Introducing the M-calculus

Jean-Bernard Stefani - INRIA

Motivations

- Unification
  - crowded space: ambients, safe ambients, join, Dπ, Seal, Klaim, DyTiCo, ...
  - can we understand these different calculi as programming with different forms of localities / domains ?
  - is it possible to have programmable localities / domains ?
- Programming model
  - can we capture formally system-level and application-level programming in distributed systems ?
  - for example, to program realistic firewalls, sandboxes and class loaders, processes and threads, components and component containers ? including dynamic binding ?

Outline

- Design principles
- Syntax
- Reduction semantics
- Examples
- Simple types
- Conclusion

Design principles

- Programmable locality / domain : cell
  - a(P)[Q]
  - a : name
  - P : membrane
  - Q : plasm
  - A cell membrane embodies the protocol obeyed by the locality (e.g. in, out, open)
- Asynchronous, higher-order communication
- Combining the join-calculus and the blue calculus
  - replicated resources and join patterns
- Including a form of dynamic binding

Syntax

$\mathcal{S}$	$::=$		Top-level configuration
		$\epsilon[P]$	root cell
	$\mid$	$\nu n.\mathcal{S}$	restriction
P	$::=$		Process
		$\mathbf{0}$	inert process
	$\mid$	V	value
	$\mid$	$a\langle P\rangle[P]$	cell
	$\mid$	$(P \mid P)$	parallel composition
	$\mid$	(PP)	application
	$\mid$	$\nu n.P$	restriction
	$\mid$	$([s = V]P, P)$	name testing
	$\mid$	$\langle D\rangle$	definition
	$\mid$	$\mathbf{pass}_a\ V$	passivation operator

Syntax

V	$::=$		Value
		$()$	void
	$\mid$	u	name
	$\mid$	$\lambda x.P$	lambda abstraction
D	$::=$		Definition
		$\bot$	empty definition
	$\mid$	$J = P$	reaction rule
	$\mid$	$D; D$	composition
J	$::=$		join pattern
		$r\tilde{x}$	message
	$\mid$	$J \mid J$	join

Syntax

s	$::=$		service name
		r	resource name
	$\mid$	$a.r$	addressed resource name
n	$::=$		resolved name
		r	resource name
	$\mid$	a	cell name
u	$::=$		name
		a	cell name
	$\mid$	x	variable
	$\mid$	s	service name

Evaluation contexts

$\mathbf{E}$	$::=$		evaluation context
		$\cdot$	hole
	$\mid$	$(\mathbf{E}V)$	function
	$\mid$	$(P\mathbf{E})$	argument
	$\mid$	$\nu n.\mathbf{E}$	restriction
	$\mid$	$(\mathbf{E} \mid P)$	parallel
	$\mid$	$a(P)[\mathbf{E}]$	plasm
	$\mid$	$a(\mathbf{E})[P]$	membrane
	$\mid$	$\epsilon[\mathbf{E}]$	top

Free names

$$\mathsf{fn}(()) = \emptyset$$
$$\mathsf{fn}(u) = \{u\} \quad \mathsf{fn}(a.r) = \{a, r\}$$
$$\mathsf{fn}(\nu n.P) = \mathsf{fn}(P) \setminus \{n\}$$
$$\mathsf{fn}(\langle D \rangle) = \mathsf{fn}(D)$$
$$\mathsf{fn}(\bot) = \emptyset$$
$$\mathsf{fn}(r_1\widetilde{x}_1 \mid \ldots \mid r_q\widetilde{x}_q) = \mathsf{fn}(r_1\widetilde{x}_1) \cup \ldots \cup \mathsf{fn}(r_q\widetilde{x}_q)$$
$$\mathsf{fn}(a(P)[Q]) = \{a\} \cup \mathsf{fn}(P) \cup \mathsf{fn}(Q)$$
$$\mathsf{fn}([s = V]P, Q) = \mathsf{fn}(s) \cup \mathsf{fn}(P) \cup \mathsf{fn}(Q) \cup \mathsf{fn}(V)$$
$$\mathsf{fn}(\nu n.\mathcal{S}) = \mathsf{fn}(\mathcal{S}) \setminus \{n\}$$

