Typing migration-control in lsdπ

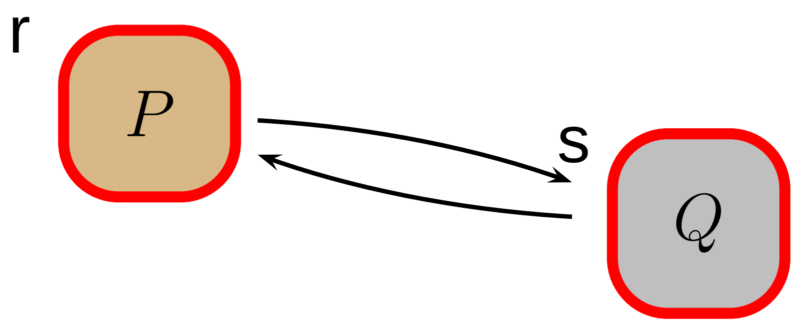
António Ravara

Dep. Mathematics, IST - Technical Univ. of Lisbon

Joint work with

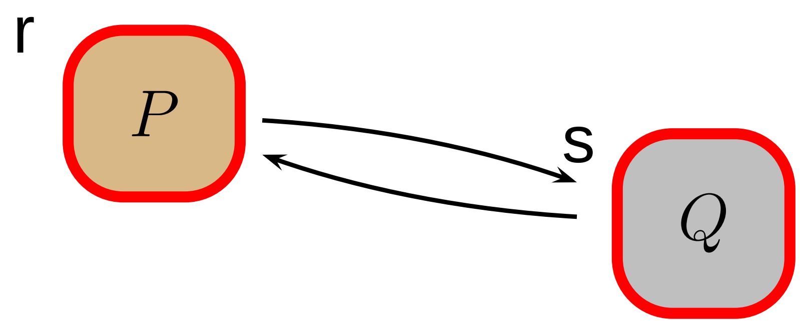
Francisco Martins, DI/FCUL

Aim of this work



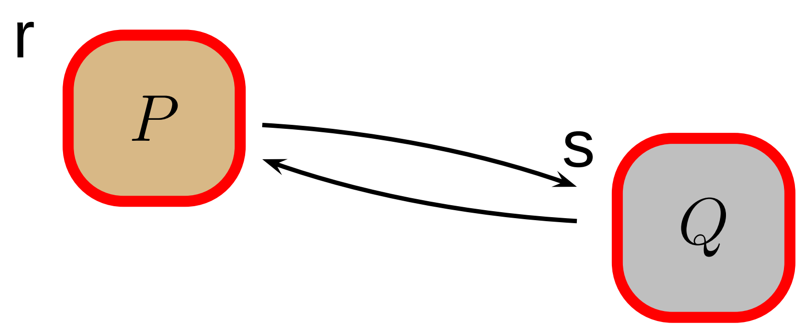
- Control code migration via static typing

Aim of this work



- Control code migration via static typing
- Ensuring security policies: sites allowed to
  - send messages (remote communicating)
  - migrate processes
  - create remote names

Aim of this work



- Control code migration via static typing
- Ensuring security policies: sites allowed to
  - send messages (remote communicating)
  - migrate processes
  - create remote names
- At runtime, no process in a well-typed network violates a security policy.

Programme of this talk

- Define a setting to study code migration
  a (lexically scoped) distributed π-calculus
- Show by example what we want to guarantee
- Discuss some rules
- Present the results

Framework: Lexically Scoped Distributed π

- Extends π, distributing processes over networks of named sites where they compute.
- Processes allowed to:
  - communicate via channels (as in π), but only locally
  - migrate from site to site.

Main concepts

In $lsd\pi$

- channels are
  - resources associated *uniquely* to sites
  - located at *creation* time
    - In $s\textbf{[}(\boldsymbol{\nu}\, c)\ \ldots\textbf{]}$, channel c is created *locally*, at s;
    - In $r\textbf{[}(\boldsymbol{\nu}\, c@s)\ \ldots\textbf{]}$, channel c is created *remotely*, to be located at s;

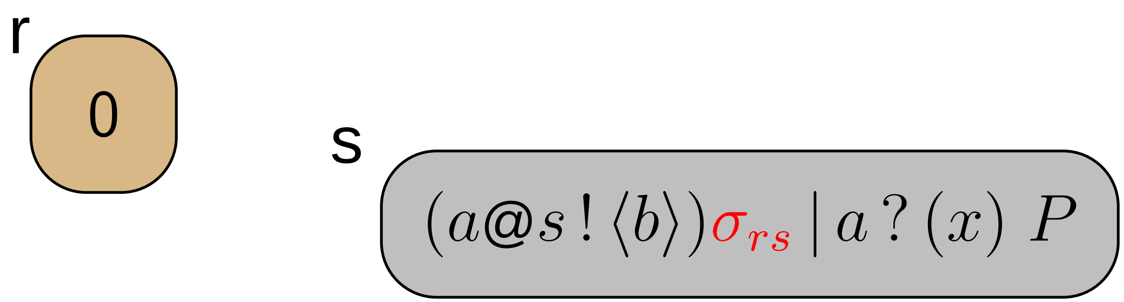
Main concepts

In $lsd\pi$

- channels are
  - resources associated *uniquely* to sites
  - located at *creation* time
    - In $s\mathbf{[}(\boldsymbol{\nu}\, c)\ \ldots\mathbf{]}$, channel c is created *locally*, at s;
    - In $r\mathbf{[}(\boldsymbol{\nu}\, c@s)\ \ldots\mathbf{]}$, channel c is created *remotely*, to be located at s;
- sites are
  - *collections* of channels;
  - *shells* of local computations.

