UNIVERSITÉ DE GRENOBLE

Soutenance de thèse

Du typage vectoriel

Alejandro Díaz-Caro

CAPP Team

LABORATOIRE D'INFORMATIQUE DE GRENOBLE

Adviser: Pablo Arrighi

Co-adviser: Frédéric Prost

23 - 09 - 2011

Lambda calculus [Church'36]

Formal system to study the definition of function

$$
\begin{aligned}
f(x) &\sim t_x \\
x \mapsto f(x) &\sim \lambda x.t_x \\
(x \mapsto f(x))r &\sim (\lambda x.t_x)\ r \\
(x \mapsto f(x))r = f(r) &\sim (\lambda x.t_x)\ r \to t_x[r/x]
\end{aligned}
$$

$$
\boxed{\begin{array}{c} \mathbf{t}, \mathbf{r} ::= x \mid \lambda x.\mathbf{t} \mid (\mathbf{t})\ \mathbf{r} \\ (\lambda x.\mathbf{t})\ \mathbf{r} \to \mathbf{t}[\mathbf{r}/x] \end{array}}
$$

Lambda calculus [Church'36]

Formal system to study the definition of function

$$f(x) \sim t_x$$
$$x \mapsto f(x) \sim \lambda x.t_x$$
$$(x \mapsto f(x))r \sim (\lambda x.t_x)\ r$$
$$(x \mapsto f(x))r = f(r) \sim (\lambda x.t_x)\ r \to t_x[r/x]$$

$$\mathbf{t}, \mathbf{r} ::= x \mid \lambda x.\mathbf{t} \mid (\mathbf{t})\ \mathbf{r}$$
$$(\lambda x.\mathbf{t})\ \mathbf{r} \to \mathbf{t}[\mathbf{r}/x]$$

Type system [Church'40]

"a tractable syntactic framework for classifying phrases according to the kinds of values they compute"

—[Pierce'02]

$$\frac{\lambda x.\mathbf{t}_x : T \to R \qquad \mathbf{r} : T}{(\lambda x.\mathbf{t}_x)\ \mathbf{r} : R}$$

Lambda calculus [Church'36]

Formal system to study the definition of function

$$\begin{aligned} f(x) &\sim t_x \\ x \mapsto f(x) &\sim \lambda x.t_x \\ (x \mapsto f(x))r &\sim (\lambda x.t_x)\ r \\ (x \mapsto f(x))r = f(r) &\sim (\lambda x.t_x)\ r \to t_x[r/x] \end{aligned}$$

$$\boxed{\begin{array}{c} \mathbf{t}, \mathbf{r} ::= x \mid \lambda x.\mathbf{t} \mid (\mathbf{t})\ \mathbf{r} \\ (\lambda x.\mathbf{t})\ \mathbf{r} \to \mathbf{t}[\mathbf{r}/x] \end{array}}$$

System F [Girard'71]

TS with a universal quantification over types

$$\begin{aligned} \lambda x.x &: Int \to Int \\ \lambda x.x &: Bool \to Bool \\ &\dots \end{aligned}$$

$$\rightsquigarrow \quad \lambda x.x : \forall X.X \to X$$

Type system [Church'40]

"a tractable syntactic framework for classifying phrases according to the kinds of values they compute"

—[Pierce'02]

$$\frac{\lambda x.\mathbf{t}_x : T \to R \qquad \mathbf{r} : T}{(\lambda x.\mathbf{t}_x)\ \mathbf{r} : R}$$

Lambda calculus [Church'36]

Formal system to study the definition of function

$$f(x) \sim t_x$$
$$x \mapsto f(x) \sim \lambda x.t_x$$
$$(x \mapsto f(x))r \sim (\lambda x.t_x)\ r$$
$$(x \mapsto f(x))r = f(r) \sim (\lambda x.t_x)\ r \to t_x[r/x]$$

$$\boxed{\begin{array}{c}\mathbf{t}, \mathbf{r} ::= x \mid \lambda x.\mathbf{t} \mid (\mathbf{t})\ \mathbf{r} \\ (\lambda x.\mathbf{t})\ \mathbf{r} \to \mathbf{t}[\mathbf{r}/x]\end{array}}$$

Type system [Church'40]

"a tractable syntactic framework for classifying phrases according to the kinds of values they compute"

—[Pierce'02]

$$\frac{\lambda x.\mathbf{t}_x : T \to R \qquad \mathbf{r} : T}{(\lambda x.\mathbf{t}_x)\ \mathbf{r} : R}$$

System F [Girard'71]

TS with a universal quantification over types

$$\lambda x.x : Int \to Int$$
$$\lambda x.x : Bool \to Bool$$
$$\dots$$
$$\rightsquigarrow \quad \lambda x.x : \forall X.X \to X$$

Curry-Howard correspondence

Correspondence between type systems and logic

$$\frac{\lambda x.\mathbf{t}_x : T \to R \qquad \mathbf{r} : T}{(\lambda x.\mathbf{t}_x)\ \mathbf{r} : R}$$
$$\Updownarrow$$
$$\frac{T \Rightarrow R \qquad T}{R}$$

Lambda calculus [Church'36]

Formal system to study the definition of function

$$f(x) \sim t_x$$
$$x \mapsto f(x) \sim \lambda x.t_x$$
$$(x \mapsto f(x))r \sim (\lambda x.t_x)\ r$$
$$(x \mapsto f(x))r = f(r) \sim (\lambda x.t_x)\ r \to t_x[r/x]$$

$$\boxed{\begin{array}{c}\mathbf{t}, \mathbf{r} ::= x \mid \lambda x.\mathbf{t} \mid (\mathbf{t})\ \mathbf{r} \\ (\lambda x.\mathbf{t})\ \mathbf{r} \to \mathbf{t}[\mathbf{r}/x]\end{array}}$$

Type system [Church'40]

"a tractable syntactic framework for classifying phrases according to the kinds of values they compute"

—[Pierce'02]

$$\frac{\lambda x.\mathbf{t}_x : T \to R \qquad \mathbf{r} : T}{(\lambda x.\mathbf{t}_x)\ \mathbf{r} : R}$$

System F [Girard'71]

TS with a universal quantification over types

$$\lambda x^{Int}.x : Int \to Int$$
$$\lambda x^{Bool}.x : Bool \to Bool$$
$$\dots$$
$$\rightsquigarrow \quad \Lambda X.\lambda x^{X}.x : \forall X.X \to X$$

Curry-Howard correspondence

Correspondence between type systems and logic

$$\frac{\lambda x.\mathbf{t}_x : T \to R \qquad \mathbf{r} : T}{(\lambda x.\mathbf{t}_x)\ \mathbf{r} : R}$$
$$\Updownarrow$$
$$\frac{T \Rightarrow R \qquad T}{R}$$

Church vs. Curry style whether the types are part of the terms or not

To capture probabilistic/quantum/quantitative constructions:

algebraic extensions

$\mathbf{t}, \mathbf{r} ::= x \mid \lambda x.\mathbf{t} \mid (\mathbf{t})\,\mathbf{r} \mid \mathbf{t} + \mathbf{r} \mid \alpha.\mathbf{t} \mid 0$ $\qquad \alpha \in (\mathcal{S}, +, \times)$, a ring.

