Linear logic
and higher-order model checking

Paul-André Melliès

CNRS & Université Paris Diderot

Abstraction and Verification in Semantics
Institut Henri Poincaré

23 → 27 June 2014

Purpose of this talk

I. Apply the ideas of linear logic to connect

- ▷ the **type-theoretic** account by Kobayashi & Ong
- ▷ the **domain-theoretic** account by Salvati & Walukiewicz

of higher-order model-checking.

II. Construct a cartesian-closed category $\mathscr{D}$ of coloured domains.

Very similar in spirit as Kazushige's talk of this morning

Higher-order recognizability

Suppose given a set $\mathscr{L}$ of Böhm trees of same type A.

Question:

When should one consider the set $\mathscr{L}$ as a recognizable language?

Higher-order recognizability

Suppose given a set $\mathscr{L}$ of Böhm trees of same type A.

Question:

When should one consider the set $\mathscr{L}$ as a recognizable language?

Tentative answer:

Use a finite domain interpretation of types.

Higher-order recognizability

Every finite domain D induces an interpretation of A as a finite domain:

$$\begin{array}{rcl} [\![o]\!] & := & D \\ [\![A \times B]\!] & := & [\![A]\!] \times [\![B]\!] \\ [\![A \to B]\!] & := & [\![A]\!] \to [\![B]\!] \end{array}$$

By continuity, every Böhm tree M of type A is interpreted as an element

$$[\![M]\!] \quad \in \quad [\![A]\!]$$

of the domain $[\![A]\!]$.

Higher-order recognizability

Now, every finite subset $\varphi \subseteq [\![A]\!]$ induces a set

$$\mathscr{L}_\varphi \quad = \quad \{\, M \mid [\![\, M \,]\!] \in \varphi \,\}$$

of Böhm trees of type A.

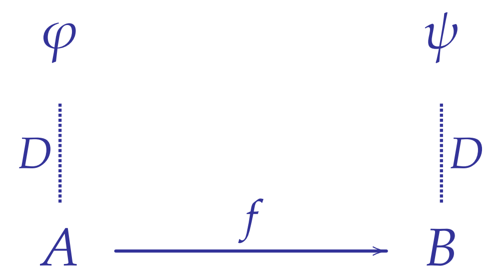
Notation: We write $\vDash M : \varphi$ to mean that $[\![M]\!] \in \varphi$.

Definition. [adapted from Salvati 2009]

A set of Böhm trees $\mathscr{L}$ is **recognizable** when it is of the form $\mathscr{L}_\varphi$.

Refinement types

Every such pair (D, φ) should be seen as a **predicate** over the type A.



Pullback operation:

Given a predicate $\psi \subseteq [\![B]\!]$ one defines the predicate

$$f^*(\psi) \quad := \quad \{ \; x \in [\![A]\!] \;\mid\; f(x) \in \psi \; \}$$

in such a way that

$$\models \; P : [\![M]\!]^*(\psi) \qquad \Longleftrightarrow \qquad \models \; MP : \psi$$

for every Böhm tree P of type A.

Refinement types

Every such pair (D, φ) should be seen as a **predicate** over the type A.

$$\begin{array}{ccc} \varphi & & \psi \\ D \vdots & & \vdots D \\ A & \xrightarrow{\;f\;} & B \end{array}$$

Pushforward operation:

Given a predicate $\varphi \subseteq [\![A]\!]$ one defines the predicate

$$f(\varphi) \quad := \quad \{\ f(x) \in [\![B]\!] \ \mid \ x \in \varphi\ \}$$

in such a way that

$$\vDash \ P : \varphi \quad \Rightarrow \quad \vDash \ MP : [\![M]\!](\varphi)$$

for every Böhm tree P of type A.

The Scott semantics of linear logic

Well-known principle.

Every preorder $(A, \leq)$ induces a domain $Domain(A)$ defined as follows:

- ▷ its elements are the ideals of the preorder,
- ▷ the ideals are ordered by inclusion.

Recall that a subset $X \subseteq A$ is called an ideal of the preorder A when

$$\forall a \in A, \ \forall x \in X, \quad a \leq x \Rightarrow a \in X.$$

The Scott semantics of linear logic

Key observation.

Suppose that the base type o is interpreted as the domain of ideals

$$[\![o]\!] \quad = \quad Domain(\,Q\,,\,\leq\,)$$

generated by a preorder Q of **atomic states**.

