Entretien avec Théo Zimmermann - Coq :

Pouvez-vous nous présenter brièvement votre parcours académique ?

Pour faire bref, j'ai fait une classe préparatoire, ensuite une formation en informatique à l'ENS, puis j'ai exploré différentes voies de recherche qui me semblaient intéressantes pour continuer plus tard. Je me suis d'abord intéressé à la vérification de programmes ou de composants matériels : ça peut être une application de Coq en particulier, ou des systèmes de preuves en général, mais ça peut également être fait avec d'autres choses qui ne sont pas forcément liées au fait de prouver des théorèmes.

Ensuite, j'ai eu envie de faire des choses qui pouvaient servir à faire de la recherche en mathématiques, et ça pour le coup ce sont spécifiquement les assistants de preuves qui le permettent, avec en tête l'idée de rendre cette recherche en mathématiques plus collaborative, en faisant participer plus de monde au travers d'interfaces numériques. Même si c'est totalement hors de portée aujourd'hui, ce que j'essaye de faire dans le cadre de Coq c'est de le rendre plus accessible pour que plus de gens soient capables de l'utiliser, afin que peut-être un jour on puisse permettre à des gens de collaborer sur des preuves mathématiques en utilisant ce genre d'outils. C'est comme ça que je suis arrivé à mon sujet de thèse actuel.

Est-ce que vous pensez que le fait que la preuve du 4CT de 1976 ait posé controverse soit lié au fait que l'informatique ne soit pas assez connue et accessible à l'époque ?

Je n'ai pas une connaissance suffisamment approfondie de ce théorème pour répondre avec certitude, mais mon impression c'est que c'est davantage le caractère "sans précédent" [qui est à l'origine de la controverse]. Jusqu'à présent, **les preuves mathématiques avaient toujours été des choses qui pouvaient être entièrement écrites par un humain et lues par d'autres humains** ; et là il y avait un composant pour prouver un certain nombre de cas de base qui était un programme informatique qui vérifiait automatiquement. **Cela veut dire d'abord qu'il n'y a pas d'humain qui a écrit la preuve entièrement, mais aussi qu'il n'y a pas d'humain qui puisse vérifier la preuve entièrement**. Si on considère que la preuve n'inclut pas le programme informatique mais uniquement la sortie du programme et encore, je crois qu'il n'y avait même pas de sortie en l'occurrence, il fallait faire confiance au fait que le programme avait vérifié les cas des bases.

Il me semble que cette controverse était légitime même si on ne se méfie pas de l'informatique, parce qu'en informatique, on sait qu'il y a des bugs et que les programmes ne donnent pas forcément quelque chose qui est vrai, dans beaucoup de cas ils font des approximations et si on veut vraiment avoir confiance il faut rentrer un peu plus dans les détails. Et donc ce qu'a fait Georges Gonthier en faisant une preuve formelle du théorème mais informatique, cela permet d'atténuer un peu ce genre de controverse, parce qu'il a beau garder un aspect informatique à la preuve, elle est entièrement vérifiée à l'aide d'un logiciel qui a été programmé spécifiquement pour ça. Du coup, **il y a beaucoup moins de risques que**

la preuve soit fausse : il faudrait qu'il y ait un bug dans le vérificateur lui-même -ce qui est possible- mais aussi que ce bug soit exploité par la preuve sans que les auteurs de la preuve ne s'en rendent compte.

Quand on avait vu Pierre-Evariste Dagand, il nous avait détaillé au tableau le noyau de Coq, et on a pu voir que ce dernier était tellement simple que si on était sûr que le noyau était correct, on était certain de la preuve.

Le noyau de Coq, on cherche à le rendre simple (et il est effectivement simple par rapport à d'autres systèmes informatiques plus complexes y compris Coq lui-même si on inclut tout), mais ça ne veut pas dire qu'il n'y a pas de bug dedans (actuellement on espère qu'il n'y en a pas). Quand je dis "bugs", je veux dire des bugs qui puissent entraîner le fait de prouver des choses fausses. De tels bugs il y en a eu par le passé, mais ils n'ont jamais été découverts par quelqu'un qui faisait une preuve et qui s'est par hasard retrouvé à prouver quelque chose de faux. Non, ils ont toujours été découverts par quelqu'un qui connaissait bien le code du noyau et qui avait une perception poussée de quels pouvaient être les problèmes d'implémentation, et qui a cherché à exploiter ces problèmes d'implémentation pour trouver des bugs. En gros, même si aujourd'hui le noyau de Coq ne peut pas être considéré sûr à 100%, **lorsqu'on fait une preuve d'un théorème par Coq et qu'on arrive à la fin de cette preuve, on peut être sûr à 100% que la preuve est valide**, parce qu'il n'y a quasiment aucune chance qu'on soit tombé dans un cas -qui probablement de toute façon n'existe pas- un de ces bugs qui se font de plus en plus rares, et qui a priori doivent être aujourd'hui en nombre très limité si ce n'est nul.

