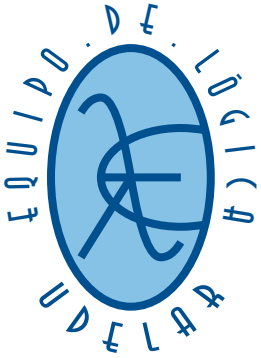
Réalisabilité et forcing

Alexandre Miquel



16 janvier 2019 – Collège de France / Sciences du logiciel

Pourquoi la logique ?

- L'activité mathématique consiste à établir des propriétés portant sur des objets idéaux. Parmi ces objets idéaux, on trouve :
  - des objets de base, finis (entiers naturels) ou infinis (nombres réels)
  - des classes (in)finies d'objets (in)finis
  - des classes (in)finies de classes (in)finies d'objets (in)finis

  Par exemple : *"Tout sous-ensemble de* $\mathbb{R}$*, non vide et majoré dans* $\mathbb{R}$*, admet une borne supérieure dans* $\mathbb{R}$

- Pour ce faire, les mathématicien.ne.s n'utilisent que des textes finis : les formules (énoncés) et les démonstrations (preuves)

  Les mathématiques sont-elles une activité bien raisonnable ?

- L'étude de la tension intrinsèque entre les ambitions infinitaires des mathématiques et leurs moyens finis est l'objet de la logique

Le prix à payer...

Premier théorème d'incomplétude [Gödel, 1931]

Toute théorie $\mathscr{T}$ récursive, arithmétique et cohérente est incomplète : il existe une formule close ϕ telle que $\mathscr{T} \nvdash \phi$ et $\mathscr{T} \nvdash \neg\phi$

Deuxième théorème d'incomplétude [Gödel, 1931]

Toute théorie $\mathscr{T}$ récursive, arithmétique et cohérente peut exprimer sa propre cohérence (par une certaine formule $\ll \mathrm{Cons}\mathscr{T} \gg$ du langage de $\mathscr{T}$), mais elle ne peut pas la démontrer : $\mathscr{T} \nvdash \mathrm{Cons}\mathscr{T}$

Rappels : une théorie $\mathscr{T}$ est :

- récursive quand son langage est dénombrable et quand son ensemble d'axiomes est récursivement énumérable (à travers un codage approprié)
- arithmétique quand elle contient l'arithmétique de Peano ou de Heyting (à travers une traduction logique appropriée)

Un prix à payer... ou un espace de liberté ?

Aucun système d'axiomes (récursif et arithmétique) ne permet de spécifier complètement l'univers mathématique

- Plusieurs manières de structurer l'univers mathématique :
  - Intuitionniste ou classique ?
  - Théorie des ensembles (ZF) ou théorie des types (Coq) ?
  - Prédicatif (CZF, MLTT) ou imprédicatif (ZF, Coq) ?
- Même dans un système donné, il y a toujours de la place pour des axiomes supplémentaires. Par exemple dans ZF :
  - L'axiome du choix (AC) ou sa négation ($\neg$AC) ?
  - L'hypothèse du continu (HC) ou sa négation ($\neg$HC) ?
- *De quels degrés de liberté dispose-t-on pour concevoir un univers mathématique ?*

Reste l'épée de Damoclès du second théorème d'incomplétude : on ne pourra jamais acquérir la certitude que les systèmes manipulés sont cohérents

Plan

Plan

Théories et modèles

Point de vue syntaxique : Une théorie $\mathscr{T}$ est définie par :

- Un langage $\mathscr{L}$ (symboles de constante, de fonction et de prédicat)
- Un système de déduction (intuitionniste ou classique)
- Un système d'axiomes

Point de vue sémantique : Un modèle $\mathscr{M}$ est défini par :

- Un ensemble $\mathscr{M} \neq \varnothing$ (domaine de l'interprétation)
- Des fonctions et des relations (ou fonctions propositionnelles) dans $\mathscr{M}$ pour interpréter les symboles du langage $\mathscr{L}$
- $\mathscr{M}$ est un modèle de la théorie $\mathscr{T}$ (notation : $\mathscr{M} \models \mathscr{T}$) lorsque $\mathscr{M}$ satisfait tous les axiomes de $\mathscr{T}$

Théorème (Correction)

Si $\mathscr{T} \vdash \phi$, alors dans tout modèle $\mathscr{M} \models \mathscr{T}$ on a $\mathscr{M} \models \phi$

Modèles de Tarski

En logique classique, on se restreint généralement aux modèles de Tarski

- Modèle de Tarski = modèle $\mathscr{M}$ dans lequel les symboles de relation ($\leq$, $\in$, etc.) sont interprétés par des relations dans $\mathscr{M}$ (au sens usuel)
- Dans un modèle de Tarski, une formule close ϕ n'a que deux valeurs de vérité possibles : 1 (vrai) ou 0 (faux)

Théorème (Complétude) [Gödel, 1929]

Une théorie est cohérente si et seulement si elle admet un modèle

(Ici : théorie = théorie classique ; modèle = modèle de Tarski)

- Comme les théories classiques usuelles (PA, ZF, etc.) sont incomplètes, elles admettent une infinité de modèles de Tarski non élémentairement équivalents (et donc non isomorphes)

Remarque : Un modèle de Tarski $\mathscr{M}$ d'une théorie $\mathscr{T}$ est une ≪complétion≫ de $\mathscr{T}$. La théorie du modèle $\mathscr{M}$ (qui complète $\mathscr{T}$) est en général non récursive

Preuves de cohérence relative

On veut démontrer que la cohérence d'une théorie $\mathscr{T}$ (par exemple ZF) implique la cohérence d'une autre théorie $\mathscr{T}'$ (par exemple ZFC)

1. On suppose que $\mathscr{T}$ est cohérente
2. Donc $\mathscr{T}$ admet un modèle $\mathscr{M}$ (d'après le théorème de complétude, $\Rightarrow$)
3. **On transforme le modèle $\mathscr{M} \models \mathscr{T}$ en un modèle $\mathscr{M}' \models \mathscr{T}'$**
4. Donc $\mathscr{T}'$ admet un modèle $\mathscr{M}'$ (d'après la construction qui précède)
5. Donc $\mathscr{T}'$ est cohérente (d'après le théorème de complétude, $\Leftarrow$)

On a ainsi démontré la cohérence relative de $\mathscr{T}'$ par rapport à $\mathscr{T}$.