In that case, the interpretation of every type A is the domain of ideals

$$[\![\,A\,]\!] \quad := \quad Domain(\,Q_A\,,\,\leq_A\,)$$

generated by a specific preorder Q_A of **higher-order states**.

The Scott semantics of linear logic

A series of new connectives on preorders, such as:

$$A^{\perp} \quad := \quad A^{op}$$

$$A \,\&\, B \quad := \quad (A + B, \leq_A + \leq_B)$$

$$A \otimes B \quad := \quad (A \times B, \leq_A \times \leq_B)$$

$$!A \quad := \quad \wp_{fin}(A)$$

where the finite sets of elements of A are ordered as:

$$\{a_1, \ldots, a_p\} \leq_{!A} \{b_1, \ldots, b_q\} \quad \Longleftrightarrow \quad \forall i \in [p]\ \exists j \in [q]\ \ a_i \leq_A b_j$$

The Scott semantics of linear logic

Given a preorder of **atomic states** for the base type o

$$Q_o \quad = \quad (\, Q \,, \leq \,)$$

the preorder Q_A of **higher-order states** is defined by induction:

$$Q_{A \times B} \quad = \quad Q_A \;\&\; Q_B$$

$$Q_{A \to B} \quad = \quad !\, Q_A \multimap Q_B$$

In particular, a state of the simple type $A \rightarrow B$ is of the form

$$\{\, q_1, \ldots, q_n \,\} \multimap q$$

where $q_1, \ldots, q_n$ are states of A and q is a state of B.

What is a higher-order automaton?

Methodological question.

Given a simple type A, a finite preorder $(Q, \leq)$ and a subset

$$\varphi \subseteq [\![A]\!]$$

can we describe the Böhm trees of the associated language

$$\mathscr{L}_\varphi \quad = \quad \{ M \mid [\![M]\!] \in \varphi \} \quad = \quad \{ M \mid \models M : \varphi \}$$

in a more direct and automata-theoretic fashion ?

What is a higher-order automaton?

Methodological question.

Given a simple type A, a finite preorder $(Q, \leq)$ and an element

$$q \in Q_A$$

can we describe the Böhm trees of the associated language

$$\mathcal{L}_q \quad = \quad \{ M \mid q \in [\![M]\!] \}$$

in a more direct and automata-theoretic fashion ?

What is a higher-order automaton?

Definition. A higher-order automaton

$$\mathcal{A} = \langle \Sigma, Q, \delta, q_0 \rangle$$

consists of:

- ▷ a finite signature $\Sigma : Type \to Set$
- ▷ a finite set of states Q
- ▷ a family of transition functions $\delta_X : \Sigma_X \longrightarrow [\![X]\!]$
- ▷ a higher-order initial state $q_0 \in [\![A]\!]$

where the interpretation $[\![-]\!]$ of types is induced by the preorder $Q_o = Q$.

What is a higher-order automaton?

Suppose given a finite preorder $(Q, \leq)$.

Adequacy Theorem.

The interpretation of a Böhm tree M is the set of its accepting states.

In other words, for every higher-order state $q \in [\![A]\!]$,

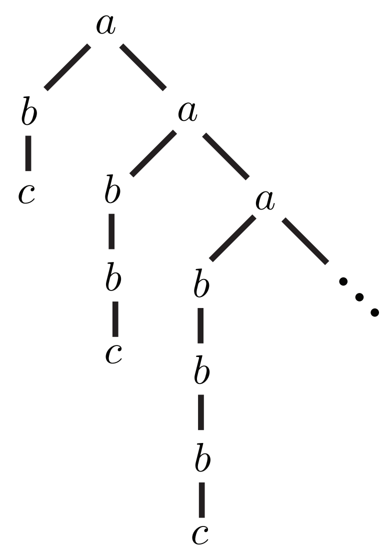
$$q \in [\![M]\!] \quad \Longleftrightarrow \quad q \text{ is accepted by the automaton } \langle \emptyset, Q, \emptyset, q \rangle$$

Corollary.

Acceptance of a Böhm tree generated by a λY-term M is decidable.

