**  $\pi$ -calculus,
- no non-deterministic choice operator  $+$ ,
- a dereferenciation operator  $!s$ ,
- inputs depend on the instant,
- outputs are persistent within the instant.

# Examples

$$\mathit{pause}.K = \nu s(\mathbf{s}(x).0, K)$$

## Examples

$$\text{pause}.K = \nu s(s(x).0, K)$$

$$A = \nu s, s', s''(\overline{s}\langle 1 \rangle \mid \overline{s}\langle 2 \rangle \mid s'(x).\overline{s''}\langle 1 \rangle, 0 \mid \text{pause}.\overline{s''}\langle !s \rangle)$$

## Examples

$$\text{pause}.K = \nu s(s(x).0, K)$$

$$A = \nu s, s', s''(\overline{s}\langle 1 \rangle \mid \overline{s}\langle 2 \rangle \mid s'(x).\overline{s''}\langle 1 \rangle, 0 \mid \text{pause}.\overline{s''}\langle !s \rangle)$$

$$\downarrow N$$

$$A = \nu s, s', s''(0 \mid 0 \mid 0 \mid \overline{s''}\langle [1; 2] \rangle)$$

## Examples

$$\text{pause}.K = \nu s(s(x).0, K)$$

$$\begin{array}{c}
 A = \nu s, s', s''(\overline{s}\langle 1 \rangle \mid \overline{s}\langle 2 \rangle \mid s'(x).\overline{s''}\langle 1 \rangle, 0 \mid \text{pause}.\overline{s''}\langle !s \rangle) \\
 \downarrow N \\
 A = \nu s, s', s''(0 \mid 0 \mid 0 \mid \overline{s''}\langle [2; 1] \rangle)
 \end{array}$$

# Within the instant

$$(out) \frac{e \Downarrow v}{\bar{s}\langle e \rangle \xrightarrow{\bar{s}\langle v \rangle} \bar{s}\langle e \rangle}$$

$$(in_{aux}) \frac{}{s(x).P, K \xrightarrow{s?v} [v/x]P}$$

$$(in) \frac{}{P \xrightarrow{sv} (P \mid \bar{s}\langle v \rangle)}$$

$$(comp) \frac{P_1 \xrightarrow{\mu} P'_1 \quad bn(\mu) \cap fn(P_2) = \emptyset}{P_1 \mid P_2 \xrightarrow{\mu} P'_1 \mid P_2}$$

$$(synch) \frac{P_1 \xrightarrow{\nu t \bar{s}\langle v \rangle} P'_1 \quad P_2 \xrightarrow{s?v} P'_2 \quad \{t\} \cap fn(P_2) = \emptyset}{P_1 \mid P_2 \xrightarrow{\tau} \nu t(P'_1 \mid P'_2)}$$

$$(=_{1}^{ind}) \frac{match(v, p) = \theta}{[v \triangleright p]P_1, P_2 \xrightarrow{\tau} \theta P_1}$$

$$(=_{2}^{ind}) \frac{match(v, p) = \uparrow}{[v \triangleright p]P_1, P_2 \xrightarrow{\tau} P_2}$$

# At the end of the instant

$$(0) \quad \frac{}{0 \xrightarrow{\emptyset, V} 0}$$

$$(\text{reset}) \quad \frac{e \Downarrow v \quad v \in V(s)}{\bar{s} \langle e \rangle \xrightarrow{[\{v\}/s], V} 0}$$

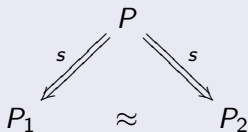
$$(\text{cont}) \quad \frac{s \notin \text{dom}(V)}{s(x).P, K \xrightarrow{\emptyset, V} V(K)}$$

$$(\text{par}) \quad \frac{P_i \xrightarrow{E_i, V} P'_i \quad i = 1, 2}{(P_1 \mid P_2) \xrightarrow{E_1 \vee E_2, V} (P'_1 \mid P'_2)}$$

$$(\text{next}) \quad \frac{P \leq \nu s P' \quad P' \xrightarrow{E, V} P'' \quad V \Vdash E}{P \xrightarrow{N} \nu s P''}$$

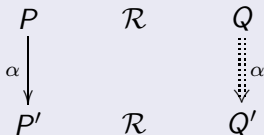
## Definition

$P$  is deterministic if for every sequence of actions  $s$  :



## Definition (Bisimulation Standard)

A symmetric relation  $\mathcal{R}$  on programs is a labelled bisimulation if  $bn(\alpha) \cap fn(Q) = \emptyset$  and :



We denote with  $\approx$  the largest labelled bisimulation.