Two origins:

- Differential λ-calculus [Ehrhard'03]: linearity *à la* Linear Logic

  Removing the differential operator: Algebraic λ-calculus (λ_{alg}) [Vaux'09]
- Quantum computing: superposition of programs

  Linearity as in algebra: Linear-algebraic λ-calculus (λ_{lin}) [Arrighi,Dowek'08]

To capture probabilistic/quantum/quantitative constructions:

algebraic extensions

$\mathbf{t}, \mathbf{r} ::= x \mid \lambda x.\mathbf{t} \mid (\mathbf{t})\ \mathbf{r} \mid \mathbf{t} + \mathbf{r} \mid \alpha.\mathbf{t} \mid 0$ $\qquad \alpha \in (\mathcal{S}, +, \times)$, a ring.

Two origins:

- Differential λ-calculus [Ehrhard'03]: linearity *à la* Linear Logic

  Removing the differential operator: Algebraic λ-calculus (λ_{alg}) [Vaux'09]
- Quantum computing: superposition of programs

  Linearity as in algebra: Linear-algebraic λ-calculus (λ_{lin}) [Arrighi,Dowek'08]

Beta reduction:

$$(\lambda x.\mathbf{t})\ \mathbf{r} \to \mathbf{t}[\mathbf{r}/x]$$

"Algebraic" reductions:

$$\begin{aligned} \alpha.\mathbf{t} + \beta.\mathbf{t} &\to (\alpha + \beta).\mathbf{t}, \\ \alpha.\beta.\mathbf{t} &\to (\alpha \times \beta).\mathbf{t}, \\ (\mathbf{t})\ (\mathbf{r}_1 + \mathbf{r}_2) &\to (\mathbf{t})\ \mathbf{r}_1 + (\mathbf{t})\ \mathbf{r}_2, \\ (\mathbf{t}_1 + \mathbf{t}_2)\ \mathbf{r} &\to (\mathbf{t}_1)\ \mathbf{r} + (\mathbf{t}_2)\ \mathbf{r}, \\ &\dots \end{aligned}$$

(oriented version of the axioms of vectorial spaces)[Arrighi,Dowek'07]

To capture probabilistic/quantum/quantitative constructions:

algebraic extensions

$\mathbf{t}, \mathbf{r} ::= x \mid \lambda x.\mathbf{t} \mid (\mathbf{t})\ \mathbf{r} \mid \mathbf{t} + \mathbf{r} \mid \alpha.\mathbf{t} \mid 0$ $\qquad \alpha \in (\mathcal{S}, +, \times)$, a ring.

Two origins:

- Differential λ-calculus [Ehrhard'03]: linearity *à la* Linear Logic

  Removing the differential operator: Algebraic λ-calculus (λ_{alg}) [Vaux'09]

- Quantum computing: superposition of programs

  Linearity as in algebra: Linear-algebraic λ-calculus (λ_{lin}) [Arrighi,Dowek'08]

Beta reduction:

$$(\lambda x.\mathbf{t})\ \mathbf{r} \to \mathbf{t}[\mathbf{r}/x]$$

"Algebraic" reductions:

$$\alpha.\mathbf{t} + \beta.\mathbf{t} \to (\alpha + \beta).\mathbf{t},$$
$$\alpha.\beta.\mathbf{t} \to (\alpha \times \beta).\mathbf{t},$$
$$(\mathbf{t})\ (\mathbf{r}_1 + \mathbf{r}_2) \to (\mathbf{t})\ \mathbf{r}_1 + (\mathbf{t})\ \mathbf{r}_2,$$
$$(\mathbf{t}_1 + \mathbf{t}_2)\ \mathbf{r} \to (\mathbf{t}_1)\ \mathbf{r} + (\mathbf{t}_2)\ \mathbf{r},$$
$$\ldots$$

(oriented version of the axioms of vectorial spaces)[Arrighi,Dowek'07]

Vectorial space of values

$\mathcal{B} = \{\mathbf{t}_i : \mathbf{t}_i \text{ var. or abs. }\}$

Set of values $::= \mathrm{Span}(\mathcal{B})$

To capture probabilistic/quantum/quantitative constructions:

algebraic extensions

$\mathbf{t}, \mathbf{r} ::= x \mid \lambda x.\mathbf{t} \mid (\mathbf{t})\ \mathbf{r} \mid \mathbf{t} + \mathbf{r} \mid \alpha.\mathbf{t} \mid 0$ $\qquad \alpha \in (\mathcal{S}, +, \times)$, a ring.

	λ_{alg}	λ_{lin}
Origin	Linear Logic	Quantum computing
Evaluation strategy	Call-by-name	Call-by-base
Algebraic part	Equalities	Rewrite system

Contribution: CPS simulation [Díaz-Caro,Perdrix,Tasson,Valiron'10]

Beta reduction:

$$(\lambda x.\mathbf{t})\ \mathbf{r} \to \mathbf{t}[\mathbf{r}/x]$$

"Algebraic" reductions:

$$\alpha.\mathbf{t} + \beta.\mathbf{t} \to (\alpha+\beta).\mathbf{t},$$
$$\alpha.\beta.\mathbf{t} \to (\alpha\times\beta).\mathbf{t},$$
$$(\mathbf{t})\ (\mathbf{r}_1 + \mathbf{r}_2) \to (\mathbf{t})\ \mathbf{r}_1 + (\mathbf{t})\ \mathbf{r}_2,$$
$$(\mathbf{t}_1 + \mathbf{t}_2)\ \mathbf{r} \to (\mathbf{t}_1)\ \mathbf{r} + (\mathbf{t}_2)\ \mathbf{r},$$
$$\dots$$

(oriented version of the axioms of vectorial spaces)[Arrighi,Dowek'07]

Vectorial space of values

$\mathcal{B} = \{\mathbf{t}_i : \mathbf{t}_i \text{ var. or abs. }\}$

Set of values $::= \mathrm{Span}(\mathcal{B})$

Example of program

Two base vectors: $\textbf{true} = \lambda x.\lambda y.x$
$\textbf{false} = \lambda x.\lambda y.y$

Example of program

Two base vectors: $\textbf{true} = \lambda x.\lambda y.x$
$\textbf{false} = \lambda x.\lambda y.y$

Linear map **U** s.t. $(\textbf{U})\textbf{true} = a.\textbf{true} + b.\textbf{false}$
$(\textbf{U})\textbf{false} = c.\textbf{true} + d.\textbf{false}$

Example of program

Two base vectors:
$$\mathbf{true} = \lambda x.\lambda y.x$$
$$\mathbf{false} = \lambda x.\lambda y.y$$

Linear map **U** s.t.
$$(\mathbf{U})\mathbf{true} = a.\mathbf{true} + b.\mathbf{false}$$
$$(\mathbf{U})\mathbf{false} = c.\mathbf{true} + d.\mathbf{false}$$

$$\mathbf{U} := \lambda x.\{((x)\ [a.\mathbf{true} + b.\mathbf{false}])\ [c.\mathbf{true} + d.\mathbf{false}]\}$$

Example of program

Two base vectors: $\textbf{true} = \lambda x.\lambda y.x$
$\textbf{false} = \lambda x.\lambda y.y$

Linear map $\mathbf{U}$ s.t. $(\mathbf{U})\textbf{true} = a.\textbf{true} + b.\textbf{false}$
$(\mathbf{U})\textbf{false} = c.\textbf{true} + d.\textbf{false}$

$$\mathbf{U} := \lambda x.\{((x)\ [a.\textbf{true} + b.\textbf{false}])\ [c.\textbf{true} + d.\textbf{false}]\}$$

Aim:

To provide a type system capturing the "vectorial" structure of terms

... to check for properties of probabilistic processes

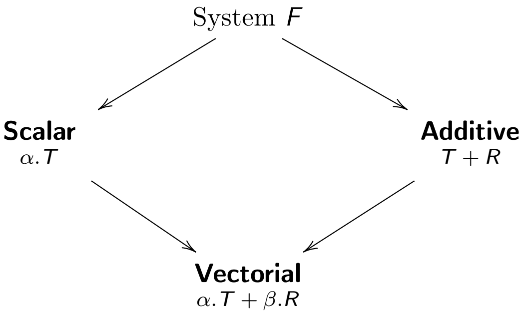
... to check for properties of quantum processes

... or whatever application needing the structure of the vector in normal form

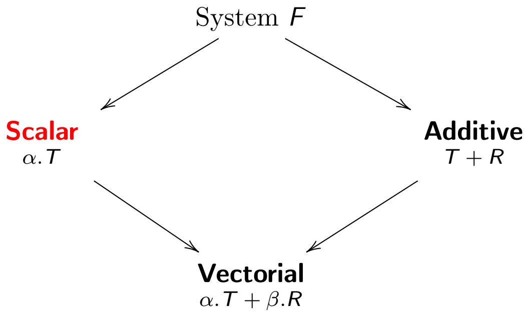
... understand what it means "linear combination of types"

... a Curry-Howard approach to defining Fuzzy/Quantum/Probabilistic logics from Fuzzy/Quantum/Probabilistic programming languages.

