La logique face à l'arbitraire

Sidney Congard

La logique face à l'arbitraire

Sidney Congard

6 juin 2022

Ce document fut réalisé dans le cadre du cours d'ouverture de philosophie en M2 au master LMFI. Il est à la fois un travail de synthèse du projet de syntaxe transcendantale de Jean-Yves Girard et une mise en contexte de ces travaux par rapport à d'autres résultats et réflexions personnelles en logique. Je vais tenter de montrer en quoi la généralité s'oppose à l'arbitraire présent en logique : rechercher une justification interne à la logique permettra de débusquer puis d'écarter cet arbitraire pour aboutir à la syntaxe transcendantale de Girard, une logique sans axiomes.

Pour cela, je commencerai par exposer la grille de lecture de Girard de la logique, puis je renouerai le lien entre la logique classique et celle de Girard en explorant ses sous-sols sémantiques :

— 0. Une grille de lecture Kantienne
— -1. Niveau aléthique : vérité & modèles
— -2. Niveau fonctionnel : preuves & catégories
— -3. Niveau interactif : tests & jeux
— -4. Niveau déontique : la négation comme format

0. Une grille de lecture Kantienne

Cette quête se place dans un esprit similaire aux arguments transcendantaux de Kant, avec cependant une différence notable : Girard ne prétend pas pouvoir révéler des conditions de possibilité nécessaires, seulement suffisantes (cela semble rejoindre ses critiques sur l'abduction [Gir2, p151]).

La logique est ici vue comme un cadre permettant de poser des questions (subjectives) et d'y apporter des réponses (objectives). Les questions comme les réponses peuvent être implicites ou explicites. Girard aboutit à cette grille de lecture d'inspiration Kantienne :

	Analytique / Réponses	Synthétique / Questions
A posteriori / Explicite	Constat	Usine
A priori / Implicite	Performance	Usage

L'espace analytique des réponses correspond à ce que l'on peut faire, sans se demander si on en a le droit. Il s'agit de la partie objective de la logique : sa part calculatoire, fidèlement représentable par un ordinateur. Les réponses sont alors des objets syntaxiques pouvant éventuellement être évalués. Selon le fait qu'on utilise une réponse telle quelle ou qu'on l'évalue, on se situe dans le constat (réponse explicite) ou la performance (réponse implicite). Une réponse n'est pas forcément implicite, cela dépend de son usage : 2^{40} peut suffire au lieu d'être "explicité" en 1099511627776. De même, le langage LISP permet de manipuler des programmes en les lisant au lieu de les exécuter, et nous pouvons exposer une boîte à musique au lieu de l'activer.

La séparation entre les deux vient du fait qu'on ne peut pas réduire la performance au constat. En effet, les constats sont sûrs mais ne peuvent que s'accumuler, alors qu'une performance ne peut pas être prédite (donc identifiée à un résultat, notamment via le problème de l'arrêt) et est destructrice (on ne conserve pas ses étapes intermédiaires). Ces risques permettent à la performance d'être très efficace : cela revient à comparer une table de logarithmes au programme 'log' d'une calculatrice.

Seule, la réponse est indiscutable : elle ne veut rien dire et ne réfère à rien. C'est donc en disant à quoi elle répond qu'on lui donne un sens. Pour cela, il va falloir interagir avec elle, en établissant un questionnement : c'est cette série de tests qui va constituer la question. Girard donne l'exemple d'un protocole entre un DVD et un lecteur de DVD [Gir2, p62] : le sens de l'appellation "DVD" est ici fondé sur des tests entre un DVD et un lecteur de DVD (affichant une langue inconnue pour exclure toute entente préexistante) menant au visionnage d'un film (cliquer là, descendre deux fois, valider, ...). Cette interaction est fondamentalement symétrique car il peut également s'agir d'un test pour le lecteur de DVD : le testeur est tout autant questionable que le testé.

