Soutenance de thèse

Sémantiques et syntaxes vectorielles de la logique linéaire

Christine Tasson
tasson@pps.jussieu.fr

le 4 décembre 2009

Syntaxes

Langages de Programmation

Sémantiques

Outils Mathématiques

Réseaux d'interaction

λ-calcul

Logique linéaire

Syntaxes différentielles

Modèle relationnel

Espaces de finitude

Espaces cohérents

Espaces topologiques

Sémantique quantitative

Au programme

1. **Introduction :** le non-déterminisme
2. **Syntaxes vectorielles :** une étude de la cohérence
3. **Sémantiques vectorielles :** une étude de la dualité
4. **Conclusion :** quelques perspectives

Aux origines de la logique linéaire

La sémantique quantitative

[Girard 88]

Quel formalisme pour le non-déterminisme ?

Définition

Un comportement est dit **non-déterministe** lorsque :
les mêmes causes ne produisent pas toujours les mêmes effets.

Il existe plusieurs situations :

- Un **environnement** non-déterministe : *sur internet, un moteur de recherche de petites annonces de logements.*
- Un **programme** non-déterministe : *un programme qui, étant données des villes, calcule le plus court chemin passant par toutes ces villes en utilisant l'algorithme probabiliste de Metropolis.*
- Un **programme** et un **environnement** non-déterministes : *un programme qui recherche les petites annonces de logements sur internet et propose un itinéraire pour aller visiter chacun des logements.*

Modélisation des environnements non-déterministes

Soit x une donnée de type A.

- Soit une **trame** $|A|$ regroupant les différentes valeurs que peuvent prendre les données de type A.
  Par exemple, $\mathbb{N}$ pour une variable entière.
- Un opérateur de **choix extérieur** $+$ permet de prendre en compte toutes les valeurs possibles de x données par l'environnement.
- x est formalisée par une **combinaison linéaire** éventuellement infinie de ses valeurs possibles :

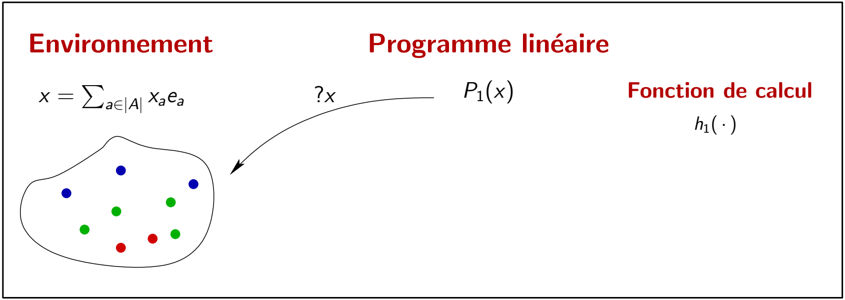
$$x = \sum_{a \in |A|} x_a e_a.$$

Par exemple, x est un entier résultant d'un tirage du loto :

$$x = \sum_{n=1}^{49} e_n.$$

Interaction programme / environnement non-déterministe

$\mathbf{P_1}$ consulte exactement 1 fois son argument :

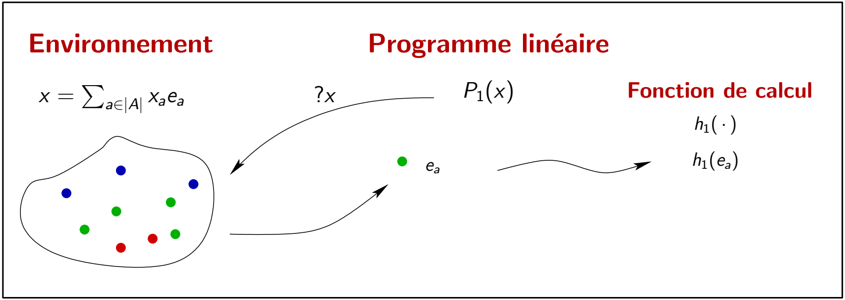


$$P_1\Big(\sum_{a \in |A|} x_a e_a\Big) = \sum_{a \in |A|} x_a h_1(e_a).$$

$\mathbf{P_n}$ consulte exactement n fois son argument :

$$P_n\Big(\sum_{a \in |A|} x_a e_a\Big) = \sum_{a_1, \ldots, a_n \in |A|} x_{a_1} \ldots x_{a_n} h_n(e_{a_1}, \ldots, e_{a_n}).$$

