Hypothèse du continu en théorie des types

Kevin Quirin,
sous la direction de Nicolas Tabareau

Mai – Août 2013

Table des matières

1 Introduction **2**

1.1 Plan 2

1.2 La preuve de Cohen 3

2 Faisceaux de Grothendieck **4**

2.1 Rappels catégoriques 4

2.2 Topologie de Grothendieck 6

2.3 Topos 8

3 L'hypothèse du continu, version catégorique **9**

4 Formalisation en Coq, les prérequis **12**

4.1 Russell 12

4.2 Préfaisceaux 12

5 Formalisation en Coq **14**

5.1 Faisceaux 14

5.2 Preuve d'indépendance de l'hypothèse du continu 16

5.3 Travaux futurs 18

Références **20**

1 Introduction

Une des avancée les plus marquantes du vingtième siècle en mathématiques est celle de l'indépendance de l'hypothèse du continu CH du reste des mathématiques. CH exprime le fait qu'il n'existe pas d'ensemble compris strictement entre $\mathbb{N}$ et $\mathbb{R}$, ou encore que toute partie infinie de $\mathbb{R}$ est soit en bijection avec $\mathbb{N}$, soit en bijection avec $\mathbb{R}$. La preuve d'indépendance a commencé par Kurt Gödel, qui a montré en utilisant *l'univers constructible* $\mathbb{L}$ qu'on pouvait ajouter l'hypothèse du continu CH aux autres axiomes de ZFC sans apporter de contradiction. Il faut ensuite attendre Paul Cohen pour obtenir l'autre moitié de l'indépendance (qu'on peut rajouter $\neg$CH à ZFC sans contradiction), avec introduction de la méthode de *forcing*, qui a en même temps ouvert aux théoriciens des ensembles un nouveau secteur de recherche.

Un autre enjeu des mathématiques modernes est la recherche d'automatisation des preuves, ou au moins leurs vérifications systématiques par un logiciel, pour éviter toute erreur ou imprécision. C'est le but de *Coq*, basé sur le *calcul des constructions* (CoC) qui est une forme de théorie des types. Ainsi, un théorème prouvé dans Coq nécessite seulement de "croire" au noyau de Coq, et non à des étapes de raisonnement plus ou moins convaincantes fréquemment utilisées en mathématiques (*Quitte à ...on peut ...*, *Il est clair que ...* ou encore *On pourrait définir ...*). On a aujourd'hui des preuves formalisées de quelques théorèmes difficiles de mathématiques :

— le théorème de Jordan (Thomas C. Hales, 2005)
— le théorème des quatre couleurs (Georges Gonthier, 2005)
— le théorème de Feit-Thompson (Georges Gonthier, 2012)
— ...

On se propose ici de tenter de donner une formalisation de l'indépendance de l'hypothèse du continu en Coq (dans la suite de ce mémoire, on appellera "indépendance de l'hypothèse du continu" la partie montrée par Cohen), ce qui nécessite l'étude du forcing en théorie des types. On verra qu'on sera confronté à un problème pour montrer certaines égalités : Coq utilise une égalité *syntaxique*, alors qu'usuellement, en mathématiques, ce qu'on appelle *égalité* est souvent un *isomorphisme* ou une *équivalence*.

On suppose le lecteur familier avec les bases de théorie des catégories.

1.1 Plan

La preuve de l'indépendance de l'hypothèse du continu en théorie des types – et *a fortiori* sa formalisation en Coq – nécessite des prérequis catégoriques. On essayera de limiter ces connaissances au strict minimum, n'utilisant parfois qu'une approche intuitive des notions ; dans tous les cas,

ce travail sur les catégories n'est pas l'objet de ce rapport, et on se contentera d'admettre les résultats. Le lecteur intéressé peut se référer à [MM92] pour des définitions précises et des preuves complètes.

La partie 1.2 résume l'idée de la preuve d'indépendance de l'hypothèse du continu de ZFC en théorie des ensembles, par la méthode de forcing de Cohen.

La partie 2 posera le cadre catégorique de l'étude, à travers des rappels sur les catégories en général (partie 2.1) et les définitions de faisceaux et topoi de Grothendieck.

La partie 3 suivante expose plus en détail la partie de la preuve étudiée et formalisée en Coq

La partie 4 place le cadre de la théorie des types utilisée (partie 4.1) et expose les travaux d'internalisation de la construction de préfaisceaux en Coq sur laquelle la suite repose (partie 4.2).

Enfin, la dernière partie 5 présente le travail effectué pendant le stage (parties 5.1 et 5.2). On conclut en proposant une suite à ce travail (partie 5.3).

1.2 La preuve de Cohen

Rappelons l'idée de la preuve de Cohen en théorie des ensembles. Pour une preuve complète, on peut se référer à [Kun]. Commençons par se donner un modèle transitif dénombrable de ZFC M.

On part d'une *notion de forcing* $\mathbb{P}$, *i.e* un ensemble partiellement ordonné, dont les éléments sont appelés *conditions*.

Définition 1.1
Une partie F de $\mathbb{P}$ est dite générique *si*

(i) G est un filtre, i.e
- G *non vide*
- $p \in F \wedge p \leqslant q \Rightarrow q \in F$
- $p, q \in F \Rightarrow \exists r \in F,\ r \leqslant p \wedge r \leqslant q$

(ii) G rencontre toutes les parties denses de $\mathbb{P}$.

On peut alors constuire un modèle $M[G]$ de ZFC, contenant M et G, et ayant les mêmes ordinaux que M ; $M[G]$ est appelée *extension générique de* M.

On donne à chaque élément de $M[G]$ un *nom* dans M : le modèle M

"voit" les objets de $M[G]$, mais sans les "comprendre", puisqu'il faudrait pour cela que M connaisse G. En revanche, M peut voir certaines propriétés des éléments de $M[G]$; par exemple, prenons pour $\mathbb{P}$ l'ensemble des fonctions finies de ω dans 2. $f_G = \bigcup G$ est alors une fonction de ω dans 2. Le modèle M le sait, mais ne peut pas déterminer la valeur de $f_G(0)$, qui dépend du générique G choisi. M sait juste que $f_G(0) = 0$ si $\{\langle 0, 0\rangle\} \in G$ et $f_G(0) = 1$ si $\{\langle 0, 1\rangle\} \in G$.

