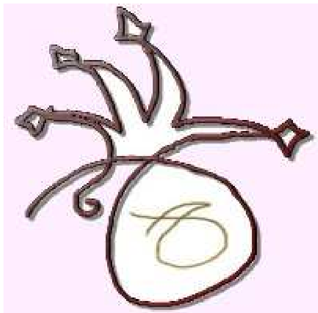
Follia project: “Logical foundations of abstract programming languages”

Soft Linear Logic, Lambda-Calculus and Intersection Types



Simona Ronchi Della Rocca, Marco Gaboardi

Dipartimento di Informatica - Università degli Studi di Torino

Outline

- Soft Linear Logic and λ-calculus
  - Our aim: a type assignment system assigning to λ-calculus types of Soft Linear Logic
  - The problem: subject reduction
  - Our solution: a linear term construction
- Soft Linear Logic and Intersection Types
  - Replacing !-modality by intersection
  - A side effect result: a nice natural deduction version of the type assigment
- Intersection Types
  - Structural complexity of λ-calculus

Motivations

Light Logics (SLL, LAL, EAL) are variants of Linear Logic where it is possible to characterize some complexity classes, in the following sense:

- the programming language is the proof-nets language
- the computational model is the proof-net normalization procedure
- the time complexity measure is the number of cut-elimination steps
- the time complexity measure is expressed as a function of the proof-net structure

Motivations

We want to use λ-calculus as programming language , and β-reduction as evaluation . So our aim is to design a type assignment system assigning to λ-terms light formulas, in such a way that:

- the computational model is the λ-calculus plus the β-reduction
- the time complexity measure is the number of β-reduction steps
- the typing supplies an upper bound to time complexity measure, which is expressed as a function of both the term and the type derivation structure.

Some problems

Light Logics are presented as sequent calculi, while λ-calculus is naturally related (through the C-H isomorphism) to logics in natural deduction style, but:

Problem 1

There is no a standard way of designing a natural deduction version of a logic in sequent calculus formulation in presence of a modality.

(some proposals: for EAL by (Coppola DalLago RdR), for LAL by (Baillot,Terui)) .

Some Problems

A type assignment system for λ-calculus can be designed as sequent calculus decoration, but in general in such systems, in presence of a modality:

Problem 2

Subject reduction property does not hold.

Problem 3

There is a mismatch between β-reduction and cut-elimination.

Lafont's SLL

SLL can be seen as a subsystem of BLL which is powerful enough to encode polynomial time.

Let us consider SLL restricted to the connectives $\multimap$ and !.

$$\frac{}{A \vdash A}\ (Id) \qquad \frac{\Gamma \vdash A \qquad \Delta, A \vdash B}{\Gamma, \Delta \vdash B}\ (cut)$$

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \multimap B}\ (\multimap R) \qquad \frac{\Gamma \vdash A \qquad B, \Delta \vdash C}{A \multimap B, \Gamma, \Delta \vdash C}\ (\multimap L)$$

$$\frac{\Gamma, \overbrace{A, \ldots, A}^{n \text{ times}} \vdash C}{\Gamma, !A \vdash C}\ (mpx) \qquad \frac{\Gamma \vdash A}{!\Gamma \vdash !A}\ (sp)$$

If $\Pi : \Gamma \vdash \sigma$ then the proof-net corresponding to Π normalizes in a number of steps $\leq |\Pi| \times n^d$ (n is the maximum rank of a multiplexor in Π and d is the maximum nesting of boxes).

A decoration for SLL

A “quite standard” decoration for SLL [MairsonTerui03] is the following:

$$\frac{}{x : A \vdash_L x : A}\ (Id) \qquad \frac{\Gamma, x : A \vdash_L M : B}{\Gamma \vdash_L \lambda x.M : A \multimap B}\ (\multimap R)$$

$$\frac{\Gamma \vdash_L M : A \quad x : B, \Delta \vdash_L N : C \quad \text{dom}(\Gamma) \cap \text{dom}(\Delta) = \emptyset \quad y \text{ fresh}}{y : A \multimap B, \Gamma, \Delta \vdash_L N[yM/x] : C}\ (\multimap L)$$

$$\frac{\Gamma \vdash_L M : A \qquad \Delta, x : A \vdash_L N : B \qquad \text{dom}(\Gamma) \cap \text{dom}(\Delta) = \emptyset}{\Gamma, \Delta \vdash_L N[M/x] : B}\ (cut)$$

$$\frac{\Gamma, x_1 : A, \ldots, x_n : A \vdash_L M : C}{\Gamma, z :!A \vdash_L M[z/x_1, \cdots, z/x_n] : C}\ (mpx) \qquad \frac{\Gamma \vdash_L M : A}{!\Gamma \vdash_L M :!A}\ (sp)$$