Plus précisément, nous avons deux batteries de tests pour l'appellation "DVD", l'une correspondant à ses droits (conférés par l'usine) et l'autre à ses devoirs (requis par l'usage). La certitude logique concerne l'adéquation entre droits et devoirs. Les droits sont donnés au cas par cas par un lecteur-test : j'ai bien un DVD s'il passe les tests avec ce lecteur. Les droits de la négation reviennent à conférer l'appellation "lecteur DVD" via les mêmes tests et un DVD-test. Les devoirs de l'appellation "DVD" correspondent aux droits de sa négation, c'est-à-dire l'obligation du DVD d'être lu par tout lecteur DVD (les objets acceptés par un DVD-test). Les devoirs sont donc plus demandeurs que les droits.

Nous pouvons illustrer partiellement cette grille de lecture par la logique classique : les réponses correspondent aux preuves, et les questions correspondent aux formules. Pour une formule A, l'équivalent de l'usine est alors l'algorithme décidant si une réponse est bien une preuve de A. Cette ébauche d'usine nous fournit un premier critère formel : la logique doit être effectivement axiomatisable, afin d'avoir au moins la certitude (à priori arbitraire) d'avoir suivi les règles. Il s'agit d'un critère minimal : une preuve non vérifiable est-elle encore une preuve ? Cela écarte des logiques "à la tête du client" qui cherchent à interdire des démonstrations en fonction de ce qu'elles permettent de déduire : il s'agit généralement de critères indécidables (via le théorème de Rice).

Sans coupures, l'activité déductive est atrophiée : impossible de réutiliser une preuve comme lemme. L'usage correspond à l'utilisation des preuves acceptées pour en former d'autres. Vérifier l'usage revient entre autre à montrer qu'on ne prouve pas trop, ce que le second théorème d'incomplétude interdit : cela marque la séparation entre l'usine et l'usage. Cependant, une certitude raisonnable est possible (que Girard distingue d'une certitude légitime, qui serait absolue), par exemple pour la logique classique du 1^{er} ordre via l'élimination des coupures qui ramène l'usage à l'usine ... le problème étant que la preuve de l'élimination des coupures nécessite plus que cette logique.

En termes plus techniques, la grille précédente se précise de cette manière :

	Non-typé	Typé
Explicite	Données	Preuves sans coupures
Implicite	Programme	Preuves déductives

L'usine est donc synthétique car elle induit un choix de format. L'usage, lui, est synthétique car son adéquation avec l'usine reste fondamentalement incertaine, elle correspond au "meaning-as-use" de Wittgenstein. La correspondance du programme de syntaxe transcendantale avec Kant et d'autres philosophes est approfondie dans [Abr1].

-1. Niveau aléthique : vérité & modèles

L'approche classique en logique, que Girard qualifie de réalisme axiomatique, consiste à postuler des axiomes (et règles), des modèles et à montrer leur adéquation (typiquement via un théorème de complétude et un théorème de correction). L'objectif est donc ici de référer à des vérités à propos de ces modèles avec la syntaxe donnée. Cela soulève immédiatement deux questions naïves : D'où viennent ces axiomes ? D'où viennent ces modèles ?

Pour permettre l'existence d'un modèle correspondant, une théorie doit être consistante. Or, le second théorème d'incomplétude dit qu'une théorie seule ne peut pas prouver sa propre consistance. Les modèles restent donc une justification complètement externe à la logique : reposer sur un modèle pré-existant (ou de manière équivalente sur un "méta-langage") mène à une régression infinie.

La justification des axiomes (en plus de leur consistance) est aussi externe : ils ne peuvent pas être vérifiés, et en cas d'erreur (c'est-à-dire ici une contradiction), on ne sait pas auquel d'entre eux renvoyer la faute.

Cette approche classique est donc l'abandon d'une recherche de justification interne (propre à la théorie, sans faire de référence à un élément extérieur comme un modèle) à la logique. La consistance est le seul critère interne retenu mais reste hors d'atteinte. Pourtant, ce critère seul est laxiste : il permet d'accepter des entiers "non-standards" ou pire encore, une théorie telle que PA avec l'axiome postulant son inconsistance.