Plan



Plan



The *Scalar* Type System

A polymorphic type system *tracking scalars*:

$$\frac{\Gamma \vdash \mathbf{t} : T}{\Gamma \vdash \alpha.\mathbf{t} : \alpha.T}$$

$$\frac{\Gamma \vdash \mathbf{t} : \alpha.T \quad \Gamma \vdash \mathbf{r} : \beta.T}{\Gamma \vdash \mathbf{t} + \mathbf{r} : (\alpha + \beta).T}$$

The *Scalar* Type System

A polymorphic type system *tracking scalars*:

$$\frac{\Gamma \vdash \mathbf{t} : T}{\Gamma \vdash \alpha.\mathbf{t} : \alpha.T}$$

$$\frac{\Gamma \vdash \mathbf{t} : \alpha.T \quad \Gamma \vdash \mathbf{r} : \beta.T}{\Gamma \vdash \mathbf{t} + \mathbf{r} : (\alpha + \beta).T}$$

Gives the "amount" of terms $\rightarrow$ Barycentric restrictions ($\sum \alpha_i = 1$)

The *Scalar* Type System

A polymorphic type system *tracking scalars*:

$$\frac{\Gamma \vdash \mathbf{t} : T}{\Gamma \vdash \alpha.\mathbf{t} : \alpha.T}$$

$$\frac{\Gamma \vdash \mathbf{t} : \alpha.T \quad \Gamma \vdash \mathbf{r} : \beta.T}{\Gamma \vdash \mathbf{t} + \mathbf{r} : (\alpha + \beta).T}$$

Gives the "amount" of terms → Barycentric restrictions ($\sum \alpha_i = 1$)

Definition (Weight function (to check barycentricity))

$\omega(\mathbf{0}) = 0$ $\quad \omega(\mathbf{b}) = 1$ $\quad \omega(\alpha.\mathbf{t}) = \alpha \times \omega(\mathbf{t})$

$\omega((\mathbf{t})\ \mathbf{r}) = \omega(\mathbf{t}) \times \omega(\mathbf{r})$ $\quad \omega(\mathbf{t} + \mathbf{r}) = \omega(\mathbf{t}) + \omega(\mathbf{r})$

The *Scalar* Type System

A polymorphic type system *tracking scalars*:

$$\frac{\Gamma \vdash \mathbf{t} : T}{\Gamma \vdash \alpha.\mathbf{t} : \alpha.T}$$

$$\frac{\Gamma \vdash \mathbf{t} : \alpha.T \quad \Gamma \vdash \mathbf{r} : \beta.T}{\Gamma \vdash \mathbf{t} + \mathbf{r} : (\alpha + \beta).T}$$

Gives the "amount" of terms $\rightarrow$ Barycentric restrictions ($\sum \alpha_i = 1$)

Definition (Weight function (to check barycentricity)**)**

$\omega(\mathbf{0}) = 0$ $\quad\omega(\mathbf{b}) = 1$ $\quad\omega(\alpha.\mathbf{t}) = \alpha \times \omega(\mathbf{t})$

$\omega((\mathbf{t})\ \mathbf{r}) = \omega(\mathbf{t}) \times \omega(\mathbf{r})$ $\quad\omega(\mathbf{t} + \mathbf{r}) = \omega(\mathbf{t}) + \omega(\mathbf{r})$

Theorem

If $\Gamma_C \vdash \boldsymbol{t} : C$ *then* $\omega(\boldsymbol{t}\downarrow) = 1$

The *Scalar* Type System

A polymorphic type system *tracking scalars*:

$$\frac{\Gamma \vdash \mathbf{t} : T}{\Gamma \vdash \alpha.\mathbf{t} : \alpha.T}$$

$$\frac{\Gamma \vdash \mathbf{t} : \alpha.T \quad \Gamma \vdash \mathbf{r} : \beta.T}{\Gamma \vdash \mathbf{t} + \mathbf{r} : (\alpha + \beta).T}$$

Gives the "amount" of terms → Barycentric restrictions ($\sum \alpha_i = 1$)

Definition (Weight function (to check barycentricity))

$\omega(\mathbf{0}) = 0$ $\quad \omega(\mathbf{b}) = 1$ $\quad \omega(\alpha.\mathbf{t}) = \alpha \times \omega(\mathbf{t})$

$\omega((\mathbf{t})\ \mathbf{r}) = \omega(\mathbf{t}) \times \omega(\mathbf{r})$ $\quad \omega(\mathbf{t} + \mathbf{r}) = \omega(\mathbf{t}) + \omega(\mathbf{r})$

Theorem

If $\Gamma_C \vdash \boldsymbol{t} : C$ *then* $\omega(\boldsymbol{t}\downarrow) = 1$

Example

$$2.(\lambda x.\frac{1}{2}.x)\ y$$

$$\omega(2.(\lambda x.\frac{1}{2}.x)\ y) = 2$$

$$y : C \vdash 2.(\lambda x.\frac{1}{2}.x)\ y : C$$

The *Scalar* Type System

A polymorphic type system *tracking scalars*:

$$\frac{\Gamma \vdash \mathbf{t} : T}{\Gamma \vdash \alpha.\mathbf{t} : \alpha.T}$$

$$\frac{\Gamma \vdash \mathbf{t} : \alpha.T \quad \Gamma \vdash \mathbf{r} : \beta.T}{\Gamma \vdash \mathbf{t} + \mathbf{r} : (\alpha + \beta).T}$$

Gives the "amount" of terms → Barycentric restrictions ($\sum \alpha_i = 1$)

Definition (Weight function (to check barycentricity))

$\omega(\mathbf{0}) = 0 \qquad \omega(\mathbf{b}) = 1 \qquad \omega(\alpha.\mathbf{t}) = \alpha \times \omega(\mathbf{t})$

$\omega((\mathbf{t})\ \mathbf{r}) = \omega(\mathbf{t}) \times \omega(\mathbf{r}) \qquad \omega(\mathbf{t} + \mathbf{r}) = \omega(\mathbf{t}) + \omega(\mathbf{r})$

Theorem

If $\Gamma_C \vdash \boldsymbol{t} : C$ *then* $\omega(\boldsymbol{t}\downarrow) = 1$