$\mathbf{P_1}$ consulte exactement 1 fois son argument :



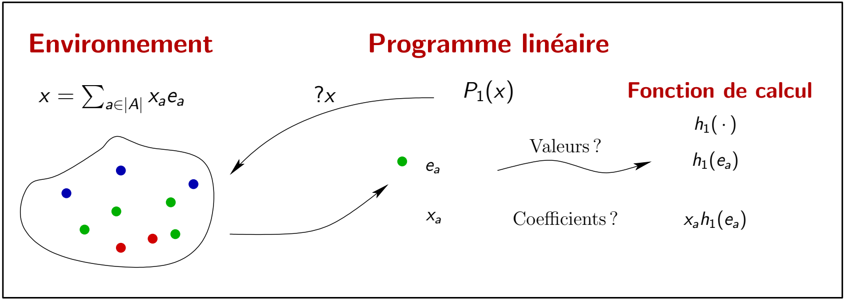
$$P_1(\sum_{a\in|A|} x_a e_a) = \sum_{a\in|A|} x_a h_1(e_a).$$

$\mathbf{P_n}$ consulte exactement n fois son argument :

$$P_n(\sum_{a\in|A|} x_a e_a) = \sum_{a_1,\ldots,a_n\in|A|} x_{a_1}\ldots x_{a_n}\, h_n(e_{a_1},\ldots,e_{a_n}).$$

Interaction programme / environnement non-déterministe

$\mathbf{P_1}$ consulte exactement 1 fois son argument :



$$P_1(\sum_{a\in|A|} x_a e_a) = \sum_{a\in|A|} x_a h_1(e_a).$$

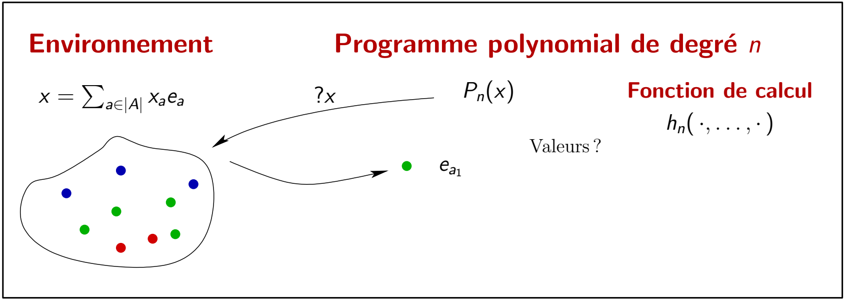
$\mathbf{P_n}$ consulte exactement n fois son argument :

$$P_n(\sum_{a\in|A|} x_a e_a) = \sum_{a_1,\dots,a_n\in|A|} x_{a_1}\dots x_{a_n}\, h_n(e_{a_1},\dots,e_{a_n}).$$

P_1 consulte exactement 1 fois son argument :

$$P_1\Big(\sum_{a\in|A|} x_a e_a\Big) = \sum_{a\in|A|} x_a h_1(e_a).$$

P_n consulte exactement n fois son argument :