$$\mathsf{fn}(0) = \emptyset$$
$$\mathsf{fn}(\lambda x.P) = \mathsf{fn}(P) \setminus \{x\}$$
$$\mathsf{fn}(PQ) = \mathsf{fn}(P) \cup \mathsf{fn}(Q)$$
$$\mathsf{fn}(D; D') = \mathsf{fn}(D) \cup \mathsf{fn}(D')$$
$$\mathsf{fn}(J = P) = (\mathsf{fn}(P) \setminus \mathsf{fn}(J)) \cup \mathsf{df}(J)$$
$$\mathsf{fn}(r\widetilde{x}) = \{r, x_1, \ldots, x_p\} \quad \widetilde{x} = (x_j)_{j \in \{1, \ldots, p\}}$$
$$\mathsf{fn}(P \mid Q) = \mathsf{fn}(P) \cup \mathsf{fn}(Q)$$
$$\mathsf{fn}(\mathtt{pass}_a\ V) = \mathsf{fn}(V)$$
$$\mathsf{fn}(\epsilon[P]) = \mathsf{fn}(P)$$

Defined names

$$\mathsf{dln}(()) = \emptyset$$
$$\mathsf{dln}(u) = \emptyset$$
$$\mathsf{dln}(\nu n.P) = \mathsf{dln}(P) \setminus \{n\}$$
$$\mathsf{dln}(\langle D \rangle) = \mathsf{dln}(D)$$
$$\mathsf{dln}(\bot) = \emptyset$$
$$\mathsf{dln}(r_1\tilde{x}_1 \mid \ldots \mid r_q\tilde{x}_q) = \{r_1, \ldots, r_q\}$$
$$\mathsf{dln}(P \mid Q) = \mathsf{dln}(P) \cup \mathsf{dln}(Q)$$
$$\mathsf{dln}(\mathrm{pass}_a\ V) = \emptyset$$

$$\mathsf{dln}(0) = \emptyset$$
$$\mathsf{dln}(\lambda x.P) = \emptyset$$
$$\mathsf{dln}(PQ) = \emptyset$$
$$\mathsf{dln}(D; D') = \mathsf{dln}(D) \cup \mathsf{dln}(D')$$
$$\mathsf{dln}(J = P) = \mathsf{dln}(J)$$
$$\mathsf{dln}(a(P)[Q]) = \emptyset$$
$$\mathsf{dln}([s = V]P, Q) = \emptyset$$
$$\mathsf{dln}(\mathcal{S}) = \emptyset$$

Defined cells

$$\mathsf{cells}(()) = \emptyset$$
$$\mathsf{cells}(u) = \emptyset$$
$$\mathsf{cells}(\nu n.P) = \mathsf{cells}(P) \setminus \{n\}$$
$$\mathsf{cells}(\langle D \rangle) = \emptyset$$
$$\mathsf{cells}(P \mid Q) = \mathsf{cells}(P) \cup \mathsf{cells}(Q)$$
$$\mathsf{cells}(\mathrm{pass}_a\ V) = \emptyset$$
$$\mathsf{cells}(c[P]) = \mathsf{cells}(P)$$

$$\mathsf{cells}(\mathbf{0}) = \emptyset$$
$$\mathsf{cells}(\lambda x.P) = \emptyset$$
$$\mathsf{cells}(PQ) = \mathsf{cells}(P) \cup \mathsf{cells}(Q)$$
$$\mathsf{cells}(a(P)[Q]) = \{a\} \cup \mathsf{cells}(P) \cup \mathsf{cells}(Q)$$
$$\mathsf{cells}([s = V]P, Q) = \mathsf{cells}(P) \cup \mathsf{cells}(Q)$$
$$\mathsf{cells}(\nu n.S) = \mathsf{cells}(S) \setminus \{n\}$$

Structural equivalence

$$\frac{n \notin \mathsf{fn}(Q)}{(\nu n.P) \mid Q \equiv \nu n.(P \mid Q)} \text{ [STRUCT.NU.PAR]}$$

$$\frac{}{\epsilon[\nu n.P] \equiv \nu n.\epsilon[P]} \text{ [STRUCT.NU.TOP]}$$

$$\frac{n \notin \mathsf{fn}(Q) \ \wedge \ n \neq a}{a(\nu n.P)[Q] \equiv \nu n.a(P)[Q]} \text{ [STRUCT.NU.MEM]}$$

$$\frac{n \notin \mathsf{fn}(P) \ \wedge \ n \neq a}{a(P)[\nu n.Q] \equiv \nu n.a(P)[Q]} \text{ [STRUCT.NU.PLASM]}$$

$$\frac{P =_\alpha Q}{P \equiv Q} \text{ [STRUCT.}\alpha\text{]}$$

$$\frac{P \equiv Q}{\mathbf{E}\{P\} \equiv \mathbf{E}\{Q\}} \text{ [STRUCT.CONTEXT]}$$