Problem: subject reduction

For the above system subject reduction is not valid. In fact, we can derive the following

$$\dfrac{\dfrac{}{s : B \multimap !A, w : B \vdash_L (\lambda z.sz)w : !A}\ (cut) \quad \dfrac{\dfrac{}{y : A \multimap A \multimap B, z : A, m : A \vdash_L yzm : B}}{y : A \multimap A \multimap B, x : !A \vdash_L yxx : B}\ (mpx)}{y : A \multimap A \multimap B, s : B \multimap !A, w : B \vdash_L yxx[(\lambda z.sz)w/x] \equiv y((\lambda z.sz)w)((\lambda z.sz)w) : B}\ (cut)$$

$$y((\lambda z.sz)w)((\lambda z.sz)w) \to_\beta y((\lambda z.sz)w)(sw)$$

but

$$y : A \multimap A \multimap B, s : B \multimap !A, w : B \not\vdash_L y((\lambda z.sz)w)(sw) : B$$

This phenomenon arises since to $(\lambda z.sz)w$ a modal type $!A$ is assigned from a not modal context, so the term is not duplicable ($(\lambda z.sz)w$ occurs twice in the term but once on the proof with a not duplicable proof).

Problem: reduction size

A mismatch between proof normalization and β-reduction [Baillot,Terui]. The following is derivable:

$$z :!A, y_1 :!A \multimap !A \multimap !A, ..., y_n :!A \multimap !A \multimap !A \vdash_L (\lambda x.y_1 xx)(...((\lambda x.y_n xx)z))...) :!A$$

by a proof Π which is linear in the size of the term.

But this term normalizes in a number of β-reductions which is $O(2^n)$!!!

Working with proof-nets instead than with λ-terms generates gives no such problem, since the dimension of Π is linear in n, and it normalizes in time linear in n.
Note that by a (m) rule from the statement before we can derive:

$$z :!A, y :!A \multimap !A \multimap !A \vdash_L (\lambda x.yxx)^n :!A$$

Linear Substitution

The problem of subject reduction we have stressed above is related to some well known facts about linear logic and sequent calculus:

> sequent calculus offers different ways to construct a term. E.g., the term $\lambda x.xMM$ can be built either as $\lambda x.xy_1y_2[M/y_1][M/y_2]$ or as $\lambda x.xyy[M/y]$, e.i, the subterm M occurs twice in the term, but it can occur either once or twice in the proof. In the presence of !, controlling duplication, the former situation can break the subject reduction.

The way we propose to overcome to this problems is by limiting the way of construction of terms to the ones which use only linear substitutions. This corresponds to restrict some rules:

$$\frac{\Gamma \vdash M : A \qquad \Delta, x : A \vdash N : C \qquad \text{dom}(\Gamma) \cap \text{dom}(\Delta) = \emptyset \qquad x \text{ occurs once in } N}{\Gamma, \Delta \vdash N[M/x] : C} \ (cut)$$

$$\frac{\Gamma \vdash M : B \qquad \Delta, x : A \vdash N : C \qquad \text{dom}(\Gamma) \cap \text{dom}(\Delta) = \emptyset \quad y \text{ fresh} \quad y \text{ occurs once in } N}{\Gamma, y : B \multimap A, \Delta \vdash N[yM/x] : C} \ (\multimap L)$$

The above side conditions in fact can be internalized and we do this in what follows.