$$r[\![a@s\,!\,\langle b\rangle]\!] \parallel s[\![a\,?\,(x)\;P]\!]$$

$lsd\pi$ ***by example***

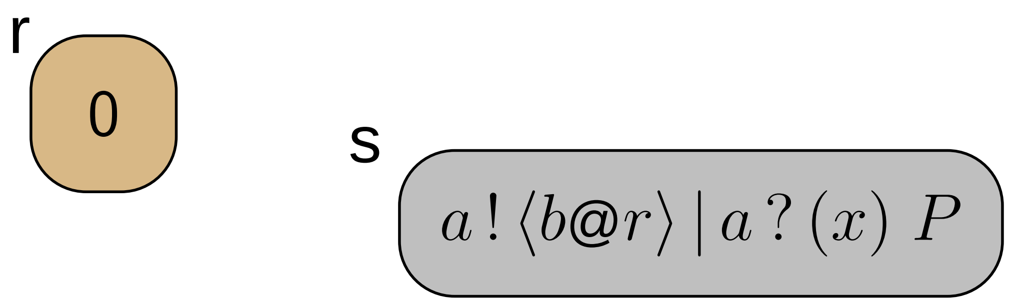


$$r[\![a@s\,!\,\langle b\rangle]\!] \parallel s[\![a\,?\,(x)\ P]\!]$$

$$\downarrow$$

$$r[\![0]\!] \parallel s[\![(a@s\,!\,\langle b\rangle)\sigma_{rs} \mid a\,?\,(x)\ P]\!]$$

$lsd\pi$ ***by example***



$$r[\![a@s\,!\,\langle b\rangle]\!] \parallel s[\![a\,?\,(x)\; P]\!]$$

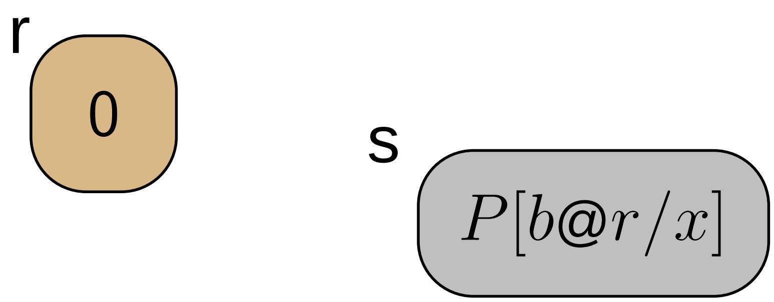
$$\downarrow$$

$$r[\![0]\!] \parallel s[\![(a@s\,!\,\langle b\rangle)\sigma_{rs} \mid a\,?\,(x)\; P]\!]$$

$$\parallel$$

$$r[\![0]\!] \parallel s[\![a\,!\,\langle b@r\rangle \mid a\,?\,(x)\; P]\!]$$

$lsd\pi$ ***by example***



$$r\mathbf{[}a@s\,!\,\langle b\rangle\mathbf{]} \parallel s\mathbf{[}a\,?\,(x)\ P\mathbf{]}$$
$$\downarrow$$
$$r\mathbf{[}0\mathbf{]} \parallel s\mathbf{[}(a@s\,!\,\langle b\rangle)\sigma_{rs} \mid a\,?\,(x)\ P\mathbf{]}$$
$$\|$$
$$r\mathbf{[}0\mathbf{]} \parallel s\mathbf{[}a\,!\,\langle b@r\rangle \mid a\,?\,(x)\ P\mathbf{]}$$
$$\downarrow$$
$$r\mathbf{[}0\mathbf{]} \parallel s\mathbf{[}P[b@r/x]\mathbf{]}$$

lsdπ motto

- Lexically Scoping Distribution:

what you see is what you get!

$\therefore$ simple names are local;
remote names are explicitly located;

lsdπ motto

- Lexically Scoping Distribution:

what you see is what you get!

$\therefore$ simple names are local;
remote names are explicitly located;

- security policies: a flavour ...

$$(\boldsymbol{\nu}\, a@s)\ (r_{G_1}[a@s\,!\,\langle b\rangle] \parallel s_{G_2}[a\,?\,(x\ :\ S)\ P])$$

Syntax (lsdπ)

- Processes

$$P, Q ::= \mathsf{0} \mid u!\langle v \rangle \mid u?(x : S)\, P \mid P \,|\, Q \mid (\boldsymbol{\nu}\, u)\, P$$

simple channels	a, b, c, x, y	channels	u, v	$::= a \mid a@s$
sites	r, s, t	set of sites	R, S	

Syntax (lsdπ)

- Processes

$$P, Q ::= \ 0 \ \mid \ u!\langle v\rangle \ \mid \ u?(x : S)\ P \ \mid \ P \mid Q \ \mid \ (\boldsymbol{\nu}\, u)\ P$$

- Networks

$$N, M ::= 0 \ \mid \ s_G[\![P]\!] \ \mid \ N \parallel M \ \mid \ (\boldsymbol{\nu}\, a@s)\ N$$

Sites are constants and may not be passed around

simple channels	a, b, c, x, y	channels	u, v	$::= a \mid a@s$
sites	r, s, t	set of sites	R, S	
$G?$	stay tunned ...			