Deux théories $\mathscr{T}$ et $\mathscr{T}'$ sont equiconsistantes (notation : $\mathscr{T} \approx \mathscr{T}'$) lorsque chacune est cohérente relativement à l'autre. Par exemple :

$$\begin{aligned} \mathrm{ZF} &\approx \mathrm{ZFC}\ (= \mathrm{ZF} + \mathrm{AC}) \approx \mathrm{ZF} + \neg\mathrm{AC} \\ &\approx \mathrm{ZFC} + \mathrm{HC} \approx \mathrm{ZFC} + \mathrm{HGC} \approx \mathrm{ZFC} + \neg\mathrm{HC} \end{aligned}$$

Transformation de modèles... ou de programmes ?

- Une preuve de cohérence relative de $\mathscr{T}'$ par rapport à $\mathscr{T}$ s'appuie sur une transformation de modèles $\quad \mathscr{M} \models \mathscr{T} \quad \mapsto \quad \mathscr{M}' \models \mathscr{T}'$
- Toutes les transformations de modèles $\mathscr{M} \mapsto \mathscr{M}'$ connues correspondent à des traductions logiques $\phi \mapsto \phi^*$ (de $\mathscr{T}'$ dans $\mathscr{T}$)

$$\mathscr{M}' \models \phi \qquad \text{ssi} \qquad \mathscr{M} \models \phi^*$$

Donc toute preuve sémantique de cohérence relative (i.e. dans les modèles)
correspond à une preuve syntaxique de cohérence relative (par traduction logique)

- Mais à travers la correspondance de Curry-Howard, toute traduction logique correspond à une transformation de programmes
  - transformation typée / (ensembles constructibles)
  - transformation non typée (¬¬-traduction, forcing)

▶ Transformer un modèle de $\mathscr{T}$ en un modèle de $\mathscr{T}'$, c'est en fait transformer un «programme» de $\mathscr{T}'$ en un «programme» de $\mathscr{T}$

Plan

Axiome du choix et hypothèse du continu (1/2)

- **Axiome du choix (AC) :** [Zermelo, 1908]

  Pour toute famille $(A_i)_{i\in I}$ d'ensembles non vides (I quelconque), il existe une famille $(x_i)_{i\in I}$ telle que $x_i \in A_i$ pour tout $i \in I$

  Nombreuses formulations équivalentes dans ZF :
  - Lemme de Zorn, Principe du bon ordre (Zermelo)
  - Tout espace vectoriel admet une base, etc.

- **Hypothèse du continu (HC) :** [Cantor, 1878]

  Tout sous-ensemble infini de $\mathbb{R}$ *est équipotent soit à* $\mathbb{N}$ *soit à* $\mathbb{R}$

  En symboles : $2^{\aleph_0} = \aleph_1$ $\quad (2^{\aleph_0} = \mathrm{card}(\mathfrak{P}(\mathbb{N})) = \mathrm{card}(\mathbb{R}))$

- **Hypothèse généralisée du continu (HGC) :**

  $$2^{\aleph_\alpha} = \aleph_{\alpha+1} \qquad \text{pour tout ordinal } \alpha$$

Axiome du choix et hypothèse du continu (1/2)

1878 : G. Cantor conjecture l'hypothèse du continu (HC)

1900 : D. Hilbert liste HC comme le 1^{er} de ses 23 problèmes

1908 : E. Zermelo introduit un système d'axiomes pour la théorie des ensembles ("théorie de Zermelo"). Il énonce l'axiome du choix (AC) et démontre le principe du bon ordre à partir de AC

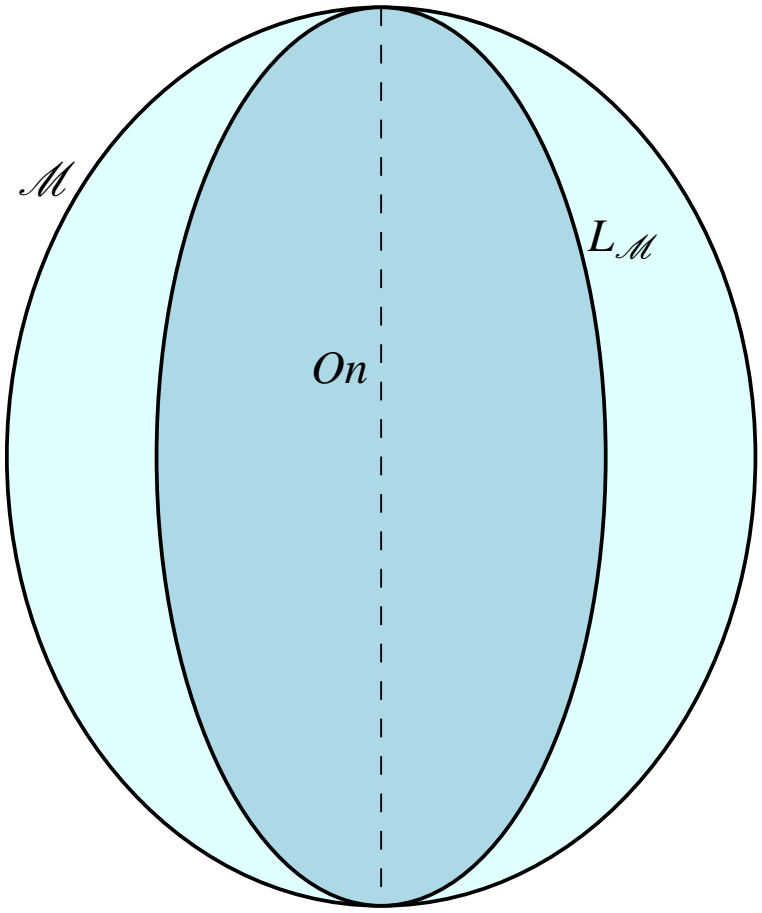
1922 : T. Skolem et A. Fraenkel complètent (indépendamment) la théorie de Zermelo avec le schéma de remplacement : système **ZF**

1938 : K. Gödel montre la cohérence relative de $\mathrm{AC} + \mathrm{HGC}$ par rapport à ZF, en introduisant les ensembles constructibles

1963 : P. Cohen montre la cohérence relative de $\neg$HC par rapport à ZFC, en introduisant la méthode du forcing. Avec la même méthode, il démontre la cohérence relative de $\neg$AC par rapport à ZF.