Example

$2.(\lambda x.\frac{1}{2}.x)\ y \to^* y$

$\omega(2.(\lambda x.\frac{1}{2}.x)\ y) = 2 \qquad \omega(y) = 1$

$y : C \vdash 2.(\lambda x.\frac{1}{2}.x)\ y : C$

The *Scalar* Type System

A polymorphic type system *tracking scalars*:

$$\frac{\Gamma \vdash \mathbf{t} : T}{\Gamma \vdash \alpha.\mathbf{t} : \alpha.T}$$

$$\frac{\Gamma \vdash \mathbf{t} : \alpha.T \quad \Gamma \vdash \mathbf{r} : \beta.T}{\Gamma \vdash \mathbf{t} + \mathbf{r} : (\alpha + \beta).T}$$

- Subject reduction (type preservation)
- Strong normalisation
  1. SN for a straightforward extension of System F
  2. verify that both systems type the same terms

Gives the "amount" of terms $\rightarrow$ Barycentric restrictions ($\sum \alpha_i = 1$)

Definition (Weight function (to check barycentricity))

$$\omega(\mathbf{0}) = 0 \qquad \omega(\mathbf{b}) = 1 \qquad \omega(\alpha.\mathbf{t}) = \alpha \times \omega(\mathbf{t})$$

$$\omega((\mathbf{t})\ \mathbf{r}) = \omega(\mathbf{t}) \times \omega(\mathbf{r}) \qquad \omega(\mathbf{t} + \mathbf{r}) = \omega(\mathbf{t}) + \omega(\mathbf{r})$$

Theorem

If $\Gamma_C \vdash \boldsymbol{t} : C$ *then* $\omega(\boldsymbol{t}\downarrow) = 1$

Example

$$2.(\lambda x.\frac{1}{2}.x)\ y \rightarrow^* y$$

$$\omega(2.(\lambda x.\frac{1}{2}.x)\ y) = 2 \qquad \omega(y) = 1$$

$$y : C \vdash 2.(\lambda x.\frac{1}{2}.x)\ y : C$$

The *Scalar* Type System

A polymorphic type system *tracking scalars*:

$$\frac{\Gamma \vdash \mathbf{t} : T}{\Gamma \vdash \alpha.\mathbf{t} : \alpha.T}$$

$$\frac{\Gamma \vdash \mathbf{t} : \alpha.T \quad \Gamma \vdash \mathbf{r} : \beta.T}{\Gamma \vdash \mathbf{t} + \mathbf{r} : (\alpha + \beta).T}$$

- Subject reduction (type preservation)
- Strong normalisation
  1. SN for a straightforward extension of System F
  2. verify that both systems type the same terms

Gives the "amount" of terms $\rightarrow$ Barycentric restrictions ($\sum \alpha_i = 1$)

Definition (Weight function (to check barycentricity))

$\omega(\mathbf{0}) = 0$ $\quad \omega(\mathbf{b}) = 1$ $\quad \omega(\alpha.\mathbf{t}) = \alpha \times \omega(\mathbf{t})$

$\omega((\mathbf{t})\ \mathbf{r}) = \omega(\mathbf{t}) \times \omega(\mathbf{r})$ $\quad \omega(\mathbf{t} + \mathbf{r}) = \omega(\mathbf{t}) + \omega(\mathbf{r})$

Theorem

If $\Gamma_C \vdash \boldsymbol{t} : C$ *then* $\omega(\boldsymbol{t}\downarrow) = 1$

Example

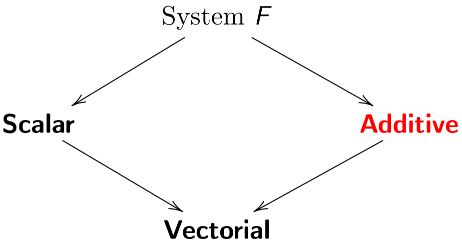
$$2.(\lambda x.\frac{1}{2}.x)\ y \to^* y$$

$$\omega(2.(\lambda x.\frac{1}{2}.x)\ y) = 2 \qquad \omega(y) = 1$$

$$y : C \vdash 2.(\lambda x.\frac{1}{2}.x)\ y : C$$

Contribution: [Arrighi,Díaz-Caro'09]

Plan



The *Additive* Type System

A polymorphic type system *with sums* (for the additive fragment of λ_{lin})

$$\frac{\Gamma \vdash \mathbf{t} : T \qquad \Gamma \vdash \mathbf{r} : R}{\Gamma \vdash \mathbf{t} + \mathbf{r} : T + R}$$

- Sums $\sim$ Assoc., comm. pairs
- distributive w.r.t. application

The *Additive* Type System

A polymorphic type system *with sums* (for the additive fragment of λ_{lin})

$$\frac{\Gamma \vdash \mathbf{t} : T \qquad \Gamma \vdash \mathbf{r} : R}{\Gamma \vdash \mathbf{t} + \mathbf{r} : T + R}$$

- Sums $\sim$ Assoc., comm. pairs
- distributive w.r.t. application

Translation into System F with pairs

The *Additive* Type System

A polymorphic type system *with sums* (for the additive fragment of λ_{lin})

$$\frac{\Gamma \vdash \mathbf{t} : T \qquad \Gamma \vdash \mathbf{r} : R}{\Gamma \vdash \mathbf{t} + \mathbf{r} : T + R}$$

- Sums $\sim$ Assoc., comm. pairs
- distributive w.r.t. application

Translation into System F with pairs

- Simplified version without AC of $+$ $\qquad |T + R| = |T| \times |R|$

The *Additive* Type System

A polymorphic type system *with sums* (for the additive fragment of λ_{lin})

$$\frac{\Gamma \vdash \mathbf{t} : T \qquad \Gamma \vdash \mathbf{r} : R}{\Gamma \vdash \mathbf{t} + \mathbf{r} : T + R}$$

- Sums $\sim$ Assoc., comm. pairs
- distributive w.r.t. application

Translation into System F with pairs

- Simplified version without AC of $+$ $\qquad |T + R| = |T| \times |R|$
- Distributivity in the translation (using the structure given by the type)

The *Additive* Type System

A polymorphic type system *with sums* (for the additive fragment of λ_{lin})

$$\frac{\Gamma \vdash \mathbf{t} : T \qquad \Gamma \vdash \mathbf{r} : R}{\Gamma \vdash \mathbf{t} + \mathbf{r} : T + R}$$

- Sums $\sim$ Assoc., comm. pairs
- distributive w.r.t. application

Translation into System F with pairs

- Simplified version without AC of $+$ $\qquad |T + R| = |T| \times |R|$
- Distributivity in the translation (using the structure given by the type)
- Equivalences given explicitly: $T \equiv R$ implies $|T| \Leftrightarrow |R|$

$$A \times B \Leftrightarrow B \times A \qquad (A \times B) \times C \Leftrightarrow A \times (B \times C)$$

The *Additive* Type System

A polymorphic type system *with sums* (for the additive fragment of λ_{lin})

$$\frac{\Gamma \vdash \mathbf{t} : T \qquad \Gamma \vdash \mathbf{r} : R}{\Gamma \vdash \mathbf{t} + \mathbf{r} : T + R}$$

- Sums $\sim$ Assoc., comm. pairs
- distributive w.r.t. application

Translation into System F with pairs

- Simplified version without AC of $+$ $\qquad |T + R| = |T| \times |R|$
- Distributivity in the translation (using the structure given by the type)
- Equivalences given explicitly: $T \equiv R$ implies $|T| \Leftrightarrow |R|$

$$A \times B \Leftrightarrow B \times A \qquad (A \times B) \times C \Leftrightarrow A \times (B \times C)$$

Theorem

If $\Gamma \vdash \boldsymbol{t} : T$ *and exists* $T' \equiv T$ *then* $|\Gamma| \vdash_F [\boldsymbol{t}]_{\mathcal{D}} : |T'|$