## 2 - Typing system 1

- At least two different values are emitted on the same signal, at the same instant :

$$\bar{s}\langle v \rangle \mid \bar{s}\langle v' \rangle \mid s(x).P, K$$

- At the end of the instant, at least two different values are available :

$$\bar{s}\langle v \rangle \mid \bar{s}\langle v' \rangle \mid \text{pause}.A(!s)$$

$$s : \text{Sig} \binom{n}{m}{p}^\omega(\sigma)$$

$$\text{où : } \begin{cases} n & \text{max of outputs} \\ m & \text{max of inputs} \\ p & \text{max of inputs at the end of the instant} \end{cases}$$

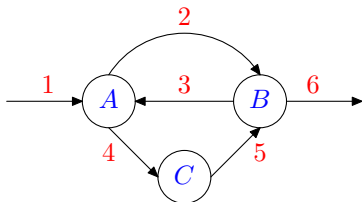
$$n, m, p \in L = \{0 < 1 < \infty\}$$

$$\text{Lifting operation : } \uparrow (u.v^\omega) = v^\omega$$

# Synchronous Data Flow

$$\nu \mathbf{s}_2, \mathbf{s}_3, \mathbf{s}_4, \mathbf{s}_5 \ (A(\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3, \mathbf{s}_4) \mid B(\mathbf{s}_2, \mathbf{s}_3, \mathbf{s}_5, \mathbf{s}_6) \mid C(\mathbf{s}_4, \mathbf{s}_5))$$

$$\begin{aligned} A(\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3, \mathbf{s}_4) &= \mathbf{s}_1(x).(\overline{\mathbf{s}_2}\langle f(x) \rangle \mid \mathbf{s}_3(y).(\overline{\mathbf{s}_4}\langle g(y) \rangle \mid \mathit{pause}.A(\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3, \mathbf{s}_4)), 0), 0 \\ B(\mathbf{s}_2, \mathbf{s}_3, \mathbf{s}_5, \mathbf{s}_6) &= \mathbf{s}_2(x).(\overline{\mathbf{s}_3}\langle i(x) \rangle \mid \mathbf{s}_5(y).(\overline{\mathbf{s}_6}\langle l(y) \rangle \mid \mathit{pause}.B(\mathbf{s}_2, \mathbf{s}_3, \mathbf{s}_5, \mathbf{s}_6)), 0), 0 \\ C(\mathbf{s}_4, \mathbf{s}_5) &= \mathbf{s}_4(x).(\overline{\mathbf{s}_5}\langle h(x) \rangle \mid \mathit{pause}.C(\mathbf{s}_4, \mathbf{s}_5)), 0 \end{aligned}$$



# Cell

$$\begin{aligned} \text{Cell}(q, s, l) &= \text{Send}(q, s, l, l) \\ \text{Send}(q, s, l, l') &= [l' \triangleright \text{cons}(s', l'')] \\ &\quad (\overline{s'} \langle q \rangle \mid \text{Send}(q, s, l, l'')), \\ &\quad \text{pause.Cell}(\text{next}(q, !s), s, l) \end{aligned}$$

$$\begin{matrix} [e] \\ \left( \begin{matrix} \infty \\ 0 \\ \infty \end{matrix} \right) \end{matrix}$$

$$\begin{matrix} [o_1] \\ \left( \begin{matrix} 1 \\ \infty \\ \infty \end{matrix} \right) \end{matrix}$$

⋮

$$\begin{matrix} [o_0] \\ \left( \begin{matrix} 0 \\ \infty \\ \infty \end{matrix} \right) \end{matrix}$$

$$\sigma ::= 1 \mid List(\sigma) \mid Set(\sigma) \mid Sig_u(\sigma)$$
$$u \in \{e^\omega, o_0^\omega\}$$

$$\rho ::= \sigma \mid Sig_u(\sigma)$$
$$u \in \{e^\omega, o_1^\omega, o_0^\omega, o_1 \cdot o_0^\omega, o_0 \cdot o_1^\omega\}$$