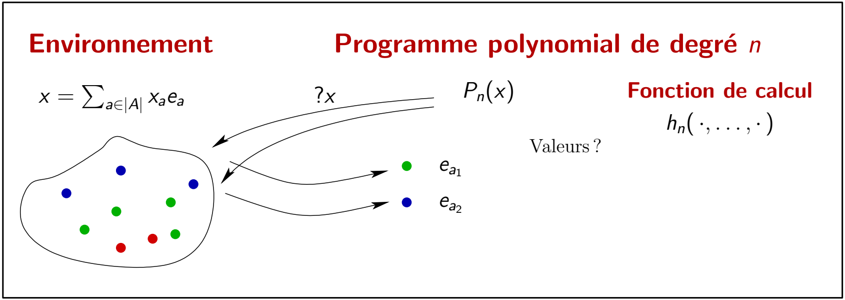
$$P_n\Big(\sum_{a\in|A|} x_a e_a\Big) = \sum_{a_1,\ldots,a_n\in|A|} x_{a_1}\ldots x_{a_n}\, h_n(e_{a_1},\ldots,e_{a_n}).$$

Interaction programme / environnement non-déterministe

$\mathbf{P_1}$ consulte exactement 1 fois son argument :

$$P_1\Big(\sum_{a\in|A|} x_a e_a\Big) = \sum_{a\in|A|} x_a h_1(e_a).$$

$\mathbf{P_n}$ consulte exactement n fois son argument :



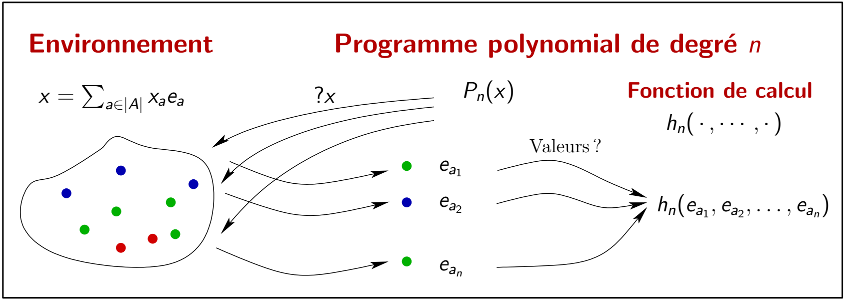
$$P_n\Big(\sum_{a\in|A|} x_a e_a\Big) = \sum_{a_1,\ldots,a_n\in|A|} x_{a_1}\ldots x_{a_n}\, h_n(e_{a_1},\ldots,e_{a_n}).$$

Interaction programme / environnement non-déterministe

$\mathbf{P_1}$ consulte exactement 1 fois son argument :

$$P_1(\sum_{a\in|A|} x_a e_a) = \sum_{a\in|A|} x_a h_1(e_a).$$

$\mathbf{P_n}$ consulte exactement n fois son argument :



$$P_n(\sum_{a\in|A|} x_a e_a) = \sum_{a_1,\ldots,a_n\in|A|} x_{a_1}\ldots x_{a_n}\, h_n(e_{a_1},\ldots,e_{a_n}).$$

Le principe

Environnements et programmes jouent des rôles interchangeables.

Dans les programmes, l'opérateur de choix permet de formaliser les **programmes non-déterministes** :

$$P = \sum_{n \in \mathbb{N}} P_n.$$

Par exemple, un programme non-déterministe $P : A \Rightarrow B$ peut être décrit comme une superposition de ses composantes :

$$\forall a \in |A|\,,\ P(e_a) = \sum_{b \in |B|} P_b(e_a) e_b.$$

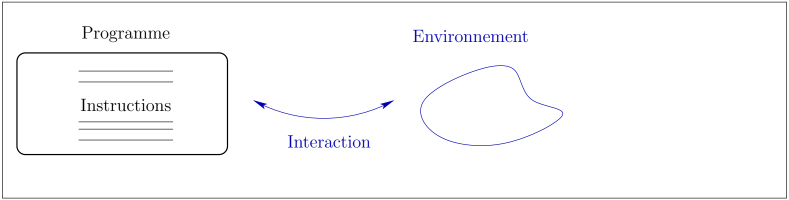
Hypothèse : *Tout programme peut être représenté par la superposition* **non-déterministe** *de ses approximations.*

Formule de Taylor syntaxique

Soit $P : A \Rightarrow B$ un programme. Pour tout $n \in \mathbb{N}$, il existe une routine polynomiale P_n de degré n telle que :

$$\forall x : A,\ P(x) = \sum_{n \in \mathbb{N}} P_n(x).$$

Idée : On déroule les différentes traces d'exécutions.