Higher-order recursion schemes

The infinite tree



is generated by the higher-order recursion scheme

$$\begin{cases} S & \mapsto & F\ a\ b\ c \\ F\ x\ y\ z & \mapsto & x\ (y\ z)\ (F\ x\ y\ (y\ z)) \end{cases}$$

Church encoding in the λ-calculus

The higher-order recursion scheme

$$\begin{cases} S & \mapsto & F\ a\ b\ c \\ F\ x\ y\ z & \mapsto & x\ (y\ z)\ (F\ x\ y\ (y\ z)) \end{cases}$$

may be seen as a λ-term of type

$$(o \to o \to o) \to (o \to o) \to o \to o.$$

in the simply-typed λ-calculus extended with a recursion operator Y.

Here, each tree-constructor a, b and c is of type:

$$a \ :\ o \to o \to o \qquad\qquad b \ :\ o \to o \qquad\qquad c \ :\ o$$

Higher-order recursion schemes

Signature

$$\mathbf{a} : o \to o \to o$$
$$\mathbf{b} : o \to o$$
$$\mathbf{c} : o$$

Non terminals

$$S : o$$
$$F : o \to o$$

Rewrite rules

$$S \mapsto F\,\mathbf{c}$$
$$F \mapsto \lambda x . \mathbf{a}\, x\, (F\, (\mathbf{b}\, x))$$

$$S \to F\,\mathbf{c} \to \mathbf{a}\,\mathbf{c}\,(F\,(\mathbf{b}\,\mathbf{c})) \to \mathbf{a}\,\mathbf{c}\,(\mathbf{a}\,(\mathbf{b}\,\mathbf{c})\,F\,(\mathbf{b}\,(\mathbf{b}\,\mathbf{c})))$$

Church encoding in linear logic

The formula

$$(o \to o \to o) \to (o \to o) \to o \to o$$

traditionally translated in linear logic as

$$A \quad = \quad !(!o \multimap !o \multimap o) \multimap !(!o \multimap o) \multimap !o \multimap o$$

may be also translated as

$$B \quad = \quad !(o \multimap o \multimap o) \multimap !(o \multimap o) \multimap !o \multimap o.$$

Church encoding in linear logic

So, the same tree may be seen as a term of type

$$A \quad = \quad !(\,!o \multimap \,!o \multimap o\,) \multimap \,!(\,!o \multimap o\,) \multimap \,!o \multimap o$$

with tree-constructors a, b and c of type

$$a \;:\; !o \multimap !o \multimap o \qquad\qquad b \;:\; !o \multimap o \qquad\qquad c \;:\; o$$

or as a term of type

$$B \quad = \quad !(\,o \multimap o \multimap o\,) \multimap \,!(\,o \multimap o\,) \multimap \,!o \multimap o$$

with tree-constructors a, b and c of type

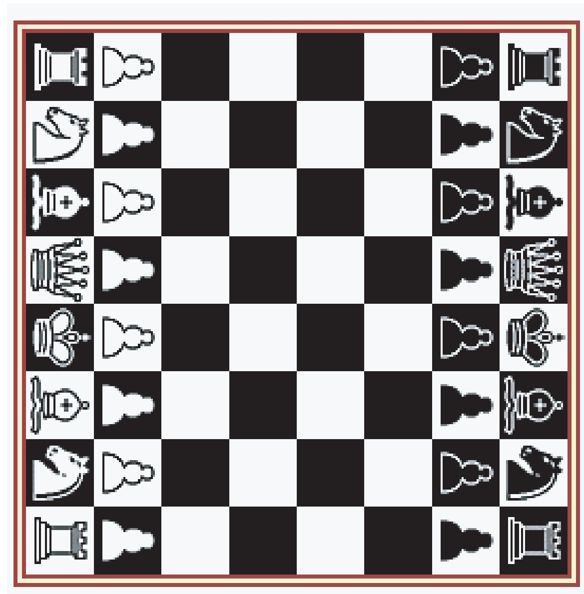
$$a \;:\; o \multimap o \multimap o \qquad\qquad b \;:\; o \multimap o \qquad\qquad c \;:\; o$$