De plus, les critères de correction et de complétude permettent d'accepter des logiques complètement arbitraires en postulant des règles au niveau syntaxique qui correspondent à la même structure au niveau sémantique. C'est ce que Girard appelle les logiques Broccoli, des échecs à rendre compte du sens des règles logiques : par exemple, [Gir1] propose l'opération ♣ "Broccoli" avec l'axiome $(A\clubsuit B) \Rightarrow ((A\clubsuit A)\clubsuit B)$ interprété dans les structures équipées de l'opération $\heartsuit$ respectant $(a\heartsuit b) \leq ((a\heartsuit a)\heartsuit b)$.

Cela n'est pas une critique contre les modèles en général, qui demeurent utiles lorsqu'ils sont plus éloignés de la syntaxe, comme par exemple les espaces cohérents qui ont mené à la découverte de la logique linéaire. De plus, Girard distingue la logique classique des logiques Broccoli malgré leurs définitions similaires ($A \wedge B$ est vrai quand A est vrai **et** B est vrai, ...), en soulignant l'importance de la notion de vérité et de résultats comme le schéma de réflexion.

Ces laxismes restreignent la portée de la logique en acceptant des systèmes chacun consistants mais incompatibles entre eux : cela correspond à ce que l'on observe actuellement, avec de nombreuses logiques modales, temporelles, ... Ce qui est à l'opposé de ce que l'on pourrait viser en logique, c'est-à-dire des principes universels.

Puisque le théorème d'incomplétude empêche la confiance absolue en un système, il faudra isoler dans la logique ce qui demande une justification externe, un acte de confiance. La notion de vérité dans un modèle devant être postulée, on peut commencer par s'intéresser à ce que produit la logique : des preuves. Et puisque le théorème d'incomplétude sépare la vérité d'une preuve dans un système donné, il faut arriver à comprendre ce qu'est une preuve hors de tout système.

-2. Niveau fonctionnel : preuves & catégories

Une lueur d'espoir arrive avec BHK, l'interprétation de la logique intuitionniste des preuves comme fonctions :

— $A \Rightarrow B$ est une fonction associant à toute preuve de A une preuve de B.

— $A \vee B$ est une paire (i, p) où soit $i = 0$ et p est une preuve de A, soit $i = 1$ et p est une preuve de B.

— ...

En particulier pour $\neg A$, comme la prouvabilité d'une formule ne commute pas avec sa négation (d'après le théorème d'incomplétude), on ne peut pas avoir en général une preuve qu'il n'existe pas de preuve de A. BHK définit alors $\neg A$ comme $A \Rightarrow \perp$ (une fonction des preuves de A vers des preuves de l'absurde).

Cette approche permet d'évaluer les preuves d'un certain énoncé, comme la preuve $f(a)$ de B (notée $f(a) : B$) construite avec les preuves $f : A \Rightarrow B$ et $a : A$. Cela revient à éliminer les coupures, c'est-à-dire l'usage de lemmes intermédiaires (ici f et a). Les propriétés de confluence (l'ordre de remplacement des lemmes n'influe pas le résultat) et de normalisation forte (dérouler les lemmes va se terminer) permettent de comprendre le résultat de cette évaluation comme étant l'unique dénotation (la forme normale) de l'expression de départ (ici $f(a)$).

Cela permet de redonner un sens à ces expressions sans avoir à passer par des modèles : au lieu de référer à un possible individu hors de portée, leur dénotation devient accessible via un calcul. Typiquement, la propriété d'existence permet de calculer le témoin d'une preuve d'une formule existentielle. Pour donner un exemple, en définissant un type (une formule) $\mathbb{N}$ avec des termes (preuves) primitifs $0 : \mathbb{N}$ et $S : \mathbb{N} \Rightarrow \mathbb{N}$, on peut manipuler syntaxiquement les entiers naturels : les preuves de $\mathbb{N}$ sont les algorithmes représentant les entiers (évalués en $0, S(0), S(S(0)), ...$). Formaliser et reconnaître la partie calculatoire de la logique la connecte naturellement avec l'informatique, notamment via l'isomorphisme de Curry-Howard.