Il reçoit la médaille Fields en 1966

Ensembles constructibles (1/3)



$L_{\mathscr{M}}$ = classe des ensembles constructibles dans $\mathscr{M}$

= plus petit sous-modèle de ZF (dans $\mathscr{M}$) contenant tous les ordinaux de $\mathscr{M}$

(modèle intérieur)

Théorème [Gödel 1938]

Si $\mathscr{M} \models \mathrm{ZF}$, alors $L_{\mathscr{M}} \models \mathrm{ZF} + \mathrm{AC} + \mathrm{HGC}$

Ensembles constructibles (2/3)

- Le sous-modèle $L_{\mathscr{M}} \subseteq \mathscr{M}$ est défini par

$$L_{\mathscr{M}} := \{a \in \mathscr{M} : \mathscr{M} \models L(a)\}$$

où $L(x)$ est la formule (de ZF) exprimant que «x est constructible»

- Formellement :
  - $\mathrm{Def}(X) :=$ ensemble des parties définissables de X

$$:= \{Y \subseteq X : \exists \phi \in \mathit{Form},\ \exists x_1, \ldots, x_n \in X,\ \forall x \in X,\ x \in Y \Leftrightarrow (X, \in) \models \phi(x, x_1, \ldots, x_n)\}$$

  - $L_0 := \varnothing, \qquad L_{\alpha+1} := \mathrm{Def}(L_\alpha), \qquad L_\alpha := \bigcup_{\beta<\alpha} L_\beta \qquad (\alpha \text{ limite})$
  - $L(x) :\equiv (\exists \alpha : \mathit{On})\ x \in L_\alpha$

Ensembles constructibles (3/3)

Traduction logique associée

- À chaque formule ϕ on associe la formule relativisée ϕ^L :

$$(\forall x\, \phi(x))^L :\equiv \forall x\, (L(x) \Rightarrow \phi^L(x)) \qquad \bot^L :\equiv \bot$$

$$(\exists x\, \phi(x))^L :\equiv \exists x\, (L(x) \wedge \phi^L(x)) \qquad (\phi \Rightarrow \psi)^L :\equiv \phi^L \Rightarrow \psi^L \qquad \text{(etc.)}$$

- Pour toute formule close ϕ, on a :

1. $L_{\mathscr{M}} \models \phi$ ssi $\mathscr{M} \models \phi^L$ (adjonction)
2. Si $\mathrm{ZF} + \mathrm{AC} + \mathrm{HGC} \vdash \phi$, alors $\mathrm{ZF} \vdash \phi^L$ (preuve syntaxique)

Transformation de programme associée ? Problème ouvert !

- Transformation typée
- Polymorphisme : implicite $\mapsto$ explicite (Curry $\mapsto$ Church)
- Filtrage/inspection sur les types (types génériques)

Qu'est-ce que le forcing ?

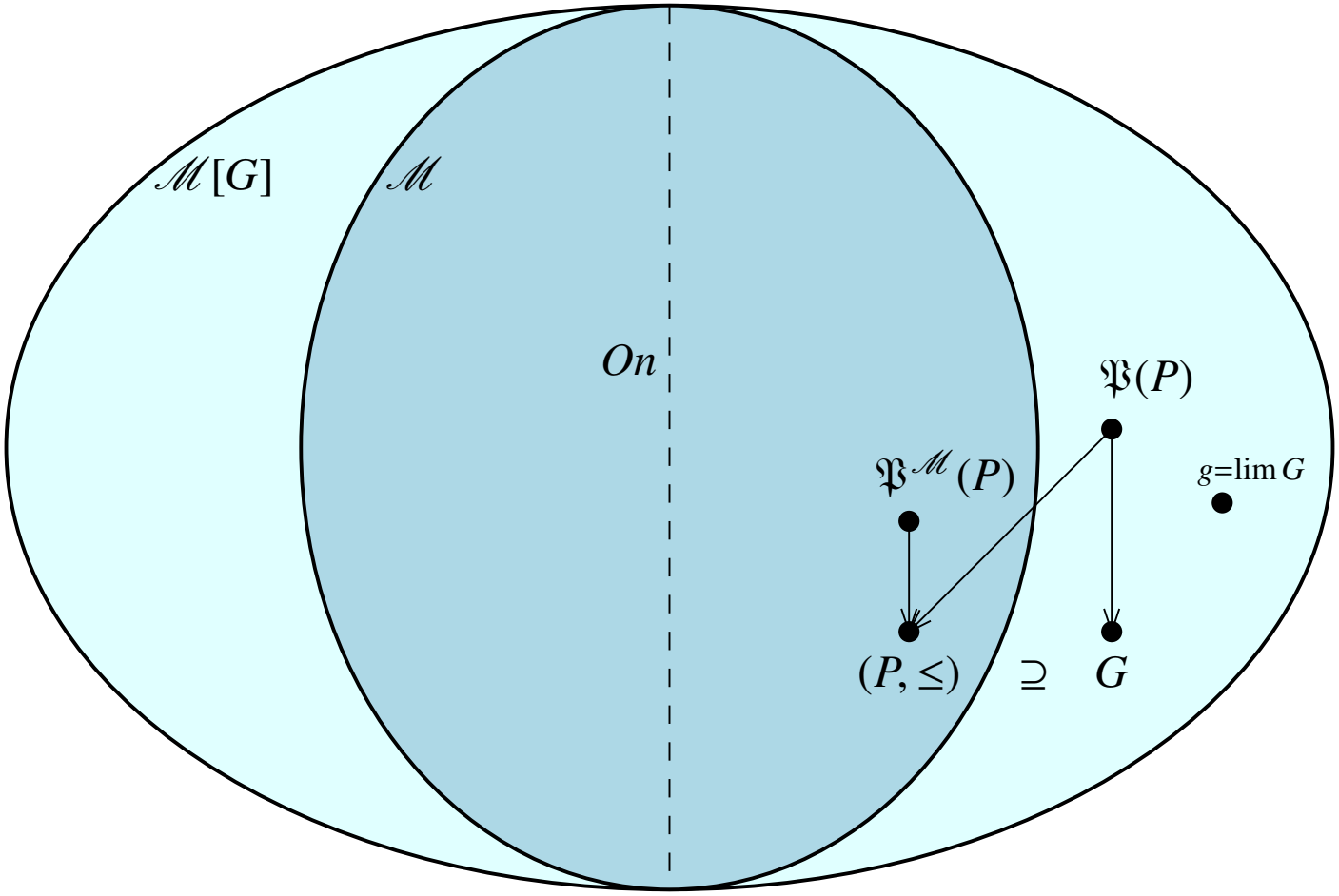
- Technique inventée par Cohen ('63) pour démontrer la cohérence relative de $\neg$HC par rapport à ZFC :

