# Sémantique dénotationnelle

Jean Goubault-Larrecq

LSV/CNRS UMR 8643 & ENS Cachan

## Une idée simple...

- Choisir un domaine de valeurs $Val$, suffisamment grand.
- Définir la valeur $[\![M]\!]$ du terme $M$ dans $Val$, par récurrence structurelle sur $M$.

Ceci donne une description mathématique simple de la sémantique des programmes, non?

## Avant de commencer, réglons quelques problèmes...

Premier problème: quelle est la valeur d'une variable $x$?

$\Rightarrow$ on a besoin d'un environnement $\rho \in Env =_{def} Var \rightarrow Val$;

Deuxième problème: comment interpréter les opérations sur la mémoire?

$\Rightarrow$ on a besoin d'une mémoire $\mu \in Mem =_{def} Addr \rightarrow Val$;

La sémantique $[\![M]\!]\,\rho\mu$ de $M$ sera paramétrée par $\rho, \mu$ et retourne un couple $\mu' \in Mem, V \in Val$.

(À ce stage, vous aurez remarqué que définir $[\![M]\!]\,\rho\mu \in Mem \times Val$ ou définir quand $\rho, \mu \vdash M \Rightarrow \mu', V$, c'est à peu près pareil.

**Exercice:** quelles sont les différences?)

## ... et simplifions un peu

Pour simplifier, nous allons considérer un langage plus restreint:

| | | | |
|---|---|---|---|
| | | | Termes (fonctions, données) |
| $M, N, P, \ldots$ | $::=$ | $x$ | variables (non modifiables) |
| | $\mid$ | $M N$ | application de $M$ à $N$ |
| | $\mid$ | $\texttt{fun}\ x \to M$ | fonction qui à $x$ associe $M$ |

(Modulo quelques différences de notations, c'est la $\lambda$-calcul de Church [années 1930].)

On a toujours besoin d'un environnement $\rho$, mais $\mu$ n'est plus nécessaire: on définit $[\![M]\!]\,\rho \in Val$.

## Allons-y donc!

$$[\![x]\!]\,\rho \quad = \quad \rho(x) \tag{1}$$

$$[\![MN]\!]\,\rho \quad = \quad f([\![N]\!]\,\rho) \quad \text{où } f = [\![M]\!]\,\rho \tag{2}$$

$$[\![\lambda x \cdot M]\!]\,\rho \quad = \quad \text{fonction } V \mapsto [\![M]\!]\,(\rho[x \mapsto V]) \tag{3}$$

Vous vous souvenez tous que ça pose un problème, non?

(voir dernier cours)

Nous avions éludé la difficulté en modélisant les fonctions par des clôtures. Ceci est insatisfaisant:

- solution inélégante: le domaine sémantique $Val$ fait appel à la syntaxe ($Terme$);
- ne permet pas de considérer les fonctions informatiques ... comme des fonctions mathématiques (un comble!)

## La théorie des domaines

---

Idée de base: trouver un domaine $Val$ tel que $(Val \to Val)$ soit inclus dans $Val$.

Problème: ça n'existe pas (Cantor).

---

Correction: un domaine sera un ensemble ayant une certaine structure, et l'espace $[Val \to Val]$ des fonctions respectant la structure seront incluses dans $Val$.

Par exemple: domaine = espace topologique, $[Val \to Val] = \{\text{fonctions continues}\}$. Plus de problème de cardinalité!

**Exercice:** si $Val = \mathbb{R}$, montrer que le cardinal de l'ensemble des fonctions continues de $\mathbb{R}$ vers $\mathbb{R}$ est celui de $\mathbb{R}$ exactement, et pas plus gros.

## L'idée de D. Scott

Domaine = dcpo ("directed complete partial order").

Fonctions considérées = fonctions continues.

**Définition**: Soit $(Val, \leq)$ ordonné. $D \subseteq Val$ est *dirigé* ssi $D \neq \emptyset$ et $\forall d, d' \in D \cdot \exists d'' \in D \cdot d, d' \leq d''$.

$(Val, \leq)$ est un *dcpo* ssi tout $D$ dirigé a un sup.

$f : Val \to Val$ est *continue* ssi $f$ préserve les sups de familles dirigées.

---

**Exercices:** si $f$ est continue, alors $f$ est *monotone*: $x \leq y \Rightarrow f(x) \leq f(y)$.

Si $B$ dcpo, $[A \to B]$ muni de l'ordre $f \leq g$ ssi $\forall x \cdot f(x) \leq g(x)$ est un dcpo.

Si $A, B$ dcpos, $A \times B$ dcpo.

La fonction $App : (f, x) \in [A \to B] \times A \mapsto f(x) \in B$ est continue.

La fonction $Cur : f \in [A \times B \to C] \mapsto g \in [A \times [B \to C]]$ telle que $g(x)(y) = f(x, y)$ est continue.

## Le théorème de Scott

**Théorème:** Pour tout dcpo $D_0$, il existe un dcpo $D$ contenant $D_0$ et tel que $D \cong [D \to D]$.

**Preuve:** technique... limite projective d'une chaîne de dcpos dont le premier est $D_0$. (Voir cours de logique et informatique, deuxième semestre!)