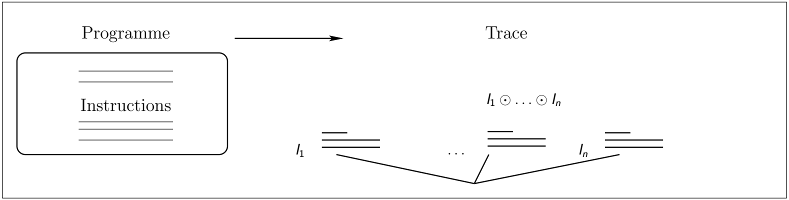
Hypothèse : *Tout programme peut être représenté par la superposition* **non-déterministe** *de ses approximations.*

Formule de Taylor syntaxique

Soit $P : A \Rightarrow B$ un programme. Pour tout $n \in \mathbb{N}$, il existe une routine polynomiale P_n de degré n telle que :

$$\forall x : A,\ P(x) = \sum_{n \in \mathbb{N}} P_n(x).$$

Idée : On déroule les différentes traces d'exécutions.



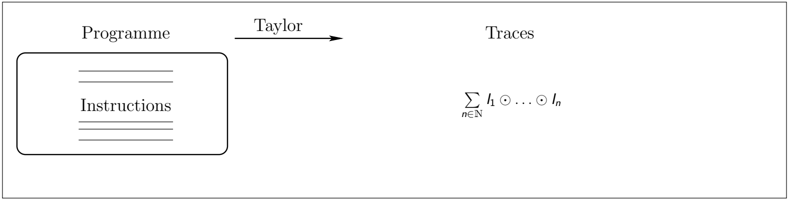
Hypothèse : *Tout programme peut être représenté par la superposition* **non-déterministe** *de ses approximations.*

Formule de Taylor syntaxique

Soit $P : A \Rightarrow B$ un programme. Pour tout $n \in \mathbb{N}$, il existe une routine polynomiale P_n de degré n telle que :

$$\forall x : A,\ P(x) = \sum_{n\in\mathbb{N}} P_n(x).$$

Idée : On déroule les différentes traces d'exécutions.



Hypothèse : *Tout programme peut être représenté par la superposition* **non-déterministe** *de ses approximations.*

Formule de Taylor syntaxique

Soit $P : A \Rightarrow B$ un programme. Pour tout $n \in \mathbb{N}$, il existe une routine polynomiale P_n de degré n telle que :

$$\forall x : A,\ P(x) = \sum_{n \in \mathbb{N}} P_n(x).$$

Idée : On déroule les différentes traces d'exécutions.

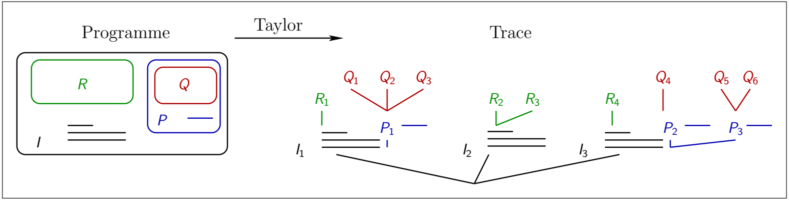
Hypothèse : *Tout programme peut être représenté par la superposition* **non-déterministe** *de ses approximations.*

Formule de Taylor syntaxique

Soit $P : A \Rightarrow B$ un programme. Pour tout $n \in \mathbb{N}$, il existe une routine polynomiale P_n de degré n telle que :

$$\forall x : A,\ P(x) = \sum_{n \in \mathbb{N}} P_n(x).$$

Idée : On déroule les différentes traces d'exécutions.