Soft Type Assignment System

We can define the set $\mathbf{T}$ of Types as:

$$
\begin{array}{lll}
A & ::= & a \mid \sigma \multimap A \quad \text{(Linear Types)} \\
\sigma & ::= & A \mid !\sigma
\end{array}
$$

and the Soft Type Assignment System by the following rules:

$$
\frac{}{x : A \vdash_T x : A}\ (Id) \qquad \frac{\Gamma, x : \sigma \vdash_T M : A}{\Gamma \vdash_T \lambda x.M : \sigma \multimap A}\ (\multimap R)
$$

$$
\frac{\Gamma \vdash_T M : \tau \qquad x : A, \Delta \vdash_T N : \rho \qquad \mathsf{dom}(\Gamma) \cap \mathsf{dom}(\Delta) = \emptyset \qquad y \text{ fresh}}{\Gamma, y : \tau \multimap A, \Delta \vdash_T N[yM/x] : \rho}\ (\multimap L)
$$

$$
\frac{\Gamma \vdash_T M : A \qquad \Delta, x : A \vdash_T N : \sigma \qquad \mathsf{dom}(\Gamma) \cap \mathsf{dom}(\Delta) = \emptyset}{\Gamma, \Delta \vdash_T N[M/x] : \sigma}\ (cut)
$$

$$
\frac{\Gamma \vdash_T M : \sigma}{!\Gamma \vdash_T M : !\sigma}\ (!) \qquad \frac{\Gamma, x_1 : \tau, \ldots, x_n : \tau \vdash_T M : \sigma}{\Gamma, x : !\tau \vdash_T M[x/x_1, \ldots, x/x_n] : \sigma}\ (m) \qquad \frac{\Gamma \vdash_T M : \sigma}{\Gamma, x : A \vdash_T M : \sigma}\ (weak)
$$

Linearity Properties of $\vdash_T$

- $\Gamma \vdash_T M : \sigma$ and $x : A \in \Gamma$ imply x occurs at most once in M;
- $\Pi : !\Gamma \vdash_T M : !\sigma$ implies Π can be tranformed into a derivation Π':

$$\frac{\Gamma \vdash_T M : \sigma}{!\Gamma \vdash_T M : !\sigma}\ (!)$$

So the modality ! is truly a witness of the possibility of duplication!

Subject Reduction for $\vdash_T$

Theorem 1 (Subject Reduction) $\Gamma \vdash_T M : \mu$ *and* $M \rightarrow_\beta M'$ *imply* $\Gamma \vdash_T M' : \mu$

Proof idea. By induction on the derivation, clearly the difficult case is

$$\dfrac{\dfrac{\Gamma, y : \sigma \vdash_T P : B}{\Gamma \vdash_T \lambda y.P : \sigma \multimap B}(\multimap R) \qquad \dfrac{\Theta \vdash_T Q : \sigma \qquad \Delta, z : B \vdash_T N\{z\} : \mu}{\Theta, \Delta, x : \sigma \multimap B \vdash_T N\{xQ\} : \mu}(\multimap L)}{\Gamma, \Theta, \Delta \vdash_T N[\lambda y.P/x] \equiv N\{(\lambda y.P)Q\} : \mu}(cut)$$

The standard cut elimination procedure would produce the cut:

$$\dfrac{\Theta \vdash_T Q : \sigma \qquad \Gamma, y : \sigma \vdash_T P : B}{\Theta, \Gamma \vdash_T P[Q/y] : B}(cut)$$

if σ is a linear type, all is OK. Otherwise this cut is not available!

Subject Reduction for $\vdash_T$

Assume $\sigma \equiv !^q C$.

- by the linearity properties, the proof can be transformed:

$$\dfrac{\dfrac{\Gamma, y :!^q C \vdash_T P : B}{\Gamma \vdash_T \lambda y.P :!^q C \multimap B}\ (\multimap R) \qquad \dfrac{\dfrac{\dfrac{\Theta' \vdash_T Q : C}{\Theta \vdash_T Q :!^q C}\ (!)^q(weak) \qquad \Delta, z : B \vdash_T N\{z\} : \mu}{\Theta, \Delta, x : \sigma \multimap B \vdash_T N\{xQ\} : \mu}\ (\multimap L)}{}}{\Gamma, \Theta, \Delta \vdash_T N[\lambda y.P/x] \equiv N\{(\lambda y.P)Q\} : \mu}\ (cut)$$

- there are $\geq q$ variables corresponding to y (introduced either by axiom or $(\multimap L)$ or $(weak)$)
- each one of such variables can be replaced by a disjoint copy of a subproof proving (an instance of) $\Theta' \vdash_T Q : C$
- all such instances can be unified through a multiplexor rule.