# Addition

$$\begin{array}{rcl}
 0 & \oplus & x = x \\
 1 & \oplus & 1 = \infty \\
 \infty & \oplus & x = \infty
 \end{array}$$

$$\begin{array}{rcl}
 \sigma & \oplus & \sigma = \sigma \\
 \text{Sig}_u(\sigma) & \oplus & \text{Sig}_{u'}(\sigma) = \text{Sig}_{u \oplus u'}(\sigma) \\
 \rho & \oplus & \rho' = \uparrow
 \end{array}$$

$$\sigma \text{ is neutral} \Leftrightarrow \forall \rho, \sigma \oplus \rho = \rho$$

## ... for expressions

$$(var) \frac{\Gamma \text{ neutral}}{\Gamma, x : \rho \vdash x : \rho}$$

$$(k) \frac{\Gamma \vdash r_i : \sigma_i \quad i = 1, \dots, n \quad \Gamma' \text{ neutral} \quad k : (\sigma_1, \dots, \sigma_n) \rightarrow \sigma \quad k = f \text{ or } k = c}{\Gamma' \oplus \Gamma \vdash k(r_1, \dots, r_n) : \sigma}$$

$$(!_{Set}) \frac{\Gamma \text{ neutral} \quad u = (\infty, 0, \infty)^\omega}{\Gamma, s : Sig_u(\sigma) \vdash !s : Set(\sigma)}$$

$$(!_{List}) \frac{\Gamma \text{ neutral} \quad u = (0, \infty, \infty)^\omega}{\Gamma, s : Sig_u(\sigma) \vdash !s : List(\sigma)}$$

## ... for programs

$$(0) \frac{\Gamma \text{ neutral}}{\Gamma \vdash 0}$$

$$(out) \frac{u \in \{e^\omega, o_1.o_0^\omega\} \quad \Gamma \vdash e : \sigma}{\Gamma, s : \text{Sig}_u(\sigma) \vdash \bar{s}\langle e \rangle}$$

$$(\nu) \frac{u \in \{e^\omega, o_1^\omega\} \quad \Gamma, s : \text{Sig}_u(\sigma) \vdash P}{\Gamma \vdash \nu s : \text{Sig}_u(\sigma) P}$$

$$(in) \frac{u \in \{o_0^\omega, o_0.o_1^\omega\} \quad \Gamma \vdash s : \text{Sig}_u(\sigma) \quad \Gamma, x : \sigma \vdash P \quad \uparrow(\Gamma) \vdash K}{\Gamma \vdash s(x).P, K}$$

$$(par) \frac{\Gamma_i \vdash P_i \quad i = 1, 2}{\Gamma_1 \oplus \Gamma_2 \vdash P_1 | P_2}$$

$$(rec) \frac{A : (\rho_1, \dots, \rho_n) \quad \Gamma_i \vdash r_i : \rho_i \quad i = 1, \dots, n}{\Gamma_1 \oplus \dots \oplus \Gamma_n \vdash A(r_1, \dots, r_n)}$$

$$(m_s) \frac{\Gamma \vdash P_i \quad i = 1, 2}{\Gamma \vdash [s_1 = s_2]P_1, P_2}$$

$$(m_c) \frac{c : (\sigma_1, \dots, \sigma_n) \rightarrow \sigma \quad \Gamma' \vdash u : \sigma \quad \Gamma \oplus \Gamma' = \Gamma \quad \Gamma, x_1 : \sigma_1, \dots, x_n : \sigma_n \vdash P_1 \quad \Gamma \vdash P_2}{\Gamma \vdash [u \triangleright c(x_1, \dots, x_n)]P_1, P_2}$$

## Proposition (Subject Reduction)

Suppose  $\Gamma \vdash P$ . Then :

- (1) If  $P \xrightarrow{sv} P'$ ,  $\Gamma' \vdash \bar{s}\langle v \rangle$  and  $\Gamma \oplus \Gamma' \downarrow$ , then  $\Gamma \oplus \Gamma' \vdash P'$ .
- (2) If  $P \xrightarrow{\nu t: \rho \bar{s}\langle v \rangle} P'$  then  $\Gamma, t: \rho \vdash P'$ .
- (3) If  $P \xrightarrow{\tau} P'$  then  $\Gamma \vdash P'$ .
- (4) If  $P \xrightarrow{N} P'$  then  $\uparrow(\Gamma) \vdash P'$ .