On définit alors la relation de *forcing* de la façon suivante : si ψ est une formule sur l'ensemble des noms des éléments de $M[G]$, et $p \in \mathbb{P}$, on écrit $p \Vdash \psi$ ("p force ψ") si pour tout G générique, si $p \in G$ alors la formule ψ est vraie dans $M[G]$.

Dans le cas de l'hypothèse du continu, on prend pour $\mathbb{P}$ l'ensemble des fonctions partielles finies de $\kappa \times \omega$ dans 2, où κ est un cardinal non dénombrable de M. Si G est un générique, alors $\bigcup G \ : \ \kappa \times \omega \to 2$, qu'on peut voir comme une κ-suite de fonctions $\omega \to 2$. On a alors

Lemme 1.2
Si G est générique pour $\mathbb{P}$, alors dans $M[G]$, $2^\omega \geqslant \kappa$.

Il suffit alors de prendre, par exemple, $\kappa = \omega_2$, et vérifier que κ n'est pas *collapsé* par le forcing : on veut que κ soit bien le ω_2 de $M[G]$. Cette vérification passe par la *condition de chaîne dénombrable*, que l'on n'explicitera pas ici.

2 Faisceaux de Grothendieck

Passons maintenant à l'adaptation de la preuve aux catégories, à travers les faisceaux de Grothendieck.

2.1 Rappels catégoriques

La méthode du forcing se base sur la notion d'ensemble partiellement ordonné. On en définit donc la version catégorique :

Définition 2.1 : Catégorie d'un ensemble partiellement ordonnée
Soit $(\mathbb{P}, \leqslant)$ un ensemble partiellement ordonné. La catégorie $\mathbb{C}$ de $\mathbb{P}$ est définie par

- *les objets de $\mathbb{C}$ sont les éléments de $\mathbb{P}$;*
- *si $p, q \in \mathbb{C}$, on a une unique flèche $q \to p$ si et seulement si $q \leqslant p$.*

On identifiera alors les flèches $q \to p$ avec les éléments q tels que $q \leqslant p$.

On notera dans la suite, pour un objet p de $\mathbb{P}$

$$\mathbb{P}_p = \{q \in \mathbb{P} \mid q \leqslant p\}.$$

On a aussi besoin, pour exprimer l'hypothèse du continu, d'utiliser l'ensemble des parties d'un ensemble. On utilise pour cela la notion de *sous-objet classifiant*, qu'on peut voir comme un ensemble de *valeurs de vérité*.

Définition 2.2 : Sous-objet classifiant
Appelons 1 *l'objet terminal de* $\mathbb{C}$. *Alors* Ω *est un* sous-objet classifiant *s'il existe un monomorphisme* true $: 1 \to \Omega$ *tel que pour tout monomorphisme* $\varphi : S \to X$ *il existe un unique* χ_φ *formant un pullback :*

$$\begin{array}{ccc} S & \xrightarrow{f_S} & 1 \\ {\scriptstyle\varphi}\downarrow & & \downarrow{\scriptstyle\text{true}} \\ X & \xrightarrow[\chi_\varphi]{} & \Omega \end{array}$$

c'est-à-dire que ce diagramme commute, et qu'il est universel dans le sens où, si T, ψ, f_T *fait commuter le diagramme, alors il existe une unique flèche* $u : T \to S$ *telle que* $f_S \circ u = f_T$ *et* $\varphi \circ u = \psi$.

La notion suivante qu'on utilise dans la suite est la notion centrale de préfaisceau.

Définition 2.3 : Préfaisceau
Soit $\mathbb{C}$ *une petite catégorie. On appelle* préfaisceau *sur* $\mathbb{C}$ *un foncteur contravariant de* $\mathbb{C}$ *dans* Set *(de façon équivalente, un foncteur* $F : \mathbb{C}^{\text{op}} \to \text{Set}$*).*

On appelle ainsi, pour S un ensemble quelconque, *préfaisceau constant égal à* S le foncteur $\Delta S : C \in \mathbb{C} \mapsto S$.

Si P est un préfaisceau sur $\mathbb{C}$, $f : D \to C$ une flèche de $\mathbb{C}$ et $x \in P(C)$, alors on appelle *restriction de* x *suivant* f l'ensemble $P(f)(x)$, noté $x \cdot f$.

Pour chaque objet C de $\mathbb{C}$, on peut définir un préfaisceau $\mathbf{y}\,C$, appelé *plongement de Yoneda*, défini sur les objets par

$$\mathbf{y}(C)(D) = \text{Hom}_{\mathbb{C}}(D, C),$$

et sur les flèches $\alpha \ : \ D' \to D$ par

$$\mathbf{y}(C)(\alpha) : \begin{array}{ccc} \mathrm{Hom}_{\mathbb{C}}(D,C) & \longrightarrow & \mathrm{Hom}_{\mathbb{C}}(D',C) \\ u & \longmapsto & u \circ \alpha \end{array} .$$

On note $\mathrm{PSh}(\mathbb{C})$ la catégorie des préfaisceaux sur $\mathbb{C}$, dont les morphismes sont les transformations naturelles entre les préfaisceaux, *i.e* des fonctions $\theta \ : \ P \to Q$ qui à un objet C de $\mathbb{C}$ associent une fonction $\theta_C \ : \ P(C) \to Q(C)$ de façon naturelle ($\forall f \ : \ D \to C, \ Qf \circ \theta_C = \theta_D \circ Pf$). Un monomorphisme de préfaisceaux est un morphisme de préfaisceaux θ tel que pour tout objet C de $\mathbb{C}$, θ_C est un monomorphisme.

