Programming = proving?

The Curry-Howard correspondence today

Seventh lecture

Forcing : just another program transformation?

Xavier Leroy

Collège de France

2019-01-09

Or voici qu'il y a huit mois Kan, travaillant sur un adjoint à lui (voir D. Kan, Adjoint Functors, *Transactions*, V, 3,18) montra par induction, croit-on, (il raisonnait — a-t-il dit à Jaulin — sur un grand cardinal, par "forcing" pour part) la

Proposition Soit G soit H soit K ($H \subset G$, $G \supset K$) trois magmas (nous suivons Kurosh) où l'on a $a(bc) = (ab)c$; où pour tout a, $x \to xa$, $x \to ax$ sont "sûrs", sont monos, alors on a $G \simeq H \times K$ si $G = H \cup K$; si H, si K sont invariants; si H, K n'ont qu'un individu commun $H \cap K =$

Las! Kan mourut avant d'avoir fini son job. Donc à la fin, l'on n'a toujours pas la solution (1).

G. Perec, *La disparition*, pp. 62–63 (1969)

I

The continuum hypothesis

Cardinals

A generalization (by Cantor) of the notion of number of elements to infinite sets.

Two sets X and Y have the same cardinal if and only if there exists a bijection h between X and Y.

Cardinals

The order between cardinals:

- $\text{card}(X) = \text{card}(Y)$ if there exists a bijection $X \to Y$.
- $\text{card}(X) \leq \text{card}(Y)$ if there exists an injection $X \to Y$.
- $\text{card}(X) < \text{card}(Y)$ if there exists an injection $X \to Y$ but no injection $Y \to X$

Theorem (Cantor, 1874, 1891)

$card(X) < card(\mathcal{P}(X)) = card(X \to \{0, 1\})$ for all set X.

Corollary: $card(\mathbb{N}) < card(\mathbb{R})$.

Two kinds of infinity

Countable infinity	Continuous infinity
$\mathbb{N}$	$\mathbb{R}$
$\mathbb{Z}$	$\mathcal{P}(\mathbb{N})$
$\mathbb{Q}$	$\mathbb{C}$
$\mathbb{N} \times \cdots \times \mathbb{N}$	$\mathbb{R} \times \cdots \times \mathbb{R}$
finite words on a finite alphabet	$\mathbb{N} \to \{0, 1, \ldots, k\}$
finite words on $\mathbb{N}$	$\mathbb{N} \to \mathbb{N}$
mathematical formulas	
computer programs	
Turing machines	
computable functions	

The continuum hypothesis (CH)

There is no cardinal between countable infinity and continuous infinity.

$$\neg\exists X,\ \ \mathrm{card}(\mathbb{N}) < \mathrm{card}(X) < \mathrm{card}(\mathcal{P}(\mathbb{N}))$$

In other words: every subset of $\mathbb{R}$ is either finite, or countable, or in bijection with $\mathbb{R}$.

The generalized continuum hypothesis (GCH)

Enumerating infinite cardinals: (uses the axiom of choice)

$$\aleph_0 = \mathrm{card}(\mathbb{N}) \qquad \aleph_{\alpha+1} = \text{the smallest cardinal} > \aleph_\alpha \qquad \aleph_\lambda = \sup_{\alpha<\lambda} \aleph_\alpha$$

By Cantor's theorem: $\aleph_{\alpha+1} \leq 2^{\aleph_\alpha}$ for all α.

Continuum hypothesis: $\aleph_1 = 2^{\aleph_0}$

Generalized continuum hypothesis: $\aleph_{\alpha+1} = 2^{\aleph_\alpha}$ for all α.

History of the problem

- 1878: G. Cantor states the continuum hypothesis. He could never prove it.
- 1900: D. Hilbert lists CH first in his list of 23 open problems.
- 1938: K. Gödel proves that GCH is consistent with ZFC set theory.
- 1964: P. Cohen proves that the negation of CH is consistent with ZFC. To this end, he develops an entirely new approach: *forcing*. He receives the Fields medal in 1966.
- 1970: W. B. Easton proves consistency of a generalization of $\neg$CH: for all α, $\aleph_{\alpha+1} < 2^{\aleph_\alpha}$.

Independence of the continuum hypothesis

(Generalized) continuum hypothesis is therefore independent of ZF, Zermelo-Fraenkel set theory, meaning:

- We can assume CH to be true (take it as an axiom) and no contradiction (logical inconsistency) follows.
- We can assume CH to be false (take its negation as an axiom) and no contradiction follows.
- As a corollary, we cannot prove CH nor $\neg$CH from the axioms of ZF.

Another example: the axiom of choice is independent of ZF.
(Proved at the same time as independence of CH by Gödel and by Cohen.)