Also we set up an inverse translation showing that it is non-trivial

The *Additive* Type System

A polymorphic type system *with sums* (for the additive fragment of λ_{lin})

$$\frac{\Gamma \vdash \mathbf{t} : T \qquad \Gamma \vdash \mathbf{r} : R}{\Gamma \vdash \mathbf{t} + \mathbf{r} : T + R}$$

- Sums $\sim$ Assoc., comm. pairs
- distributive w.r.t. application

Translation into System F with pairs

- Simplified version without AC of $+$ $\qquad |T + R| = |T| \times |R|$
- Distributivity in the translation (using the structure given by the type)
- Equivalences given explicitly: $T \equiv R$ implies $|T| \Leftrightarrow |R|$

$$A \times B \Leftrightarrow B \times A \qquad (A \times B) \times C \Leftrightarrow A \times (B \times C)$$

Theorem

If $\Gamma \vdash \mathbf{t} : T$ and exists $T' \equiv T$ then $|\Gamma| \vdash_F [\mathbf{t}]_{\mathcal{D}} : |T'|$

Also we set up an inverse translation showing that it is non-trivial

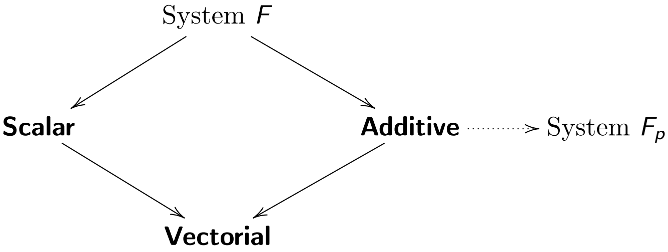
---

Subject reduction ✓

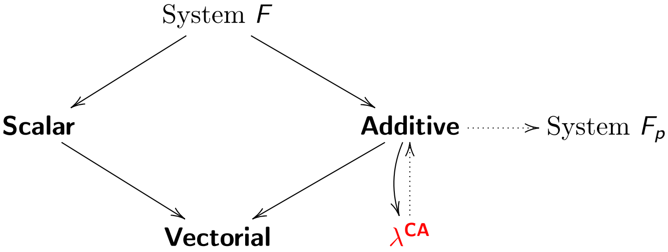
Strong normalisation (using the one from System F_p) ✓

Contribution: `[Díaz-Caro,Petit'10]`

Plan



Plan



System F
Scalar
Additive
System F_p
Vectorial
λ^{CA}

The *Complete Additive* System (λ^{CA})

Extending sums to the whole calculus (with positive reals scalars)

$$\frac{\Gamma \vdash \mathbf{t} : T}{\Gamma \vdash \alpha.\mathbf{t} : \lfloor\alpha\rfloor.T} = \underbrace{T + \cdots + T}_{\lfloor\alpha\rfloor}$$

- More general than Additive
- Less complex than Vectorial
- "Amounts" approximated

The *Complete Additive* System (λ^{CA})

Extending sums to the whole calculus (with positive reals scalars)

$$\frac{\Gamma \vdash \mathbf{t} : T}{\Gamma \vdash \alpha.\mathbf{t} : \lfloor\alpha\rfloor.T} \quad \lfloor\alpha\rfloor.T = \underbrace{T + \cdots + T}_{\lfloor\alpha\rfloor}$$

- More general than Additive
- Less complex than Vectorial
- "Amounts" approximated

If $\vdash \mathbf{t} : T$, then $\vdash (0.9).\mathbf{t} + (1.1).\mathbf{t} : T$

$(0.9).\mathbf{t} + (1.1).\mathbf{t} \to 2.\mathbf{t}$ and $\vdash 2.\mathbf{t} : 2.T$

The *Complete Additive* System (λ^{CA})

Extending sums to the whole calculus (with positive reals scalars)

$$\frac{\Gamma \vdash \mathbf{t} : T}{\Gamma \vdash \alpha.\mathbf{t} : \lfloor \alpha \rfloor . T} = \underbrace{T + \cdots + T}_{\lfloor \alpha \rfloor}$$

- More general than Additive
- Less complex than Vectorial
- "Amounts" approximated

$$\text{If } \vdash \mathbf{t} : T, \quad \text{then } \vdash (0.9).\mathbf{t} + (1.1).\mathbf{t} : T$$
$$(0.9).\mathbf{t} + (1.1).\mathbf{t} \to 2.\mathbf{t} \quad \text{and} \quad \vdash 2.\mathbf{t} : 2.T$$

Weak subject reduction: $\mathbf{t} \to \mathbf{r}$, $\Gamma \vdash \mathbf{t} : T \Rightarrow \Gamma \vdash \mathbf{r} : R$ with $T \preceq R$

The *Complete Additive* System (λ^{CA})

Extending sums to the whole calculus (with positive reals scalars)

$$\frac{\Gamma \vdash \mathbf{t} : T}{\Gamma \vdash \alpha.\mathbf{t} : \lfloor\alpha\rfloor.T} \qquad \lfloor\alpha\rfloor.T = \underbrace{T + \cdots + T}_{\lfloor\alpha\rfloor}$$

- More general than Additive
- Less complex than Vectorial
- "Amounts" approximated

$$\text{If } \vdash \mathbf{t} : T, \quad \text{then } \vdash (0.9).\mathbf{t} + (1.1).\mathbf{t} : T$$
$$(0.9).\mathbf{t} + (1.1).\mathbf{t} \to 2.\mathbf{t} \quad \text{and} \quad \vdash 2.\mathbf{t} : 2.T$$

Weak subject reduction: $\mathbf{t} \to \mathbf{r},\ \Gamma \vdash \mathbf{t} : T \Rightarrow \Gamma \vdash \mathbf{r} : R$ with $T \preceq R$

Abstract interpretation (theorem)

$$\begin{array}{ccccc}
\lambda^{\mathrm{CA}} & \xrightarrow{\tau} & \lambda_{add} & \xrightarrow{[\cdot]} & F_p \\
\downarrow & & \downarrow_a\ (\preceq) & & \downarrow_F\ (\preceq) \\
\lambda^{\mathrm{CA}} & \xrightarrow[\tau]{} & \lambda_{add} & \xrightarrow[{[\cdot]}]{} & F_p
\end{array}$$

The *Complete Additive* System (λ^{CA})

Extending sums to the whole calculus (with positive reals scalars)

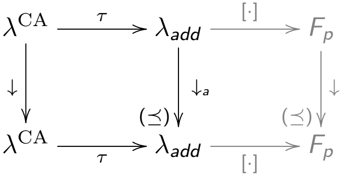
$$\frac{\Gamma \vdash \mathbf{t} : T}{\Gamma \vdash \alpha.\mathbf{t} : \lfloor\alpha\rfloor.T = \underbrace{T + \cdots + T}_{\lfloor\alpha\rfloor}}$$

- More general than Additive
- Less complex than Vectorial
- “Amounts” approximated

$$\text{If } \vdash \mathbf{t} : T, \quad \text{then } \vdash (0.9).\mathbf{t} + (1.1).\mathbf{t} : T$$
$$(0.9).\mathbf{t} + (1.1).\mathbf{t} \to 2.\mathbf{t} \quad \text{and} \quad \vdash 2.\mathbf{t} : 2.T$$

Weak subject reduction: $\mathbf{t} \to \mathbf{r}$, $\Gamma \vdash \mathbf{t} : T \Rightarrow \Gamma \vdash \mathbf{r} : R$ with $T \preceq R$