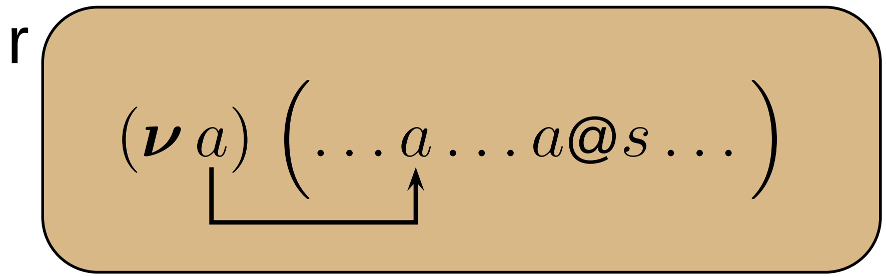
Syntax (types)

Γ	$::=$	$\{s_1 : (\varphi_1, G_1), \ldots, s_n : (\varphi_n, G_n)\}$	typings
φ	$::=$	$\{a_1 : \gamma_1, \ldots, a_n : \gamma_n\}$	site types
G	$::=$	$\{\mathrm{rem} : S_1, \mathrm{mig} : S_2, \mathrm{new} : S_3\}$	site policies

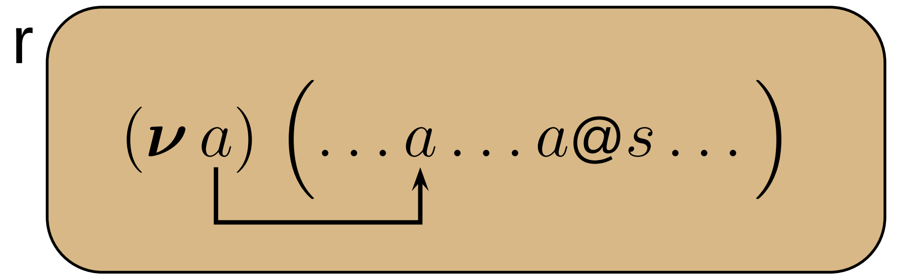
Syntax (types)

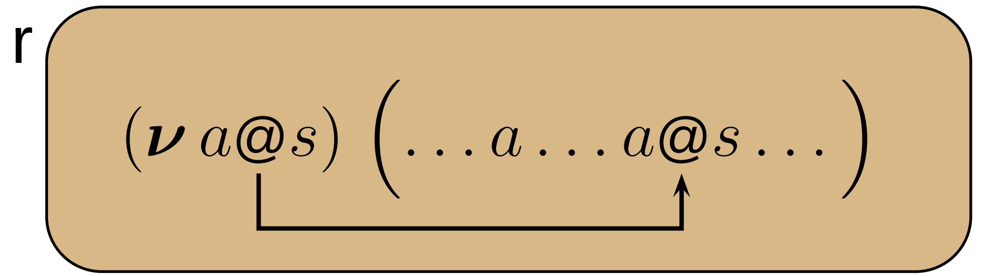
Γ	$::=$	$\{s_1 : (\varphi_1, G_1), \ldots, s_n : (\varphi_n, G_n)\}$	typings
φ	$::=$	$\{a_1 : \gamma_1, \ldots, a_n : \gamma_n\}$	site types
G	$::=$	$\{\mathrm{rem} : S_1, \mathrm{mig} : S_2, \mathrm{new} : S_3\}$	site policies
γ	$::=$	$\mathrm{ch}(\gamma)@S^t \mid \mathrm{val}$	channel types
t	$::=$	$o \mid i \mid b$	site tags