On définit enfin l'exponentiation entre préfaisceaux par

$$\forall C \in \mathbb{C}, \ Q^P(C) = \mathrm{Hom}_{\mathrm{PSh}(\mathbb{C})}(\mathbf{y}(C) \times P, Q).$$

2.2 Topologie de Grothendieck

Commençons cette partie par regarder les bien connues notions d'*ouverts* et de *fonctions continues*. Prenons donc un espace topologique X, et notons $\mathscr{O}(X)$ la catégorie dont les objets sont les ouverts de X, et les flèches les inclusions entre ces ouverts. On considère maintenant une application $C \ : \ \mathscr{O}(X) \to \mathrm{Set}$ qui à un ouvert U de X associe l'ensemble des fonctions continues de U dans $\mathbb{R}$. C définit donc un préfaisceau sur $\mathscr{O}(X)$, puisqu'on peut bien définir des restrictions compatibles avec la continuité : si f est continue sur U et $V \subseteq U$, alors $f\,|_V$ est continue sur V. Mais C a une autre propriété que n'aurait pas, par exemple, le foncteur qui à U associe l'ensemble des fonctions bornées sur U. Si on recouvre un ouvert U par des ouverts U_i inclus dans U, et si on définit des fonctions continues f_i sur U_i telles que pour tous i et j, $f_i|_{U_i \cap U_j} = f_j\,|_{U_i \cap U_j}$, alors on peut définir une unique fonction continue f sur U dont la restriction à tout U_i est f_i. Cette propriété, la *propriété de recollement*, fait de C un *faisceau*. On veut maintenant étendre cette notion de faisceau dans le cas d'une catégorie quelconque (disons une petite catégorie $\mathbb{C}$) ; on doit donc donner aux objets de $\mathbb{C}$ le même comportement que les ouverts d'un espace topologique. On remplace ainsi les ouverts stables par intersection par des *cribles* :

Définition 2.4 : Crible
Soit C un objet de $\mathbb{C}$. On appelle crible sur C *toute famille de flèches de $\mathbb{C}$ ayant toutes pour codomaine C, et telle que*

$$f \in S \Rightarrow f \circ g \in S$$

pour toute flèche g autorisant la composition.

On note alors que si S est un crible sur un objet C, et $h : D \to C$ une flèche de $\mathbb{C}$, alors

$$h^*(S) = \{g \mid \mathrm{cod}(g) = D \text{ et } hg \in S\}$$

est un crible sur l'objet D.

On peut alors définir les topologies de Grothendieck :

Définition 2.5 : Topologie de Grothendieck
Une topologie de Grothendieck *sur $\mathbb{C}$ est un foncteur J qui à tout objet C de $\mathbb{C}$ associe une collection $J(C)$ de cribles sur C vérifiant, pour C et D objets quelconques de $\mathbb{C}$:*

(i) le crible $\{f \mid \mathrm{cod}(f) = C\}$ est dans $J(C)$;

(ii) J est stable : si $S \in J(C)$ alors pour tout $h : D \to C$, $h^*(S) \in J(D)$;

(iii) J est transitif : si $S \in J(C)$ et R est un crible sur C tel que $h^*(R) \in J(D)$ pour tout $h : D \to C$, alors $R \in J(C)$.

Si $S \in J(C)$, on dit que S couvre C ; si $f^*(S)$ couvre D pour une flèche $f : D \to C$, on dit que S couvre f.

Exemple – Soit $\mathbb{P}$ un ensemble partiellement ordonné, et soit $p \in Pp$. Une partie D de $\{q \in P \mid q \leqslant p\}$ est dite *dense sous p* si pour tout $r \leqslant p$, on peut trouver $q \in D$ tel que $q \leqslant r$.

On définit alors la topologie dense par

$$J(p) = \{D \mid \forall q \in D,\ q \leqslant p \text{ et } D \text{ est un crible dense sous } p\}.$$

Alors J définit bien une topologie de Grothendieck sur $\mathbb{P}$.

On va maintenant définir la notion de faisceau de Grothendieck de la même façon que sur un espace topologique. Soit P un préfaisceau sur $\mathbb{C}$.

Définition 2.6
Soit C un objet de $\mathbb{C}$ et S un crible couvrant C. On appelle famille coïncidante *sur S d'éléments de P toute fonction qui à $f : D \to C$ de S associe $x_f \in P(D)$ telle que*

$$\forall g : E \to D \in \mathbb{C},\ x_f \cdot g = x_{fg}.$$

On appelle amalgamation *d'une telle* famille coïncidante *un élément $x \in P(C)$ tel que*

$$\forall f \in S,\ x \cdot f = x_f.$$

Définition 2.7 : Faisceau de Grothendieck
On appelle alors (J-)faisceau (de Grothendieck) *un préfaisceau P tel que toute* famille coïncidante *de tout crible couvrant de tout élément de $\mathbb{C}$ ait une unique amalgamation. On note* $\mathrm{Sh}(\mathbb{C}, J)$ *la famille des faisceaux pour la topologie J.*

On a alors un théorème fort, qui permet de transformer tout préfaisceau sur $\mathbb{C}$ en un J-faisceau, plus précisemment :

Théorème 2.8
Le foncteur d'inclusion $i : \mathrm{Sh}(\mathbb{C}, J) \to \mathrm{Set}^{\mathbb{C}^{\mathrm{op}}}$ *a un adjoint à gauche*

$$\mathbf{a} : \mathrm{Set}^{\mathbb{C}^{\mathrm{op}}} \to \mathrm{Sh}(\mathbb{C}, J),$$

qu'on appelle foncteur de sheafification.

Ce foncteur permet notamment de définir les *faisceaux constants* (qui, contrairement à leur nom, ne sont pas constants) comme les images par $\mathbf{a}$ des préfaisceaux constants (qui eux sont vraiment constants). Si $\mathbf{a}(P) = F$, on dit aussi que F est le *sheafifié* de P.

2.3 Topos

On appellera *topos* toute catégorie isomorphe à $\mathrm{Sh}(\mathbb{C}, J)$ pour une certaine petite catégorie $\mathbb{C}$ et une certaine topologie J sur $\mathbb{C}$.