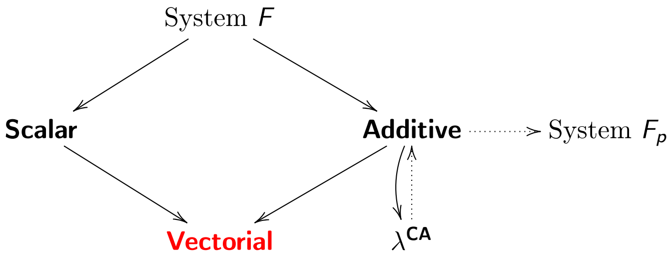
Abstract interpretation (theorem)



Strong normalisation (using *Additive*)

Contribution: `[Buiras,Díaz-Caro,Jaskelioff'11]`

Plan


System F
Scalar
Additive
System F_p
Vectorial
λ^{CA}

The *Vectorial* system

Types:

$$T, R, S := U \mid T + R \mid \alpha.T$$
$$U, V, W := X \mid U \to T \mid \forall X.U$$

(U, V, W reflect the basis terms)

Equivalences:

$$\begin{array}{rcl} 1.T & \equiv & T \\ \alpha.(\beta.T) & \equiv & (\alpha \times \beta).T \\ \alpha.T + \alpha.R & \equiv & \alpha.(T + R) \\ \alpha.T + \beta.T & \equiv & (\alpha + \beta).T \\ T + R & \equiv & R + T \\ T + (R + S) & \equiv & (T + R) + S \end{array}$$

(reflect the vectorial spaces axioms)

Typing rules

$$\frac{}{\Gamma, x : U \vdash x : U}\ ax \qquad \frac{\Gamma \vdash \mathbf{t} : T}{\Gamma \vdash 0 : 0.T}\ 0_I \qquad \frac{\Gamma \vdash \mathbf{t} : T}{\Gamma \vdash \alpha.\mathbf{t} : \alpha.T}\ s_I$$

$$\frac{\Gamma \vdash \mathbf{t}\colon \sum_{i=1}^{n} \alpha_i.\forall \vec{X}.(U \to T_i) \qquad \Gamma \vdash \mathbf{r}\colon \sum_{j=1}^{m} \beta_j.V_j \qquad \begin{array}{c}\forall V_j, \exists \vec{W}_j\ / \\ U[\vec{W}_j/\vec{X}] = V_j\end{array}}{\Gamma \vdash (\mathbf{t})\ \mathbf{r}\colon \sum_{i=1}^{n}\sum_{j=1}^{m} \alpha_i \times \beta_j.T_i[\vec{W}_j/\vec{X}]}\ \to_E$$

$$\frac{\Gamma, x : U \vdash \mathbf{t} : T}{\Gamma \vdash \lambda x.\mathbf{t} : U \to T}\ \to_I \qquad \frac{\Gamma \vdash \mathbf{t} : T \qquad \Gamma \vdash \mathbf{r} : R}{\Gamma \vdash \mathbf{t} + \mathbf{r} : T + R}\ +_I$$

$$\frac{\Gamma \vdash \mathbf{t} : \sum_{i=1}^{n} \alpha_i.U_i \qquad X \notin FV(\Gamma)}{\Gamma \vdash \mathbf{t} : \sum_{i=1}^{n} \alpha_i.\forall X.U_i}\ \forall_I \qquad \frac{\Gamma \vdash \mathbf{t} : \sum_{i=1}^{n} \alpha_i.\forall X.U_i}{\Gamma \vdash \mathbf{t} : \sum_{i=1}^{n} \alpha_i.U_i[V/X]}\ \forall_E$$

Typing rules

$$\frac{}{\Gamma, x : U \vdash x : U}\ ax \qquad \frac{\Gamma \vdash \mathbf{t} : T}{\Gamma \vdash 0 : 0.T}\ 0_I \qquad \frac{\Gamma \vdash \mathbf{t} : T}{\Gamma \vdash \alpha.\mathbf{t} : \alpha.T}\ s_I$$

$$\frac{\Gamma \vdash \mathbf{t} : \sum_{i=1}^{n} \alpha_i.\forall \vec{X}.(U \to T_i) \qquad \Gamma \vdash \mathbf{r} : \sum_{j=1}^{m} \beta_j.V_j \qquad \begin{smallmatrix}\forall V_j, \exists \vec{W}_j \ / \\ U[\vec{W}_j/\vec{X}] = V_j\end{smallmatrix}}{\Gamma \vdash (\mathbf{t})\ \mathbf{r} : \sum_{i=1}^{n}\sum_{j=1}^{m} \alpha_i \times \beta_j.T_i[\vec{W}_j/\vec{X}]}\ \to_E$$

$$\frac{\Gamma, x : U \vdash \mathbf{t} : T}{\Gamma \vdash \lambda x.\mathbf{t} : U \to T}\ \to_I \qquad \frac{\Gamma \vdash \mathbf{t} : T \qquad \Gamma \vdash \mathbf{r} : R}{\Gamma \vdash \mathbf{t} + \mathbf{r} : T + R}\ +_I$$

$$\frac{\Gamma \vdash \mathbf{t} : \sum_{i=1}^{n} \alpha_i.U_i \qquad X \notin FV(\Gamma)}{\Gamma \vdash \mathbf{t} : \sum_{i=1}^{n} \alpha_i.\forall X.U_i}\ \forall_I \qquad \frac{\Gamma \vdash \mathbf{t} : \sum_{i=1}^{n} \alpha_i.\forall X.U_i}{\Gamma \vdash \mathbf{t} : \sum_{i=1}^{n} \alpha_i.U_i[V/X]}\ \forall_E$$

Strong normalisation: Reducibility candidates ✓
Main difficulty: show that $\{\mathbf{t}_i\}_i$ SN $\Rightarrow \sum_i \mathbf{t}_i$ SN (algebraic measure)

Typing rules

$$\frac{}{\Gamma, x : U \vdash x : U}\ ax \qquad \frac{\Gamma \vdash \mathbf{t} : T}{\Gamma \vdash 0 : 0.T}\ 0_I \qquad \frac{\Gamma \vdash \mathbf{t} : T}{\Gamma \vdash \alpha.\mathbf{t} : \alpha.T}\ s_I$$

$$\frac{\Gamma \vdash \mathbf{t} : \sum_{i=1}^{n} \alpha_i.\forall \vec{X}.(U \to T_i) \qquad \Gamma \vdash \mathbf{r} : \sum_{j=1}^{m} \beta_j.V_j \qquad {\scriptstyle \forall V_j, \exists \vec{W}_j \,/\, U[\vec{W}_j/\vec{X}] = V_j}}{\Gamma \vdash (\mathbf{t})\ \mathbf{r} : \sum_{i=1}^{n}\sum_{j=1}^{m} \alpha_i \times \beta_j.T_i[\vec{W}_j/\vec{X}]}\ \to_E$$

$$\frac{\Gamma, x : U \vdash \mathbf{t} : T}{\Gamma \vdash \lambda x.\mathbf{t} : U \to T}\ \to_I \qquad \frac{\Gamma \vdash \mathbf{t} : T \qquad \Gamma \vdash \mathbf{r} : R}{\Gamma \vdash \mathbf{t} + \mathbf{r} : T + R}\ +_I$$

$$\frac{\Gamma \vdash \mathbf{t} : \sum_{i=1}^{n} \alpha_i.U_i \qquad X \notin FV(\Gamma)}{\Gamma \vdash \mathbf{t} : \sum_{i=1}^{n} \alpha_i.\forall X.U_i}\ \forall_I \qquad \frac{\Gamma \vdash \mathbf{t} : \sum_{i=1}^{n} \alpha_i.\forall X.U_i}{\Gamma \vdash \mathbf{t} : \sum_{i=1}^{n} \alpha_i.U_i[V/X]}\ \forall_E$$

Strong normalisation: Reducibility candidates ✓
Main difficulty: show that $\{\mathbf{t}_i\}_i$ SN $\Rightarrow \sum_i \mathbf{t}_i$ SN (algebraic measure)
Subject reduction **a challenge**

The case of the factorisation rule

System F à la Curry: a term can have different, unrelated types

$$\frac{\Gamma \vdash \mathbf{t} : T \qquad \Gamma \vdash \mathbf{t} : T'}{\Gamma \vdash \alpha.\mathbf{t} + \beta.\mathbf{t} : \alpha.T + \beta.T'}$$

However, $\alpha.\mathbf{t} + \beta.\mathbf{t} \to (\alpha + \beta).\mathbf{t}$... one of the two types must be chosen!