Free names

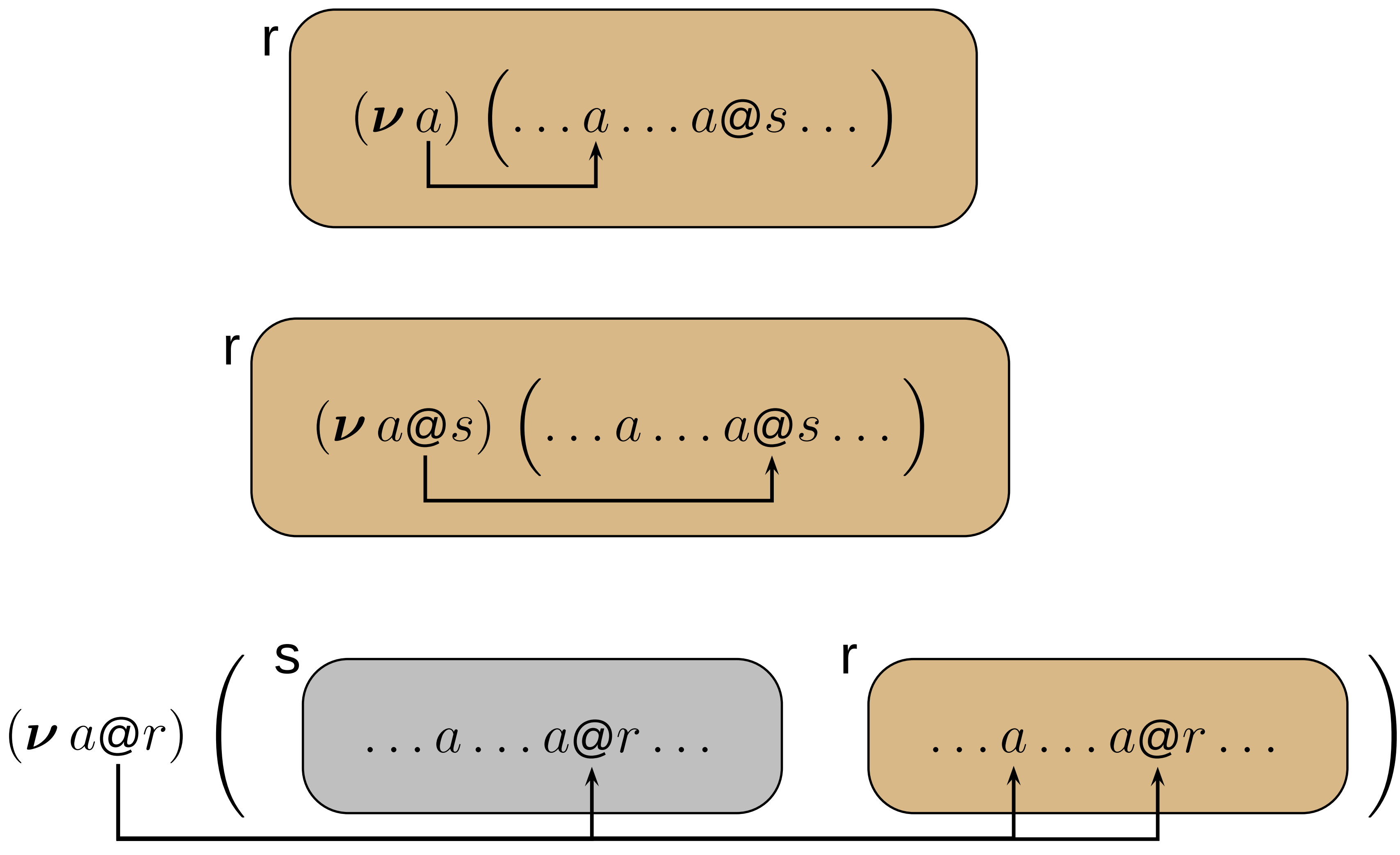


Free names


r
$(\boldsymbol{\nu}\, a)\left(\ldots a \ldots a@s \ldots\right)$


r
$(\boldsymbol{\nu}\, a@s)\left(\ldots a \ldots a@s \ldots\right)$

Free names


r
$(\boldsymbol{\nu}\, a) \Big(\ldots a \ldots a@s \ldots \Big)$
r
$(\boldsymbol{\nu}\, a@s) \Big(\ldots a \ldots a@s \ldots \Big)$
s
r
$(\boldsymbol{\nu}\, a@r) \Big(\ldots a \ldots a@r \ldots \quad \ldots a \ldots a@r \ldots \Big)$

Free names

$$s_G\mathbf{[}(\boldsymbol{\nu}\, a)\ a\,!\,\langle b\rangle \mid a@r\,?\,(x\ :\ S)\ P\mathbf{]}$$

Free names

$$s_G[\!\![(\boldsymbol{\nu}\, a)\ a\,!\,\langle b\rangle \mid a@r\,?\,(x : S)\ P]\!\!]$$

$$s_G[\!\![(\boldsymbol{\nu}\, a@r)\ a\,!\,\langle b\rangle \mid a@r\,?\,(x : S)\ P]\!\!]$$

Free names

$$s_G[\![(\boldsymbol{\nu}\, a)\; a\,!\,\langle b\rangle \mid a@r\,?\,(x : S)\; P]\!]$$

$$s_G[\![(\boldsymbol{\nu}\, a@r)\; a\,!\,\langle b\rangle \mid a@r\,?\,(x : S)\; P]\!]$$

$$(\boldsymbol{\nu}\, a@s)\; r_{G_1}[\![a\,?\,(x : S_1)\; P]\!] \parallel s_{G_2}[\![a\,?\,(y : S_2)\; Q]\!]$$

Free names

$$s_G[\![(\boldsymbol{\nu}\, a)\ a\,!\,\langle b\rangle \mid a@r\,?\,(x : S)\ P]\!]$$

$$s_G[\![(\boldsymbol{\nu}\, a@r)\ a\,!\,\langle b\rangle \mid a@r\,?\,(x : S)\ P]\!]$$

$$(\boldsymbol{\nu}\, a@s)\ r_{G_1}[\![a\,?\,(x : S_1)\ P]\!] \parallel s_{G_2}[\![a\,?\,(y : S_2)\ Q]\!]$$

$$s_G[\![a\,?\,(x : S)\ x@r\,!\,\langle b\rangle \mid x\,!\,\langle b\rangle]\!]$$