## Theorem (Typing implies determinacy)

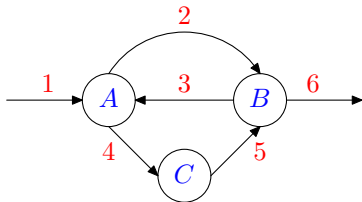
*If the program  $P$  is typable in a context  $\Gamma$ , then  $P$  is determinate.*

# Synchronous Data Flow

$$\nu s_2, s_3, s_4, s_5 \ (A(s_1, s_2, s_3, s_4) \mid B(s_2, s_3, s_5, s_6) \mid C(s_4, s_5))$$

$$I = \text{Sig}_{o_0^\omega}(\text{Data}) \quad O = \text{Sig}_{o_1^\omega}(\text{Data})$$

$$A : (I, O, I, O) \quad B : (I, O, I, O) \quad C : (I, O)$$

$$s_1 : I \quad s_6 : O$$


## Cell

$$\begin{aligned} \text{Cell}(q, s, l) &= \text{Send}(q, s, l, l) \\ \text{Send}(q, s, l, l') &= [l' \triangleright \text{cons}(s', l'')] \\ &\quad (\overline{s'} \langle q \rangle \mid \text{Send}(q, s, l, l'')), \\ &\quad \text{pause.Cell}(\text{next}(q, !s), s, l) \end{aligned}$$

$$S_1 = \text{Sig}_{e^\omega}(\sigma) \quad L_1 = \text{List}(S_1)$$

$$\text{Cell} : (\text{State}, S_1, L_1) \quad \text{Send} : (\text{State}, S_1, L_1, L_1)$$

$$\text{next} : (\text{State}, \text{Set}(\text{State})) \rightarrow \text{State}$$

# Client-Server


$$\begin{aligned} \text{Server}(s) &= \text{pause.Handle}(s, !s) \\ \text{Handle}(s, l) &= [l \triangleright \text{cons}(\text{req}(s', x), l')] \\ &\quad (\overline{s'} \langle f(x) \rangle \mid \text{Handle}(s, l')), \\ &\quad \text{Server}(s) \end{aligned}$$

$$\begin{aligned} \text{Client}(x, s, t) &= \nu s' ( \\ &\quad \overline{s} \langle \text{req}(s', x) \rangle \\ &\quad \mid \text{pause}.s'(x).\overline{t} \langle x \rangle, 0 \\ &\quad ) \end{aligned}$$

# Client-Server

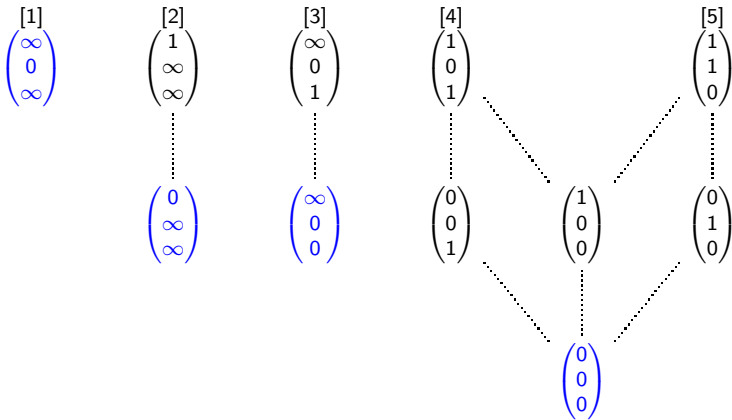
$$\begin{aligned} \text{Server}(s) &= \text{pause.Handle}(s, !s) \\ \text{Handle}(s, l) &= [l \triangleright \text{cons}(\text{req}(s', x), l')] \\ &\quad (\overline{s'} \langle f(x) \rangle \mid \text{Handle}(s, l')), \\ &\quad \text{Server}(s) \end{aligned}$$

$$\begin{aligned} \text{Client}(x, s, t) &= \nu s' ( \\ &\quad \overline{s} \langle \text{req}(s', x) \rangle \\ &\quad \mid \text{pause}.s'(x).\overline{t} \langle x \rangle, 0 \\ &\quad ) \end{aligned}$$