$$\mathrm{Cons}(\mathrm{ZFC} + \neg\mathrm{CH}), \qquad \text{ou :} \qquad \mathrm{ZFC} \nvdash \mathrm{CH}$$

- Le forcing peut être vu comme :
  - Une transformation de modèles de ZF(C) (extensions génériques)
  - Une technique pour construire des modèles booléens de ZF [Scott, Solovay, Vopěnka]
  - Une traduction logique de ZF dans ZF (formules & preuves)
- Un outil standard de théorie des ensembles :
  - Permet de démontrer la cohérence relative et/ou l'indépendance de nombreux axiomes (HC, HGC, AC, Axiome de Solovay, etc.)

Comment fonctionne le forcing ?

Exploiter la sous-spécificiation de l'ensemble $\mathfrak{P}(X)$ (avec X infini)



Forcing : la traduction logique sous-jacente

- Traduction paramétrée par un ensemble ordonné $(P, \leq)$ $\quad (P \neq \varnothing)$
  - Les éléments de P sont les conditions de forcing
  - $q \leq p$ signifie : $\ll q$ est plus forte que $p \gg$
- Traduction des formules : $\phi \quad \mapsto \quad p \Vdash \phi$

$$p \Vdash q \in G \;:\equiv\; p \leq q \qquad p \Vdash \phi \wedge \psi \;:\equiv\; (p \Vdash \phi) \wedge (p \Vdash \psi)$$
$$p \Vdash \neg\phi \;:\equiv\; \forall q \leq p\ (q \not\Vdash \phi) \qquad p \Vdash \forall x\, \phi(x) \;:\equiv\; \forall x\ (p \Vdash \phi(x))$$

(traduction de $\vee$, $\Rightarrow$ et $\exists$ déduite des lois de De Morgan)

Correction : $\quad \mathscr{M}[G] \models \phi \quad \Leftrightarrow \quad \exists p \in G,\ \mathscr{M} \models (p \Vdash \phi)$

Lien avec les modèles booléens :

- Pour toute formule ϕ : $\quad [\![\phi]\!] \;:=\; \{p \in P : p \Vdash \neg\neg\phi\} \;\in\; \mathcal{B}_P$
  où $\mathcal{B}_P$ est l'algèbre de Boole (complète) induite par P

Traduction logique & transformation de programmes

- **Forcing en réalisabilité classique** [Krivine '09]
  - Combinaison du forcing et de la réalisabilité classique
  - Découverte de la transformation de programmes sous-jacente
  - Cas d'étude : comment forcer un bon ordre sur $\mathbb{R}$
- **Analyse calculatoire du forcing** [M. '11]
  - Reformulation en arithmétique classique d'ordre supérieur (PAω)
  - Termes de preuves classiques à la Curry

$$t, u \quad ::= \quad x \quad | \quad \lambda x\,.\,t \quad | \quad tu \quad | \quad \mathfrak{cc} \qquad (= \text{call/cc})$$

  - Traduction logique + transformation de programmes (non typée) :

$$t \ : \ \phi \qquad \mapsto \qquad t^* \ : \ (p \Vdash \phi) \qquad (\text{dans PA}\omega)$$

  - Internalisation de $t \mapsto t^*$ dans la machine abstraite

Aperçu de la transformation de programmes $t \mapsto t^*$

- La transformation de programmes $t \mapsto t^*$ est paramétrée par des combinateurs de preuves $\alpha_*, \alpha_1, \ldots, \alpha_{15}$ exprimant la bonne formation de l'ensemble des conditions de forcing

$$
\begin{aligned}
x^* &\equiv x \\
(t\,u)^* &\equiv \gamma_3\, t^*\, u^* \\
(\lambda x\,.\,t)^* &\equiv \gamma_1\,(\lambda x\,.\,t^*\,\underbrace{\{x := \beta_4 x\}}_{\text{var. liée}}\,\underbrace{\{x_i := \beta_3 x_i\}_{i=1}^{n}}_{\text{var. libres de } t})
\end{aligned}
$$

$$
\begin{aligned}
\text{cc}^* &\equiv \lambda c x\,.\,\text{cc}\,(\lambda k\,.\,x\,(\alpha_{14}\,c)\,(\gamma_4\,k))
\end{aligned}
$$

$$
\begin{aligned}
\gamma_3 &\equiv \lambda x y c\,.\,x\,(\alpha_{11}\,c)\,y \\
\gamma_1 &\equiv \lambda x c y\,.\,x\,y\,(\alpha_6\,c) \\
\beta_3 &\equiv \lambda x c\,.\,x\,(\alpha_9\,c) \\
\beta_4 &\equiv \lambda x c\,.\,x\,(\alpha_{10}\,c) \\
\gamma_4 &\equiv \lambda x c y\,.\,x\,(y\,(\alpha_{15}\,c))
\end{aligned}
$$

Correction

Si $t : \phi$, alors $t^* : p \Vdash \phi$ (dans PAω)

Le forcing comme transformation de modèle... de calcul !