Structural congruence (some rules)

$$s_G[\![P \mid Q]\!] \equiv s_G[\![P]\!] \parallel s_G[\![Q]\!]$$

Structural congruence (some rules)

$$s_G[\![P \mid Q]\!] \equiv s_G[\![P]\!] \parallel s_G[\![Q]\!]$$

$$(\boldsymbol{\nu}\, a@r)\ s_G[\![P]\!] \equiv s_G[\![(\boldsymbol{\nu}\, a@r)\ P]\!] \qquad r \neq s$$

Structural congruence (some rules)

$$s_G[\![P \mid Q]\!] \equiv s_G[\![P]\!] \parallel s_G[\![Q]\!]$$

$$(\boldsymbol{\nu}\, a@r)\ s_G[\![P]\!] \equiv s_G[\![(\boldsymbol{\nu}\, a@r)\ P]\!] \qquad r \neq s$$

$$(\boldsymbol{\nu}\, a@s)\ s_G[\![P]\!] \equiv s_G[\![(\boldsymbol{\nu}\, a@s)\ P]\!] \qquad a \notin \mathrm{fn}(P)$$

- Example:

$$(\boldsymbol{\nu}\, a@s)\ s_G[\![a\,!\langle b\rangle \mid a@s\,!\langle c\rangle]\!] \not\equiv s_G[\![(\boldsymbol{\nu}\, a@s)\ a\,!\langle b\rangle \mid a@s\,!\langle c\rangle]\!]$$

Structural congruence (some rules)

$$s_G[P \mid Q] \equiv s_G[P] \parallel s_G[Q]$$

$$(\boldsymbol{\nu}\, a@r)\ s_G[P] \equiv s_G[(\boldsymbol{\nu}\, a@r)\ P] \qquad r \neq s$$

$$(\boldsymbol{\nu}\, a@s)\ s_G[P] \equiv s_G[(\boldsymbol{\nu}\, a@s)\ P] \qquad a \notin \mathrm{fn}(P)$$

$$(\boldsymbol{\nu}\, a@s)\ s_G[P] \equiv s_G[(\boldsymbol{\nu}\, a)\ P] \qquad a@s \notin \mathrm{fn}(P)$$

- Example:

$$(\boldsymbol{\nu}\, a@s)\ s_G[a\,!\,\langle b\rangle \mid a@s\,!\,\langle c\rangle] \not\equiv s_G[(\boldsymbol{\nu}\, a)\ a\,!\,\langle b\rangle \mid a@s\,!\,\langle c\rangle]$$

Reduction rules (some rules)

Local communication

$$s_G\mathbf{[}a\,!\,\langle v\rangle \mid a\,?\,(x\ :\ S)\ P\mathbf{]} \rightarrow P[v/x]$$

Reduction rules (some rules)

Local communication

$$s_G[\![a\,!\,\langle v\rangle \mid a\,?\,(x : S)\ P]\!] \to P[v/x]$$

Migration: σ_{rs} translates free names from r to s

$$\begin{aligned} & s_{G_1}[\![P]\!] \parallel r_{G_2}[\![a@s\,?\,(x : S)\ b\,!\,\langle x\rangle]\!] \\ & \quad \to s_{G_1}[\![P \mid (a@s\,?\,(x : S)\ b\,!\,\langle x\rangle)\sigma_{rs}]\!] \parallel r_{G_2}[\![\mathbf{0}]\!], \quad r \neq s \\ & \quad = s_{G_1}[\![P \mid a\,?\,(x : S)\ b@r\,!\,\langle x\rangle]\!] \parallel r_{G_2}[\![\mathbf{0}]\!], \quad r \neq s \end{aligned}$$

$$\begin{array}{ll} \sigma_{rs}(s) = s & \sigma_{rs}(a@s) = a \\ \sigma_{rs}(a) = a@r & \sigma_{rs}(a@t) = a@t \quad t \notin \{r, s\} \end{array}$$

Security policies violation

Let $r \neq t$

- Remote communication

$$s_{\{\mathrm{rem}:\{t\}\}}[\![P]\!] \parallel r_{G_1}[\![a@s\,!\,\langle x\rangle]\!]$$

Security policies violation

Let $r \neq t$

- Remote communication

$$s_{\{\mathrm{rem}:\{t\}\}}[\![P]\!] \parallel r_{G_1}[\![a@s\,!\,\langle x \rangle]\!]$$

$$s_{\{\mathrm{rem}:\{t\}\}}[\![b@r\,?\,(x:S)\; a@s\,!\,\langle x \rangle]\!] \parallel r_{\{\mathrm{mig}:\{s\}\}}[\![0]\!]$$

Security policies violation

Let $r \neq t$

- Remote communication

$$s_{\{\mathrm{rem}:\{t\}\}}[\![P]\!] \parallel r_{G_1}[\![a@s\,!\,\langle x\rangle]\!]$$

$$s_{\{\mathrm{rem}:\{t\}\}}[\![b@r\,?\,(x\ :\ S)\ a@s\,!\,\langle x\rangle]\!] \parallel r_{\{\mathrm{mig}:\{s\}\}}[\![0]\!]$$

$$s_{\{\mathrm{rem}:\{r\}\}}[\![a\,?\,(x\ :\ \{t\})\ 0]\!] \parallel r_{G_1}[\![a@s\,!\,\langle b@r\rangle]\!]$$