Principle of duality

Proponent
Program

plays the formula

A



Opponent
Environment

plays the formula

$A^{\perp}$

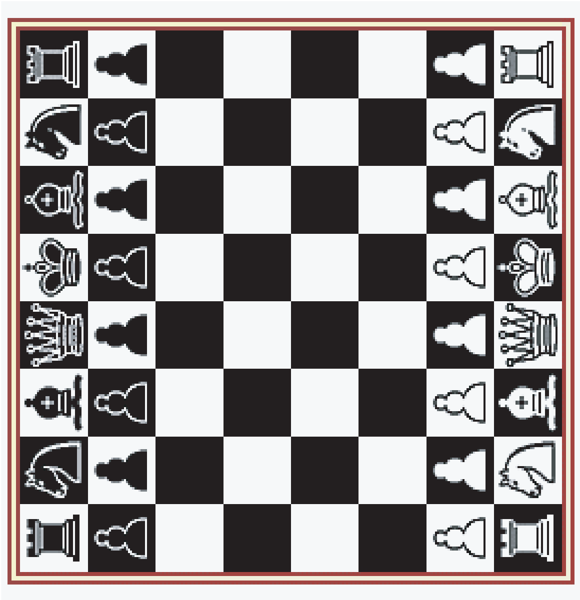
Negation permutes the rôles of Proponent and Opponent

Principle of duality

Opponent
Environment

plays the formula

$A^{\perp}$



Proponent
Program

plays the formula

A

Negation permutes the rôles of Opponent and Proponent

Duality applied to the Church encoding

Question: So, what is the dual of a tree ?

Answer: Well, it should be a tree automaton !

Duality applied to the Church encoding

The formulas A and B have counter-formulas:

$$A^{\perp} \quad = \quad !(\,!o \multimap !o \multimap o\,) \otimes !(\,!o \multimap o\,) \otimes !o \otimes o^{\perp}$$

$$B^{\perp} \quad = \quad !(\,o \multimap o \multimap o\,) \otimes !(\,o \multimap o\,) \otimes o \otimes o^{\perp}$$

Claim:

- ▷ the counter-formula $B^{\perp}$ is the type of **tree automata**
- ▷ the counter-formula $A^{\perp}$ is the type of **alternating tree automata**

What is a linear higher-order automaton?

Suppose given a finite preorder $(Q, \leq)$.

Adequacy Theorem.

The interpretation of a Böhm tree M is the set of its accepting states.

In other words, for every higher-order state $q \in [\![A]\!]$,

$$q \in [\![M]\!] \quad \Longleftrightarrow \quad q \text{ is accepted by the automaton } \langle \emptyset, Q, \emptyset, q \rangle$$

Corollary.

Acceptance of a Böhm tree generated by a LL_Y-term M is decidable.

The modal nature of priorities

A proof-theoretic account of parity tree automata

An intersection type system equivalent to the modal μ-calculus

The grammar of kinds κ

$$\kappa \quad :: \quad o \quad | \quad \kappa \Rightarrow \kappa$$

Naoki Kobayashi and Luke Ong [LICS 2009]

An intersection type system equivalent to the modal μ-calculus

The grammar of **atomic** types θ and **intersection** types τ

$$\overline{q_i \ ::_{atomic} \ o}$$

$$\frac{\theta_1 \ ::_{atomic} \ \kappa \quad \ldots \quad \theta_n \ ::_{atomic} \ \kappa}{(\theta_1, m_1) \wedge \ldots \wedge (\theta_n, m_n) \ :: \ \kappa}$$

$$\frac{\tau_1 \ :: \ \kappa_1 \quad \ldots \quad \tau_n \ :: \ \kappa_n \qquad q \ ::_{atomic} \ o}{\tau_1 \Rightarrow \cdots \tau_k \Rightarrow q \ ::_{atomic} \ \kappa_1 \Rightarrow \ldots \Rightarrow \kappa_k \Rightarrow o}$$

Naoki Kobayashi and Luke Ong [LICS 2009]

A type system equivalent to the modal μ-calculus

$$\frac{}{x : (\theta, \Omega[\theta]) \vdash x : \theta}$$

$$\frac{\{(i, q_{ij}) \mid 1 \leq i \leq n, 1 \leq j \leq k_i\} \text{ satisfies } \delta_A(q, a)}{a : \bigwedge_{j=1}^{k_1}(q_{1j}, m_{1j}) \Rightarrow \ldots \Rightarrow \bigwedge_{j=1}^{k_n}(q_{nj}, m_{nj}) \Rightarrow q}$$