Syntaxes vectorielles

Une étude de la cohérence

En ajoutant aux langages de programmation usuels :

$+$: un opérateur de choix ;

$\Bbbk$: des coefficients ;

∂ : un opérateur de dérivation,

on obtient des langages non-déterministes satisfaisant la formule de Taylor syntaxique.

Quelles sont les propriétés de ces syntaxes étendues ?

Deux pistes :

1. Dans le fragment algébrique, une étude sémantique pour comprendre la totalité-cohérence.
2. Dans le fragment différentiel, une caractérisation des programmes partiels approchant un programme déterministe.

Cadre : le λ-calcul et son extension algébrique [Vaux 09].

Totalité-cohérence

La composition de deux λ-termes simplement typés est **totale** (sans erreur et sans boucle) et **cohérente** (propriété sémantique).

Idée : le λ-calcul barycentrique est formé de termes du λ-calcul algébrique vérifiant la propriété sémantique de totalité-cohérence.

Théorème ([T. 09])

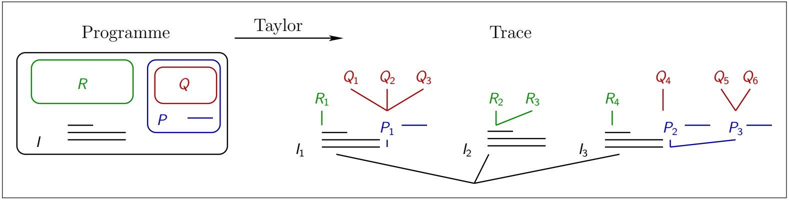
- *Les* **espaces de totalité** *forment une sémantique dénotationnelle du λ-calcul.*
- *Les espaces de totalité forment une sémantique du λ-calcul barycentrique complète pour le type $\mathcal{B}^n \Rightarrow \mathcal{B}$.*

Cadre : les réseaux de la logique linéaire et les réseaux différentiels [Ehrhard - Regnier 06].

Uniformité et formule de Taylor syntaxique

Les programmes partiels issus d'une même formule de Taylor syntaxique possèdent une structure similaire, ils sont **uniformes**.

Idée : un algorithme de fusion s'appuyant sur la réécriture.



Cadre : les réseaux de la logique linéaire et les réseaux différentiels [Ehrhard - Regnier 06].

Uniformité et formule de Taylor syntaxique

Les programmes partiels issus d'une même formule de Taylor syntaxique possèdent une structure similaire, ils sont **uniformes**.

Idée : un algorithme de fusion s'appuyant sur la réécriture.

Théorème ([Pagani - T. 09])

Soient π un réseau de la logique linéaire et $\alpha_1, \ldots, \alpha_n$ des réseaux différentiels.

- *Si $(\alpha_i)_{i \leq n} \xrightarrow{\texttt{mrg}*} \pi$ alors pour tout $i \leq n$, $\alpha_i \in \mathcal{T}(\pi)$.*
- *Si pour tout $i \leq n$, $\alpha_i \in \mathcal{T}(\pi)$, alors il existe $\pi_0 \ll \pi$ tel que $(\alpha_i)_{i \leq n} \xrightarrow{\texttt{mrg}*} \pi_0$*

Sémantique vectorielle

de la logique linéaire

- Un **type** A est représenté par un espace vectoriel topologique.
- Une **donnée** $x : A$ est représentée par un vecteur :

$$x = \sum_{a \in |A|} x_a e_a$$

- Un **programme** $P : A \Rightarrow B$ est interprété par une fonction $f : A \rightarrow B$ développable en série de Taylor.

Formule de Taylor sémantique

Soit $f : A \rightarrow B$ une fonction régulière. Pour tout $n \in \mathbb{N}$ il existe un polynôme f_n homogène de degré n tel que :

$$f(x) = \sum_{n \in \mathbb{N}} f_n(x).$$

$$
\begin{tikzcd}
 & & \text{Espaces vectoriels topologiques} \arrow[dl] \arrow[dr] & \\
\text{Corps} & \text{Discret} \arrow[d] & & \mathbb{R} \text{ ou } \mathbb{C} \arrow[dl] \arrow[d] \\
\text{Espace} & \text{Topologie linéarisée (ETL)} & \text{Normé (EVN)} \arrow[r, "\subseteq"] & \text{Localement convexe (ELC)}
\end{tikzcd}
$$

Le choix linéarisé est guidé par l'exemple des espaces de finitude.