In general $\alpha.T + \beta.T' \neq (\alpha + \beta).T \neq (\alpha + \beta).T'$

(and since we are working in System F, there is no principal types neither)

Several possible solutions:

- Remove factorisation rule (Done. SR and SN both work)
  - $+$ in scalars not used anymore. Scalars $\Rightarrow$ Monoid
  - It works!... but it is no so expressive ("vectorial" structure lost)

Several possible solutions:

- Remove factorisation rule (Done. SR and SN both work)
  - $+$ in scalars not used anymore. Scalars $\Rightarrow$ Monoid
  - It works!... but it is no so expressive ("vectorial" structure lost)
- Add the typing rule $$\frac{\Gamma \vdash \mathbf{t} : T \qquad \Gamma \vdash \mathbf{t} : T'}{\Gamma \vdash (\alpha + \beta).\mathbf{t} : \alpha.T + \beta.T'}$$
  - As soon as we add this one, we have to add many others
  - Too complex and inelegant (subject reduction by axiom)

Several possible solutions:

- Remove factorisation rule (Done. SR and SN both work)
  - $+$ in scalars not used anymore. Scalars $\Rightarrow$ Monoid
  - It works!... but it is no so expressive ("vectorial" structure lost)
- Add the typing rule $$\frac{\Gamma \vdash \mathbf{t} : T \qquad \Gamma \vdash \mathbf{t} : T'}{\Gamma \vdash (\alpha + \beta).\mathbf{t} : \alpha.T + \beta.T'}$$
  - As soon as we add this one, we have to add many others
  - Too complex and inelegant (subject reduction by axiom)
- Weak subject reduction
  - If $\Gamma \vdash \mathbf{t} : T$ and $\mathbf{t} \to_R \mathbf{r}$, then
    - if R is not the factorisation rule: $\Gamma \vdash \mathbf{r} : T$
    - if R is the factorisation rule: $\exists S \sqsubseteq T \;/\; \Gamma \vdash \mathbf{r} : S$

  where $(\alpha + \beta).T \sqsubseteq \alpha.T + \beta.T'$ if $\exists \mathbf{t} \;/\; \Gamma \vdash \mathbf{t} : T$ and $\Gamma \vdash \mathbf{t} : T'$

Contribution: `[Arrighi,Díaz-Caro,Valiron'11]`

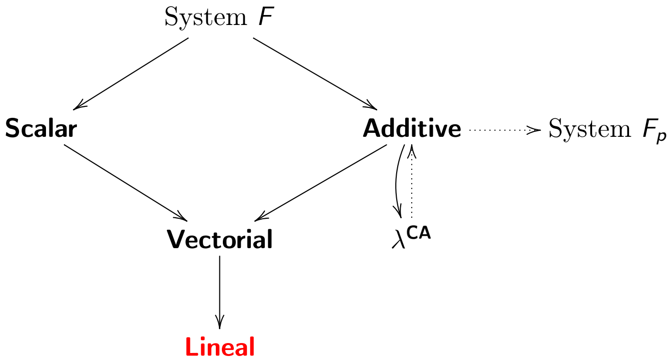
Several possible solutions:

- Remove factorisation rule (Done. SR and SN both work)
  - $+$ in scalars not used anymore. Scalars $\Rightarrow$ Monoid
  - It works!... but it is no so expressive ("vectorial" structure lost)
- Add the typing rule $$\frac{\Gamma \vdash \mathbf{t} : T \qquad \Gamma \vdash \mathbf{t} : T'}{\Gamma \vdash (\alpha + \beta).\mathbf{t} : \alpha.T + \beta.T'}$$
  - As soon as we add this one, we have to add many others
  - Too complex and inelegant (subject reduction by axiom)
- Weak subject reduction
  - If $\Gamma \vdash \mathbf{t} : T$ and $\mathbf{t} \to_R \mathbf{r}$, then
    - if R is not the factorisation rule: $\Gamma \vdash \mathbf{r} : T$
    - if R is the factorisation rule: $\exists S \sqsubseteq T \ / \ \Gamma \vdash \mathbf{r} : S$

  where $(\alpha + \beta).T \sqsubseteq \alpha.T + \beta.T'$ if $\exists \mathbf{t} \ / \ \Gamma \vdash \mathbf{t} : T$ and $\Gamma \vdash \mathbf{t} : T'$

  Contribution: `[Arrighi,Díaz-Caro,Valiron'11]`
- Church style
  - Seems to be the *natural* solution: the type is part of the term, if the types are different, the terms are different (no factorisation rule)

Plan


System F
Scalar
Additive
System F_p
Vectorial
λ^{CA}
Lineal

The system *Lineal*

Types:

$$T, R, S := U \mid T + R \mid \alpha.T$$
$$U, V, W := X \mid U \to T \mid \forall X.U \mid U@(\textstyle\sum_i V_i)$$

(U, V, W reflect the basis terms)

Equivalences:

$$\begin{array}{rcl}
1.T & \equiv & T \\
\alpha.(\beta.T) & \equiv & (\alpha \times \beta).T \\
\alpha.T + \alpha.R & \equiv & \alpha.(T + R) \\
\alpha.T + \beta.T & \equiv & (\alpha + \beta).T \\
T + R & \equiv & R + T \\
T + (R + S) & \equiv & (T + R) + S \\
(\forall X.U)@V & \equiv & U[V/X]
\end{array}$$

(reflect the vectorial spaces axioms)

Typing rules

$$\frac{}{\Gamma, x : U \vdash x : U}\ ax \qquad \frac{\Gamma \vdash \mathbf{t} : T}{\Gamma \vdash 0 : 0.T}\ 0_I \qquad \frac{\Gamma \vdash \mathbf{t} : T}{\Gamma \vdash \alpha.\mathbf{t} : \alpha.T}\ s_I$$

$$\frac{\Gamma \vdash \mathbf{t} : \sum_{i=1}^{n} \alpha_i.(\langle \forall X \rangle_k.(U \to T_i))@\langle \sum_{j=1}^{m+\delta} W_j \rangle_k \quad \Gamma \vdash \mathbf{r} : \sum_{j=1}^{m} \beta_j.V_j \quad \begin{array}{c}\forall V_j, \exists j_1, \dots, j_k \ / \\ U\langle [W_j/X] \rangle_k = V_j\end{array}}{\Gamma \vdash (\mathbf{t})\ \mathbf{r} : \sum_{i=1}^{n} \sum_{j=1}^{m} \alpha_i \times \beta_j.T_i \langle [W_j/X] \rangle_k}\ \to_E$$

$$\frac{\Gamma, x : U \vdash \mathbf{t} : T}{\Gamma \vdash \lambda x : U.\mathbf{t} : U \to T}\ \to_I \qquad \frac{\Gamma \vdash \mathbf{t} : T \qquad \Gamma \vdash \mathbf{r} : R}{\Gamma \vdash \mathbf{t} + \mathbf{r} : T + R}\ +_I$$

$$\frac{\Gamma \vdash \mathbf{t} : \sum_{i=1}^{n} \alpha_i.U_i \quad X \notin FV(\Gamma)}{\Gamma \vdash \Lambda X.\mathbf{t} : \sum_{i=1}^{n} \alpha_i.\forall X.U_i}\ \forall_I \qquad \frac{\Gamma \vdash \mathbf{t} : \sum_{i=1}^{n} \alpha_i.\forall X.U_i}{\Gamma \vdash \mathbf{t}@(\sum_{j=1}^{m} V_j) : \sum_{i=1}^{n} \alpha_i.(\forall X.U_i)@(\sum_{j=1}^{m} V_j)}\ @_I$$