Reduction : evaluation

$$\frac{}{(\lambda x.P)V \to P\{^V/_x\}}\ [\text{RED.BETA}] \qquad \frac{}{([n=n]P,Q) \to P}\ [\text{RED.IF.THEN}]$$

$$\frac{n \neq V}{([n=V]P,Q) \to Q}\ [\text{RED.IF.ELSE}] \qquad \frac{}{a(\mathbf{pass}_a\ V \mid P)[Q] \to V(\lambda.P)(\lambda.Q)}\ [\text{RED.PASSIV}]$$

$$\frac{\langle D\rangle = \langle D_0 : r_1\widetilde{x_1} \mid \ldots \mid r_n\widetilde{x_n} = P\rangle}{\langle D\rangle \mid r_1\widetilde{V_1} \mid \ldots \mid r_n\widetilde{V_n} \to \langle D\rangle \mid P\{\widetilde{V_i}/\widetilde{x_i}\}}\ [\text{RED.RES}] \qquad \frac{P \to Q}{\mathbf{E}\{P\} \to \mathbf{E}\{Q\}}\ [\text{RED.CONTEXT}]$$

$$\frac{P \equiv P' \qquad P' \to Q' \qquad Q' \equiv Q}{P \to Q}\ [\text{RED.PROC.EQUIV}]$$

$$\frac{\mathcal{S}_1 \equiv \mathcal{S}_1' \qquad \mathcal{S}_1' \to \mathcal{S}_2' \qquad \mathcal{S}_2' \equiv \mathcal{S}_2}{\mathcal{S}_1 \to \mathcal{S}_2}\ [\text{RED.TOP.EQUIV}]$$

Reduction : routing (1/3)

$$\frac{r \notin \mathsf{dln}(P) \qquad r \in \mathsf{dln}(Q)}{a(P \mid r\tilde{V})[Q] \to a(P)[Q \mid r\tilde{V}]} \text{ [RED.MESS.PLASM.IN]}$$

$$\frac{r \in \mathsf{dln}(P) \qquad r \notin \mathsf{dln}(Q)}{a(P)[Q \mid r\tilde{V}] \to a(P \mid r\tilde{V})[Q]} \text{ [RED.MESS.PLASM.OUT]}$$

$$\frac{r \notin \mathsf{dln}(P) \qquad r \notin \mathsf{dln}(Q)}{a(P)[Q \mid r\tilde{V}] \to a(P \mid \mathbf{o}(\lambda.r\tilde{V}))[Q]} \text{ [RED.MESS.FILTER.OUT]}$$

$$\frac{r \notin \mathsf{dln}(P) \qquad r \notin \mathsf{dln}(Q)}{b(a(P \mid r\tilde{V})[Q] \mid R)[S] \to b(a(P)[Q] \mid r\tilde{V} \mid R)[S]} \text{ [RED.MEM.MESS.OUT]}$$

$$\frac{r \notin \mathsf{dln}(P) \qquad r \notin \mathsf{dln}(Q)}{b(R)[a(P \mid r\tilde{V})[Q] \mid S] \to b(R)[a(P)[Q] \mid r\tilde{V} \mid S]} \text{ [RED.PLASM.MESS.OUT]}$$

$$\frac{r \notin \mathsf{dln}(P) \qquad r \notin \mathsf{dln}(Q)}{\epsilon[a(P \mid r\tilde{V})[Q] \mid R] \to \epsilon[a(P \mid \mathbf{e}(\lambda.r\tilde{V}))[Q] \mid R]} \text{ [RED.MESS.ERR]}$$