Security policies violation

Let $r \neq t$

- Remote communication
- Migration

$$s_{\{\mathrm{mig}:\{t\}\}}[\![P]\!] \parallel r_{G_1}[\![a@s\,?\,(x:S)\;Q]\!]$$

Security policies violation

Let $r \neq t$

- Remote communication
- Migration

$$s_{\{\mathrm{mig}:\{t\}\}}[\![P]\!] \parallel r_{G_1}[\![a@s\,?\,(x : S)\; Q]\!]$$

$$s_{\{\mathrm{rem}:\{r\}\}}[\![a\,?\,(x : \{r, t\})\; x\,!\,\langle c\rangle]\!] \parallel r_{\emptyset}[\![a@s\,!\,\langle b\rangle]\!]$$

Security policies violation

Let $r \neq t$

- Remote communication
- Migration
- Name creation

$$s_{\{\mathrm{new}:\{t\}\}}[\![P]\!] \parallel r_{G_1}[\![(\boldsymbol{\nu}\; a@s)\; Q]\!]$$

Security policies violation

Let $r \neq t$

- Remote communication
- Migration
- Name creation

$$s_{\{\mathrm{new}:\{t\}\}}[\![P]\!] \parallel r_{G_1}[\![(\boldsymbol{\nu}\, a@s)\ Q]\!]$$

$$(\boldsymbol{\nu}\, a@s)\ s_{\{\mathrm{new}:\{r\}\}}[\![P]\!] \parallel r_{G_1}[\![a@s\,!\,\langle b\rangle]\!] \parallel t_{G_2}[\![a@s\,!\,\langle c\rangle]\!]$$

Subtyping relation

$$b \leq i \qquad b \leq o$$

$$\frac{R \subseteq S}{S^o \leq R^o} \qquad \frac{S \subseteq R}{S^i \leq R^i} \qquad \frac{t \leq t'}{S^t \leq S^{t'}}$$

$$\gamma \leq \gamma \qquad \frac{\gamma_1 \leq \gamma_2 \qquad \gamma_2 \leq \gamma_3}{\gamma_1 \leq \gamma_3}$$

$$\frac{\gamma_1 \leq \gamma_2 \qquad S^t \leq R^{t'}}{\mathrm{ch}(\gamma_1)@S^t \leq \mathrm{ch}(\gamma_2)@R^{t'}}$$

Subtyping relation

$$b \leq i \qquad b \leq o$$

$$\frac{R \subseteq S}{S^o \leq R^o} \qquad \frac{S \subseteq R}{S^i \leq R^i} \qquad \frac{t \leq t'}{S^t \leq S^{t'}}$$

$$\gamma \leq \gamma \qquad \frac{\gamma_1 \leq \gamma_2 \qquad \gamma_2 \leq \gamma_3}{\gamma_1 \leq \gamma_3}$$

$$\frac{\gamma_1 \leq \gamma_2 \qquad S^t \leq R^{t'}}{\mathrm{ch}(\gamma_1)@S^t \leq \mathrm{ch}(\gamma_2)@R^{t'}}$$

$\therefore$ Outputs may grow

Inputs may shrink

Judgments

- Typing names

$$\Gamma \vdash_s n : \gamma$$

  - if n is a simple name, then it belongs to s

Judgments

- Typing names

$$\Gamma \vdash_s n : \gamma$$

  - if n is a simple name, then it belongs to s

- Typing processes

$$\Gamma \vdash_{s,S} P$$

  - simple names of P considered of s
  - at runtime, P might be in any site of S

- Example

$$s_G\mathbf{[}a\,?\,(x\ :\ \{r,t\})\ x\,?\,(\ldots)\ P\mathbf{]}$$

$\uparrow$ $S{=}\{r{,}t\}$

Judgments

- Typing names

$$\Gamma \vdash_s n : \gamma$$

  - if n is a simple name, then it belongs to s

- Typing processes

$$\Gamma \vdash_{s,S} P$$

  - simple names of P considered of s
  - at runtime, P might be in any site of S

- Typing networks

$$\Gamma \vdash N$$

Typing located outputs

- Rule

$$\frac{\Gamma \vdash_s v : \gamma_2 \qquad \gamma_2 \leq \gamma_1 \qquad \Gamma(r)_1(a) = \mathrm{ch}(\gamma_1)@\{r\}^b \qquad S \subseteq \Gamma(r)_2(\mathrm{rem})}{\Gamma \vdash_{s,S} a@r\,!\,\langle v \rangle}$$

Typing located outputs

- Rule

$$\frac{\Gamma \vdash_s v : \gamma_2 \quad \gamma_2 \leq \gamma_1 \quad \Gamma(r)_1(a) = \mathrm{ch}(\gamma_1)@\{r\}^b \quad S \subseteq \Gamma(r)_2(\mathrm{rem})}{\Gamma \vdash_{s,S} a@r\,!\,\langle v \rangle}$$