Models of set theory

ZF set theory:

A symbol "$\in$" and 8 axioms:

Extensionality

Pairing

Comprehension

Union

Power set

Infinity

Replacement

Foundation

The structure of groups:

Three symbols "1", "$\cdot$" and "$^{-1}$" and three identities:

$$(x \cdot y) \cdot z = x \cdot (y \cdot z)$$

$$1 \cdot x = x = x \cdot 1$$

$$x \cdot x^{-1} = 1 = x^{-1} \cdot x$$

A model of set theory:

A collection of objects and a predicate $\in$ that satisfy the 8 axioms.

A group:

A set G and operations $(1, \cdot, ^{-1})$ that satisfy the 3 identities.

Models of set theory

The existence of a model of ZF proves the consistency of ZF axioms (we cannot prove absurdity $\perp$).

Conversely: if ZF is consistent, it has a model (Gödel, 1930).

The existence of a model of ZF satisfying an hypothesis H shows that ZF $+ H$ is consistent, and therefore that we cannot prove $\neg H$ from ZF axioms.

Gödel's 1938 proof: given a model M of ZF, build an inner model $L \subseteq M$ that satisfies CH.

Cohen's 1964 proof: given a model M of ZF, build an extension of this model $M[G] \supset M$ that satisfies $\neg$CH.

Models of set theory

The existence of a model of ZF proves the consistency of ZF axioms (we cannot prove absurdity $\perp$).

Conversely: if ZF is consistent, it has a model (Gödel, 1930).

The existence of a model of ZF satisfying an hypothesis H shows that ZF $+$ H is consistent, and therefore that we cannot prove $\neg H$ from ZF axioms.

Gödel's 1938 proof: given a model M of ZF, build an inner model $L \subseteq M$ that satisfies CH.

Cohen's 1964 proof: given a model M of ZF, build an extension of this model $M[G] \supset M$ that satisfies $\neg$CH.

Models of set theory

The existence of a model of ZF proves the consistency of ZF axioms (we cannot prove absurdity $\perp$).

Conversely: if ZF is consistent, it has a model (Gödel, 1930).

The existence of a model of ZF satisfying an hypothesis H shows that ZF $+ H$ is consistent, and therefore that we cannot prove $\neg H$ from ZF axioms.

Gödel's 1938 proof: given a model M of ZF, build an inner model $L \subseteq M$ that satisfies CH.

Cohen's 1964 proof: given a model M of ZF, build an extension of this model $M[G] \supset M$ that satisfies $\neg$CH.

Models of set theory

The existence of a model of ZF proves the consistency of ZF axioms (we cannot prove absurdity $\perp$).

Conversely: if ZF is consistent, it has a model (Gödel, 1930).

The existence of a model of ZF satisfying an hypothesis H shows that ZF $+ H$ is consistent, and therefore that we cannot prove $\neg H$ from ZF axioms.

Gödel's 1938 proof: given a model M of ZF, build an inner model $L \subseteq M$ that satisfies CH.

Cohen's 1964 proof: given a model M of ZF, build an extension of this model $M[G] \supset M$ that satisfies $\neg$CH.

Gödel's constructible sets

Let $(M, \in)$ be a model of ZF.

If X is a set from this model, we write $\mathit{Def}(X)$ the set of sets definable by logical formulas Φ where all variables (quantified or free) range over X:

$$\mathit{Def}(X) = \left\{ \, \{x \in X \mid (X, \in) \models \Phi(x) \} \, \right\}$$

Define by transfinite induction:

$$L_0 = \emptyset \qquad L_{\alpha+1} = \mathit{Def}(L_\alpha) \qquad L_\lambda = \bigcup_{\alpha<\lambda} L_\alpha$$

In other words: L_α is all the sets that we can construct using only members of L_β with $\beta < \alpha$.

Gödel's constructible sets

If $(M, \in)$ is a model of ZF, and *Ord* the collection of its ordinals, we define $L = \bigcup_{\alpha \in Ord} L_\alpha$. Then, $(L, \in)$ is a model of ZF. Moreover:

- L satisfies the axiom of choice.
  (Every set A of L is well ordered by an order induced by ordinal order.)

- L satisfies the generalized continuum hypothesis.
  (For all α, $\mathcal{P}(L_\alpha) \cap L \subseteq L_\beta$ for a β "not much bigger than" α. It follows that $2^{\aleph_\gamma} \leq \aleph_{\gamma+1}$ and therefore $\aleph_{\gamma+1} = 2^{\aleph_\gamma}$.)

Cohen's generic extensions

In Gödel's approach, we start from a model M and we keep only the "well-behaved" sets of M (those that are constructible), thus eliminating "wild" sets that could have intermediate cardinals and thus invalidate CH.