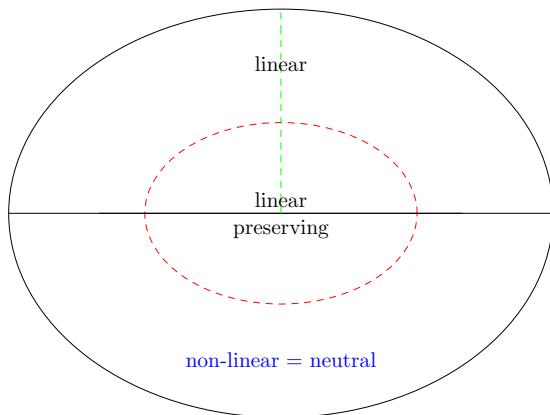
 more usages !

## 3 - Extended system

## Signals' classes



# Classification



$$\kappa ::= 1 \mid List_{\infty}(\kappa) \mid Set_{\infty}(\kappa) \mid Sig_u(\kappa)$$

u is neutral

$$\lambda ::= List_1(\sigma) \mid Set_1(\sigma) \mid Sig_u(\kappa) \mid Sig_v(\lambda)$$

u is uniform and linear

$$\sigma ::= \kappa \mid \lambda$$

v is uniform and linear preserving

$$\rho ::= \sigma \mid Sig_u(\kappa) \mid Sig_v(\lambda)$$

u is neutral and v is linear

# Addition

$$\begin{aligned}
 0 &\oplus x = x \\
 1 &\oplus 1 = \uparrow \\
 \infty &\oplus x = \infty
 \end{aligned}$$

$$u \oplus u' \text{ is defined} \iff [u, u' \in \text{Class}(n) \wedge u \oplus u' \in \text{Class}(n)]$$

$$\begin{array}{lll}
 \kappa & \oplus & \kappa = \kappa \\
 Op_{\infty}(\kappa_1, \dots, \kappa_n) & \oplus & Op_{\infty}(\kappa_1, \dots, \kappa_n) = Op_{\infty}(\kappa_1, \dots, \kappa_n) \\
 Sig_u(\sigma) & \oplus & Sig_v(\sigma) = Sig_{u \oplus v}(\sigma) \\
 \rho & \oplus & \rho' = \uparrow
 \end{array}$$

## ... for expressions

$$(var) \frac{\Gamma \text{ neutral}}{\Gamma, x : \rho \vdash x : \rho}$$

$$(!_{Set_i}) \frac{\Gamma \text{ neutral} \quad \begin{array}{l} i = \pi_3(u[0]) \\ u \in U_i(2,3) \end{array}}{\Gamma, s : Sig_u(\sigma) \vdash !s : Set_i(\sigma)}$$

$$(k) \frac{\begin{array}{l} \Gamma_i \vdash r_i : \sigma_i \quad i = 1, \dots, n \quad \Gamma_0 \text{ neutral} \\ k : (\sigma_1, \dots, \sigma_n) \rightarrow \sigma \quad k = f \text{ or } k = c \end{array}}{\Gamma_0 \oplus \Gamma_1 \oplus \dots \oplus \Gamma_n \vdash k(r_1, \dots, r_n) : \sigma}$$

$$(!_{List_i}) \frac{\Gamma \text{ neutral} \quad \begin{array}{l} i = \pi_3(u[0]) \\ u \in U_i(2,4) \end{array}}{\Gamma, s : Sig_u(\sigma) \vdash !s : List_i(\sigma)}$$