The reduction corresponds to a global proof transformation!

Comparing $\vdash_L$ and $\vdash_T$

- $\Gamma \vdash_T M : \sigma$ implies there are Γ', σ' such that $\Gamma' \vdash_L M : \sigma'$
- $\Gamma \vdash_L M : \sigma$ implies there is M', which is a partial linearization of M, and Γ', σ' such that $\Gamma' \vdash_T M : \sigma'$

Examples of partial linearization :

- $\lambda x_1 \ldots x_n . y x_1 \ldots x_n$ is a partial linearization of $\lambda x.y \underbrace{x \ldots x}_{n}$
- $\lambda x.yxxxzt$ is a partial linearization of $\lambda x.yxxxzz$

Examples:

- $y : A \multimap A \multimap B, s : B \multimap !A, w : B \vdash_L y((\lambda z.sz)w)((\lambda z.sz)w) : B$
  and
  $y : !A \multimap !A \multimap B, s : !(B \multimap A), w : !B \vdash_T y((\lambda z.sz)w)((\lambda z.sz)w) : B$
- $z : !A, y_1 : !A \multimap !A \multimap !A, \ldots, y_n : !A \multimap !A \multimap !A \vdash_L (\lambda x.y_1 xx)(\ldots((\lambda x.y_n xx)z))\ldots) : !A$
  and
  $z : !^n A, y_i : !^i (A \multimap A \multimap A)(1 \leq i \leq n) \vdash_T (\lambda x.y_1 xx)(\ldots((\lambda x.y_n xx)z))\ldots) : A$
- $(\lambda x.yxx)^n z$ can be typed in $\vdash_L$ but it cannot be typed in $\vdash_T$.

Complexity in $\vdash_T$

Lemma 1 *For all derivation Π in $\vdash_T$ with subject M,* $|M| \leq |\Pi|$

Theorem 2 *Let M be typable in $\vdash_T$ and let $M \rightarrow^m_\beta M'$. For all derivation Π in $\vdash_T$ with subject M,*

$$m \leq W(\Pi) \leq |M|^{d(\Pi)+1}$$

where $d(\Pi)$ is the maximal nesting of (!) rule in Π.

This result is based on the definition of weight of Π ($W(\Pi)$), measuring the possible growth of Π during the reduction. In particular:

$$W\left(\frac{\Pi' : \Gamma \vdash_T M : \sigma}{!\Gamma \vdash_T M : !\sigma}\ (!)\right) = n \times W(\Pi')$$

where n is the maximum rank of a multiplexor in Π.

Intersection Types

The intersection connective $\cap$, in a linear setting, splits naturally into a multiplicative version:

$$\frac{\Gamma \vdash M : A \quad \Delta \vdash M : B}{\Gamma \uplus \Delta \vdash M : A \cap B} \ (\cap R) \qquad \frac{\Gamma, x : A, y : B \vdash M : C}{\Gamma, z : A \cap B \vdash M[z/x, z/y] : C} \ (\cap L)$$

and an additive version:

$$\frac{\Gamma \vdash M : A \quad \Gamma \vdash M : B}{\Gamma \vdash M : A \,\&\, B} \ (\& R) \qquad \frac{\Gamma, x : A \vdash M : C}{\Gamma, x : A \,\&\, B \vdash M : C} \ (\& L)$$

Multiplicative intersection connective contains contraction modulo the equivalence

$$!A \cong !A \cap !A$$

So we will use multiplicative intersection types without idempotency!

Replacing ! by intersection

The set $\mathbf{T}$ of Intersection Types is defined as follows:

$$
\begin{array}{lcll}
A & ::= & a \mid \sigma \multimap A & \text{(strict types)} \\
\sigma & ::= & A \mid \sigma \cap \ldots \cap \sigma &
\end{array}
$$