where $m_{ij} = max(\Omega[q_{ij}], \Omega[q])$

$$\frac{\Delta \vdash t : (\theta_1, m_1) \wedge \ldots \wedge (\theta_k, m_k) \Rightarrow \theta \qquad \Delta_1 \vdash u : \theta_1 \quad \cdots \quad \Delta_k \vdash u : \theta_k}{\Delta, \Delta_1 \Uparrow m_1, \ldots, \Delta_k \Uparrow m_k \vdash t\,u : \theta}$$

where $\Delta \Uparrow m = \{F : (\theta, max(m, m')) \mid F : (\theta, m) \in \Delta\}$

$$\frac{\Delta, x : \bigwedge_{i \in I}(\theta_i, m_i) \vdash t : \theta \qquad I \subseteq J}{\Delta \vdash \lambda x . t : \bigwedge_{i \in J}(\theta_i, m_i) \Rightarrow \theta}$$

Emulation theorem

Let $\mathcal{G}$ be a higher-order recursion scheme.

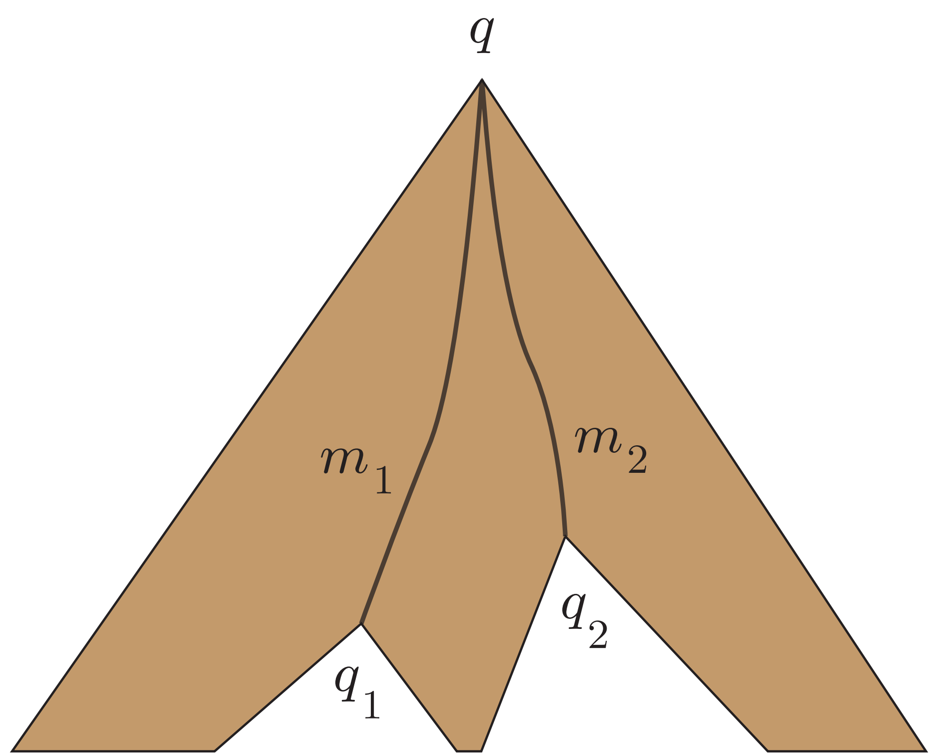
Let $\mathcal{A}$ be an alternating parity tree automaton.

Theorem [Kobayashi & Ong]

The tree generated by $\mathcal{G}$ is recognized by $\mathcal{A}$ $\iff$ The higher-order recursion scheme $\mathcal{G}$ is typable.