Justement, vous dites "quasiment", "a priori"... N'avez-vous pas l'impression que depuis la preuve de 1976, voire depuis les théorèmes d'incomplétude de Gödel, les mathématiciens ont renoncé à être dans la certitude absolue ?

Ah si, absolument, c'est une question très intéressante. **C'est beaucoup plus les théorèmes de Gödel qui ont ébranlé les mathématiciens que le fait de concevoir des preuves avec des ordinateurs**. Ces théorèmes montrent qu'il n'y a pas une vérité absolue, et donc qu'on peut remettre en cause des certitudes qu'on avait initialement. Coq lui-même est basé sur une logique constructiviste ou intuitionniste, c'est-à-dire qu'il n'inclut pas des axiomes assez courants quand on fait des mathématiques, comme le principe du tiers exclu (pour toute proposition A, on a soit A soit non-A), qui est une proposition que l'on n'inclut pas par défaut dans Coq. Ça ne veut pas dire qu'on ne peut pas l'ajouter, et évidemment pour des tas des théorèmes on va l'ajouter. Daniel de Rauglaudre [son collègue] parlait tout à l'heure du paradoxe de Banach-Tarski, lui il ajoute aussi l'axiome du choix qui est un axiome qui a été beaucoup plus controversé, même dans le milieu des mathématiques classiques. Ce que ça signifie, ces histoires d'axiomes qu'on choisit de ne pas inclure par défaut, ça signifie qu'il n'y a pas qu'une vérité absolue : **on peut démontrer des théorèmes en se basant sur un**

certain nombre d'axiomes, mais si on part d'autres axiomes, d'autres postulats, on peut démontrer d'autres théorèmes éventuellement incompatibles.

Tout à l'heure, vous parliez du fait qu'en informatique le côté humain était moins présent, mais tout de même l'humain conserve-t-il une part importante dans les preuves informatiques ?

La place de l'humain reste énorme dans les preuves formelles informatisées, parce que même si par ailleurs des gens ont développé des outils permettant de trouver des preuves de manière automatique, cela reste des théorèmes qui sont assez simples. Lorsqu'on s'attaque vraiment à des théorèmes difficiles tel que le théorème des 4 couleurs, mais même des théorèmes qui restent moins difficiles que ça, à la portée d'étudiants en mathématiques, on tombe dans des cas que les machines ne sont pas capables de trouver automatiquement, **on a besoin d'exploiter la connaissance accumulée par les mathématiciens au fur et à mesure de leur recherche, et retraduire cette connaissance dans une preuve formelle qui va pouvoir être vérifiée. Il reste une grande part à l'humain**.

Par contre, on donne une plus grande part au calcul également. La preuve du théorème des 4 couleurs avait besoin de calculer un certain nombre de cas de base, et de vérifier que tous ces cas de bases étaient corrects. Certaines parties d'une preuve peuvent être réduites à du calcul, et dans ces cas-là l'humain devient moins important dans le sens où le calcul va se faire, et l'ordinateur sera content et dira "OK c'est correct". **Ceci dit, pour que le calcul se fasse, il subsiste la nécessité que l'humain ait réfléchi avant à comment on allait faire le calcul**, et même dans la preuve originale du théorème des 4 couleurs, avant qu'elle soit formalisée, il y avait quand même un humain qui avait réfléchi, le programme ne s'est pas fait tout seul.

On a pu remarquer qu'il y avait une séparation entre mathématiques et informatique, au niveau de la place de l'expérimentation. Est-ce que vous pensez qu'il y ait encore aujourd'hui un cloisonnement à faire entre les deux, que l'informatique soit moins légitime car fondée sur l'expérimentation ?