En s'intéressant aux preuves apportées à une question et non seulement à sa prouvabilité, on étend la logique en permettant de poser des questions qui attendent des réponses implicites : on passe de questions qui acceptent comme réponse "oui" (il existe une preuve de cet énoncé) ou "non" (il existe une preuve de la négation de l'énoncé) à une demande de produire un objet utilisable répondant à certaines exigences ou de produire une absurdité à partir d'un tel objet. La logique classique seule ne permet donc pas de répondre à "combien font $2 + 2$?" mais seulement par exemple à "est-ce que $2 + 2$ est égal à 4 ?" (si le domaine est constitué d'entiers) ou à "existe-t-il un entier en forme normale qui corresponde à l'évaluation de $2 + 2$" (si le domaine est constitué d'expressions à évaluer) et manque donc un pan essentiel de l'activité déductive. On peut remarquer que c'est essentiellement dû au fait que la seconde question n'empêche pas de répondre par l'absurde : "supposons qu'un tel entier n'existe pas ...".

Ce sont des questions très courantes, par exemple si un médecin sait soigner les maladies A et B, il peut traiter une personne apportant une preuve (disons un autotest) qu'elle est atteinte de A ou de B ... sous réserve que le médecin puisse observer le résultat de l'autotest indiquant quelle maladie fut détectée : il s'agit là du "ou" intuitionniste décrit précédemment (sinon, il devra lui appliquer les deux traitements). On peut aussi remarquer que répondre simplement "oui" à la question "tu as l'heure ?" est de mauvais goût.

Cette approche dénotationnelle qui égalise les preuves ayant la même dénotation ne peut pas bien fonctionner avec la négation définie précédemment : en effet, les réponses négatives $\neg A$ produisent une preuve de l'absurde, or celles-ci permettent de déduire n'importe quoi, notamment n'importe quelle autre preuve de l'absurde. On égalise donc toutes les preuves de $\neg A$, en particulier une preuve de $\neg\neg A$ n'a pas le contenu calculatoire d'une preuve de A.

La logique classique n'a pas de procédure (confluente) permettant d'identifier une preuve à sa dénotation, et n'a donc qu'une sémantique dénotationnelle triviale : toutes les preuves d'une même formule sont égalisées, comme les énoncés négatifs ci-dessus. C'est pour cela qu'elle peut être inclue dans la logique intuitionniste via une double négation. Une fois que les questions acceptant des réponses implicites sont prises en compte, la logique intuitionniste (à priori une restriction de la logique classique) est donc une extension de la logique classique : tant qu'on est seulement intéressé par la cohérence de A, prouver $\neg\neg A$ suffit (où l'on retrouve le raisonnement par l'absurde).

Une sémantique dénotationnelle des preuves se retrouve naturellement dans les catégories, où une preuve d'une formule $A \Rightarrow B$ est un morphisme de A vers B et la normalisation consiste à composer les morphismes (si il n'y en avait au plus qu'un de A vers B, on aurait pas besoin de calculer lequel : la logique classique dégénère une catégorie en ordre partiel). Il s'agit d'un domaine mature et fécond où de nombreuses extensions et correspondances existent comme celle entre géométrie, calcul et catégories ([NL1]), dont la théorie des types homotopiques est un bon exemple.

En isolant la partie calculatoire de la logique, l'approche dénotationnelle rend possible la vérification concrète de preuves mathématiques, ce qui témoigne d'une architecture plus fine. Auparavant, cette vérification ne valait qu'en principe ([FM1]) : les assistants de preuve de ZFC comme Isabelle ajoutent une notion de calcul externe, typiquement en permettant d'orienter les égalités prouvées pour les utiliser automatiquement comme des règles de réécriture.