Cohen's approach is dual: we start from a model M and we adjoin it a new set G that will "inflate $\mathcal{P}(\mathbb{N})$" so much that $\aleph_0 < \aleph_1 < 2^{\aleph_0}$ in the resulting model $M[G]$.

Extension of an algebraic structure

A familiar mathematical concept. For instance:

- If we add an element X to a ring A, we also add $2X$, $-X$, X^2, X^3, $\ldots$, and we get $A[X]$, the ring of polynomials over A.
- If we extend the field $\mathbb{R}$ with an element i such that $i^2 = -1$, we also add all the $x + iy$, and we get $\mathbb{C}$.

Careful! An extension can be inconsistent! For instance:

- If we extend the ordered field $\mathbb{R}$ with an element i such that $i^2 = -1$, we contradict the property $\forall x,\ x^2 \geq 0$ which was true before the extension.

Extension of an algebraic structure

A familiar mathematical concept. For instance:

- If we add an element X to a ring A, we also add $2X$, $-X$, X^2, X^3, $\ldots$, and we get $A[X]$, the ring of polynomials over A.
- If we extend the field $\mathbb{R}$ with an element i such that $i^2 = -1$, we also add all the $x + iy$, and we get $\mathbb{C}$.

Careful! An extension can be inconsistent! For instance:

- If we extend the ordered field $\mathbb{R}$ with an element i such that $i^2 = -1$, we contradict the property $\forall x,\ x^2 \geq 0$ which was true before the extension.

Cohen's proof

- Let M be a transitive countable model of ZF.
- Let k be a set of M such that $M \models \mathrm{card}(k) = \aleph_2$.
- Extend M with a new element G that is a "generic" function from k to $\mathcal{P}(\mathbb{N})$, giving $M[G]$.
- Show that $M[G]$ is a model of ZF.
- Show that $M[G] \models$ "function G is injective", and therefore that $M[G] \models \mathrm{card}(k) \leq \mathrm{card}(\mathcal{P}(\mathbb{N})) = 2^{\aleph_0}$.
- Show that cardinals are preserved by the extension, and therefore that $M[G] \models \mathrm{card}(k) = \aleph_2$.
- Conclude $M[G] \models \aleph_0 < \aleph_1 < \aleph_2 \leq 2^{\aleph_0}$, and therefore $M[G] \models \neg\mathrm{CH}$.

II

Forcing

Forcing conditions

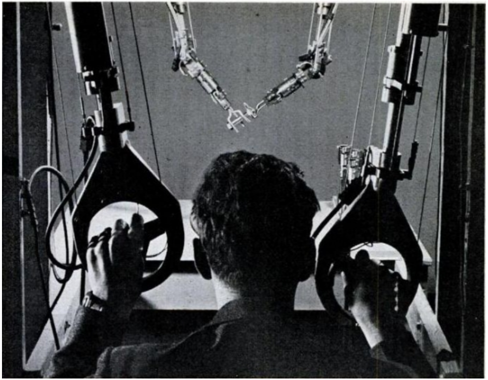
Constructing the model extension $M[G]$ is not very hard; but how can we reason in this model?

What are the properties of G?

How to prove that a logical formula is true in $M[G]$?

Cohen's idea: we can describe G and its properties through finite approximations (but as precise as we want) that live in M and that we call forcing conditions.

Forcing conditions



Dangerous object being handled: $M[G]$.
Handles of the remote manipulator: forcing conditions.

Forcing conditions

Definition

A set of forcing conditions is a partially-ordered set $(\mathcal{C}, \preccurlyeq)$.
$q \preccurlyeq p$ means that condition q is "finer" than condition p, or equivalently that q implies p.

Example

If the generic element G is a set of integers, we take as forcing conditions p the finite functions from integers to $\{0, 1\}$, such as $\{4 \mapsto 1,\ 13 \mapsto 0\}$.

- $p(n) = 1$ means "n belongs to G"
- $p(n) = 0$ means "n does not belong to G"

We order conditions by reverse inclusion: $q \preccurlyeq p \stackrel{def}{=} p \subseteq q$.

Forcing predicates

Given a logical formula A that mentions elements of $M[G]$, we say that A is forced by condition p, and write $p \Vdash_W A$, if:

$$
\begin{aligned}
p \Vdash_W n \in \bar{G} &\text{ iff } p(n) = 1 \\
p \Vdash_W A \wedge B &\text{ iff } p \Vdash_W A \text{ and } p \Vdash_W B \\
p \Vdash_W \neg A &\text{ iff } \forall q \preccurlyeq p,\ \neg(q \Vdash_W A) \\
p \Vdash_W \forall x \in X. A(x) &\text{ iff } p \Vdash_W A(x) \text{ for all } x \in X
\end{aligned}
$$

Remark: if $p \Vdash_W A$ then $q \Vdash_W A$ for all $q \preccurlyeq p$.