The Soft Intersection Type Assignment System is composed by the following rules:

$$
\frac{}{x : A \vdash_I x : A}\ (Id) \qquad \frac{\Gamma, x : \sigma \vdash_I M : A}{\Gamma \vdash_I \lambda x.M : \sigma \multimap A}\ (\multimap R)
$$

$$
\frac{\Gamma \vdash_I M : \tau \qquad x : A, \Delta \vdash_I N : \rho \qquad \mathsf{dom}(\Gamma) \cap \mathsf{dom}(\Delta) = \emptyset \qquad y \text{ fresh}}{\Gamma, y : \tau \multimap A, \Delta \vdash_I N[yM/x] : \rho}\ (\multimap L)
$$

$$
\frac{\Gamma \vdash_I M : A \qquad \Delta, x : A \vdash_I N : \sigma \qquad \mathsf{dom}(\Gamma) \cap \mathsf{dom}(\Delta) = \emptyset}{\Gamma, \Delta \vdash_I N[M/x] : \sigma}\ (cut)
$$

$$
\frac{\Gamma_i \vdash_I M : \sigma \quad (1 \leq i \leq n)}{\uplus_i \Gamma_i \vdash_I M : \sigma \cap \ldots \cap \sigma}\ (\cap R)
$$

$$
\frac{\Gamma, x_1 : \sigma, \ldots, x_n : \sigma \vdash_I M : \rho}{\Gamma, z : \sigma \cap \ldots \cap \sigma \vdash_I M[z/x_1, \ldots, z/x_n] : \rho}\ (\cap L) \qquad \frac{\Gamma \vdash_I M : \sigma}{\Gamma, x : A \vdash_I M : \sigma}\ (weak)
$$

Complexity of $\vdash_I$

Lemma 2 *For all derivation* Π *in* $\vdash_I$ *with subject* M, $|M| \leq |\Pi|$

Theorem 3 *Let* M *be typable in* $\vdash_I$ *and let* $M \to_\beta^m M'$. *For all derivation* Π *in* $\vdash_I$ *with subject* M,

$$m \leq |M|^{d(\Pi)+1}$$

where $d(\Pi)$ *is the maximal nesting of* $(\cap R)$ *rule in* Π.

The definition of weight of Π ($W(\Pi)$) now can measure exactly the growth of Π during the reduction. In particular:

$$W\left(\frac{\Pi_i : \Gamma_i \vdash_I M : \sigma \quad (1 \leq i \leq n)}{\uplus_i \Gamma_i \vdash_I M : \sigma \cap \ldots \cap \sigma} \ (\cap R)\right) = \sum_{1 \leq i \leq n} W(\Pi_i)$$

A natural deduction for $\vdash_I$

Let $\mathfrak{A}$ and $\mathfrak{B}$ denote multisets of variable assignments.

$$\frac{}{x : A \vdash_N x : A}\ (Id) \qquad \frac{\mathfrak{A} \vdash_N M : \sigma \multimap A \quad \mathfrak{B} \vdash_N N : \sigma}{\mathfrak{A}, \mathfrak{B} \vdash_N MN : A}\ (\multimap E)$$

$$\frac{\mathfrak{A}, x : \sigma \cap \ldots \cap \sigma \vdash_N M : A \qquad x \notin \mathrm{dom}(\mathfrak{A})}{\mathfrak{A} \vdash_N \lambda x.M : \sigma \cap \ldots \cap \sigma \multimap A}\ (\multimap I)$$

$$\frac{\Gamma_i \vdash_N M : \sigma \quad (1 \leq i \leq n)}{\uplus_i \Gamma_i \vdash_N M : \sigma \cap \ldots \cap \sigma}\ (\cap I)$$

$$\frac{\mathfrak{A}, x : \sigma \ldots, x : \sigma \vdash_N M : \rho}{\mathfrak{A}, x : \sigma \cap \ldots \cap \sigma \vdash_N M : \rho}\ (m) \qquad \frac{\mathfrak{A} \vdash_N M : \sigma}{\mathfrak{A}, x : A \vdash_N M : \sigma}\ (weak)$$

$\Gamma \vdash_I M : \sigma$ if and only if $\mathfrak{A} \vdash_N M : \sigma$

where $\mathfrak{A} = \underbrace{x : \sigma, \ldots, x : \sigma}_{n}, \mathfrak{B}$ and $x \notin dom(\mathfrak{B})$ implies $\Gamma = \Gamma', x : \underbrace{\sigma \cap \ldots \cap \sigma}_{n}$

Extending the intersection

Let types be considered modulo commutativity of $\cap$.