De plus, l'exigence de dénotation rajoute un critère calculatoire pour justifier des axiomes, ce qui peut éclairer des concepts existants (comme les réels ici [NL2]). En effet, alors qu'une référence à un monde externe peut parler d'objets infinis (du moins de leur cohérence) sans problèmes, internaliser cet objet revient à décrire ses utilisations possibles et exige donc de ramener un infini actuel à un infini potentiel.

Or, il y a des limitations qui ne rendent pas cette approche tout à fait satisfaisante, autant d'un point de vue de la justification de la logique que de sa portée :

Premièrement, abandonner les modèles (au niveau logique, pas mathématique) nous fait perdre la complétude, c'est-à-dire la précieuse dualité entre preuves et contre-modèles qui n'a pas vraiment de remplaçant à ce niveau.

Deuxièmement, les règles logiques restent primitives. Il existe pourtant plusieurs règles à la fois justifiées (de manière calculatoire), admissibles et incompatibles : comme au niveau précédent, cela témoigne de l'enfermement dans un système parmi d'autres. On peut notamment le remarquer au sein des différentes écoles intuitionnistes qui acceptent (ou refusent) chacune moult règles supplémentaires spécifiques.

Troisièmement, comment s'assure-t-on du comportement des preuves ? Il nécessite une autre preuve et on sent vite le problème arriver : certains proposent une méta-preuve (et le même problème s'applique à elle) tandis que d'autres comme Kreisel proposait un système donné en avance dans lequel se déroulerait cette méta-preuve. Autrement dit, une nouvelle impasse par diagonalisation.

Enfin, la symétrie dont bénéficiait la logique classique est perdue, en même temps que le contenu des preuves d'énoncés négatifs. On voit pourtant passer de belles constructions lors d'une preuve par l'absurde ... Cela suggère une sémantique opérationnelle, où l'on intéresse au déroulement de l'explicitation d'une preuve. Cependant, cette fois l'inspiration ne peut pas venir directement de Curry-Howard : il faudrait justifier pourquoi employer une stratégie d'évaluation plutôt qu'une autre, c'est-à-dire un certain ordre de déroulement des lemmes, ce qui semble purement dogmatique.

Si l'on revient à la grille de lecture de départ, ce niveau illustre la case des réponses implicites, auparavant invisibilisées (donc inutilisables). Il révèle également l'usage en permettant de décrire le comportement d'une preuve : cela correspond au satisfaisant typage à la Curry (partant des règles d'élimination). Cependant, l'usine reste absente : la vérification d'une preuve reste axiomatique, elle ne fait que vérifier que l'on a bien suivi les règles. Il s'agit ici du typage à la Church, qui nous fournit une confiance en limitant les preuves acceptées (où les règles d'introduction viennent en premier) mais nous enferme dans un nouveau système.

Puisque nous avons une vision plus fine de ce qui constitue une preuve, il faudra nous intéresser à la case usine, où les propriétés qui nous intéressent sont garanties par la procédure de vérification au lieu de simplement suivre des règles données. Autrement dit, il faudra établir de véritables tests pour une preuve.

-3. Niveau interactif : tests & jeux

Les règles logiques semblent chercher à garantir des propriétés sur les preuves construites. Par exemple, si la confluence n'est pas présente en logique classique, la normalisation forte reste obligatoire : elle correspond à une définition bien fondée (dérouler les lemmes se terminera forcément). La non-terminaison de l'explicitation d'une expression est fatale car cela indique qu'elle ne peut pas être définie. Cependant, prédire l'explicitation ne peut pas être adressé en toute généralité, comme le montre le problème de l'arrêt.

On peut détecter ce problème avec le théorème du point fixe de Lawvere, qui généralise le théorème de Cantor dans les catégories. La notion de cardinalité y est remplacée par l'existence d'un point fixe : au lieu de montrer que les ensembles X et 2^X n'ont pas la même cardinalité, ce théorème montre qu'avoir un représentant dans X pour chaque élément de 2^X implique un point fixe pour toutes les fonctions $2 \to 2$ (ce qui casse la négation). On retrouve alors sous une forme procédurale la contradiction que l'on retrouve par exemple dans le paradoxe de Russell, où la valeur de vérité d'une propriété ne peut pas être explicitée (donc définie) : $x = non(x) \rightsquigarrow x = non(non(x)) \rightsquigarrow ....$ On peut retrouver de nombreuses applications de ce théorème dans [Law1], notamment à des paradoxes.