Dans un espace vectoriel topologique, la topologie est engendrée par une base de voisinages de zéro :

- EVN boule unité ;
- ELC ensembles absolument convexes et absorbants ;
- ETL sous-espaces vectoriels.

Définition

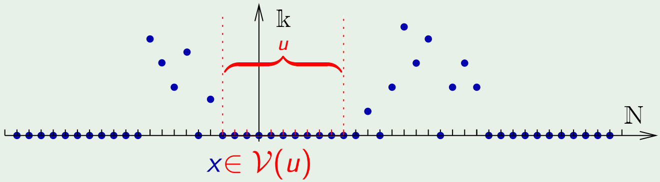
Soit $\Bbbk$ un corps discret. Un espace de Lefschetz E est un $\Bbbk$-espace vectoriel topologique dont la topologie est engendrée par une base $\mathcal{V}$ d'ouverts linéaires telle que $\bigcap \mathcal{V} = \{0\}$.

Dans un espace vectoriel topologique, la topologie est engendrée par une base de voisinages de zéro :

EVN	boule unité ;
ELC	ensembles absolument convexes et absorbants ;
ETL	sous-espaces vectoriels.

Exemple

On note $\mathbb{k}^{(\mathbb{N})}$ l'espace des suites presque partout nulles, muni de la topologie engendrée par les $\mathcal{V}(u) = \{x\ ;\ |x| \cap u = \emptyset\}$ pour toute partie u finie de $\mathbb{N}$.



La topologie induit une bornologie, une partie est bornée si elle est

EVN	absorbée par la boule unité ;
ELC	absorbée par tous les ouverts absolument convexes ;
ETL	linéairement absorbée par les ouverts linéaires.

Définition

Soit E un espace de Lefschetz. Un sous-espace vectoriel B de E est dit **linéairement borné** lorsque pour tout ouvert linéaire V de E, il existe un sous-espace vectoriel D de dimension finie tel que :

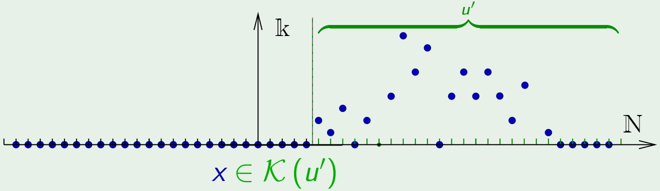
$$B = (V \cap B) \oplus D$$

La topologie induit une bornologie, une partie est bornée si elle est

- EVN absorbée par la boule unité ;
- ELC absorbée par tous les ouverts absolument convexes ;
- ETL linéairement absorbée par les ouverts linéaires.

Exemple

Dans $\mathbb{k}^{(\mathbb{N})}$, $\mathcal{K}(u') = \{x\ ;\ |x| \subseteq u'\}$ est linéairement borné pour toute partie u' de $\mathbb{N}$.

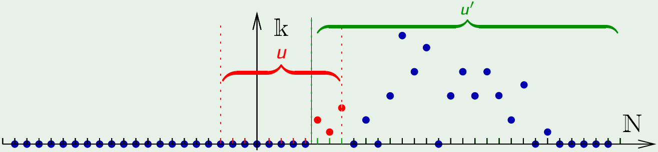


La topologie induit une bornologie, une partie est bornée si elle est

- EVN absorbée par la boule unité ;
- ELC absorbée par tous les ouverts absolument convexes ;
- ETL linéairement absorbée par les ouverts linéaires.

Exemple

Dans $\mathbb{k}^{(\mathbb{N})}$, $\mathcal{K}(u') = \{x \; ; \; |x| \subseteq u'\}$ est linéairement borné pour toute partie u' de $\mathbb{N}$.