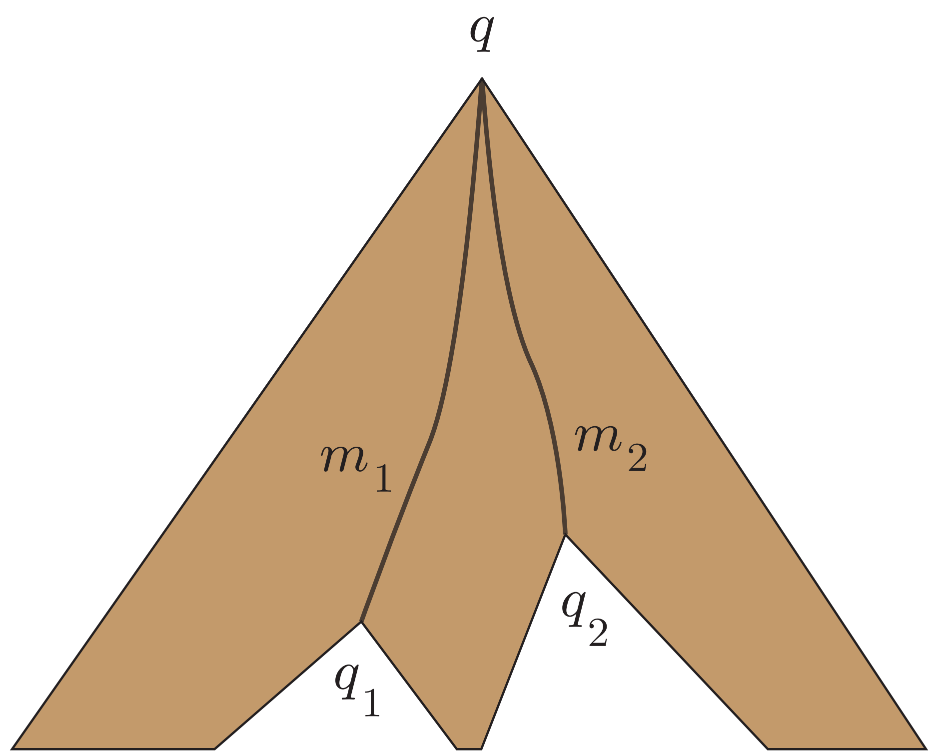
Guiding idea of Kobayashi and Ong

$$(q_1, m_1) \wedge (q_2, m_2) \Rightarrow q$$



Modal reformulation

$$\boxed{m_1}\; q_1 \wedge \boxed{m_2}\; q_2 \Rightarrow q$$



Collecting colours works in the same way as collecting levels of copies

A colour modality for intersection types

Definition. A parametric modality is a family of functors

$$\Box_m \quad : \quad \mathscr{C} \quad \longrightarrow \quad \mathscr{C} \qquad\qquad m \in \mathbb{N}$$

each of them lax monoidal:

$$\Box_m A \;\otimes\; \Box_m B \;\longrightarrow\; \Box_m (A \otimes B)$$

$$1 \;\longrightarrow\; \Box_m 1$$

and defining together a parametric comonad

$$\Box_{max(m,m')}\, A \;\longrightarrow\; \Box_m\, \Box_{m'}\, A$$

$$\Box_0\, A \;\longrightarrow\; A$$

The structure of **copy management** in linear logic

The exponential modality

$$!A \otimes !B \longrightarrow !(A \otimes B)$$

$$!A \longrightarrow !!A$$

$$!A \longrightarrow A$$

The structure of **copy management** in linear logic

Translation

$$\frac{\Delta \vdash t : (\theta_1, m_1) \wedge \ldots \wedge (\theta_k, m_k) \Rightarrow \theta \qquad \Delta_i \vdash u : \theta_i}{\Delta\,,\, \Delta_1 \Uparrow m_1\,,\, \ldots\,,\, \Delta_k \Uparrow m_k \ \vdash\ t\,u : \theta}$$

where $\quad \Delta \Uparrow m = \{F : (\theta\,,\, max(m, m') \mid F : (\theta, m) \in \Delta\}$

is translated as

$$\frac{\Delta \vdash t : \Box_{m_1} \theta_1 \wedge \ldots \wedge \Box_{m_k} \theta_k \Rightarrow \theta \qquad \dfrac{\Delta_i \vdash u : \theta_i}{\Box\, m_i\ \Delta_i \vdash u : \Box\, m_i\, \theta_i}}{\Delta\,,\, \Box_{m_1} \Delta_1\,,\, \ldots\,,\, \Box_{m_k} \Delta_k \ \vdash\ t\,u : \theta}$$

Linear logic with colours

A domain-theoretic account of parity tree automata

A colour modality for domains

Suppose given a specific number n of colours.