## ... for programs

$$(0) \frac{\Gamma \text{ neutral}}{\Gamma \vdash 0}$$

$$(\nu) \frac{\Gamma, s : \text{Sig}_u(\sigma) \vdash P}{\Gamma \vdash \nu s P}$$

$$(\text{par}) \frac{\Gamma_i \vdash P_i \quad i = 1, 2}{\Gamma_1 \oplus \Gamma_2 \vdash P_1 \mid P_2}$$

$$(m_s) \frac{\Gamma \vdash P_i \quad i = 1, 2}{\Gamma \vdash [s_1 = s_2] P_1, P_2}$$

$$(\text{out}) \frac{u \in U_{\text{out}} \quad \Gamma_1 \vdash s : \text{Sig}_u(\sigma) \quad \Gamma_2 \vdash e : \sigma}{\Gamma_1 \oplus \Gamma_2 \vdash \bar{s} \langle e \rangle}$$

$$(\text{in}) \frac{u \in U_{\text{in}} \quad \Gamma_1 \vdash s : \text{Sig}_u(\sigma) \quad \Gamma_2, x : \sigma \vdash P \quad \uparrow (\Gamma_1 \oplus \Gamma_2) \vdash K}{\Gamma_1 \oplus \Gamma_2 \vdash \mathbf{s}(x).P, K}$$

$$(\text{rec}) \frac{A : (\rho_1, \dots, \rho_n) \quad \Gamma_i \vdash r_i : \rho_i \quad i = 1, \dots, n}{\Gamma_1 \oplus \dots \oplus \Gamma_n \vdash A(r_1, \dots, r_n)}$$

$$(m_c) \frac{c : (\sigma_1, \dots, \sigma_n) \rightarrow \sigma \quad \Gamma' \vdash u : \sigma \quad \Gamma, x_1 : \sigma_1, \dots, x_n : \sigma_n \vdash P_1 \quad \Gamma \oplus \Gamma' \vdash P_2}{\Gamma \oplus \Gamma' \vdash [\mathbf{u} \sqsupseteq c(x_1, \dots, x_n)] P_1, P_2}$$

## Client-Server

$$\begin{aligned} \text{Server}(s) &= \text{pause.Handle}(s, !s) \\ \text{Handle}(s, l) &= [l \triangleright \text{cons}(\text{req}(s', x), l')] \\ &\quad (\overline{s'} \langle f(x) \rangle \mid \text{Handle}(s, l')), \\ &\quad \text{Server}(s) \end{aligned}$$

$$\begin{aligned} \text{Client}(x, s, t) &= \nu s' ( \\ &\quad \overline{s} \langle \text{req}(s', x) \rangle \\ &\quad \mid \text{pause}.s'(x).\overline{t} \langle x \rangle, 0 \\ &\quad ) \end{aligned}$$

$$s : \text{Sig} \begin{pmatrix} \infty \\ 0 \\ 1 \end{pmatrix}^{\omega}(\text{Req}) \quad l : \text{Set}_1(\text{Req}) \quad s' : \text{Sig} \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}^{\omega}(\text{Req})$$

## Client-Server

$$\begin{aligned} \text{Server}(s) &= \text{pause.Handle}(s, !s) \\ \text{Handle}(s, l) &= [l \triangleright \text{cons}(\text{req}(s', x), l')] \\ &\quad (\overline{s'} \langle f(x) \rangle \mid \text{Handle}(s, l')), \\ &\quad \text{Server}(s) \end{aligned}$$

$$\begin{aligned} \text{Client}(x, s, t) &= \nu s' ( \\ &\quad \overline{s} \langle \text{req}(s', x) \rangle \\ &\quad \mid \text{pause}.s'(x).\overline{t} \langle x \rangle, 0 \\ &\quad ) \end{aligned}$$

 It's not typable!

# Client-Server

$$\begin{aligned} \text{Server}(s) &= \text{pause.Handle}(s, !s) \\ \text{Handle}(s, l) &= [l \triangleright \text{cons}(\text{req}(s', x), l')] \\ &\quad (\overline{s'} \langle f(x) \rangle \mid \text{Handle}(s, l')), \\ &\quad \text{Server}(s) \end{aligned}$$

$$\begin{aligned} \text{Client}(x, s, t) &= \nu s' ( \\ &\quad \overline{s} \langle \text{req}(s', x) \rangle \\ &\quad \mid \text{pause}.s'(x).\overline{t} \langle x \rangle, 0 \\ &\quad ) \end{aligned}$$

Solution? separate !s from the rest.

# 4 - Future work

- 1 Client-Server example,
- 2 subject reduction lemma.

Thank you ...