Le forcing peut être vu comme :

une transformation de modèles	$\mathscr{M}$	$\mapsto$	$\mathscr{M}[G]$
une traduction logique	ϕ	$\mapsto$	$p \Vdash \phi$
une transformation de programmes	t	$\mapsto$	t^*
une transformation de modèle de calcul	KAM	$\mapsto$	KFAM

Idée : Pousser la transformation de programmes $t \mapsto t^*$ dans la machine abstraite (le "hardware")

Au lieu d'évaluer le programme transformé (t^*) dans la machine usuelle (KAM)
on évalue le programme source (t) dans une machine transformée (KFAM)

$$\langle t^* \mid \mathrm{KAM} \rangle \quad \rightsquigarrow \quad \langle t \mid \mathrm{KFAM} \rangle$$

Krivine Forcing Abstract Machine (KFAM)

Termes	t, u	$::=$	x	$\mid$	$\lambda x\,.\,t$	$\mid$	tu	$\mid$ $\mathfrak{cc}$
Environnements	e	$::=$	$\emptyset$	$\mid$	$e; x := c$			
Fermetures	c	$::=$	$t[e]$	$\mid$	k_π	$\mid$	$\underbrace{t[e]^*}$	$\mid$ $\underbrace{\mathsf{k}_\pi^*}$
Piles	π	$::=$	$\diamond$	$\mid$	$c \cdot \pi$		fermetures de forcing	

- Mode réel :

$$
\begin{array}{rcl@{\qquad}l}
x[e; y =: c] \star \pi & \succ & x[e] \star \pi & (y \not\equiv x) \\
x[e; x =: c] \star \pi & \succ & c \star \pi & \\
(\lambda x\,.\,t)[e] \star c \cdot \pi & \succ & t[e; x := c] \star \pi & \\
(tu)[e] \star \pi & \succ & t[e] \star u[e] \cdot \pi & \\
\mathfrak{cc}[e] \star uc \cdot \pi & \succ & c \star \mathsf{k}_\pi \cdot \pi & \\
\mathsf{k}_\pi \star c \cdot \pi' & \succ & c \star \pi &
\end{array}
$$

- Mode forcing :

$$
\begin{array}{rcl@{\qquad}l}
x[e; y := c]^* \star c_0 \cdot \pi & \succ & x[e]^* \star \alpha_9\, c_0 \cdot \pi & (y \not\equiv x) \\
x[e; x := c]^* \star c_0 \cdot \pi & \succ & c \star \alpha_{10}\, c_0 \cdot \pi & \\
(\lambda x\,.\,t)[e]^* \star c_0 \cdot c \cdot \pi & \succ & t[e; x := c]^* \star \alpha_6\, c_0 \cdot \pi & \\
(tu)[e]^* \star c_0 \cdot \pi & \succ & t[e]^* \star \alpha_{11}\, c_0 \cdot u[e]^* \cdot \pi & \\
\mathfrak{cc}[e]^* \star c_0 \cdot c \cdot \pi & \succ & c \star \alpha_{14}\, c_0 \cdot \mathsf{k}_\pi^* \cdot \pi & \\
\mathsf{k}_\pi^* \star c_0 \cdot c \cdot \pi' & \succ & c \star \alpha_{15}\, c_0 \cdot \pi &
\end{array}
$$

Plan

Différentes notions de modèles (1/2)

- **Modèles de Tarski :** $[\![\phi]\!] \in \{0,1\}$
  - Interprète la prouvabilité classique (correction/complétude)
- **Modèles Booléens :** $[\![\phi]\!] \in \mathcal{B}$
  - $\mathcal{B} =$ algèbre de Boole complète
  - Forcing de Cohen, modèles produits
  - Souvent utilisés à travers un quotient par un ultrafiltre $\mathcal{U} \subseteq \mathcal{B}$

    $$\mathscr{M}/\mathcal{U} = \text{modèle de Tarski}$$

    (extensions génériques, modèles de l'analyse non standard)
- **Modèles de Heyting :** $[\![\phi]\!] \in \mathcal{H}$
  - $\mathcal{H} =$ algèbre de Heyting complète
  - Forcing de Kripke, *step indexing*
  - Interprète la prouvabilité intuitionniste

Différentes notions de modèles (2/2)

- **Réalisabilité intuitionniste :** $[\![\phi]\!] \in \mathfrak{P}(\Lambda)$ [Kleene '45]
  - Λ = langage de programmation fonctionnelle
  - Interprète les preuves intuitionnistes
  - Modèle basé sur la sémantique opérationnelle des programmes (preuves) intuitionnistes
  - Généralise les modèles de Tarski (cas dégénéré : $\Lambda = \{\bullet\}$)
- **Réalisabilité classique :** $[\![\phi]\!] \in \mathfrak{P}(\Lambda_c)$ [Krivine '94]
  - Λ_c = langage de programmation fonctionnelle avec contrôle
  - Interprète les preuves classiques (call/cc : loi de Peirce)
  - Modèle basé sur la sémantique opérationnelle des programmes (preuves) classiques
  - Généralise les modèles de Tarski... et le forcing de Cohen

L'interprétation de Brouwer-Heyting-Kolmogorov (BHK)