Reduction : routing (2/3)

$$\frac{r \notin \mathsf{dln}(P) \qquad r \in \mathsf{dln}(Q)}{a.r\tilde{V} \mid a(P)[Q] \rightarrow a(P \mid \mathbf{i}(\lambda.r\tilde{V}))[Q]} \;[\text{RED.ADDR.FINAL.PLASM}]$$

$$\frac{r \in \mathsf{dln}(P) \vee r \notin \mathsf{dln}(Q)}{a.r\tilde{V} \mid a(P)[Q] \rightarrow a(P \mid r\tilde{V})[Q]} \;[\text{RED.ADDR.FINAL.MEM}]$$

$$\frac{}{a(P \mid a.r\tilde{V})[Q] \rightarrow a(P \mid r\tilde{V})[Q]} \;[\text{RED.ADDR.MEM}]$$

$$\frac{}{a(P)[Q \mid a.r\tilde{V}] \rightarrow a(P)[Q \mid r\tilde{V}]} \;[\text{RED.ADDR.PLASM}]$$

Reduction : routing (3/3)

$$\frac{b \in \mathsf{cells}(P) \qquad b \neq a}{a(P)[Q] \mid b.r\tilde{V} \to a(P \mid b.r\tilde{V})[Q]} \;[\text{RED.ADDR.MEM.IN}]$$

$$\frac{b \notin \mathsf{cells}(P) \qquad b \in \mathsf{cells}(Q) \qquad b \neq a}{a(P)[Q] \mid b.r\tilde{V} \to a(P \mid \mathbf{i}(\lambda.b.r\tilde{V}))[Q]} \;[\text{RED.ADDR.FILTER.IN}]$$

$$\frac{b \notin \mathsf{cells}(P) \qquad b \in \mathsf{cells}(Q) \qquad b \neq a}{a(P \mid b.r\tilde{V})[Q] \to a(P)[Q \mid b.r\tilde{V}]} \;[\text{RED.ADDR.PLASM.IN}]$$

$$\frac{b \notin \mathsf{cells}(P) \cup \mathsf{cells}(Q) \qquad b \neq a}{a(P \mid b.r\tilde{V})[Q] \to a(P)[Q] \mid b.r\tilde{V}} \;[\text{RED.ADDR.MEM.OUT}]$$

$$\frac{b \notin \mathsf{cells}(Q) \qquad b \in \mathsf{cells}(P) \qquad b \neq a}{a(P)[Q \mid b.r\tilde{V}] \to a(P \mid b.r\tilde{V})[Q]} \;[\text{RED.ADDR.PLASM.OUT}]$$

$$\frac{b \notin \mathsf{cells}(P) \cup \mathsf{cells}(Q) \qquad b \neq a}{a(P)[Q \mid b.r\tilde{V}] \to a(P \mid \mathbf{o}(\lambda.b.r\tilde{V}))[Q]} \;[\text{RED.ADDR.FILTER.OUT}]$$

Example (1/10) : transparent router

$$Fwd = \langle \mathbf{i}\, m = m() \,;\, \mathbf{o}\, m = m() \rangle$$

Example (2/10) : updatable library

$$a\langle\langle \mathsf{l}\ \widetilde{x} = L_1\rangle \mid P(\mathsf{l})\rangle^*[Q] \mid \langle a.\mathsf{update}\ \mathsf{l}\ \lambda\widetilde{x}.L_2\rangle \to^* a\langle\langle \mathsf{l}\ \widetilde{x} = L_2\rangle \mid P(\mathsf{l})\rangle^*[Q]$$

$$\begin{array}{rcl}
(\langle \mathsf{l}\ \widetilde{x} = L\rangle \mid P(\mathsf{l}))^* & = & \nu r.(\langle r\ \lambda\widetilde{x}.L\rangle \mid \langle r\ f \mid \mathsf{l}\ \widetilde{y} = f\ \widetilde{y} \mid r\ f\rangle \mid \langle r\ f' \mid \langle \mathsf{update}\ x\ f\rangle = A(\mathsf{l}, r, x, f, f')\rangle) \\
A(\mathsf{l}, r, x, f, f') & = & ([\mathsf{l} = x]\ \langle r\ f\rangle,\ \langle r\ f'\rangle)
\end{array}$$