Dans le cas de la catégorie des ouverts d'un espace topologique, on peut noter qu'on a un sous-objet classifiant, qui est le faisceau Ω défini par

$$\Omega(U) = \{V \mid V \text{ ouvert inclu dans } U\}.$$

On peut modifier cette définition en remplaçant V par son *crible principal* $\{W \mid W \subseteq V\}$, et on obtient alors

$$\Omega(U) = \{S \mid S \text{ crible principal sur } U\}.$$

On peut à nouveau généraliser cette construction pour obtenir un sous-objet classifiant pour une topologie de Grothendieck $\mathrm{Sh}(\mathbb{C}, J)$ quelconque. On remplace les cribles principaux par des cribles clos ; un crible S sur C est dit *clos* si pour toute flèche $f : D \to C$ de $\mathbb{C}$, on a

$$M \text{ couvre } f \Longrightarrow f \in M.$$

On peut alors montrer que l'application Ω qui à un objet C de $\mathbb{C}$ associe l'ensemble des cribles clos sur C est un faisceau, et même un sous-objet classifiant pour la catégorie $\mathrm{Sh}(\mathbb{C}, J)$.

Définition 2.9
On appelle topos booléen *un topos pour lequel le sous-objet classifiant* Ω *est isomorphe à* $2 \simeq 1 + 1$.

On peut alors montrer que le topos pour la topologie dense est un topos booléen ; on peut voir que la notion de topos booléen revient à dire que la logique dans le topos en question est classique : on peut montrer le tiers-exclu. On peut aussi montrer que le topos de la topologie dense satisfait l'axiome du choix.

Pour terminer cette section, et ainsi montrer qu'un topos de Grothendieck vérifie bien la définition usuelle de topos élémentaire, on va définir des exponentiations dans le topos.

On pose, pour F, G des préfaisceaux sur $\mathbb{C}$

$$F^G = \mathrm{Hom}_{\mathrm{PSh}\mathbb{C}}(\mathbf{y}(\cdot) \times P, Q).$$

On vérifie alors que F^G est bien un préfaisceau, et on transporte cette définition de l'exponentielle aux faisceaux, *via* le lemme

Lemme 2.10
Soit F *un faisceau pour une topologie* J*. Alors pour tout préfaisceau* G *sur* $\mathbb{C}$*, le préfaisceau* F^G *est un faisceau.*

3 L'hypothèse du continu, version catégorique

On va maintenant expliciter la preuve d'indépendance de l'hypothèse du continu version catégorique (qu'on trouve dans [MM92]), qui suit en fait largement la preuve de Cohen.

Commençons par fixer un ensemble B de cardinal plus grand que $2^{\mathbb{N}}$; on prendra dans la formalisation de la partie suivante $B = \mathfrak{pp}(\mathbb{N})$, mais pourvu qu'il soit assez grand, l'ensemble B importe peu. On considère ensuite l'ensemble partiellement ordonné $(\mathbb{P}, \leqslant)$, où

— $\mathbb{P}$ est l'ensemble des *conditions*, qui sont des fonctions finies $B \times \mathbb{N} \to 2$, *i.e* des couples (F_p, p) où F_p est une partie finie de $B \times \mathbb{N}$ et p une fonction totale de $\mathbb{F}_p$ dans 2 ;
— $q \leqslant p$ (on dit "q est plus fort que p") si F_q est inclu dans F_p et la restriction de q à F_p coïncide avec p.

On muni cet ensemble partiellement ordonné de la topologie dense J_d, pour construire le *topos de Cohen* $\mathrm{Sh}(\mathbb{P}, J_d)$. C'est alors un topos booléen, satisfaisant l'axiome du choix ; notons Ω son sous-objet classifiant. On

cherche donc à montrer qu'il existe un objet K et des monomorphismes $\mathbb{N} \hookrightarrow K \hookrightarrow \Omega^{\mathbb{N}}$, et qu'il n'y a pas d'épimorphismes $\mathbb{N} \twoheadrightarrow K$ ni $K \twoheadrightarrow \Omega^{\mathbb{N}}$, c'est-à-dire :

Théorème 3.1
Il existe un topos booléen, satisfaisant l'axiome du choix et la négation de l'hypothèse du continu.

Ici, nous ne nous intéresserons en fait qu'à la construction des monomorphismes, la partie concernant les épimorphismes faisant partie des travaux à venir.

On veut donc construire dans le topos $\mathrm{Sh}(\mathbb{P}, J_d)$ un monomorphisme $B \times \mathbb{N} \to 2$, c'est-à-dire un sous-objet de $B \times \mathbb{N}$. On construira tout d'abord ces objets dans la catégorie $\mathrm{Set}^{\mathbb{P}^{\mathrm{op}}}$ des préfaisceaux, et on utilisera ensuite le foncteur de sheafification. On veut donc un sous objet du préfaisceau constant $\Delta(B \times \mathbb{N})$, qu'on définit pour $p \in \mathbb{P}$ par

$$A(p) = \{(b, n) \mid p(b, n) = 0\}. \tag{1}$$

Lemme 3.2
Le foncteur A est un sous-objet clos de $\Delta(B \times \mathbb{N})$ pour la topologie dense ; en d'autres termes, pour tout $p \in \mathbb{P}$

$$(b, n) \in A(p) \iff \forall q \leqslant p,\ \exists r \leqslant q,\ (b, n) \in A(r).$$

Démonstration. Prenons donc $p \in \mathbb{P}$, $b \in B$ et $n \in \mathbb{N}$. Supposons que $(b, n) \notin A(p)$. Deux cas se présentent alors :

- $p(b, n) = 1$: dans ce cas, quelque soit $r \leqslant p$, $r(b, n) = 1$, et donc en prenant $q = p$, on a bien le résultat ;
- $p(b, n)$ n'est pas défini : on peut alors étendre p en q qui vérifie $q(b, n) = 1$. Alors $r(b, n) = 1$ pour tout $r \leqslant q$.