Une apparition de la notion de test est soulignée par Girard chez Herbrand ([Gir3, p18]), lorsqu'il souhaitait établir la différence entre $A := \forall x \exists y P[x, y]$ et $B := \exists y \forall x P[x, y]$, dont les preuves contiennent un témoin $y = t[x]$ tel que $P[x, t[x]]$. Alors que Gentzen interdit simplement de dériver B de A lorsque $t[x]$ dépend de x via les règles du calcul des séquents, Herbrand établit un test qui montre l'impossibilité de la dépendance pour B lors de la recherche d'une contradiction entre une preuve supposée $P[u, t[u]]$ de B et une autre $\neg P[f(v), v]$ de $\neg B$. Cela revient à résoudre les équations $u = f(v)$ et $v = t[u]$, soit $u = f(t[u])$, qui diverge lorsque t dépend de u.

L'usine ramène donc l'aspect performatif, à priori imprédictible de l'usage des preuves à des constats en utilisant des cas suffisamment représentatifs des témoins potentiels des preuves.

Le test d'Herbrand est satisfaisant pour un cas particulier, mais comment intégrer la notion de test à la logique ? Comme présenté dans [Gir1, p8], on peut s'intéresser à ce qu'est un test de manière générale, dans le même esprit que BHK :

— Un test de $A \vee B$ est une paire d'un test de A et d'un test de B.
— Un test de $A \wedge B$ est soit un test de A, soit un test de B. En particulier pour $A \wedge A$, les tests différencient les deux preuves de A.
— Un test de $f : A \Rightarrow B$ est une paire d'une preuve $p : A$ et d'un test de $f(p) : B$.

On obtient $test(A \wedge B) = test(A) \vee test(B)$, $test(A \Rightarrow B) = A \wedge test(B)$, ... Ce qui suggère une identification des tests de A aux preuves de $\neg A$. Cependant, puisque des tests de A et $\neg A$ peuvent cohabiter mais pas des preuves contradictoires, il faut comprendre le statut de ces "pré-preuves". La sémantique des jeux offre une excellente métaphore :

— La négation correspond à l'échange des joueurs.
— Une pré-preuve de A (ou test de $\neg A$) est une stratégie visant à prouver A.
— Une preuve de A est une stratégie gagnante face à toutes les stratégies pour $\neg A$ (tests de A).
— Une stratégie de $A \otimes B$ est une stratégie pour gagner A et B en parallèle.
— ...

On peut rapprocher le rôle d'un test à celui d'un contre-modèle : une preuve doit se conformer au test ou au modèle donné sous peine d'être réfutée. Un résultat de complétude (interne cette fois) permet alors ici d'avoir une dualité entre tests et preuves : pour tout A, il existe soit une preuve de A, soit pour toute pré-preuve de A un test l'invalidant.

Pour cela, il faudrait une négation à la fois involutive et constructive, ce qui écarte la logique classique et la logique intuitionniste. La solution se trouve dans la restriction des contractions (qui autorisent la réutilisation des preuves) : on aboutit alors à la logique linéaire.

Les involutions constructives sont souvent des opérations centrales, comme l'opposée d'une catégorie, la symétrie valeur/continuation en informatique, l'échange des joueurs en sémantique des jeux ou la dualité donner/recevoir en focalisation.

En logique linéaire, les pré-preuves correspondent aux structures de preuves, les preuves aux réseaux de preuves (des structures de preuve connexes et acycliques) et les tests d'une structure de preuve de A aux "switch" possibles de celle-ci, qui correspondent à des structures de preuve de $\neg A$ ([Gir2, p226]). La cyclicité est une manifestation syntaxique directe de l'impossibilité de l'explicitation d'une expression, dû à des dépendances similaires à celles révélées par le test de Herbrand.