Example (3/10) : creating a new cell

$$a.n\ () \mid a(\langle n\ () = New\rangle \mid P_1)[Q_1] \quad \rightarrow^* \quad b(P_2)[Q_2] \mid a(\langle n\ () = New\rangle \mid P_1)[Q_1]$$

$$New = \mathbf{pass}_a\ \lambda p\, q.(b(P_2)[Q_2] \mid a(p())[q()])$$

Example (4/10) : adding a new cell

$$a.\mathsf{add}\,(\lambda.P) \mid a(\langle \mathsf{add}\ f = Add(a,f)\rangle)[Q] \quad \rightarrow^{*} \quad a(\langle \mathsf{add}\ f = Add(a,f)\rangle)[P \mid Q]$$

$$Add(a,f) = \mathrm{pass}_a\ \lambda p\,q.a(p())[f() \mid q()]$$

$$\begin{array}{rcl}
a.\mathsf{add}\,(\lambda.P) \mid a(\langle \mathsf{add}\ f = Add(a,f)\rangle)[Q] & \rightarrow & a(\mathsf{add}\,(\lambda.P) \mid \langle \mathsf{add}\ f = Add(a,f)\rangle)[Q] \\
& \rightarrow & a(\langle \mathsf{add}\ f = Add(a,f)\rangle \mid \mathrm{pass}_a\ \lambda p\,q.p()[(\lambda.P)() \mid q()])[Q] \\
& \rightarrow & \lambda p\,q.a(p())[(\lambda.P)() \mid q()](\lambda.(\langle \mathsf{add}\ f = Add(a,f)\rangle))(\lambda.Q) \\
& \rightarrow^{*} & a(\langle \mathsf{add}\ f = Add(a,f)\rangle)[P \mid Q]
\end{array}$$

Example (5/10) : move

$$a(P)[(\text{go } u) \mid Q] \to^* u.\text{add}\,(a(P)[Q]^*)$$

$$\begin{aligned} P &= \langle Fwd \mid \langle \text{go } u = Go(a,u) \rangle \rangle \\ Go(a,u) &= \text{pass}_a\ \lambda p\, q. u.\text{add}\,(\lambda . a(p())[q()]) \end{aligned}$$

Example (6/10) : objective move

$$(a.\mathsf{move}\ u\ v) \mid a(P)[Q(u) \mid R] \ \rightarrow^* \ (v.\mathsf{add}\ \ Q^*(u)) \mid a(P)[R]$$

$$Q(b) = b(C_b)[Q]$$

$$C_b = (Fwd \mid \langle \mathsf{go}\ v = Go(b, v) \rangle)$$

$$P = (Fwd \mid \langle \mathsf{move}\ x\ y = (x.\mathsf{go}\ y) \rangle)$$

$$\begin{array}{lll} ((a.\mathsf{move}\ bv) \mid a(P)[Q(b) \mid R]) & \rightarrow & a((\mathsf{move}\ bv) \mid P)[Q(b) \mid R] \\ & \rightarrow^* & a()P[(b.\mathsf{go}\ v) \mid Q(b) \mid R] \\ & \rightarrow^* & a(P)[v.\mathsf{add}\ (\lambda.b(\lambda.C_b)()[(\lambda.Q)()] \mid R] \\ & \rightarrow^* & a(v.\mathsf{add}\ (\lambda.b(\lambda.C_b)()[(\lambda.Q)()]) \mid P)[R] \\ & \rightarrow & (v.\mathsf{add}\ (\lambda.b(\lambda.C_b)()[(\lambda.Q)()]) \mid a(P)[R]) \end{array}$$

Example (7/10) : components

$$
\begin{aligned}
Q^{i}(b) &= \nu r\,\mathsf{on}.b(Fwdr(\mathsf{on}) \mid \langle \mathsf{suspend} \mid \mathsf{on} = Suspend_b;\mathsf{resume} \mid r\ x = Add(b,x) \mid \mathsf{on}\rangle \mid \mathsf{on})[Q] \\
Fwdr(\mathsf{on}) &= \langle \mathsf{on} \mid \mathtt{i}\ m = m() \mid \mathsf{on}\ ;\ \mathsf{on} \mid \mathtt{o}\ m = m() \mid \mathsf{on}\rangle \\
Suspend_b &= \mathtt{pass}_b\ \ \lambda p\ q.b(p\ () \mid (r\ q))[0]
\end{aligned}
$$

$$
\begin{aligned}
Q^{e}(b) &= b(P \mid \langle \mathsf{update}\ f = Update_b(f)\rangle)[Q] \\
Update_b(f) &= \mathtt{pass}_b\ \ \lambda p\ q.b(f())[q()]
\end{aligned}
$$