Réciproquement, si $(b, n) \in A(p)$, il est clair que $\forall q \leqslant p,\ \exists r \leqslant q,\ (b, n) \in A(r)$: il suffit de prendre par exemple $r = q$. □

On note Ω (resp. Ω_d) le sous-objet classifiant de $\mathrm{Set}^{\mathbb{P}^{\mathrm{op}}}$ (resp. $\mathrm{Sh}(\mathbb{P}, J_d)$). On peut factoriser la fonction caractéristique de A χ_A *via* Ω_d en

$$f \ :\ \Delta B \times \Delta \mathbb{N} \to \Omega_d,$$

puis transposer f en morphisme de préfaisceaux

$$g \ :\ \Delta B \to \Omega_d^{\Delta \mathbb{N}}.$$

Lemme 3.3
g est un monomorphisme de préfaisceaux.

Démonstration. Par définition de monomorphisme de préfaisceaux, il suffit de vérifier que pour tout $p \in \mathbb{P}$, $g_p \ : \ \Delta(B)(p) \to \left(\Omega_d^{\Delta\mathbb{N}}\right)(p)$ est un monomorphisme d'ensembles. On a tout d'abord $\Delta(B)(p) = B$, et $\left(\Omega_d^{\Delta\mathbb{N}}\right)(p)$ est l'ensemble des transformations naturelles $\mathbf{y}(p) \times \Delta\mathbb{N} \to \Omega_d$. Pour $b \in B$, on a donc

$$g_p(b) \ : \ \mathbf{y}(p) \times \Delta\mathbb{N} \to \Omega_d,$$

défini, d'après la définition de A, par

$$g_p(b)(q,n) = \{r \in \mathbb{P} \mid r \leqslant q \text{ et } r(b,n) = 0\}. \tag{2}$$

Soit donc $b \neq c$ deux éléments de B. p est une fonction finie, et donc il existe un n_0 tel que ni $p(b,n_0)$, ni $p(c,n_0)$ ne soient définis ; on peut alors étendre p en r tel que $r(b,n_0) = 0$ et $r(c,n_0) = 1$, et alors $r \in g_p(b)(p,n_0)$ mais $r \notin g_p(c)(p,n_0)$.

Donc $g_p(b) \neq g_p(c)$. □

Le foncteur de sheafification $\mathbf{a}$ envoie donc g sur un monomorphisme $m \ : \ \widehat{B} \to \mathbf{a}\left(\Omega_d^{\Delta\mathbb{N}}\right)$. Les propriétés de $\mathbf{a}$ permettent de montrer qu'on a $\mathbf{a}\left(\Omega_d^{\Delta\mathbb{N}}\right) \simeq \Omega_d^{\widehat{\mathbb{N}}}$. On a donc un monomorphisme

$$m \ : \ \widehat{B} \to \Omega_d^{\widehat{\mathbb{N}}}.$$

Comme on avait au départ une injection $\mathbb{N} \to B$, qui est préservée par $\mathbf{a}$, on a donc finalement deux injections

$$\widehat{N} \to \widehat{B} \to \Omega_d^{\widehat{\mathbb{N}}}.$$

La fin de la démonstration passe par la preuve de $\widehat{\mathbb{N}} < \widehat{2^{\mathbb{N}}} < \widehat{B}$, avec des inégalités strictes, grâce à un équivalent de la condition de chaîne dénombrable. On aura alors en combinant les deux inégalités

$$\widehat{\mathbb{N}} < \widehat{\Omega^{\mathbb{N}}} < \widehat{B} \leqslant \Omega_d^{\widehat{\mathbb{N}}}.$$

En oubliant le $\widehat{B}$ qu'on vient de construire, on obtient bien notre résultat, en prenant $K = \widehat{\Omega^{\mathbb{N}}}$:

$$\mathbb{N} < \widehat{\Omega^{\mathbb{N}}} < \Omega_d^{\widehat{\mathbb{N}}}.$$

4 Formalisation en Coq, les prérequis

Coq étant basé sur le calcul des constructions, qui est une théorie des types dépendants, on espère pouvoir formaliser la preuve précédente.

4.1 Russell

On utilisera en fait une extension du calcul des constructions, Russel [Soz08], qui permet de manipuler des *subset types*, et utilise la *proof irrelevance*.

Les termes de Russell sont donnés par la grammaire

$$\begin{aligned} T, U, M, N :=& \text{Prop} \mid \text{Type}_i \mid \Pi x : T.U \mid \Sigma x : T.U \mid \\ & \{x : T | U\} \mid x \mid \lambda x : T.M \mid MN \mid \pi_1 M \mid \pi_2 M, \end{aligned}$$

les axiomes $\mathscr{A} = \{(\text{Prop}, \text{Type}_0), (\text{Type}_i, \text{Type}_{i+1})\}$ et les règles sont données dans la figure 4.1 où $\leqslant$ est définie comme la plus petite relation réflexive, transitive contenant les règles

$$\frac{\Gamma \vdash T \leqslant T'}{\Gamma \vdash T \leqslant \{x : T' | U\}} \qquad \frac{\Gamma \vdash T \leqslant T'}{\Gamma \vdash \{x : T | U\} \leqslant T'}$$

4.2 Préfaisceaux

On part de la formalisation de la construction des préfaisceaux données dans [JTS12], les fichiers de code pouvant être trouvés sur `https://github.com/mattam82/Forcing`. On construit donc, pour une notion de forcing $\mathbb{P}$, les préfaisceaux sur s (Type ou Prop), à un niveau $p \in \mathbb{P}$ comme un couple composé d'une fonction $f : \mathbb{P}_p \to s$ et d'un ensemble de fonctions de restrictions $\theta_{q \to r}$ vérifiant des propriétés de transitivité et reflexivité ; formellement

$$\Sigma f : \mathbb{P}_p \to s.\{\theta : \Pi q : \mathbb{P}_p.\Pi r : \mathbb{P}_q.fq \to fr \mid \mathbf{trans}(\theta, p) \wedge \mathbf{refl}(\theta, p)\},$$

avec

$$\begin{aligned} \mathbf{refl}(\theta, p) &= \Pi q : \mathbb{P}_p.\theta_{q \to q} = \text{id} \\ \mathbf{trans}(\theta, p) &= \Pi q : \mathbb{P}_p.\Pi r : \mathbb{P}_q.\Pi s : \mathbb{P}_r.(\theta_{r \to s})(\theta_{q \to r}) = \theta_{q \to s} \end{aligned}$$