Sans la règle de contraction, l'infini potentiel est écarté et on se retrouve dans un monde d'actions. Cela correspond bien à la notion de tests (qui sont appliqués une seule fois) ou de jeux (dans une partie où on ne revient pas en arrière) et permet une analyse sans problèmes de ce fragment de la logique linéaire. Avec les contractions, on réintroduit l'infini (avec ses incertitudes, traduites par des problèmes de prédictions) mais aussi des notions centrales comme celle de vérité pérenne $!A$.

Ce niveau est donc satisfaisant pour adresser des limitations apparues au niveau précédent : une symétrie est regagnée avec la complétude sans faire appel aux modèles, tout en conservant le contenu des (pré-)preuves (c'est-à-dire en restant constructif). De plus, les règles logiques sont justifiées par des batteries de tests, et l'incertitude logique est internalisée : elle apparaît lorsqu'une infinité de tests est requise.

De plus, le problème par diagonalisation est enfin adressé : chercher le modèle d'un modèle ou la preuve d'une preuve mène à une régression infinie, mais les tests d'un test de A sont internalisés par la négation : ce sont des (pré-)preuves de A ! C'est ce qui justifie l'appellation "dualité moniste" de Girard entre tests et preuves.

La logique linéaire est fructueuse en informatique, notamment avec la prise en compte des ressources, de la focalisation ou des types sessions (un terme de ce type correspond à la possibilité de communiquer, par exemple entre un client et un serveur). La notion de ressources est également importante en physique quantique avec le théorème de non-clonage, où l'on retrouve des formalismes basés sur la logique linéaire (comme les catégories monoïdales). Enfin, des logiques linéaires légères permettent de capturer des classes de complexité via la gestion des exponentielles.

Cependant, il reste la problématique récurrente du choix d'un système : on a encore le risque de s'enfermer dans un jeu parmi d'autres, avec des règles fixées.

-4. Niveau déontique : la négation comme format

Comment sortir des systèmes ? Une piste se trouve dans une lecture que fait Girard de la propriété de sous-formule ([Gir3, p17]) : tout ce qui est donné dans une question (c'est-à-dire une formule avec les définitions de ses composants) suffit à y répondre, sans faire appel à un système externe à la formule.

Cela rejoint des remarques antérieures de Kreisel sur le fait que la différence entre logique classique et intuitionniste n'est pas dûe à des systèmes différents mais à l'utilisation d'une connective différente pour "$\vee$", ce qui est clarifié en logique linéaire (OU classique "$\parr$" versus OU intuitionniste "$\oplus$").

L'ingrédient majeur pour y parvenir provient de la correspondance d'une règle avec une batterie de tests sur des objets plus généraux que les preuves. C'est un point central de la syntaxe transcendantale, qui permet de définir des concepts logiques via des batteries de tests sur des objets syntaxiques quelconques (tests qui sont également des objets quelconques).

La partie analytique de la logique correspond à la syntaxe pure, c'est-à-dire à tout ce qui peut être dit ou fait. Elle peut être structurée en choisissant un modèle de calcul particulier : cela n'empêche pas de considérer tout objet syntaxique, car il permettra d'encoder au besoin un autre modèle (une machine de Turing, le lambda-calcul, ...). Celui choisi ici est la résolution stellaire, qui facilite la mise en place de tests symétriques et la reconstruction de la logique linéaire. Elle manipule des constellations : des ensembles finis d'étoiles, elle-mêmes des ensembles finis de rayons, c'est-à-dire de termes partageant les mêmes variables et éventuellement associés à une couleur.

Une constellation est normalisée lorsqu'elle n'est plus colorée. Pour cela, on génère tous les diagrammes possibles par taille croissante, c'est-à-dire les constellations de N+1 étoiles (possiblement plusieurs fois les mêmes) provenant de la constellation de départ et connectées N fois par des rayons ayant des couleurs complémentaires. Les étoiles connectées sont fusionnées deux à deux par unification de leurs variables. Une reconstruction d'automates non-déterministes et de circuits booléens est détaillée dans [Eng1] (qui illustre avec d'autres exemples les notions de typage ci-dessous).