Theorem

1- For every extension $M[G]$ and every formula A,

$M[G] \models A$ if and only if there exists $p \in G$ such that $M \models (p \Vdash_W A)$.

2- For every p, there exists an extension $M[G]$ such that $p \in G$.

Example of use

Lemma

The generic set of integers G contains infinitely many prime numbers.

Proof.

We have to show $M[G] \models \forall m, \exists n, n \in \bar{G} \wedge n \geq m \wedge n$ prime.
By the forcing theorem, it suffices to show (in M)

$$\begin{aligned} & \emptyset \Vdash_W \forall m, \exists n, n \in \bar{G} \wedge n \geq m \wedge n \text{ prime} \\ \text{that is } & \emptyset \Vdash_W \forall m, \neg(\forall n, \neg(n \in \bar{G} \wedge n \geq m \wedge n \text{ prime})) \\ \text{that is } & \forall m, \forall p, \exists q \preccurlyeq p, \exists n,\ q(n) = 1 \wedge n \geq m \wedge n \text{ prime} \end{aligned}$$

The function p being finite and the set of prime numbers infinite, we can always find an $n \geq m$ prime and outside the domain of p. We then take $q = p \cup \{n \mapsto 1\}$ and we have $q \preccurlyeq p$ and $q(n) = 1$.

$\square$

Example of use

If G is the function $k \to \mathcal{P}(\mathbb{N})$ from Cohen's proof, we take as forcing conditions the finite functions $k \times \mathbb{N} \to_{fin} \{0, 1\}$, ordered by reverse inclusion.

We define $p \Vdash_W n \in \bar{G}(x)$ iff $p(x, n) = 1$.

Exercise: show that G is injective: $M[G] \models \forall x_1, x_2,\ x_1 \neq x_2 \Rightarrow G(x_1) \neq G(x_2)$.

Ideas that resonate

- Forcing (Cohen, 1963–1964)
  Set theory; classical logic.

- Kripke models (Kripke, 1959–1965)
  Modal logics, intuitionistic logic.

- The (pre-)sheave constructions (Lawvere and Tierney, 1971–1972)
  Category theory, topos.

Kripke models

A relation $p \Vdash_K A$, “formula A is true in world p”.

A world $p \approx$ a set of facts (atomic propositions).

Worlds are ordered: $q \preccurlyeq p$,
reads as “world q is accessible from world p”
and implies that q contains all the facts of p.

Intuitionistic Kripke models

$$\begin{aligned}
p \Vdash_K F(a_1, \ldots, a_n) &\text{ iff } F(a_1, \ldots, a_n) \in \mathit{Facts}(p) \\
p \Vdash_K A \wedge B &\text{ iff } p \Vdash_K A \text{ and } p \Vdash_K B \\
p \Vdash_K A \vee B &\text{ iff } p \Vdash_K A \text{ or } p \Vdash_K B \\
p \Vdash_K A \Rightarrow B &\text{ iff } \text{for all } q \preccurlyeq p,\ q \Vdash_K A \text{ implies } q \Vdash_K B \\
p \Vdash_K \neg A &\text{ iff } \forall q \preccurlyeq p,\ \neg(q \Vdash_K A) \\
p \Vdash_K \forall x. A(x) &\text{ iff } \text{for all } x,\ p \Vdash_K A(x) \\
p \Vdash_K \exists x. A(x) &\text{ iff } \text{there exists } x \text{ such that } p \Vdash_K A(x)
\end{aligned}$$

Monotonicity property:

$$p \Vdash_K A \wedge q \preccurlyeq p \Rightarrow q \Vdash_K A$$

(In red, the "minimal modification" that ensures monotonicity.)

Intuitionistic Kripke models

$$
\begin{array}{rcl}
p \Vdash_K F(a_1, \ldots, a_n) & \text{iff} & F(a_1, \ldots, a_n) \in \mathit{Facts}(p) \\
p \Vdash_K A \wedge B & \text{iff} & p \Vdash_K A \text{ and } p \Vdash_K B \\
p \Vdash_K A \vee B & \text{iff} & p \Vdash_K A \text{ or } p \Vdash_K B \\
p \Vdash_K A \Rightarrow B & \text{iff} & \text{for all } q \preccurlyeq p\text{, } q \Vdash_K A \text{ implies } q \Vdash_K B \\
p \Vdash_K \neg A & \text{iff} & \forall q \preccurlyeq p,\ \neg(q \Vdash_K A) \\
p \Vdash_K \forall x.\, A(x) & \text{iff} & \text{for all } x\text{, } p \Vdash_K A(x) \\
p \Vdash_K \exists x.\, A(x) & \text{iff} & \text{there exists } x \text{ such that } p \Vdash_K A(x)
\end{array}
$$

Monotonicity property:

$$p \Vdash_K A \wedge q \preccurlyeq p \Rightarrow q \Vdash_K A$$

(In red, the “minimal modification” that ensures monotonicity.)