On a donc, en Coq, le code suivant :

$$\frac{(x : T) \in \Gamma}{\Gamma \vdash x : T} \qquad \frac{(s_1, s_2) \in \mathscr{A}}{\Gamma \vdash s_1 : s_2}$$

$$\frac{\Gamma, x : T \vdash M : U}{\Gamma \vdash \lambda x : T.M : \Pi x : T.U} \qquad \frac{\Gamma \vdash M : \Pi x : T.U \quad \Gamma \vdash N : T}{\Gamma \vdash MN : U\{N/x\}}$$

$$\frac{\Gamma \vdash T : s_1 \quad \Gamma, x : T \vdash U : s_2}{\Gamma \vdash \Pi x : T.U : \max(s_1, s_2)} \qquad \frac{\Gamma \vdash T : \mathrm{Type}_i \quad \Gamma, x : T \vdash U : \mathrm{Type}_i}{\Gamma \vdash \Sigma x : T.U : \mathrm{Type}_i}$$

$$\frac{\Gamma \vdash M : T \quad \Gamma \vdash N : U\{M/x\}}{\Gamma \vdash (M, N) : \Sigma x : T.U} \qquad \frac{\Gamma \vdash M : \Sigma x : T.U}{\Gamma \vdash \pi_1 M : T}$$

$$\frac{\Gamma \vdash M : \Sigma x : T.U}{\Gamma \vdash \pi_2 M : U\{\pi_1 M/x\}} \qquad \frac{\Gamma \vdash T : s \quad \Gamma, x : T \vdash U : \mathrm{Prop}}{\Gamma \vdash \{x : T | U\} : s}$$

$$\frac{\Gamma \vdash M : T \quad \Gamma \vdash T \leqslant U}{\Gamma \vdash M : U}$$

FIGURE 1 – Règles de Russel

```
Definition transport {p} (f : subp p → Type) :=
  ∀ q : subp p, ∀ r : subp q, (f q) → (f (ι r)).
Definition refl (Θ : transport f) := ∀ q : subp p, ∀ x : f q, Θ q (ι q)
x = x.
Definition trans (Θ : transport f) :=
  ∀ q : subp p, ∀ r : subp q, ∀ s : subp r,
    ∀ x, ((Θ (ι r) s) ∘ (Θ q r)) x = Θ q (ι s) x.
Definition presheaf (p : P) :=
  {f : subp p → Type & {Θ : transport f | refl Θ ∧ trans Θ }}.
Definition presheaf_f {p : P} (s : presheaf p) : subp p → Type := π1
s.
Program Definition Θ {p : P} (s : presheaf p) : transport (presheaf_f
s) := π2 s.
```

La fonction ι permet de pouvoir considérer, si r est de type subp q et s de type subp r, que $\iota\ s$ est de type subp q.

5 Formalisation en Coq

À partir de maintenant, on se permet de rajouter deux axiomes à notre théorie :

— l'extensionnalité pour les fonctions dépendantes : pour montrer que deux fonctions dépendantes f et g sont égales, il suffit de montrer que pour tout x, $f(x) = g(x)$;

— l'extensionnalité pour les formules : pour montrer que deux formules sont égales, il suffit de montrer qu'elles sont équivalentes.

```
Axiom FunExtDep : ∀ {A} {B : A → Type} {f g : ∀ x : A, B x}, (∀ x : A, f x = g x) → f = g .
Axiom Prop_ext : ∀ A B :Prop, (A ↔ B) → A = B.
```

5.1 Faisceaux

On définit alors les notions de recouvrement pour la topologie dense, en donnant une définition de partie dense, de famille coïncidante et d'amalgamation, puis on donne une définition de faisceaux pour la topologie dense, à un niveau p.

On rappelle la topologie de Grothendieck pour la topologie dense, vue dans la partie 2.2 :

$$J(p) = \{D \mid \forall q \in D,\ q \leqslant p \text{ et } D \text{ est un crible dense sous } p\}.$$

On adapte donc cette définition en Coq, en utilisant les subset-types :

```
Record dense_cover_prop {p :P} (q :subp p) (D :subp q → Prop) : Type :=
  mkCover
    {
      denseCover : ∀ r :subp q, ∃ s :subp r, D (ι s) ;
      sieveCover : ∀ r :subp q, D (ι r) → ∀ s :subp r, D (ι s)
    }.
Definition dense_cover {p :P} (q :subp p) := {D:subp q → Prop & dense_cover_prop q D}.
Definition in_dense_cover {p :P} (q :subp p) (D :dense_cover q) :=
  { r:subp q | (π1 D) (ι r)}.
```

On adapte ensuite les définitions de famille coïncidante et d'amalgamation au cas de la topologie dense, en utilisant la fonction Θ qui récupère les fonctions de restriction d'un préfaisceau :

```
Definition matching_family_prop {p :P} (q :subp p) (F :presheaf p)
(S :dense_cover q) (M :∀ r :in_dense_cover q S, presheaf_f F (ι r)) :=
```

```
  ∀ r :in_dense_cover q S, ∀ s :subp r, (Θ F (ι r) (s)) (M r) = M
(dense_cover_dclose_proj q S r s).

  Definition matching_family {p :P} (q :subp p) (F :presheaf p)
    (S :dense_cover q) :=
    {M:∀ r :in_dense_cover q S, presheaf_f F (ι r) | matching_family_prop
q F S M }.

  Definition amalgamation_prop {p :P} (q :subp p) (F :presheaf p)
    (S :dense_cover q) (M :matching_family q F S) :=
    ∃! x :presheaf_f F q, ∀ r :in_dense_cover q S, (Θ F q r) x = π_1 M r.
```