Subject reduction ✓

Strong normalisation (using Vectorial) ✓

Most important properties of *Lineal*

Theorem

If $\Gamma \vdash \boldsymbol{t} \colon \sum_i \alpha_i.U_i$ *then* $\boldsymbol{t} \to^* \sum_i \alpha_i.\boldsymbol{b}_i$ *where* $\Gamma \vdash \boldsymbol{b}_i : U_i$

(where U_i is not a type abstraction or application)

Theorem

If $\boldsymbol{t} \downarrow = \sum_i \alpha_i.\boldsymbol{b}_i$ *then* $\Gamma \vdash \boldsymbol{t} \colon \sum_i \alpha_i.U_i + 0.T,$ *where* $\Gamma \vdash \boldsymbol{b}_i : U_i$

Confluence as a side effect

Confluence

In the original untyped setting: “confluence by restrictions”:

$$Y_{\mathbf{b}} = (\lambda x.(\mathbf{b} + (x)x))\ \lambda x.(\mathbf{b} + (x)x)$$

$$Y_{\mathbf{b}} \to \mathbf{b} + Y_{\mathbf{b}} \to \mathbf{b} + \mathbf{b} + Y_{\mathbf{b}} \to \dots$$

Confluence

In the original untyped setting: "confluence by restrictions":

$$Y_{\mathbf{b}} = (\lambda x.(\mathbf{b} + (x)x))\ \lambda x.(\mathbf{b} + (x)x)$$

$$Y_{\mathbf{b}} \to \mathbf{b} + Y_{\mathbf{b}} \to \mathbf{b} + \mathbf{b} + Y_{\mathbf{b}} \to \dots$$

$$\begin{array}{ccc} Y_{\mathbf{b}} + (-1).Y_{\mathbf{b}} & \longrightarrow (1-1).Y_{\mathbf{b}} & \longrightarrow^* 0 \\ \downarrow & & \\ \mathbf{b} + Y_{\mathbf{b}} + (-1).Y_{\mathbf{b}} & & \\ \downarrow_* & & \\ \mathbf{b} & & \end{array}$$

Confluence

In the original untyped setting: "confluence by restrictions":

$$Y_{\mathbf{b}} = (\lambda x.(\mathbf{b} + (x)x))\ \lambda x.(\mathbf{b} + (x)x)$$

$$Y_{\mathbf{b}} \to \mathbf{b} + Y_{\mathbf{b}} \to \mathbf{b} + \mathbf{b} + Y_{\mathbf{b}} \to \dots$$

$$\begin{array}{ccccc} Y_{\mathbf{b}} + (-1).Y_{\mathbf{b}} & \longrightarrow & (1-1).Y_{\mathbf{b}} & \longrightarrow^* & 0 \\ \downarrow & & & & \\ \mathbf{b} + Y_{\mathbf{b}} + (-1).Y_{\mathbf{b}} & & & & \\ \downarrow_* & & & & \\ \mathbf{b} & & & & \end{array}$$

Solution in the untyped setting:

$\alpha.\mathbf{t} + \beta.\mathbf{t} \to (\alpha + \beta).\mathbf{t}$

only if $\mathbf{t}$ is closed-normal

Confluence

In the original untyped setting: "confluence by restrictions":

$$Y_{\mathbf{b}} = (\lambda x.(\mathbf{b} + (x)x))\ \lambda x.(\mathbf{b} + (x)x)$$

$$Y_{\mathbf{b}} \to \mathbf{b} + Y_{\mathbf{b}} \to \mathbf{b} + \mathbf{b} + Y_{\mathbf{b}} \to \dots$$

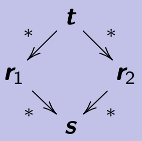
$$\begin{array}{ccccc} Y_{\mathbf{b}} + (-1).Y_{\mathbf{b}} & \longrightarrow & (1-1).Y_{\mathbf{b}} & \longrightarrow^* & 0 \\ \downarrow & & & & \\ \mathbf{b} + Y_{\mathbf{b}} + (-1).Y_{\mathbf{b}} & & & & \\ \downarrow_* & & & & \\ \mathbf{b} & & & & \end{array}$$

Solution in the untyped setting:
$\alpha.\mathbf{t} + \beta.\mathbf{t} \to (\alpha + \beta).\mathbf{t}$
only if $\mathbf{t}$ is closed-normal

In the typed setting: **Strong normalisation solves the problem**

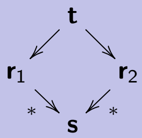
Theorem (Confluence)

$\forall \boldsymbol{t} \;/\; \Gamma \vdash \boldsymbol{t} : T$



Proof.

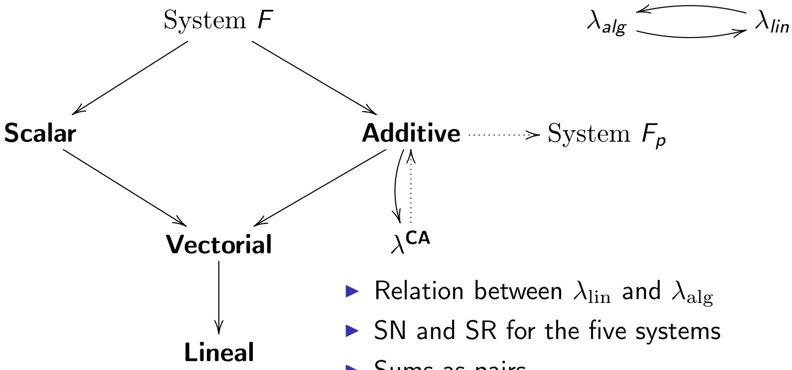
1) **local confluence**:



- Algebraic fragment: Coq proof `[Valiron'10]`
- Beta-reduction: Straightforward extension
- Commutation: Induction

2) Local confluence + Strong normalisation $\Rightarrow$ Confluence □

Contributions



- Relation between $\lambda_{\rm lin}$ and $\lambda_{\rm alg}$
- SN and SR for the five systems
- Sums as pairs
- Types $\leftrightarrow$ vectorial structure of terms
- Extra: No cloning theorem

Papers
Díaz-Caro,Perdrix,Tasson,Valiron HOR'10 (journal version in preparation)
Arrighi,Díaz-Caro QPL'09 (journal version submitted)
Díaz-Caro,Petit (in preparation)
Buiras,Díaz-Caro,Jaskelioff LSFA'11
Arrighi,Díaz-Caro,Valiron DCM'11 (journal version in preparation)

Future work

- Invariability of models of λ_{alg} through the CPS simulation
- Differential λ-calculus $\leftrightarrow$ Linear-algebraic λ-calculus
- Algebraic Linearity $\leftrightarrow$ Linear logic resources
- Quantum language (orthogonality issues)
- Relations with Probabilistic/Quantum/Fuzzy Logics