Kripke models and modal logic

Kripke introduced these models (classical or intuitionistic) to study modal logics. Indeed, modalities have a natural interpretation in terms of quantification over accessible worlds:

$$p \Vdash_K \Box A \ \text{ iff } \ \forall q \preccurlyeq p,\ q \Vdash_K A$$

$$p \Vdash_K \Diamond A \ \text{ iff } \ \exists q \preccurlyeq p,\ q \Vdash_K A$$

Intuitionistic Kripke models

Intuitionistic Kripke models are also "the right model" for intuitionistic logic, because:

- Every formula A provable in intuitionistic logic is true in every world of every Kripke model: $p \Vdash_K A$.
- Classical laws (excluded middle, double negation elimination) are invalid in some worlds of some Kripke models.

Example

Let F be an atomic formula. Consider the two worlds p_0, p_1

$$p_1 \preccurlyeq p_0 \qquad Facts(p_0) = \emptyset \qquad Facts(p_1) = \{F\}$$

We have

$$\begin{array}{ll} p_0 \nVdash_K F & \\ p_0 \nVdash_K \neg F & \text{(because } p_1 \Vdash_K F\text{)} \\ p_0 \nVdash_K F \vee \neg F & \end{array}$$

Kripke models and forcing

There are striking similarities between

- forcing conditions and worlds;
- the relation $p \Vdash_W A$, "condition p forces formula A"
  and the relation $p \Vdash_K A$, "world p satisfies formula A".
  (To the point that some authors read $p \Vdash_K A$ as "p forces A".)

This leads to a theory of intuitionistic forcing based on Kripke models that proves Cohen's independence results for intuitionistic set theory.

(M. Fitting, *Intuitionistic logic model theory and forcing*, 1969)

Kripke models and forcing

Example

We take as worlds p the finite functions $\mathbb{N} \rightarrow_{fin} \{0, 1\}$, interpreted by $Facts(p) = \{\text{“}n \in G\text{”} \mid p(n) = 1\}$.

We cannot show directly $\emptyset \Vdash_K$ “G contains infinitely many prime numbers”, but we can show one of its double negations,

$$\emptyset \Vdash_K \forall m, \neg\neg(\exists n, n \in G \wedge n \geq m \wedge n \text{ prime})$$

$$\text{that is } \forall m, \forall p, \exists q \preccurlyeq p,\ q \Vdash_K \exists n, n \in G \wedge n \geq m \wedge n \text{ prime}$$

$$\text{that is } \forall m, \forall p, \exists q \preccurlyeq p, \exists n,\ q(n) = 1 \wedge n \geq m \wedge n \text{ prime}$$

Double negation and forcing

More generally, we recover the laws of the forcing predicate $\Vdash_W$ by composing $\Vdash_K$ with the Gödel-Gentzen negative translation (see lecture of Dec 5th 2018):

$$\llbracket A \Rightarrow B \rrbracket = \llbracket A \rrbracket \Rightarrow \llbracket B \rrbracket$$

$$\llbracket A \wedge B \rrbracket = \llbracket A \rrbracket \wedge \llbracket B \rrbracket \qquad \llbracket A \vee B \rrbracket = \neg\neg(\llbracket A \rrbracket \vee \llbracket B \rrbracket)$$

$$\llbracket \forall x.\, A \rrbracket = \forall x.\, \llbracket A \rrbracket \qquad \llbracket \exists x.\, A \rrbracket = \neg\neg\exists x.\, \llbracket A \rrbracket$$

Defining $p \Vdash_W A$ as $p \Vdash_K \llbracket A \rrbracket$, we have, as expected,

$$p \Vdash_W A \wedge B \;\text{ iff }\; p \Vdash_W A \text{ and } p \Vdash_W B$$

$$p \Vdash_W A \vee B \;\text{ iff }\; \forall q \preccurlyeq p, \exists r \preccurlyeq q,\; r \Vdash_W A \text{ or } p \Vdash_W B$$

Moreover, $\llbracket A \rrbracket \Leftrightarrow \neg\neg A$, and therefore

$$\emptyset \Vdash_K \neg\neg A \text{ if and only if there exists } p \text{ such that } p \Vdash_W A$$

III

Internalizing forcing in a type theory

Forcing and type theory

What forcing / Kripke models / the pre-sheave construction bring to type theory:

- Independence results.
  (E.g. of Voevodsky's univalence axiom.)
- Tools for categorical logic.
  (E.g the "cubical" model for univalence by Coquand et al.)
- Tools for programming and semantics.
  (E.g. general recursive types or *step-indexing*.)