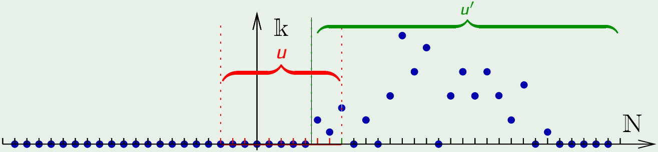
$x_2 \in D(u \cap u')$ $\quad x_1 \in \mathcal{K}(u') \cap \mathcal{V}(u)$

La topologie induit une bornologie, une partie est bornée si elle est

- EVN absorbée par la boule unité ;
- ELC absorbée par tous les ouverts absolument convexes ;
- ETL linéairement absorbée par les ouverts linéaires.

Exemple

Dans $\mathbb{k}^{(\mathbb{N})}$, $\mathcal{K}(u') = \{x \; ; \; |x| \subseteq u'\}$ est linéairement borné pour toute partie u' de $\mathbb{N}$.



$$x = x_1 + x_2 \in \mathcal{K}(u') = (\mathcal{K}(u') \cap \mathcal{V}(u)) \oplus D(u \cap u')$$

La bornologie induit une topologie engendrée par les bornivores, qui

EVN/ELC absorbent tous les bornés ;

ETL absorbent linéairement tous les linéairement bornés.

Définition

Soit E un espace de Lefschetz. Un **linéairement bornivore** est un sous-espace vectoriel V de E si pour tout linéairement borné B de E, il existe un sous-espace vectoriel D de dimension finie tel que :

$$B = (V \cap B) \oplus D$$

La dualité bornologie linéaire et topologie linéarisée joue un rôle :

- dans la dualité de la logique linéaire ;
- dans la réflexivité des espaces de Lefschetz.

La **négation** est interprétée par le dual topologique muni de la topologie de la convergence uniforme sur les linéairement bornés :

$$\left[\!\left[A^{\perp} \right]\!\right]_{\mathbf{Lef}} = \neg_{\mathfrak{B}} \left[\!\left[A \right]\!\right]_{\mathbf{Lef}}$$

Dualité de la logique linéaire classique

La négation est involutive : $A^{\perp\perp} \equiv A$.

Définition

Un espace de Lefschetz E est dit **réflexif** lorsque l'évaluation est un homéomorphisme linéaire :

$$\begin{array}{rcl} \mathrm{ev} : E & \longrightarrow & \neg_{\mathfrak{B}} \neg_{\mathfrak{B}} E \\ x & \longmapsto & x' \mapsto \langle x', x \rangle \end{array}$$

Bornés, Ouverts et Dual

Soit E un espace de Lefschetz.

Si B est un linéairement borné de E,

alors $B^{\perp}$ est un ouvert linéaire de $\neg_{\mathfrak{B}} E$.

Si V est un ouvert linéaire de E,

alors $V^{\perp}$ est un linéairement borné de $\neg_{\mathfrak{B}} E$.

Pour avoir la réflexivité il faut la relation duale, on ne l'a pas en général avec les ouverts, mais avec les bornivores.

Bornés, Bornivores et Dual

		E	$\neg_{\mathfrak{B}} E$		E	$\neg_{\mathfrak{B}} E$
Si	linéairement borné	B	B'	alors,	${}^{\perp}V'$	$V^{\perp}$
	linéairement bornivore	V	V'		${}^{\perp}B'$	$B^{\perp}$

Définition

Un espace de Lefschetz E est dit **linéairement bornologique** lorsque sa topologie coïncide avec la topologie des linéairement bornivores.

Théorème

Un espace de Lefschetz est linéairement bornologique et complet si et seulement s'il est réflexif pour la topologie de la convergence uniforme sur les linéairement bornés.

Schéma de la démonstration pour le sens direct :

- L'évaluation ev est linéaire et injective.
- Si E est linéairement bornologique, ev est continue.
- Si E est complet, alors ev est surjective.