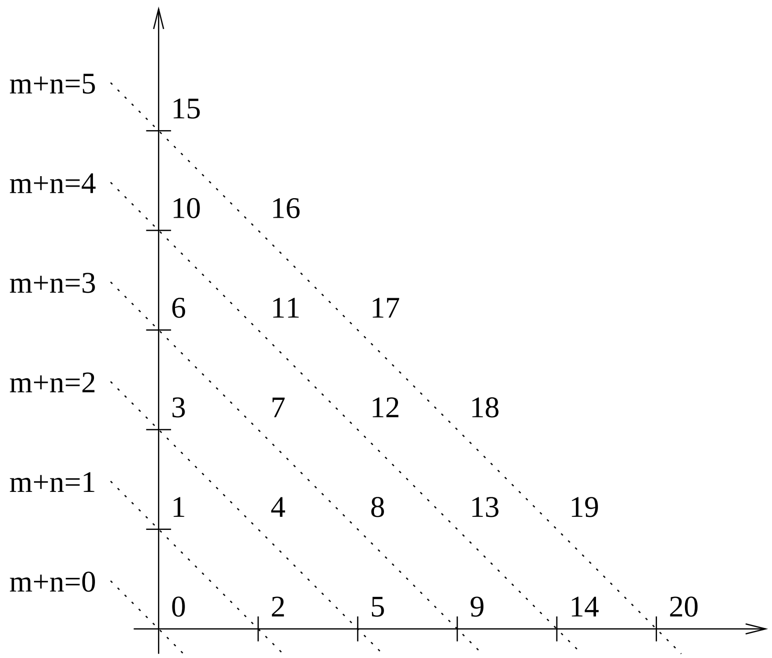
Dans ce cadre, $[\![M]\!]\,\rho$ a un sens... mais le domaine des valeurs est tordu.

## Un modèle plus simple: le modèle $\mathbb{P}\omega$ de Plotkin

$\mathbb{P}\omega$ est le dcpo des parties de $\mathbb{N}$, ordonné par inclusion $\subseteq$.

On utilise quelques astuces pour voir que $[\mathbb{P}\omega \to \mathbb{P}\omega]$ s'injecte naturellement dans $\mathbb{P}\omega$...

D'abord, on peut coder toute paire d'entiers $m, n$ comme un entier $\langle m, n\rangle$:




**Exercice:** écrire une formule explicite calculant $\langle m, n\rangle$, et une autre retrouvant $m$ et $n$ à partir de $\langle m, n\rangle$.

Ensuite, on peut coder tout ensemble fini $e = \{n_1, \ldots, n_k\}$ d'entiers par le nombre binaire $[e] = \sum_{i=1}^{k} 2^{n_i}$:

| | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| {1, 3, 4, 7, 9, 10} = | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | = 1690 |

Réciproquement, on note $e_m$ l'ensemble tel que $[e_m] = m$.

---

Enfin, toute fonction continue de $\mathbb{P}\omega$ vers $\mathbb{P}\omega$ est déterminée de façon unique par sa valeur sur les ensembles finis d'entiers:

$$f(E) = \sup_{e \text{ fini } \subseteq E} f(e)$$

(Note: le sup est juste l'union $\bigcup$.)

On code donc $f \in [\mathbb{P}\omega \to \mathbb{P}\omega]$ par

$$r(f) \in \mathbb{P}\omega = \{\langle m, n\rangle \mid n \in f(e_m)\}$$

Étant donné $E \in \mathbb{P}\omega$, on peut retrouver la fonction $f = i(E)$ correspondante par:

$$i(E)(m) = \{n \in \mathbb{N} \mid \exists e \text{ fini } \subseteq m \cdot \langle [e], n\rangle \in E\}$$

**Exercice:** Vérifier que $i \circ r$ est l'identité sur $[\mathbb{P}\omega \to \mathbb{P}\omega]$, et que $i$ et $r$ sont continues.

La fonction $r$ permet de projeter $[\mathbb{P}\omega \to \mathbb{P}\omega]$ sur $\mathbb{P}\omega$, et $i$ injecte $\mathbb{P}\omega$ dans $[\mathbb{P}\omega \to \mathbb{P}\omega]$.

On peut donc définir:

$$\begin{aligned} [\![x]\!]\,\rho &= \rho(x) & (4)\\ [\![MN]\!]\,\rho &= f([\![N]\!]\,\rho) \quad \text{où } f = i([\![M]\!]\,\rho) & (5)\\ [\![\lambda x \cdot M]\!]\,\rho &= r(\text{fonction } V \mapsto [\![M]\!]\,(\rho[x \mapsto V])) & (6) \end{aligned}$$

**Exercice:** Vérifier que $[\![(\lambda x \cdot M)N]\!]\,\rho = [\![M[x := N]]\!]\,\rho$.
Par contre, en général $[\![\lambda x \cdot Mx]\!]\,\rho \neq [\![M]\!]\,\rho$, même lorsque $x$ n'apparaît pas libre dans $M$.