Example (8/10) : coding the π_{1l} calculus

$$
\begin{aligned}
PP(a) \quad &= \quad \nu \mathsf{on}\ \mathsf{off}.(\langle \mathsf{on} \mid \mathbf{i}\ m = (\mathsf{on} \mid m()) \ ;\ \mathsf{on} \mid \mathbf{o}\ m = (\mathsf{on} \mid m())\rangle \\
&\qquad \mid \langle \mathsf{on} \mid \mathsf{add}\ f = Augment(a, f) \\
&\qquad\quad \mathsf{on} \mid \mathsf{stop} = \mathsf{off} \\
&\qquad\quad \mathsf{on} \mid \mathsf{ping}\ (y, n) = \mathsf{on} \mid y() \\
&\qquad\quad \mathsf{off} \mid \mathsf{ping}\ (y, n) = \mathsf{off} \mid n()\rangle) \\
Augment(a, f) \quad &= \quad \mathtt{pass}_a\ \lambda p\ q.a(\mathsf{on} \mid p())[q() \mid f()]
\end{aligned}
$$

Example (9/10) : coding the join-calcul

$$
\begin{array}{rcl}
PJ(a) & = & (Fwd \mid \langle \mathsf{add}\ f = Add(a,f)\rangle \mid \langle \mathsf{go}\ b = Send(a,b)\rangle) \\
Add(a,f) & = & \mathbf{pass}_{\square}\ \lambda p\ q.a(p())[q() \mid f()] \\
Send(a,b) & = & \mathbf{pass}_{\square}\ \lambda p\ q.(b.\mathsf{add}\ \lambda.a(p())[q()])
\end{array}
$$

Example (10/10) : coding the distribued join

$$
\begin{aligned}
PJF(a) &= \nu\mathsf{on}.(\langle \mathsf{on} \mid \mathbf{i}\ m = (\mathsf{on} \mid m)\rangle \\
&\qquad \mid \langle \mathsf{on} \mid \mathsf{o}\ m = (\mathsf{on} \mid m)\rangle \\
&\qquad \mid \langle \mathsf{on} \mid \mathsf{ins}\ f = Insert(a, f)\rangle \\
&\qquad \mid \langle \mathsf{on} \mid \mathsf{go}\ b = Send(a, b)\rangle \\
&\qquad \mid \langle \mathsf{on} \mid \mathsf{halt} = Halt(a)\rangle \\
&\qquad \mid \langle \mathsf{on} \mid \mathsf{ping}\ (y, n) = (\mathsf{on} \mid y\ ())\rangle) \\
Insert(a, f) &= \mathsf{pass}_a\ \lambda p\ q.a(\mathsf{on} \mid p())[q() \mid f()] \\
Send(a, b) &= \mathsf{pass}_a\ \lambda p\ q.(b.\mathsf{ins}\ \lambda.a(\mathsf{on} \mid p())[q()]) \\
Halt(a) &= \mathsf{pass}_a\ \lambda p\ q.\ a(\langle \mathsf{ping}\ (y, n) = n\ ()\rangle \mid \langle \mathbf{i}\ (\mathbf{b.ping}, (y, n)) = n\ ()\rangle \mid \langle \mathsf{o} - = 0\rangle)[q()]
\end{aligned}
$$

Types : syntax

τ	$::=$		type
		Δ	process type
	$\mid$	σ	value type
s	$::=$		type scheme
		$\forall\widetilde{\alpha}\widetilde{\rho}.\sigma$	type scheme
σ	$::=$		value type
		`unit`	unit type
	$\mid$	α	type variable
	$\mid$	`dom`	cell name type
	$\mid$	$\sigma \rightarrow \tau$	function
Δ	$::=$		multiset of cell names
		$\emptyset$	empty multiset
	$\mid$	ρ	multiset variable
	$\mid$	a	cell name
	$\mid$	Δ, Δ	union

Types : auxiliary definitions (1/2)

$$
\begin{array}{rcl}
a, \Delta \wedge a, \Delta' & = & a, (\Delta \wedge \Delta') \\
\rho, \Delta \wedge \rho, \Delta' & = & \rho, (\Delta \wedge \Delta') \\
\Delta \wedge \Delta' & = & \Delta, \Delta' \text{ if } \Delta \cap \Delta' = \emptyset \\
\mathtt{unit} \wedge \mathtt{unit} & = & \mathtt{unit} \\
\mathtt{dom} \wedge \mathtt{dom} & = & \mathtt{dom} \\
\alpha \wedge \alpha & = & \alpha \\
\widetilde{\sigma} \wedge \widetilde{\sigma'} & = & \widetilde{(\sigma \wedge \sigma')} \text{ with tuples of identical size} \\
\sigma \to \tau \wedge \sigma \to \tau' & = & \sigma \to (\tau \wedge \tau')
\end{array}
$$