- Example

$$\frac{\Gamma \vdash_s v : \mathrm{ch}(\gamma)@\{s\}^b \quad \mathrm{ch}(\gamma)@\{s\}^b \leq \mathrm{ch}(\gamma)@\{s,r\}^i \quad \Gamma(r)_1(a) = \mathrm{ch}(\mathrm{ch}(\gamma)@\{s,r\}^i)@\{r\}^b \quad \{t\} \subseteq \{s,t\}}{\Gamma \vdash_{s,\{t\}} a@r\,!\,\langle v \rangle}$$

Typing located outputs

- Rule

$$\frac{\Gamma \vdash_s v : \gamma_2 \quad \gamma_2 \leq \gamma_1 \quad \Gamma(r)_1(a) = \mathrm{ch}(\gamma_1)@\{r\}^b \quad S \subseteq \Gamma(r)_2(\mathrm{rem})}{\Gamma \vdash_{s,S} a@r\,!\,\langle v \rangle}$$

- Example

$$\frac{\Gamma \vdash_s v : \mathrm{ch}(\gamma)@\{s\}^b \quad \mathrm{ch}(\gamma)@\{s\}^b \leq \mathrm{ch}(\gamma)@\{s,r\}^i \quad \Gamma(r)_1(a) = \mathrm{ch}(\mathrm{ch}(\gamma)@\{s,r\}^i)@\{r\}^b \quad \{t\} \subseteq \{s,t\}}{\Gamma \vdash_{s,\{t\}} a@r\,!\,\langle v \rangle}$$

Typing located outputs

- Rule

$$\frac{\Gamma \vdash_s v : \gamma_2 \quad \gamma_2 \leq \gamma_1 \quad \Gamma(r)_1(a) = \mathrm{ch}(\gamma_1)@\{r\}^b \quad S \subseteq \Gamma(r)_2(\mathrm{rem})}{\Gamma \vdash_{s,S} a@r\,!\,\langle v \rangle}$$

- Example

$$\frac{\Gamma \vdash_s v : \mathrm{ch}(\gamma)@\{s\}^b \quad \mathrm{ch}(\gamma)@\{s\}^b \leq \mathrm{ch}(\gamma)@\{s,r\}^i \quad \Gamma(r)_1(a) = \mathrm{ch}(\mathrm{ch}(\gamma)@\{s,r\}^i)@\{r\}^b \quad \{t\} \subseteq \{s,t\}}{\Gamma \vdash_{s,\{t\}} a@r\,!\,\langle v \rangle}$$

Typing located outputs

- Rule

$$\frac{\Gamma \vdash_s v : \gamma_2 \qquad \gamma_2 \le \gamma_1 \qquad \Gamma(r)_1(a) = \mathrm{ch}(\gamma_1)@\{r\}^b \qquad S \subseteq \Gamma(r)_2(\mathrm{rem})}{\Gamma \vdash_{s,S} a@r\,!\,\langle v \rangle}$$

- Example

$$\frac{\Gamma \vdash_s v : \mathrm{ch}(\gamma)@\{s\}^b \qquad \mathrm{ch}(\gamma)@\{s\}^b \le \mathrm{ch}(\gamma)@\{s,r\}^i \qquad \Gamma(r)_1(a) = \mathrm{ch}(\mathrm{ch}(\gamma)@\{s,r\}^i)@\{r\}^b \qquad \{t\} \subseteq \{s,t\}}{\Gamma \vdash_{s,\{t\}} a@r\,!\,\langle v \rangle}$$

Typing located inputs

- Rule

$$\frac{\begin{array}{c}\Gamma \vdash_{s,\{r\}} P \\ \Gamma(r)_1(a) = \mathrm{ch}(\gamma_1)@\{r\}^b \\ \Gamma(s)_1(x) = \mathrm{ch}(\gamma_2)@R^b \\ \mathrm{ch}(\gamma_2)@R^b \leq \gamma_1 \\ S \subseteq \Gamma(r)_2(\mathrm{mig})\end{array}}{\Gamma \setminus x@s \vdash_{s,S} a@r\,?\,(x \ : \ R)\ P}$$

Typing located inputs

- Rule

$$\frac{\Gamma \vdash_{s,\{r\}} P \quad \Gamma(r)_1(a) = \mathrm{ch}(\gamma_1)@\{r\}^b \quad \Gamma(s)_1(x) = \mathrm{ch}(\gamma_2)@R^b \quad \mathrm{ch}(\gamma_2)@R^b \leq \gamma_1 \quad S \subseteq \Gamma(r)_2(\mathrm{mig})}{\Gamma \setminus x@s \vdash_{s,S} a@r\,?\,(x \;:\; R)\; P}$$

- Example

$$s_\emptyset\mathbf{[}a@r\,?\,(x \;:\; \{t\})\; x\,!\,\langle c\rangle\mathbf{]} \parallel r_{\{\mathrm{mig}:\{s\}\}}\mathbf{[}a\,!\,\langle b@t\rangle\mathbf{]} \parallel t_{\{\mathrm{rem}:\{r\}\}}\mathbf{[0]}$$

Typing located names (nets)