- **Principe philosophique :** la signification d'une formule ϕ est donnée par l'ensemble $[\![\phi]\!]$ des preuves ("*evidences*") de ϕ :

$$
\begin{aligned}
[\![\phi \wedge \psi]\!] &= [\![\phi]\!] \times [\![\psi]\!] && \text{(produit cartésien)} \\
[\![\phi \vee \psi]\!] &= [\![\phi]\!] + [\![\psi]\!] && \text{(union disjointe)} \\
[\![\bot]\!] &= \varnothing && \text{(ensemble vide)} \\
[\![\phi \Rightarrow \psi]\!] &= [\![\phi]\!] \to [\![\psi]\!] && \text{(fonctions calculables)} \\
[\![(\forall x \in \mathbb{N})\ \phi(x)]\!] &= \prod_{n \in \mathbb{N}} [\![\phi(n)]\!] && \text{(produit dépendant)} \\
[\![(\exists x \in \mathbb{N})\ \phi(x)]\!] &= \sum_{n \in \mathbb{N}} [\![\phi(n)]\!] && \text{(somme dépendante)}
\end{aligned}
$$

- **Exemple typique :** $(\forall x \in \mathbb{N})(\exists y \in \mathbb{N})\ \phi(x, y)$

De la philosophie aux mathématiques

On peut donner à l'interprétation philosophique de BHK un contenu mathématique : la théorie de la réalisabilité [Kleene '45]

- Chaque formule ϕ de l'arithmétique (intuitionniste) est interprétée par l'ensemble $[\![\phi]\!]$ des programmes qui réalisent ϕ
- **Théorème :** Si $\mathrm{HA} \vdash \phi$, alors ϕ est réalisable

Note : HA = arithmétique de Heyting = arithmétique intuitionniste du premier ordre

La théorie de la réalisabilité s'étend à :

- L'arithmétique (intuitionniste) du second ordre, d'ordre supérieur
- La théorie des ensembles intuitionniste (IZF) [Myhill-Friedman '73]
- Interprétation catégorique de la réalisabilité + lien avec les topos [Hyland-Johnstone-Pitts '80 ; Hyland '82]

Algèbres combinatoires partielles

Chaque modèle de réalisabilité (de HA, HA2, HAω, IZF, ...) est paramétré par un langage de programmation abstrait :

Définition (Algèbre combinatoire partielle)

Une algèbre combinatoire partielle (PCA) est un ensemble P muni d'une fonction partielle $(p, q) \mapsto p\,q$ de P^2 dans P ($\ll$ application $\gg$) et de deux éléments ($\ll$ combinateurs $\gg$) $\mathbf{K}, \mathbf{S} \in P$ tels que :

- $\mathbf{K}\,p\downarrow$, $(\mathbf{K}\,p)\,q\downarrow$, et $(\mathbf{K}\,p)\,q = p$
- $\mathbf{S}\,p\downarrow$, $(\mathbf{S}\,p)\,q\downarrow$; et
  $((\mathbf{S}\,p)\,q)\,r\downarrow$ ssi $(p\,r)\,(q\,r)\downarrow$, auquel cas : $((\mathbf{S}\,p)\,q)\,r = (p\,r)\,(q\,r)$

Exemples :

- $P := \mathbb{N}$, muni de l'application de Kleene $n \cdot m := f_n(m)$
  (où f_n est la $n$ième fonction récursive partielle)
- $P := \Lambda/=_\beta$ (ensemble des λ-termes clos, modulo β-équivalence), muni de l'application du λ-calcul

Interprétation de l'arithmétique intuitionniste

Étant donnée une algèbre combinatoire partielle P, on pose :

$$[\![n = m]\!] := \begin{cases} \{\overline{0}\} & \text{si } n = m \\ \varnothing & \text{si } n \neq m \end{cases} \qquad [\![\bot]\!] := \varnothing$$

$$[\![\phi \wedge \psi]\!] := \{\langle p, q\rangle \ : \ p \in [\![\phi]\!] \ \text{ et } \ q \in [\![\psi]\!]\}$$

$$[\![\phi \vee \psi]\!] := \{\langle \overline{0}, p\rangle \ : \ p \in [\![\phi]\!]\} \cup \{\langle \overline{1}, q\rangle \ : \ q \in [\![\psi]\!]\}$$

$$[\![\phi \Rightarrow \psi]\!] := \{p \in P \ : \ \forall q \in [\![\phi]\!], \ \ p\,q\downarrow \text{ et } p\,q \in [\![\psi]\!]\}$$

$$[\![\forall x\, \phi(x)]\!] := \{p \in P \ : \ \forall n \in \mathbb{N}, \ \ p\,\overline{n}\downarrow \text{ et } p\,\overline{n} \in [\![\phi(n)]\!]\}$$

$$[\![\exists x\, \phi(x)]\!] := \{\langle \overline{n}, p\rangle \ : \ n \in \mathbb{N} \ \text{ et } \ p \in [\![\phi(n)]\!]\}$$

La relation de réalisabilité est définie par : $p \Vdash \phi$ ssi $p \in [\![\phi]\!]$

Théorème (Adéquation) [Kleene '45]

Si $\mathsf{HA} \vdash \phi$, alors $p \Vdash \phi$ pour un certain $p \in P$ (dans toute PCA P)

Du forcing à la réalisabilité

Forcing	**Réalisabilité**
p, q = conditions de forcing	p, q = programmes
$p \Vdash \phi$	$p \Vdash \phi$
$\dfrac{p \Vdash \phi \Rightarrow \psi \qquad q \Vdash \phi}{\underbrace{p\,q}_{\text{borne inf.}} \Vdash \psi}$	$\dfrac{p \Vdash \phi \Rightarrow \psi \qquad q \Vdash \phi}{\underbrace{p\,q}_{\text{application}} \Vdash \psi}$
$\dfrac{p \Vdash \phi \qquad q \Vdash \psi}{p\,q \Vdash \phi \wedge \psi}$	$\dfrac{p \Vdash \phi \qquad q \Vdash \psi}{\langle p, q\rangle \Vdash \phi \wedge \psi}$
$[\![\phi \wedge \psi]\!] = [\![\phi]\!] \cap [\![\psi]\!]$	$[\![\phi \wedge \psi]\!] = [\![\phi]\!] \times [\![\psi]\!]$