Enfin, on utilise à nouveau les subset-types pour définir les faisceaux, qui sont donc les préfaisceaux pour lesquels toute famille coïncidante a une unique amalgamation.

```
  Definition sheaf_prop {p :P} (F :presheaf p) :=
    ∀ q :subp p, ∀ S :dense_cover q, ∀ M :matching_family q F S, amalga-
mation_prop q F S M.

  Definition sheaf (p :P) :=
    { F:presheaf p | sheaf_prop (p :=p) F }.
```

On définit les types des morphismes :

— presheaf_mor (resp. sheaf_mor) est le type des morphismes de préfaisceaux (resp. faisceaux) ;

— presheaf_mor_presheaf (resp. sheaf_mor_presheaf) le type des exponentielles de préfaisceaux (resp. faisceaux) :

```
  Definition presheaf_mor {p :P} (F :presheaf p) (G :presheaf p) :=
    {m : ∀ q :subp p, (presheaf_f F) q → (presheaf_f G) q |
     ∀ (q :subp p) (r : subp q) (x :(presheaf_f F) q) , m (ι r) (Θ F q r
x) = Θ G q r (m q x)}.

  Definition sheaf_mor {p :P} (F :sheaf p) (G :sheaf p) :=
    presheaf_mor (sheaf_f F) (sheaf_f G).

  Program Definition presheaf_mor_presheaf {p :P} (F G :presheaf
p) : presheaf p :=
    existT (fun q :subp p ⇒ ∀ r :subp q, (presheaf_f F) (ι r) → (presheaf_f
G) (ι r))
           (λ s :subp p, λ t :subp s, λ x : ∀ r :subp s, (presheaf_f F) (ι
r) → (presheaf_f G) (ι r), λ r :subp t, λ y :(presheaf_f F) r, x (ι r) y).

  Program Definition sheaf_mor_presheaf {p :P} (F G :sheaf p) : sheaf
p :=
    existT (fun q :subp p ⇒ ∀ r :subp q, (presheaf_f (sheaf_f F)) (ι r) →
(presheaf_f (sheaf_f G)) (ι r)) (λ s :subp p, λ t :subp s, λ x : ∀ r :subp s,
```

(presheaf$_f$ (sheaf_f F)) (ι r) $\to$ (presheaf$_f$ (sheaf_f G)) (ι r), λ r :subp t, λ y :(presheaf$_f$ (sheaf_f F)) r, x (ι r) y).

On notera un morphisme de préfaisceau par $\cdot \xrightarrow{PSh} \cdot$, et un morphisme de faisceau par $\cdot \xrightarrow{Sh} \cdot$. On définit d'abord les préfaisceaux constants, pour ensuite les sheafifier pour obtenir les faisceaux constants.

```
Program Definition Δ {p :P} T : presheaf p :=
  existT (fun q ⇒ T) (λ s : subp p, λ t : subp s, λ (x :T), x).
```

On a besoin dans la preuve d'indépendance du foncteur de sheafification **a**, qu'on se contente de définir dans le cas précis des préfaisceaux constants, et de notre monomorphisme g défini par l'équation (2).

```
Program Definition constant_sheaf_transport (T :Type) :
  ∀ (q : subp p) (r : subp q),
    {C : dense_cover q & in_dense_cover q C → T} →
    {C : dense_cover (ι r) & in_dense_cover (ι r) C → T}.

Program Definition constant_sheaf (T :Type) : sheaf p :=
  existT (fun q :subp p ⇒ {C:dense_cover q & ∀ r :in_dense_cover q
C, T}) _.

  Program Definition sheaf_g (p :P) (g : (Δ B) --PSh-->
    (presheaf_mor_presheaf (Δ nat) (Omega))) :
    (constant_sheaf B) --Sh--> (sheaf_mor_presheaf (constant_sheaf nat)
Omega_d).
```

5.2 Preuve d'indépendance de l'hypothèse du continu

On fixe notre ensemble $B = \mathfrak{pp}(\mathbb{N})$, puis on définit la notion de forcing. On choisit de représenter les conditions de forcing p – les fonctions partielles finies de $B \times \mathbb{N}$ dans 2 – comme couples de listes de couples ; la première liste contient les couples (b, n) tels que $p(b, n) = 1$, et la seconde ceux pour lesquels $p(b, n) = 0$.

```
Definition B := nat → Prop → Prop.
Definition P :=
  { x : (list (B×nat)) × (list (B×nat)) | disjoint_list x}.

Definition le (x :P) (y :P) :=

(∀ a : B × nat, (y (a) = 1) → x (a) = 1) ∧
```

```
(∀ a : B × nat, (y (a) = 0) → x (a) = 0).

Instance le_pre : PreOrder le.
```

On va ici changer un peu la preuve vue dans la partie 3 ; on ne considère plus $g : \Delta(B) \to \Omega_d^{\Delta(\mathbb{N})}$, mais plutôt comme $g : \Delta(B) \to \Omega^{\Delta(\mathbb{N})}$, pour le sheafifier ensuite. On prend donc pour Ω le préfaisceau constant égal à Prop.

```
Definition Omega := Δ (p :=p) Prop.
```

On ne construit pas l'ensemble A (cf. (1)) pour ensuite transposer sa fonction caractéristique : il est plus simple de partir directement de la définition de g vue en (2), et de montrer par la suite que g est bien un monomorphisme de préfaisceaux.

```
Program Definition g (p :P) : (Δ B) ⟶^PSh (presheaf_mor_presheaf (Δ nat) (Omega)).
```

Pour la preuve d'injectivité de g, on a besoin de construire, pour chaque condition de forcing p, un entier n_0 pour lequel $q(n,b)$ n'est défini pour aucun b ; on choisit pour n_0 la somme de tous les entiers pour lesquels p est définie, à laquelle on ajoute 1 (ce que fait la fonction sum_condition).

```
Lemma sum_lemma (p :P) : ∀ b :B, undefined p (b,sum_condition p).
```