D'abord, toute l'informatique n'est pas comme ça, **il y a de l'informatique extrêmement théorique, en particulier dans ces laboratoires il y beaucoup de gens qui ne font pas du tout d'expérimentation**. Mais effectivement, on peut considérer que cette branche-là de l'informatique se rapproche beaucoup plus des mathématiques telles qu'on les entend, alors qu'une autre partie de l'informatique pourrait se rapprocher davantage de la physique et des autres sciences expérimentales. **Ceci dit, l'informatique, en permettant plus d'expérimentation et notamment des simulations, a tendance à imposer aux autres sciences aujourd'hui de faire plus d'expérimentation**. Cela a un impact en physique et en biologie, où une large part du travail est aujourd'hui basée sur des simulations informatiques, mais **même en mathématiques l'expérimentation peut s'avérer extrêmement utile pour comprendre ce**

qui se passe. Les gens qui font de la géométrie aujourd'hui sont très contents de la possibilité d'avoir des programmes qui leur permettent de mieux visualiser des structures sur des exemples, parce que l'expérimentation ce sera toujours sur des exemples mais ça permet de se construire une idée de ce qui se passe et ensuite d'essayer de généraliser et d'en tirer des conclusions universelles.

Vous parliez de votre objectif de rendre Coq plus accessible. Pensez-vous que des mathématiciens n'utilisent pas Coq par soucis d'accessibilité et qu'une certaine dépendance vis-à-vis des informaticiens pourrait être créée ?

Oui, absolument. Difficile à aborder, donc la plupart des mathématiciens n'en sont pas encore au stade d'utilisation de Coq. Ceci dit, en étant motivé, il peut tout à fait apprendre et ne pas avoir besoin de se reposer sur des informaticiens. L'exemple le plus connu est sans doute celui de **Vladimir Voevodsky**, dont vous entendrez encore parler certainement. Il est l'exemple typique d'un **vrai mathématicien qui ait découvert l'intérêt des preuves formelles en se rendant compte d'erreurs dans ses publications scientifiques antérieures** et que donc le système de revue par les pairs qui a encore cours aujourd'hui montrait ses limites. Des erreurs pouvaient se glisser sans s'en apercevoir, d'où l'intérêt de la formalisation. Il y a eu beaucoup de contacts entre informaticiens et mathématiciens, ce qui a donné lieu à de nombreux domaines de recherche, qui en définitive ont fait s'estomper la frontière informatique/mathématique. Pas besoin d'être médaillé Fields pour comprendre comment utiliser Coq. Beaucoup d'étudiants en thèse assistent à des universités d'été pour apprendre à s'en servir et formaliser une partie de leurs résultats. Mais ça ne veut pas dire qu'on ne peut pas rendre le système beaucoup plus accessible. Il faudrait qu'un mathématicien puisse utiliser le système sans se poser la moindre question sur le système lui-même. Ce qui n'est pas le cas aujourd'hui !

Comment contribuez-vous à rendre Coq plus accessible ?

Beaucoup de choses. Les interfaces utilisateur sont un des aspects. Dans mon travail de thèse, je me concentre sur une chose en particulier : **rendre les preuves Coq plus lisibles**. Une fois qu'une preuve est écrite en Coq, il faudrait qu'un mathématicien puisse vérifier qu'elle soit correcte manuellement, sans avoir besoin d'utiliser Coq. À partir des scripts de preuves, un enchaînement de blocs de bases (diptyque), relativement illisibles, on devrait pouvoir générer un texte, semblable à une publication de revue dans un langage qui soit plus compréhensible par des mathématiciens sans formation particulière sur Coq. C'est un objectif relativement difficile, mais c'est un objectif intermédiaire avant de rendre coq totalement accessible par les mathématiciens. **Il n'y aura au moins pas besoin de comprendre Coq pour s'en servir**.

Le fait de rajouter des couches pour rendre Coq plus accessible, n'y a-t-il pas ici un risque de multiplier les bugs et rendre le système moins fiable ?