Forcing and type theory

What type theory and similar Curry-Howard approaches bring to forcing:

- A presentation based on transformations (encodings) of an extended type theory $TT[G]$ to the initial type theory TT.
  (Like the negative translations to encode classical logic in intuitionistic logic, lecture of Dec 5th 2018.)
- The transformation also applies to proof terms, thus guaranteeing the logical consistency of the approach.
  (Like Bernardy et al's encoding of parametricity, lecture of Dec 19th 2018.)

Forcing and type theory

Recent work: (references at end of lecture)

- A. Miquel (2011) and L. Rieg (2014), inspired by J.-L. Krivine: classical forcing for the logic PAω ($\approx$ Fω + `call/cc`).

  $\Rightarrow$ seminar of Jan 16th 2019
- G. Jaber, N. Tabareau and M. Sozeau (2012): intuitionistic forcing for CC + universes + Σ, internalization of the presheave construction.
- G. Jaber, G. Lewertowski, P.-M. Pédrot, N. Tabareau, M. Sozeau (2016): intuitionistic forcing for Coq, quasi-monadic transformation, in call by name.

Outline of the transformation

(Following Jaber, Tabareau, Sozeau, LICS 2012)

Assume given a type $\mathbb{P}$ of worlds (a.k.a. forcing conditions) and a preorder $\preccurlyeq$.

- To each proposition A we associate a proposition $[\![A]\!]_p$ indexed by a world p, similar to $p \Vdash_K A$ ("A holds in world p").
- To each proof $\vdash a : A$ we associate a proof $p : \mathbb{P} \vdash [a]_p : [\![A]\!]_p$.

The translation is directed by the usual property of implication:

$$p \Vdash_K A \Rightarrow B \text{ iff } \forall q \preccurlyeq p, q \Vdash_K A \Rightarrow q \Vdash_K B$$

Expressed with dependent products: (with $P_p \stackrel{def}{=} \{q : \mathbb{P} \mid q \preccurlyeq p\}$)

$$[\![\Pi x : A.\, B]\!]_p \;=\; \Pi(q : P_p).\, \Pi(x : [\![A]\!]_q).\, [\![B]\!]_q$$

The forcing monad

Let's try to express this as a monadic transformation in a higher-order monad T. We can write

$$[\![A \to B]\!]_p = T\ (\lambda q.\ [\![A]\!]_q \to [\![B]\!]_q)\ p$$

where

$$T\ A = \lambda(p : \mathbb{P}).\ \Pi(q : P_p).\ A\ q$$

We can view this "forcing monad" as an asynchronous I/O monad:

- p is the log of inputs already received;
- $q \preccurlyeq p$ means that we can have received 0, 1 or several new inputs;
- every computation in this monad must be ready to receive new inputs, hence $\Pi(q : P_p) \ldots$

The forcing monad

$$T\,A = \lambda(p : \mathbb{P}).\ \Pi(q : P_p).\ A\,q$$

This is not the environment monad

$$T\,A = \mathbb{P} \to A$$

because the environment p changes during computation, non-deterministically but monotonically.

This is not the monotonic state monad

$$T\,A = \Pi(p : \mathbb{P}).\ \{(a, q) : A \times \mathbb{P} \mid q \preccurlyeq p\}$$

because, in the state monad, the state change $p \to q$ is initiated by the computation, while in the forcing monad the state change is imposed by the outside world.

Towards a translation

$$[\lambda(x : A).\ B]_p = \lambda(q : P_p).\ \lambda(x : [\![A]\!]_q).\ [B]_q$$
$$[A\ B]_p = [A]_p\ p\ [B]_p$$

Since types are terms, we must define $[\![\cdot]\!]$ as a function of $[\cdot]$:

$$[\![A]\!]_p = [A]_p\ p$$
$$[\Pi(x : A).\ B]_p = \lambda(q : P_p).\ \Pi(r : P_q).\ \Pi(x : [\![A]\!]_r).\ [\![B]\!]_r$$
$$[U]_p = \lambda(q : P_p).\ U$$

What about variables $[x]_q$?

A variable can be used in a different world q than the world p where it was bound!

Morphisms

The interpretation $[A]_p$ of a type is not just a function $f : P_p \to \square$ but also a morphism $\theta\ q\ r : f\ q \to f\ r$ that maps the interpretation at world q to the interpretation at world $r \preccurlyeq q$.

$$\begin{tikzcd}
q \arrow[r, "f"] \arrow[d, "\preccurlyeq"'] & f\ q \arrow[d, "\theta\ q\ r"] \\
r \arrow[r, "f"] & f\ r
\end{tikzcd}$$

In the case where A is a proposition, θ is the proof of monotonicity of forcing: $p \Vdash_K A \wedge q \preccurlyeq p \Rightarrow q \Vdash_K A$.