On peut aussi, pour b et c distincts, de type B, et pour une condition de forcing q qui ne soit pas définie en (b,n) et (c,n) pour un entier n fixé, étendre q pour avoir $q(b,n)$=1 et $q(c,n)$=0.

```
Program Definition extension (q :P) (n :nat) (b :B|undefined q (b,n))
(c :B|c≠b∧undefined q (c,n)) : (subp q) :=
    exist (P :=Psub q) ((‘b,n)::(fst (π1 q)),(‘c,n)::(snd (π1 q))) _.
```

Le lemme suivant permet donc de montrer que g est bien un monomorphisme (cf. lemme 3.3) de préfaisceaux en utilisant sum_lemma et extension pour construire les r et n demandés.

```
Program Lemma lemma3.3 : ∀ (q :subp p), ∀ b :B, ∀ c :B, (b ≠ c) → ∃
r :subp q, ∃ n :nat, ((g p q b r n) ≠ (g p q c r n)).
```

On a en fait montré la contraposée de l'injectivité de g. On utilise ici un peu de logique classique pour montrer le "bon sens".

```
Axiom not_forall : ∀ (A :Type), ∀ (P :A → Prop), ¬ (∀ x :A, P x) → ∃
x :A, ¬ P x.
```

Axiom *exists_not* : ∀ (*A B* :Type), ∀ (*P* :*A* → *B* → Prop), (∃ *x* :*A*, ∃ *y* :*B*, ¬ *P x y*) → ¬ (∀ *x* :*A*, ∀ *y* :*B*, *P x y*).

Program Lemma lemma3.3' : ∀ (*q* :subp *p*), ∀ *b* :B, ∀ *c* :B, ¬¬ (∀ *r* :subp *q*, ∀ *n* :nat, (*g p q b r n*)=(*g p q c r n*)) → ¬¬ (*b*=*c*).

Axiom *NN* : ∀ *P* :Prop, ¬¬ *P* ↔ *P*.

Program Lemma lemma3.3" : ∀ (*q* :subp *p*), ∀ *b* :B, ∀ *c* :B, (∀ *r* :subp *q*, ∀ *n* :nat, (*g p q b r n*)=(*g p q c r n*)) → (*b*=*c*).

Il faut alors montrer que l'image par le foncteur de sheafification de g reste un monomorphisme, et on aura alors bien les injections voulues.

Lemma injective_a {*p* :P} :
∀ *g* : (Δ B) $\xrightarrow{PSh}$ (presheaf_mor_presheaf (Δ nat) (Omega)),
(∀ *q* :subp *p*, ∀ *b c* :B, (∀ *r* :subp *q*, ∀ *n* :nat, π_1 *g q b r n*= π_1 *g q c r n*) → *b*=*c*) →
(∀ *q* :subp *p*, ∀ *b c* :presheaf$_f$ (sheaf_f (constant_sheaf B)) *q*,
π_1 (sheaf_g *p g*) *q b* = π_1 (sheaf_g *p g*) *q c* → *b*=*c*).

On a finalement l'injection voulue, sheaf_g *g*.

Theorem ch {*p* :P} :
∀ *q* :subp *p*, ∀ *b c* :presheaf$_f$ (sheaf_f (constant_sheaf B)) *q*, (π_1 (sheaf_g *p* (*g p*))) *q b* = (π_1 (sheaf_g *p* (*g p*)))*q c* → *b*=*c*.

5.3 Travaux futurs

On remarque dans le code qu'on a admis les deux preuves :
— du fait que le sheafifié du préfaisceau constant est bien un faisceau ;
— du fait qu'un morphisme de préfaisceau entre deux faisceaux est bien un faisceau.
Ces deux faits sont prouvables, mais leur preuve passerait par la construction du foncteur de sheafification dans le cas général, ce qu'on a voulu éviter par manque de temps.

D'autres *admit* ont une raison plus profonde. On travaille dans cette preuve avec des types dépendants, dont on doit gérer l'égalité. Or Coq ne sait pas bien gérer l'égalité : il utilise une égalité syntaxique, et nous utilisons usuellement une égalité sémantique. Si T et U sont deux types, et $t : T$ et $u : U$, on aimerait pouvoir dire $u=v$ si T et U sont deux types isomorphes.

Cela devient possible avec la *théorie des types homotopiques*, qui repose sur l'axiome d'univalence de Voevodsky, qui affirme justement que deux types isomorphes sont égaux. Dans les grandes lignes, un jugement $a : A$

est interprété par "A est un espace, et a est un point de A". La preuve d'une égalité entre deux points a et b de A est un chemin de a à b, et deux de ces preuves sont considérées égales si les chemins correspondants sont homotopes.

Pour terminer la preuve d'indépendance, on peut donc étudier la théorie des types homotopiques, pour pouvoir gérer les problèmes d'égalité entre types dépendants. La prochaine étape est aussi de formaliser la seconde partie de la preuve, pour montrer qu'il n'existe pas d'épimorphismes entre les objets considérés ; il faut pour cela formaliser la condition de chaîne dénombrable, et montrer qu'elle permet bien de conclure.

Références

[Jec] Thomas Jech. *Set theory.*

[JTS12] G. Jaber, N. Tabareau, and M. Sozeau. Extending type theory with forcing. pages 395–404, 2012.

[Kun] Kenneth Kunen. *Set theory.*

[MM92] Saunders MacLane and Ieke Mœrdijk. *Sheaves in Geometry and Logic, a First Introduction to Topos Theory.* Springer-Verlag, 1992.

[Pro10] Alain Prouté. *Introduction à la Logique Catégorique.* 2010.

[Pro11] Alain Prouté. Pourquoi les topologies de grothendieck intéressent-elles les logiciens ? 2011.

[Soz08] Matthieu Sozeau. *An environment for programming with dependent types.* PhD thesis, 2008.

[Tie72] Myles Tierney. Sheaf theory and the continuum hypothesis. 1972.