Si, c'est intéressant. Je ne veux pas répondre "oui" trop vite. Tant que l'on multiplie les bugs à l'extérieur du noyau, une preuve vérifiée par le noyau peut être correcte. Mais si on multiplie les bugs dans les surfaces avec lesquelles il interagit, le risque est de croire avoir prouvé quelque chose qui ne correspond pas en réalité à ce qui a été prouvé. On a mal perçu ce qu'on a perçu *in fine*. Cela s'appelle l**'inconsistance de Pollack**. Il avait remarqué que dans un certain nombre de systèmes, on pouvait faire croire au lecteur qu'on avait prouvé faux. Normalement cela ne devrait pas être possible mais à cause des couches supérieures d'affichage. D'une certaine manière, l'ensemble peut être rendu moins fiable sous certains aspects, il est vrai. Ceci dit, on peut trouver des mesures pour éviter cela. Par exemple, on peut interpréter le texte généré et le revérifier dans Coq, hors de portée de vérifier tous les textes mathématiques, mais les textes sous la forme de ceux pouvant être générés par cet outil de traduction. Si on a ce genre d'étapes, c'est une sécurité. On peut demander à Coq de vérifier une dernière fois que la preuve est correcte.

Cela ajoute donc une notion de "réplicabilité" qu'on retrouve dans les sciences expérimentales ?

Oui tout à fait. Il y a aussi une volonté d'avoir plusieurs implémentations du vérificateur du noyau. Ce n'est pas vraiment le cas aujourd'hui, mais rien n'empêche de faire un programme indépendant qui vérifie le même genre de preuve que Coq vérifie. Donc on aura encore plus de certitude que la preuve soit valide. C'est profondément lié à la réplicabilité.

Est-ce que la preuve formelle par informatique génère des problèmes que l'on ne rencontre pas dans le cas de preuves mathématiques ?

Oui, il y a de multiples problèmes rentrant dans cette catégorie. Des notions triviales en mathématiques ne le sont plus du tout lorsqu'on veut les intégrer dans un ordinateur. **L'ordinateur s'améliore de manière constante mais il y a beaucoup de preuves mathématiques qu'il faut beaucoup détailler pour les faire comprendre à Coq**. Il y a aussi des problèmes d'implémentation. *Grosso modo* lorsqu'on manipule des objets mathématiques, on ne perd pas trop de temps à réfléchir comment ils sont construits. Les mathématiciens vont parler des nombres entiers, des nombres réels sans se demander ce que c'est réellement, c'est une connaissance intuitive, ils ont juste besoin de connaître leurs propriétés de base. Cependant, on part de zéro par informatique. Les choix à l'implémentation sont donc cruciaux qui vont rendre la preuve plus ou moins facile à réaliser. Par ailleurs, plusieurs implémentations sont possibles pour représenter le même objet. La question se pose ainsi de savoir s'il en existe une meilleure que l'autre. Cela dépend en fait de ce que l'on veut prouver.

On peut lier aussi les deux ensembles, transférer de la connaissance issue d'une structure à celle obtenue d'une autre structure, représentant le même objet mais de manière différente.

Pour revenir à la notion de preuve, les critiques apportées par Tymoczko s'attachaient à la notion d'a priori : la preuve n'était pas faite uniquement dans la tête mais faisait intervenir un ordinateur. Que pensez-vous de cela ?

Je pense que cela se rapproche d'un prolongement, ne changeant pas fondamentalement le statut des mathématiques. À l'origine, les preuves n'étaient pas faites entièrement dans la tête, on avait au moins besoin de papier et de stylo. Cela faisait bien longtemps qu'on ne pouvait plus avoir de preuve entière dans la tête, mais cette époque a t-elle au moins existé ? J'ai vu à propos de cela un argumentaire intéressant. Une preuve est avant tout **un objet de communication**. **Il faut pouvoir convaincre son assistance que le théorème est correct. Cela tient plus au langage et à la communication avec les autres.** Si des outils externes (tableau ou ordinateur) permettent de convaincre l'assistance qu'une preuve est correcte, il n'y a pas forcément de problème philosophique de ce côté-là. Cette idée de communication s'est illustrée récemment par des controverses sur des travaux mathématiques n'ayant rien à voir avec la preuve informatisée. Il s'agit d'une personne qui avait travaillé dans l'isolement des décennies qui avait construit toute une théorie sur laquelle il s'était appuyé pour démontrer une conjecture. Tout le monde a trouvé cela formidable mais personne n'a été capable de comprendre la preuve. Est-ce qu'elle est plus valable qu'une autre parce qu'un mathématicien est capable de la comprendre entièrement ? En fait, on peut se demander si c'est vraiment une preuve. **Il s'agit de Shinichi Mochizuki à propos de la conjecture ABC**. On n'a pas encore réussi à la comprendre mais des gens y travaillent. Le problème est que ce mathématicien a refusé de donner des séminaires pour expliquer, on ne peut uniquement se référer aux écrits.