In the case where A is a "type that computes", we additionally want functoriality properties for θ, namely: $\theta\ q\ q = id$ and $\theta\ q\ s = \theta\ r\ s \circ \theta\ q\ r$.

Morphisms

We simultaneously define the translation of types $[\![A]\!]_p$ and the morphisms $\theta(A)_{p\to q}$ from $[\![A]\!]_p$ to $[\![A]\!]_q$.

$$\begin{aligned}[A]_p : \Sigma f : P_p \to \Box.\\ &\{\theta : \Pi(q : P_p).\, \Pi(r : P_q).\, f\; q \to f\; r \mid \texttt{functorial}_p(\theta)\}\end{aligned}$$

$$\begin{aligned}[\![A]\!]_p &= \pi_1([A]_p)\\ \theta(A)_{p\to q} &= \pi_2([A]_p)\; p\; q\end{aligned}$$

Finally, the translation of a variable is

$$[x]_p^\sigma = \theta(\texttt{type}(\sigma, x))_{\texttt{world}(\sigma,x)\to p}(x)$$

in an environment σ : variable $\to$ type $\times$ world.

Technical issues

These morphisms are obvious in category theory but raise equality-related issues in type theory.

In particular: if two types are convertible $A =_{\beta\eta} B$,
their translations are generally not convertible.

$$\frac{\Gamma \vdash M : A \qquad A =_{\beta\eta} B}{\Gamma \vdash M : B}$$

Translation, version 2

(Jaber, Lewertowski, Pédrot, Tabareau, Sozeau, LICS 2016)

We can get rid of these morphisms by translating Π function types in "call by name", that is, by leaving flexible the world of the argument.

$$\begin{array}{rl} \text{by value} & [\![\Pi x : A.\, B]\!]_p = \Pi q : P_p.\, \Pi x : [\![A]\!]_q.\, [\![B]\!]_q \\ \text{by name} & [\![\Pi x : A.\, B]\!]_p = \Pi x : (\Pi q : P_p.\, [\![A]\!]_q).\, [\![B]\!]_p \end{array}$$

Translating variables:

$$\begin{array}{rl} \text{by value} & [x]_p^\sigma = \theta(\mathtt{type}(\sigma, x))_{\mathtt{world}(\sigma,x)\to p}(x) \\ \text{by name} & [x]_p^\sigma = x\, p \end{array}$$

No need for morphisms θ; it suffices that the σ environment proves that $p \preccurlyeq \mathtt{world}(\sigma, x)$.

Additional benefit: if $A =_{\beta\eta} B$ then $[\![A]\!]_p =_{\beta\eta} [\![B]\!]_p$.

Using the translation for forcing

The translations $[\cdot]$ make it possible to mechanically transport the definitions and theorems of *TT* (the initial type theory, e.g. Coq) to $\mathrm{TT}[G]$ (its extension).

(Coq plug-ins have been developed to automate this process.)

To declare a generic element G of type A in the extension, it suffices to manually define (in TT) a term G^f of type $\forall p,\ [\![A]\!]_p$

Example

To get a generic set of integers $G : \mathtt{nat} \to \mathtt{Prop}$, we take $\mathbb{P} = \mathtt{Finfun.t\ nat\ bool}$ and we define

$$G^f = \lambda(p : \mathbb{P}).\ \lambda(q : P_p).\ \lambda(n : \mathtt{nat}).\ \mathtt{Finfun.app}\ q\ n = \mathtt{Some\ true}$$

IV

Forcing on natural numbers

Forcing on natural numbers

(Also called "internal logic of the topos of trees" by Birkedal et al)

A simple example of forcing conditions / Kripke worlds is

$$\mathbb{P} \stackrel{\textit{def}}{=} \mathbb{N} \quad \text{naturally ordered by } q \preccurlyeq p \stackrel{\textit{def}}{=} q \leq p$$

An intuitive interpretation in terms of time:
$p \Vdash_K A$ reads "A is true now and during p days".

The $\triangleright$ modality and Löb's rule

The $\triangleright A$ modality reads "later A" and is defined by

$$0 \Vdash_K \triangleright A \qquad p+1 \Vdash_K \triangleright A \text{ if } p \Vdash_K A$$

In other words: $\triangleright A$ is true today for p days if A is true tomorrow for $p-1$ days.

In this modal logic, Löb's rule is valid:

$$\frac{\triangleright A \Rightarrow A}{A}$$

Proof.