Définition

Un espace de Lefschetz E est dit **linéairement bornologique** lorsque sa topologie coïncide avec la topologie des linéairement bornivores.

Théorème

Un espace de Lefschetz est linéairement bornologique et complet si et seulement s'il est réflexif pour la topologie de la convergence uniforme sur les linéairement bornés.

La catégorie des espaces linéairement bornologiques et complets et des fonctions linéaires continues est stable par :

- Négation
- Coproduits
- Produits

? Structure monoïdale close

Exemple

L'espace $\mathbb{k}^{(\mathbb{N})}$ des suites presque partout nulles muni de la topologie engendrée par les $\mathcal{V}(u)$ où $u \in \mathcal{P}_{\mathrm{fin}}(\mathbb{N})$ est un espace linéairement bornologique, mais pas complet.

Son complété est l'espace $\mathbb{k}^{\mathbb{N}}$ de toutes les suites muni de la topologie produit.

Exemple (**LinFin** [Ehrhard 05])

Les espaces de Lefschetz engendrés par un espace de finitude sont des espaces linéairement bornologiques et complets.

Soient E et F deux espaces de Lefschetz.

Définition (Polynômes)

Un **polynôme homogène de degré** n est une fonction de E dans F telle qu'il existe une fonction n-linéaire continue h_n telle que

$$\forall x \in E,\ P(x) = h_n(\underbrace{x, \ldots, x}_{n}).$$

Définition (Fonctions entières)

$\mathbf{Ser}_{\Bbbk}(\mathbf{E}\ ;\ \mathbf{F})$, espace des fonctions entières de E à valeurs dans F, est le **complété** de l'espace des polynômes de E dans F muni de la topologie de la convergence uniforme sur les linéairement bornés.

Soient E et F deux espaces de Lefschetz réflexifs.

Théorème (Formule de Taylor)

Toute série formelle $f \in \mathrm{Ser}_{\Bbbk}(E, F)$, vérifie la formule de Taylor

$$\forall x \in E,\ f(x) = \sum_{n=0}^{\infty} \frac{1}{n!} \partial^n f(0) \cdot x^{\otimes n}.$$

Théorème

Soit $f \in \mathrm{Ser}_{\Bbbk}(E, \Bbbk)$. Pour tout B linéairement borné de E, il existe $N \in \mathbb{N}$ et des formes n-linéaires h_n hypocontinues sur E telles que

$$\forall x \in B,\ f(x) = \sum_{n \leq N} h_n(\underbrace{x, \ldots, x}_{n}).$$

Dans **LinFin**, $[\![A \Rightarrow B]\!] = \mathrm{Ser}_{\Bbbk}([\![A]\!], [\![B]\!])$.

Syntaxes vectorielles

Totalité-cohérence

- Peut-on généraliser le théorème de complétude à d'autres types ?
- Un terme algébrique dont la sémantique est totale est-il barycentrique ?

Uniformité

- Peut-on caractériser un programme déterministe par un nombre fini de ses approximations ?
- Comment se comporte la formule de Taylor syntaxique vis-à-vis du calcul ?

Sémantiques vectorielles

- Quelle est la bonne structure monoïdale ?
- Comment transposer les constructions des ETL aux ELC ?
- Quelle exponentielle induit la formule de Taylor ?

Bibliographie

- T. Ehrhard et L. Regnier, *Differential Interaction Nets*, Electronic Notes in Theoretical Computer Science, 2006
- T. Ehrhard, *Finiteness spaces*, Mathematical Structures in Computer Science, 2005
- J.Y. Girard, *Normal functor power series*, Theoretical Computer Science, 1988
- S. Lefschetz, *Algebraic topology*, American Mathematical Society, 1942
- M. Pagani, C. Tasson, *The inverse Taylor expansion problem in linear logic*, Lics 2009
- C. Tasson, *Algebraic totality, towards completeness*, TLCA 2009
- L. Vaux, *The algebraic lambda-calculus*, Mathematical Structures in Computer Science, 2009