- Rule

$$\frac{\Gamma \vdash N \qquad S \setminus s \subseteq \Gamma(s)_2(\mathrm{new}) \qquad S \text{ is the set of sites where } a@s \text{ occurs free in } N}{\Gamma \setminus a@s \vdash (\boldsymbol{\nu}\, a@s)\ N}$$

Typing located names (nets)

- Rule

$$\frac{\Gamma \vdash N \qquad S \setminus s \subseteq \Gamma(s)_2(\mathrm{new}) \qquad S \text{ is the set of sites where } a@s \text{ occurs free in } N}{\Gamma \setminus a@s \vdash (\boldsymbol{\nu}\, a@s)\ N}$$

- Example

$$(\boldsymbol{\nu}\, a@s)\ s_{\{\mathrm{new}:\{r\},\mathrm{rem}:\{r\}\}}\mathbf{[}0\mathbf{]} \parallel r_{\emptyset}\mathbf{[}a@s\,!\,\langle b\rangle\mathbf{]}$$

Typing located names (nets)

- Rule

$$\frac{\Gamma \vdash N \qquad S \setminus s \subseteq \Gamma(s)_2(\text{new}) \qquad S \text{ is the set of sites where } a@s \text{ occurs free in } N}{\Gamma \setminus a@s \vdash (\boldsymbol{\nu}\, a@s)\ N}$$

- Example

$$(\boldsymbol{\nu}\, a@s)\ s_{\{\text{new}:\{r\},\text{rem}:\{r\}\}}[0] \parallel r_{\emptyset}[a@s\,!\,\langle b\rangle]$$

$$S = \{r\}$$
$$\Gamma(s) = \{(\emptyset, \{\text{new} : \{r\}, \text{rem} : \{r\}\})\}$$
$$\Gamma(r) = \{(b : \text{ch}(\gamma)@\{r\}^{b}, \emptyset)\}$$

Runtime errors

$$\mathcal{E} = \{N | N \longrightarrow^{\star} \nu\vec{X}(M' \parallel M)\}$$

and M of the form

- Remote communication

$$r_{G_1}\mathbf{[}P\mathbf{]} \parallel s_{G_2}\mathbf{[}a@r\,!\,\langle v\rangle\mathbf{]}, \qquad s \notin G_1(\mathrm{rem})$$

$$s_G\mathbf{[}a\,!\,\langle b@r\rangle \mid a\,?\,(x\ :\ S)\ P\mathbf{]}, \qquad r \notin S$$

Runtime errors

$$\mathcal{E} = \{N | N \rightarrow^{\star} \nu\vec{X}(M' \parallel M)\}$$

and M of the form

- Remote communication

$$r_{G_1}\mathbf{[}P\mathbf{]} \parallel s_{G_2}\mathbf{[}a@r\,!\,\langle v\rangle\mathbf{]}, \qquad s \notin G_1(\mathrm{rem})$$

$$s_G\mathbf{[}a\,!\,\langle b@r\rangle \mid a\,?\,(x : S)\;P\mathbf{]}, \qquad r \notin S$$

- Migration

$$r_{G_1}\mathbf{[}P\mathbf{]} \parallel s_{G_2}\mathbf{[}a@r\,?\,(x : S)\;P\mathbf{]}, \qquad s \notin G_1(\mathrm{mig})$$

Runtime errors

$$\mathcal{E} = \{N | N \rightarrow^\star \nu\vec{X}(M' \,\|\, M)\}$$

and M of the form

- Remote communication

$$r_{G_1}[\![P]\!] \,\|\, s_{G_2}[\![a@r\,!\,\langle v\rangle]\!], \qquad s \notin G_1 (\mathrm{rem})$$

$$s_G[\![a\,!\,\langle b@r\rangle \,|\, a\,?\,(x : S)\ P]\!], \qquad r \notin S$$

- Migration

$$r_{G_1}[\![P]\!] \,\|\, s_{G_2}[\![a@r\,?\,(x : S)\ P]\!], \qquad s \notin G_1 (\mathrm{mig})$$

- Name creation

$$r_{G_1}[\![P]\!] \,\|\, s_{G_2}[\![(\boldsymbol{\nu}\, a@r)\ P]\!], \qquad s \notin G_1 (\mathrm{new})$$

The usual properties

- Subject reduction

  if $\Gamma \vdash N$ and $N \rightarrow M,$ then $\Gamma \vdash M$

- Well-typed networks free of runtime errors

  if $\Gamma \vdash N$ and $N \rightarrow^{\star} M,$ then $M \notin \mathcal{E}$

Conclusions and further work

- we propose a type system to control:
  - remote communication
  - process migration
  - name creation

Conclusions and further work

- we propose a type system to control:
  - remote communication
  - process migration
  - name creation
- But,
  - In $u?(x : S)\ P$, type S is fixed.
  - Sites are constants (used explicitly in types!)

Conclusions and further work

- we propose a type system to control:
  - remote communication
  - process migration
  - name creation
- But,
  - In $u\,?\,(x\;:\;S)\;P$, type S is fixed.
  - Sites are constants (used explicitly in types!)
- Further work
  - Solve the above limitations :))
  - Specify security policies at channel level
  - Adjust security policies dynamically