Assume $p \Vdash_K \triangleright A \Rightarrow A$. We have $(q \Vdash_K \triangleright A) \Rightarrow (q \Vdash_K A)$ for all $q \leq p$.
We show $q \Vdash_K A$ for all $q \leq p$ by induction over q:
$0 \Vdash_K A$ since $0 \Vdash_K \triangleright A$.
If $q < p$ and $(q \Vdash_K A)$, then $(q+1 \Vdash_K \triangleright A)$, therefore $(q+1 \Vdash_K A)$. $\square$

Generalization: a fixed-point operator

We can declare the following terms in the forcing extension, just by giving terms that inhabit the translations of their types:

$$\begin{aligned}
\triangleright &: \mathtt{Type} \to \mathtt{Type} \\
\mathtt{fix} &: \forall (A : \mathtt{Type}),\ (\triangleright A \to A) \to A \\
\mathtt{next} &: \forall (A : \mathtt{Type}),\ A \to \triangleright A \\
\mathtt{fix_eq} &: \forall (A : \mathtt{Type}). \forall (f : \triangleright A \to A).\ \mathtt{fix}\, A\, f = f\, (\mathtt{next}\, A\, (\mathtt{fix}\, A\, f))
\end{aligned}$$

(Constructions: by induction over p.)

`fix` is therefore the proof term for Löb's rule, but it also gives an interesting fixed-point operator.

General recursive types

By specializing `fix` on a universe, say $A = \mathtt{Set}$, we can construct

$$\begin{aligned} \mu &: (\mathtt{Set} \to \mathtt{Set}) \to \mathtt{Set} \\ \mathtt{unfold} &: \forall (F : \mathtt{Set} \to \mathtt{Set}),\ \mu\, F \to F\,(\triangleright\, \mu\, F) \\ \mathtt{fold} &: \forall (F : \mathtt{Set} \to \mathtt{Set}), F\,(\triangleright\, \mu\, F) \to \mu\, F \end{aligned}$$

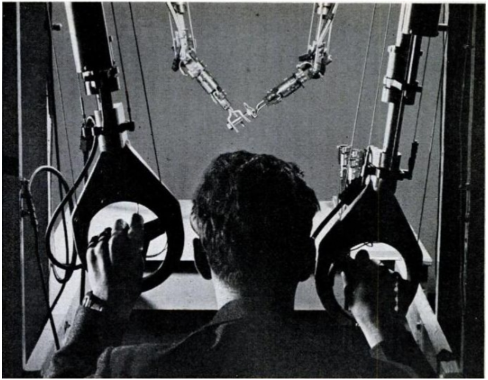
as well as proofs of $\mu\, F = F(\triangleright\, \mu\, F)$ and $\mathtt{fold}\, F \circ \mathtt{unfold}\, F = \mathtt{id}$ and $\mathtt{unfold}\, F \circ \mathtt{fold}\, F = \mathtt{id}$.

The type $\mu\, F$ is therefore equivalent to the recursive Caml type `t`

```
type t = C of t F       unfold (C x) = x      fold x = C x
```

No hypotheses are made on the F type constructor: it is not necessarily increasing, nor contractive.

General recursive types



Dangerous object being handled: a recursive type such as $T = T \to T$, which endangers termination.

Handles of the remote manipulator: the terms produced by translation $[\cdot]$.

Going further

The general recursive types obtained by forcing make it possible to give simple denotational semantics to Turing-complete languages (no strong normalization). For instance:

- $D = D \to D$ for pure λ-calculus;
- $D = (Loc \to D) \to \mathcal{P}(Val)$ for mutable references.

More generally: the naive idea of "counting days" and the less naive idea of the "later" modality ($\triangleright$) resonate with a powerful semantic technique: *step-indexing*, described in the next lecture.

V

Further reading

Further reading

Introductions to forcing in set theory:

- Timothy Y. Chow, *A beginner's guide to forcing*, Contemporary Mathematics (479), 2008. `https://arxiv.org/abs/0712.1320`
- Robert S. Wolf, *A tour through mathematical logic*, chapter 6. Carus Mathematical Monographs, 2005.

Forcing as a translation for propositions and proofs:

- A. Miquel, *Forcing as a Program Transformation*, LICS 2011. `https://www.fing.edu.uy/~amiquel/publis/lics11.pdf`
- G. Jaber, N. Tabareau, M. Sozeau, *Extending Type Theory with Forcing*, LICS 2012. `https://hal.inria.fr/hal-00685150/`
- G. Jaber, G. Lewertowski, P.-M. Pédrot, N. Tabareau, M. Sozeau, *The Definitional Side of the Forcing*, LICS 2016. `https://hal.inria.fr/hal-01319066`