Definition. The colour modality on preorders is defined as

$$\Box A \quad := \quad \underbrace{A \,\&\, \cdots \,\&\, A}_{n}$$

As a consequence, note that

$$\mathit{Domain}(\Box A) \quad := \quad \mathit{Domain}(A) \times \cdots \times \mathit{Domain}(A)$$

The colour modality

Two preliminary observations

▷ The modality $\Box$ defines a comonad.

$$\begin{array}{cccccc} \varepsilon_A & : & \Box A & \longrightarrow & A \\ & & (1, q) & \mapsto & q \\[1em] \delta_A & : & \Box A & \longrightarrow & \Box \Box A \\ & & (\mathbf{max}\,(m_1, m_2), q) & \mapsto & (m_1, (m_2, q)) \end{array}$$

▷ The comonad $\Box$ commutes with finite products:

$$\begin{array}{ccc} \Box\,(A \,\&\, B) & \cong & \Box A \,\&\, \Box B \\[1em] \Box\,\top & \cong & \top \end{array}$$

The colour modality

A third observation

▷ There exists a distributivity law

$$\lambda \quad : \quad ! \,\Box \quad \Rightarrow \quad \Box \, ! \quad : \quad \mathbf{ScottL} \quad \longrightarrow \quad \mathbf{ScottL}$$

defined as follows:

$$\lambda_A \quad : \quad \{(m_1, q_1), \ldots, (m_k, q_k)\} \quad \mapsto \quad (\,\mathbf{max}\,(m_1, \ldots, m_k), \{q_1, \ldots, q_k\})$$

A colour modality

An important consequence: The composite modality

$$! \, \Box \quad : \quad \mathbf{ScottL} \quad \longrightarrow \quad \mathbf{ScottL}$$

defines an exponential modality of linear logic.

From this follows that the Kleisli category

$$\mathscr{D} \quad := \quad \mathit{Kleisli}\,(\,\mathbf{ScottL}\,,\,!\,\Box\,)$$

is a cartesian closed category.

A domain-theoretic formulation

The category $\mathscr{D}$ has

- finite prime algebraic domains as objects
- continous functions $f : D^n \longrightarrow E$ as morphisms.

Two morphisms of the category $\mathscr{D}$

$$f : D^n \longrightarrow E \qquad\qquad g : E^n \longrightarrow F$$

are composed as follows:

$$D^n \xrightarrow{D^{\mathbf{max}}} D^{n \times n} \xrightarrow{f^n} E^n \xrightarrow{g} E$$

A domain-theoretic formulation

In the case $n = 2$

$$g \circ f \; : \; (x_1, x_2) \quad \mapsto \quad g\,(\,f\,(x_1, x_2)\,,\; f\,(x_2, x_2)\,)$$

In the case $n = 3$

$$g \circ f \; : \; (x_1, x_2, x_3) \quad \mapsto \quad g\,(\,f\,(x_1, x_2, x_3)\,,\; f\,(x_2, x_2, x_3)\,,\; f\,(x_3, x_3, x_3)\,)$$

More generally:

$$\begin{pmatrix} 1 & 2 \\ 2 & 2 \end{pmatrix} \quad \begin{pmatrix} 1 & 2 & 3 \\ 2 & 2 & 3 \\ 3 & 3 & 3 \end{pmatrix} \quad \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 2 & 3 & 4 \\ 3 & 3 & 3 & 4 \\ 4 & 4 & 4 & 4 \end{pmatrix} \quad \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 2 & 3 & 4 & 5 \\ 3 & 3 & 3 & 4 & 5 \\ 4 & 4 & 4 & 4 & 5 \\ 5 & 5 & 5 & 5 & 5 \end{pmatrix}$$

An inductive-coinductive fixpoint

For simplicity, let us assume that the number n of colours is even.

Given a morphism in the category $\mathscr{D}$

$$f \quad : \quad D^n \quad \longrightarrow \quad D$$

one defines the fixpoint

$$Y(f) \quad = \quad \nu x_n \, . \, \mu x_{n-1} \, . \, \nu x_{n-2} \, \ldots \, \nu x_2 \, . \, \mu x_1 \, . \, f(x_1, \cdots, x_n)$$

Theorem. This defines a categorical interpretation of the λY-calculus.

What is a higher-order automaton?

Suppose given a finite preorder $(Q, \leq)$.

Adequacy Theorem.

The interpretation of a Böhm tree M is the set of its accepting states.

In other words, for every higher-order state $q \in [\![A]\!]$,

$$q \in [\![M]\!] \quad \Longleftrightarrow \quad q \text{ is accepted by the parity automaton } \langle \emptyset, Q, \emptyset, q \rangle$$

Corollary.

Acceptance of a Böhm tree generated by a λY-term M is decidable.

Thank you !