Types : auxiliary definitions (2/2)

$$
\begin{array}{rcl}
\mathtt{unit} \subseteq \mathtt{unit} & & \\
\mathtt{dom} \subseteq \mathtt{dom} & & \\
\alpha \subseteq \alpha & & \\
\widetilde{\sigma_i}^{i\in[1..n]} \subseteq \widetilde{\sigma'_i}^{i\in[1..n]} & \Leftarrow & (\sigma_i \subseteq \sigma'_i)^{i\in[1..n]} \\
\sigma \rightarrow \tau \subseteq \sigma \rightarrow \tau' & \Leftarrow & \tau \subseteq \tau'
\end{array}
$$

Types : typing contexts

C	$::=$		context
		$\cdot : \tau$	hole for a process
	$\mid$	$\cdot$	hole for a definition
	$\mid$	$\cdot : \Delta$	hole for a top-level configuration
	$\mid$	$\epsilon[C]$	top-level cell
	$\mid$	$\nu r : s.C$	resource restriction
	$\mid$	$\nu a.C$	cell restriction
	$\mid$	$\lambda x.C$	function
	$\mid$	$a(C)[Q]$	cell membrane
	$\mid$	$a(P)[C]$	cell plasm
	$\mid$	$(C \mid P)$	left process parallel
	$\mid$	$(P \mid C)$	right process parallel
	$\mid$	$\mathrm{pass}_a\ C$	passivation
	$\mid$	(CQ)	left application
	$\mid$	(PC)	right application
	$\mid$	$([n = C]P, Q)$	test value
	$\mid$	$([n = V]C, P)$	test true
	$\mid$	$([n = V]P, C)$	test false
	$\mid$	$\langle C \rangle$	definition
	$\mid$	C, D	left definition composition
	$\mid$	D, C	right definition composition
	$\mid$	$J = C$	join guarded process

Types : types for special names i, o et e

$$\mathbf{i} : \forall\rho.(\mathtt{unit} \rightarrow \rho) \rightarrow \rho$$
$$\mathbf{o} : \forall\rho.(\mathtt{unit} \rightarrow \rho) \rightarrow \rho$$
$$\mathbf{e} : \forall\rho.(\mathtt{unit} \rightarrow \rho) \rightarrow \rho$$

Typing rules (1/4)

$$\frac{}{\Gamma \vdash \mathbf{0} : \emptyset} \text{[NIL]} \qquad \frac{}{\Gamma \vdash () : \texttt{unit}} \text{[VOID]} \qquad \frac{u : s \in \Gamma \qquad \sigma = Inst(s)}{\Gamma \vdash u : \sigma} \text{[NAME]}$$

$$\frac{}{\Gamma \vdash (\cdot : \tau) : \tau} \text{[PROC.HOLE]} \qquad \frac{\Gamma \vdash a : \texttt{dom} \qquad \Gamma \vdash r : \sigma \to \Delta}{\Gamma \vdash a.r : \sigma \to \Delta} \text{[ADDR]}$$

$$\frac{\Gamma + x : \sigma \vdash P : \tau \qquad x \notin fn(\Gamma)}{\Gamma \vdash \lambda x.P : \sigma \to \tau} \text{[FUN]} \qquad \frac{\Gamma \vdash a : \texttt{dom} \qquad \Gamma \vdash P : \Delta_1 \qquad \Gamma \vdash Q : \Delta_2}{\Gamma \vdash a(P)[Q] : a, \Delta_1, \Delta_2} \text{[DOM]}$$