$$\frac{}{x : A \vdash_N x : A}\ (Id) \qquad \frac{\mathfrak{A} \vdash_\cap M : \sigma \multimap A \quad \mathfrak{B} \vdash_\cap N : \sigma}{\mathfrak{A}, \mathfrak{B} \vdash_\cap MN : A}\ (\multimap E)$$

$$\frac{\mathfrak{A}, x : \sigma_1 \cap \ldots \cap \sigma_n \vdash_\cap M : A \qquad x \notin \mathrm{dom}(\mathfrak{A})}{\mathfrak{A} \vdash_\cap \lambda x.M : \sigma_1 \cap \ldots \cap \sigma_n \multimap A}\ (\multimap I)$$

$$\frac{\Gamma_i \vdash_\cap M : \sigma_i \quad (1 \leq i \leq n)}{\uplus_i \Gamma_i \vdash_\cap M : \sigma_1 \cap \ldots \cap \sigma_n}\ (\cap I)$$

$$\frac{\mathfrak{A}, x : \sigma_1 \ldots, x : \sigma_n \vdash_\cap M : \rho}{\mathfrak{A}, x : \sigma_1 \cap \ldots \cap \sigma_n \vdash_\cap M : \rho}\ (m) \qquad \frac{\mathfrak{A} \vdash_\cap M : \sigma}{\mathfrak{A}, x : A \vdash_\cap M : \sigma}\ (weak)$$

- M can be typed in $\vdash_\cap$ if and only if M is strongly normalizing
- Let M be typable in $\vdash_\cap$ and let $M \rightarrow_\beta^m M'$. For all derivation Π in $\vdash_\cap$ with subject M,

$$m \leq |M|^{d(\Pi)+1}$$

where $d(\Pi)$ is the maximal nesting of $(\cap I)$ rule in Π.

Redundancy

In order to derive:

$$z : ? \vdash_S (\lambda xy.yxxx)z : (A \multimap B \multimap C \multimap D) \multimap D$$

which type could be assigned to z?

Each of the following types:

$$z : (A \cap B \cap C)$$

$$z : (A \cap B) \cap C$$

$$z : A \cap (B \cap C)$$

$$\vdots$$

$$z : (C \cap B \cap A)$$

The simplest system

$$\frac{}{\mathfrak{A}, x : A \vdash_S x : A} \; (Id)$$

$$\frac{\mathfrak{A} \vdash_S M : B_1 \cap \ldots \cap B_n \multimap A \quad \mathfrak{B}_i \vdash_S N : B_i \quad (1 \leq i \leq n)}{\mathfrak{A}, \mathfrak{B}_1, \ldots, \mathfrak{B}_n \vdash_S MN : A} \; (\multimap E)$$

$$\frac{\mathfrak{A}, x : B_1, \ldots, x : B_n \vdash_S M : A \quad x \notin dom(\mathfrak{A})}{\mathfrak{A} \vdash_S \lambda x.M : B_1 \cap \ldots \cap B_n \multimap A} \; (\multimap I)$$

The system $\vdash_S$ gives type to all and only the strongly normalizing terms

$\vdash_S$ and structural complexity

Every type assignment derivation gives an upper-bound on both space and time complexity of the β-reduction.

In particular, the following properties hold:
Let $\Pi : \Gamma \vdash_S M : \sigma$.

- Let $M \rightarrow_\beta N$. Then there is $\Pi' : \Gamma \vdash_S N : \sigma$ and $|\Pi| > |\Pi'|$.
- Let $M \rightarrow_\beta^m N$. Then $m \leq |\Pi|$.
- Let $M_1 \rightarrow_\beta M_2 \rightarrow_\beta \ldots \rightarrow_\beta M_n \equiv fn(M)$. Then $|M_i| \leq |\Pi|$

where $|\Pi|$ is the dimension of Π, i.e., the number of used rules.

These results has been independently proved by Kazushige Terui.
Similar results have been proved by Mairson and Møller Neergaard, but they use particular proof-nets as model of computation.

Future work

- Polynomial completeness for $\vdash_T$
  - adding second order types
  - adding suitable constants to λ-calculus
- Polynomial completeness for $\vdash_\cap$
  - advantage with respect to $\vdash_T$: we can write more programs, so gaining in expressivity
  - problem: there is not a uniform encoding of integers
- More investigation on the structural complexity of λ-calculus