Vous parlez de l'importance de la communication, pensez-vous que le critère d'élégance est important dans une preuve ?

C'est une question assez intéressante que nombre de personnes se sont posés. Le point de vue général est que l'élégance est secondaire. Le plus important est que la preuve soit juste. Il arrive qu'une seconde preuve soit publiée parce que plus élégante et c'est celle-ci qu'on choisit pour la présenter ensuite. **L'élégance permet aussi de rendre la preuve plus claire. Dans les preuves formelles, l'élégance passe encore plus au second plan parce que la relecture devient inutile**. Pour la définition, on regarde les définitions et les énoncés des théorèmes et les hypothèses, pas les preuves, que l'ordinateur va vérifier. Il peut y avoir certaines formes d'élégance mises en avant par les auteurs, avec des méthodes qui peuvent être appliquées par la suite. Mais si l'auteur ne fait pas cet effort, cela passe vraiment au second plan. Le projet sur lequel je travaille, pour rendre les preuves plus lisibles contribue à ce que le critère d'élégance puisse compter pour la suite.

À votre avis, est-ce c'est pour cette raison que l'on continue à chercher des preuves manuelles du théorème ?

L'Homme n'a toujours pas vérifié entièrement la preuve. Chercher manuellement permettrait de comprendre de A à Z, ce qui serait satisfaisant. **Par ailleurs, ce n'est jamais du temps perdu que de chercher des preuves de théorèmes déjà connus. Cela permet de tirer de nouvelles intuitions et de comprendre ce qu'il se passe.**

Selon le domaine de recherche, les personnes que nous avons rencontrées faisaient intervenir différents critères de preuve. Quels critères principaux selon vous ?

Fondamentalement, la preuve doit être cohérente et respecter les éléments essentiels de la logique. **Le critère de vérifiabilité est très important** mais on peut le définir de différentes manières, vérifier par ordinateur est pour certains le seul moyen d'affirmer que la preuve soit vraie. Cette position est raisonnable. Mais il y a des exemples qui montrent que la vérification ne garantit pas que la preuve soit vraie non plus.

Ordinateur plus fiable que l'humain ? Se pourrait-il qu'un jour on se fie complètement à l'ordi ?

Cela m'étonnerait. En fait, la question de l'intelligence artificielle est intéressante. Cette notion n'a plus rien à voir avec la définition qu'on lui donnait dans les années 1950. D'une certaine manière, les systèmes de preuve qui existent aujourd'hui sont les héritiers directs de ces systèmes d'Intelligence Artificielle. À l'époque, on a cherché à inventer des systèmes intelligents manipulant des règles de raisonnement, on a écrit des langages de programmation pour cela. Cela est difficile mais on a réussi à faire des "prouveurs automatiques". Cela est simple au niveau technique mais peut s'avérer complexe car très long avec beaucoup d'étapes laborieuses. On a aussi des systèmes qui peuvent vérifier des preuves plus complexes mais au prix que l'humain ne puisse plus interagir**. Aujourd'hui, l'Intelligence Artificielle n'est plus basée sur des raisonnements logiques mais plutôt sur les inférences statistiques**. Les systèmes qu'on ne comprend pas vraiment vont s'adapter aux données du problème et prédire de mieux en mieux les résultats. Mais cela est approximatif et est voué à le rester. Cela signifie que même si Google classe bien les images par catégorie, il peut y avoir des anomalies, idem pour Spotify etc… Ce n'est pas gênant sur ce type d'application mais cela l'est davantage sur des applications critiques (pilotage d'avion, de fusée...). Il faut savoir ce qu'on lui confie. Si on revient sur la question des preuves vérifiées par ordinateur, il y a des gens qui s'intéressent aux applications modernes de l'Intelligence Artificielle (IA), avec **les réseaux de neurones** ou bien l'apprentissage statistique appliquées à la preuve de théorèmes. Ces applications sont intéressantes car elles vont peut être en mesure de trouver des preuves qui auraient demandé une certaine forme d'intelligence de la part des humains pour les trouver. Mais on ne peut pas se satisfaire du fait que l'Intelligence Artificielle ait trouvé la preuve, il faut trouver un système qui, lui, n'a rien d'intelligent et est fondamentalement bête pour la

vérifier. **Car l'implémentation doit être la plus petite possible. Il y aura toujours la dualité entre la recherche de preuve avec des techniques empiriques et la vérification en appliquant des règles de base limitées.**