Slogan : Réalisabilité = Forcing non commutatif

Interprétation catégorique de la réalisabilité

- Logique catégorique :
  - Hyperdoctrines, Topos élémentaires [Lawvere '63, '68]
  - Lien entre le forcing et les topos de faisceaux [Fourman, Scott '77]
  - Topos effectif, théorie des tripos [Hyland, Johnstone, Pitts '80]

- **Tripos** = modèle catégorique de la logique d'ordre supérieur
  - On peut construire un tripos à partir :
    - d'une algèbre de Heyting complète (forcing intuitionniste)
    - d'une algèbre de Boole complète (forcing classique)
    - d'une algèbre combinatoire partielle (réalisabilité intuitionniste)
  - Tout tripos engendre un topos ("tripos-to-topos construction")
  - Ce qui permet de construire :
    - Des topos de forcing (intuitionnistes et classiques)
    - Des topos de réalisabilité intuitionniste (à partir des PCA)

Quid de la réalisabilité classique ? (1/2)

Théorème : La logique d'un topos de réalisabilité est classique si et seulement si la PCA sous-jacente est dégénérée : $P = \{\bullet\}$

Une réalisabilité classique impossible ? Et pourtant...

La réalisabilité classique [Krivine '94, '00, '03, '11, ...]

- Reformulation complète des principes de la réalisabilité pour les rendre compatibles avec la logique classique
- Basée sur la correspondence entre les principes de raisonnement classiques et les opérateurs de contrôle découverte par Griffin '90 :

$$\text{call/cc} \ : \ ((A \Rightarrow B) \Rightarrow A) \Rightarrow A \qquad \text{(loi de Peirce)}$$

- S'étend à la théorie des ensembles (ZF)
- Interprète l'axiome des choix dépendants (DC)
- Liens profonds avec le forcing de Cohen

Quid de la réalisabilité classique ? (2/2)

- Streicher'12 : *Krivine's classical realizability from a categorical perspective*

  La réalisabilité de Krivine ne dérive pas d'une algèbre combinatoire partielle (PCA), mais d'une algèbre combinatoire ordonnée (OCA)

Définition (Algèbre combinatoire ordonnée)

Une algèbre combinatoire ordonnée (OCA) est un ensemble P muni d'une fonction totale $(p, q) \mapsto p\,q$ de P^2 dans P (« application ») et de deux éléments (« combinateurs ») $\mathbf{K}, \mathbf{S} \in P$ tels que :

$$(\mathbf{K}\,p)\,q \;\leq\; p \qquad \text{et} \qquad ((\mathbf{S}\,p)\,q)\,r \;\leq\; (p\,r)\,(q\,r)$$

- Ferrer *et al* '15 : *Ordered combinatory algebras and realizability*
- **Intuition :** PCA $=$ point de vue de la β-équivalence
  OCA $=$ point de vue de la β-réduction
- À l'instar des PCA, chaque OCA induit un tripos (intuitionniste ou classique), qui engendre à son tour un topos

Les cardinaux hérétiques de la réalisabilité classique

Dans *Realizability algebras II : new models of ZF + DC* (2012), Krivine présente un modèle de réalisabilité de ZF + DC dans lequel :

1. Il existe un ensemble infini $S \subseteq \mathbb{R}$ qui n'est pas équipotent à $S \times S$
2. Il existe des ensembles infinis $S_1, S_2 \subseteq \mathbb{R}$ dont les cardinaux (intuitifs) sont incomparables
3. Il existe une famille d'ensembles infinis $(S_q)_{q \in \mathbb{Q}}$ indicée par $\mathbb{Q}$ dont les cardinaux intuitifs sont strictement croissants (ordre dense !)

Chacun de ces trois énoncés implique $\neg$AC et $\neg$HC

Une telle construction n'est pas possible par forcing (seul)

(Pourtant, l'algèbre de réalisabilité sous-jacente est dénombrable...)

Plan

Quelle algèbre de valeurs de vérité

Qu'est-ce qu'une algèbre de valeurs de vérité ?

... pour la logique intuitionniste ou la logique classique

- Une algèbre de Heyting complète ? (forcing de Kripke)
- Une algèbre de Boole complète ? (forcing de Cohen)
- Un ensemble de parties d'une PCA ? (réalisabilité intuitionniste)
- Un ensemble de parties d'une OCA ? (réalisabilité int. et classique)

$\rightsquigarrow$ Algèbres implicatives

Algèbres implicatives

Définition (Algèbre implicative) [M. 2016]

- Une structure implicative est un triplet $(\mathscr{A}, \preccurlyeq, \to)$ où
  - (1) $(\mathscr{A}, \preccurlyeq)$ est un treillis complet
  - (2) $(\to) : \mathscr{A}^2 \to \mathscr{A}$ est une opération binaire telle que :
    - (2a) $a' \preccurlyeq a,\ \ b \preccurlyeq b'$ implique $(a \to b) \preccurlyeq (a' \to b')$ $\qquad (a, a', b, b' \in \mathscr{A})$
    - (2b) $\bigwedge_{b \in B} (a \to b) \ = \ a \to \bigwedge_{b \in B} b$ $\qquad (a \in \mathscr{A},\ B \subseteq \mathscr{A})$
- Un séparateur de $\mathscr{A}$ est un sous-ensemble $S \subseteq \mathscr{A}$ tel que :
  - (1) Si $a \in S$ et $a \preccurlyeq b$, alors $b \in S$ (clôture supérieure)
  - (2) $\bigwedge_{a,b \in \mathscr{A}} (a \to b \to c) \ \ (= \mathbf{K}^{\mathscr{A}}) \ \in \ S$ et
    $\bigwedge_{a,b,c \in \mathscr{A}} ((a \to b \to c) \to (a \to b) \to a \to c) \ \ (= \mathbf{S}^{\mathscr{A}}) \ \in \ S$
  - (3) Si $(a \to b) \in S$ et $a \in S$, alors $b \in S$ (clôture par modus ponens)
- Algèbre implicative $=$ structure implicative + séparateur