Typing rules (2/4)

$$\frac{\Gamma \vdash P : \Delta_1 \qquad \Gamma \vdash Q : \Delta_2}{\Gamma \vdash P \mid Q : \Delta_1, \Delta_2} \; [\text{PAR}]$$

$$\frac{\begin{array}{c}\Gamma + r : \forall \widetilde{\alpha}\widetilde{\rho}.\sigma \to \Delta \vdash P : \Delta_1 \\ r \notin fn(\Gamma) \qquad ftv(\forall \widetilde{\alpha}\widetilde{\rho}.\sigma \to \Delta) = fsv(\forall \widetilde{\alpha}\widetilde{\rho}.\sigma \to \Delta) = \emptyset\end{array}}{\Gamma \vdash \nu r : \forall \widetilde{\alpha}\widetilde{\rho}.\sigma \to \Delta.P : \Delta_1} \; [\text{NU.RES}]$$

$$\frac{\Gamma + a : \texttt{dom} \vdash P : \Delta \qquad a \notin fn(\Gamma) \qquad a \notin (\Delta - a)}{\Gamma \vdash \nu a.P : \Delta - a} \; [\text{NU.DOM}]$$

$$\frac{\begin{array}{c}\Gamma \vdash V : (\texttt{unit} \to \rho_1) \to (\texttt{unit} \to \rho_2) \to \Delta \\ \rho_1, \rho_2 \text{ do not occur in } \Gamma \qquad \rho_1, \rho_2 \text{ occur each at most once in } \Delta\end{array}}{\Gamma \vdash \mathbf{pass}_a\, V : \Delta - (a, \rho_1, \rho_2)} \; [\text{PASS}]$$

Typing rules (3/4)

$$\frac{\Gamma \vdash P : \sigma \to \tau \qquad \Gamma \vdash Q : \sigma' \qquad \sigma' \subseteq \sigma}{\Gamma \vdash PQ : \tau}\;[\text{APP}] \qquad \frac{\Gamma \vdash P : \tau_1 \qquad \Gamma \vdash Q : \tau_2}{\Gamma \vdash [n = V]P, Q : \tau_1 \wedge \tau_2}\;[\text{TEST}]$$

$$\frac{\Gamma \vdash D}{\Gamma \vdash \langle D \rangle : \emptyset}\;[\text{DEF}] \qquad \frac{\Gamma \vdash D_1 \qquad \Gamma \vdash D_2}{\Gamma \vdash D_1, D_2}\;[\text{AND}] \qquad \frac{}{\Gamma \vdash \cdot}\;[\text{DEF.HOLE}] \qquad \frac{}{\Gamma \vdash \bot}\;[\text{DEF.}\bot]$$

$$\frac{\begin{array}{c}(r_i : s_i = \forall \widetilde{\alpha}_i \widetilde{\rho}_i . \widetilde{\sigma}_i \to \Delta_i \in \Gamma)^{i \in [1..n]} \\ \Delta' \subseteq \Delta_1, \ldots, \Delta_n \qquad \Gamma + \widetilde{x_1} : \widetilde{\sigma_1} + \ldots + \widetilde{x_n} : \widetilde{\sigma_n} \vdash P : \Delta' \\ (\widetilde{x}_i)^i \cap fn(\Gamma) = \emptyset \qquad \forall i \in [1..n]. fv(\widetilde{\sigma}_i \to \Delta_i) \cap fv(\Gamma) \cap (\widetilde{\alpha}_i \cup \widetilde{\rho}_i) = \emptyset \\ \forall i, j \in [1..n]^2 . i \neq j \implies fv(\widetilde{\sigma}_i \to \Delta_i) \cap fv(\widetilde{\sigma}_j \to \Delta_j) \cap (\widetilde{\alpha}_i \cup \widetilde{\rho}_i \cup \widetilde{\alpha}_j \cup \widetilde{\rho}_j) = \emptyset\end{array}}{\Gamma \vdash r_1 \widetilde{x_1} \mid \ldots \mid r_n \widetilde{x_n} = P}\;[\text{JOIN}]$$

Typing rules (4/4)

$$\frac{\Gamma \vdash P : \Delta \qquad set(\Delta)}{\Gamma \vdash \epsilon[P] : \Delta}\ [\text{TOP}]$$

$$\frac{\Gamma + r : s \vdash \mathcal{S} : \Delta \qquad r \notin fn(\Gamma) \qquad fsv(s) = ftv(s) = \emptyset}{\Gamma \vdash \nu r : s.\mathcal{S} : \Delta}\ [\text{TOP.NU.RES}]$$

$$\frac{\Gamma + a : \mathtt{dom} \vdash \mathcal{S} : \Delta \qquad a \notin fn(\Gamma)}{\Gamma \vdash \nu a.\mathcal{S} : \Delta - a}\ [\text{TOP.NU.DOM}] \qquad \frac{set(\Delta)}{\Gamma \vdash (\cdot : \Delta) : \Delta}\ [\text{TOP.HOLE}]$$