Valeurs de vérité = réalisateurs généralisés

1. Les éléments de $\mathscr{A}$ représentent des valeurs de vérité ou des types. Mais comme la λ-abstraction et l'application sont définissables dans $\mathscr{A}$ (cf diapo suivante), on peut voir :
   - chaque réalisateur comme une valeur de vérité (≪type principal≫)
   - chaque valeur de vérité comme un réalisateur généralisé
2. On obtient alors l'identification de Curry-Howard ultime :

   Réalisateur = Programme = Formule = Type
3. Dans ce cadre, la relation $a \preccurlyeq b$ peut se lire :
   - a est un sous-type de b (a, b valeurs de vérité)
   - a est de type b (a réalisateur, b valeur de vérité)
   - a est plus défini que b (a, b réalisateurs)
4. En particulier : sous-typage ($\preccurlyeq$) = ordre de Scott inversé ($\sqsupseteq$)

Representation des λ-termes comme des valeurs de vérité

Soit $\mathscr{A} = (\mathscr{A}, \preccurlyeq, \to)$ une structure implicative

1. Étant donnés $a, b \in \mathscr{A}$ et une function $f : \mathscr{A} \to \mathscr{A}$, on pose :

$$ab := \bigwedge \{c \in \mathscr{A} \ : \ a \preccurlyeq (b \to c)\} \qquad \text{(application)}$$

$$\lambda f := \bigwedge_{a \in \mathscr{A}} (a \to f(a)) \qquad \text{(abstraction)}$$

Remarque : Ces constructions sont monotones par rapport à a, b et f

2. On peut donc représenter chaque λ-terme t (à paramètres dans $\mathscr{A}$) comme une valeur de vérité $t^{\mathscr{A}} \in \mathscr{A}$. La correspondance $t \mapsto t^{\mathscr{A}}$ n'est *pas* injective en général (sur les formes $\beta\eta$-normales) ; mais :
   - β-réduction : Si $t \twoheadrightarrow_\beta t'$, alors $(t)^{\mathscr{A}} \preccurlyeq (t')^{\mathscr{A}}$
   - η-réduction : Si $t \twoheadrightarrow_\eta t'$, alors $(t)^{\mathscr{A}} \succcurlyeq (t')^{\mathscr{A}}$

3. Adjonction fondamentale : $ab \preccurlyeq c \quad \Leftrightarrow \quad a \preccurlyeq (b \to c)$

Identités remarquables

- Dans toute structure implicative $\mathscr{A} = (\mathscr{A}, \preccurlyeq, \to)$:

$$\mathbf{I}^{\mathscr{A}} := (\lambda x\,.\,x)^{\mathscr{A}} = \bigwedge_{a}(a \to a)$$

$$\mathbf{K}^{\mathscr{A}} := (\lambda xy\,.\,x)^{\mathscr{A}} = \bigwedge_{a,b}(a \to b \to a)$$

$$\mathbf{S}^{\mathscr{A}} := (\lambda xyz\,.\,xz(yz))^{\mathscr{A}} = \bigwedge_{a,b,c}((a \to b \to c) \to (a \to b) \to a \to c)$$

  + égalités similaires pour $\mathbf{C} \equiv \lambda xyz\,.\,xzy$ et $\mathbf{W} \equiv \lambda xy\,.\,xyy$

- Par analogie, on pose :

$$\mathrm{cc}^{\mathscr{A}} := \bigwedge_{a,b}(((a \to b) \to a) \to a) \qquad \text{(Peirce's law)}$$

L'interprétation du λ-calcul s'étend aux λ-termes avec call/cc

Propriétés des séparateurs

Rappel : Algèbre implicative $=$
structure implicative $(\mathscr{A}, \preccurlyeq, \to)$ $+$ séparateur $S \subseteq \mathscr{A}$

- Un séparateur $S \subseteq \mathscr{A}$ définit un critère de vérité. Il est :
  - cohérent si $\bot \notin S$
  - classique si $\mathfrak{cc}^{\mathscr{A}} \in S$
- Intuitions : Séparateur $=$ ensemble des «types prouvables»
  $=$ ensemble des «preuves légales»
  $=$ filtre non commutatif
- Les constructions usuelles sur les filtres s'étendent aux séparateurs :
  lemme de déduction, quotient par un séparateur, ultraséparateur, etc.
- Un séparateur est un filtre (situation de forcing) si et seulement si

$$\pitchfork^{\mathscr{A}} := \bigwedge_{a,b} (a \to b \to a \curlywedge b) \in S \qquad \text{(choix non déterministe)}$$

Slogan : Forcing $=$ réalisabilité non déterministe

Le tripos implicatif

- Chaque algèbre implicative induit un tripos : le tripos implicatif
- Cette construction factorise (et généralise) toutes les constructions de tripos connues jusqu'ici :
  - tripos de forcing (Kripke, Cohen)
  - tripos de réalisabilité (intuitionniste et classique)

Théorème (Caractérisation des tripos de forcing) [M. 2016]

Le tripos induit par une algèbre implicative $(\mathscr{A}, S)$ est un tripos de forcing (à iso. près) si et seulement si S est finiment engendré et $\pitchfork^{\mathscr{A}} \in S$

Théorème (Tripos implicatifs classiques) [Guillermo, Malherbe, Ferrer 2016]

Les tripos implicatifs classiques sont équivalents aux tripos de Krivine

Théorème (Complétude) [M. 2018]

Tous les tripos (sur **Set**) sont des tripos implicatifs (à isomorphisme près)

- **Corollaire :** Tous les tripos classiques sont des tripos de Krivine