Un premier choix pour reconstruire la logique en partant du calcul est de décider ce qu'est une interaction réussie entre deux constellations orthogonales A et B (un test réussi, noté $A \perp B$). La normalisation de l'évaluation est un choix naturel, mais le modèle de calcul utilisé peut suggérer d'autres choix, comme ici en exigeant une unique forme normale. La normalisation correspond ici à un nombre fini de formes normales, et permet ensuite de capturer la logique linéaire multiplicative avec la règle MIX. L'ensemble des constellations interagissant correctement avec A est noté $A^{\perp}$.

Il y a ensuite deux manières de typer les objets, ce qui correspond à la partie synthétique de la logique :

L'usage fournit un typage "existentialiste" (à la Curry) qui décrit comment interagissent des objets avec des techniques proches de la réalisabilité en considérant des comportements (formules), c'est-à-dire des ensembles de constellations A tels que $A = A^{\perp\perp}$. Cette condition signifie que l'ensemble est caractérisé par les tests associés, il est clos par interactions. Cela permet par exemple de définir $A \otimes B$ en fonction des comportements A et B. Cependant, ce typage empêche de pouvoir reconnaître de manière effective ses habitants (il faudrait pouvoir observer toutes les interactions possibles d'un objet).

L'usine fournit un typage "néo-essentialiste" (à la Church), le préfixe "néo-" venant du fait qu'il n'y a plus de règles ou sémantiques postulées, seulement un choix de tests. Il permet d'associer un objet à un comportment A en se basant sur les tests $A^{\perp}$. Cette batterie de tests (généralement infinie) peut alors souvent être ramenée à des tests suffisants, non nécessaires pour conserver la finitude des tests à passer. C'est notamment ce qu'il se passe avec le critère de correction de la logique linéaire.

Ces deux notions de typages sont exprimables de manière finie et interne. Seule leur adéquation (prouvant les prédictions de l'usine quand à l'usage, ce qui correspond à l'élimination des coupures) est reléguée à un système externe, telle que la théorie des ensembles.

Les preuves sont alors constituées de deux parts : le véhicule, contenant sa partie objective, calculatoire (correspondant aux liens axiomes en logique linéaire) et le gabarit, établissant les tests du comportement associé.

Les opérateurs logiques définis de cette manière sont rangés dans le 1^{er} ordre de Girard qui définit une partie complètement sûre et désystématisée de la logique alors "moralement apodictique" : chacun peut l'étendre avec ses propres définitions, qui seront compatibles (et interchangeables en cas de définitions équivalentes, à la manière du typage structurel en informatique).

Ce 1^{er} ordre est plus restreint que le 1^{er} ordre usuel et n'inclue notamment pas le calcul des prédicats, bien qu'il continue d'incorporer des éléments (peut-être les connecteurs additifs comme présentés dans [Gir6] à 27m11s). C'est dû à la reconnaissance du 2^{d} ordre parfois implicite dans le 1^{er} ordre usuel, notamment via des schémas d'axiomes ou des constantes de prédicat : Girard donne comme exemple le théorème $\forall x.P(x) \Rightarrow P(x)$, implicitement $\forall P \forall x.P(x) \Rightarrow P(x)$. La négation n'est donc pas internalisée, car elle révèlerait sinon un quantificateur existentiel qui ne peut pas être implicite : celui-ci se traduit de manière externe par l'existence d'un contre-modèle ([Gir4]).

La complétude s'appliquant aux énoncés Π^1, le changement opérant au 2^d ordre arrive avec la quantification existentielle. Les preuves d'existence ($\exists X.A$) doivent embarquer des tests orthogonaux pour les occurrences de X et $X^\perp$. Cette troisième partie de la preuve (appelée le "moule") change son statut : ce qui est testé devient en partie synthétique car les tests embarqués doivent se tester eux-mêmes (ce qui implique un conflit d'intérêt).

Le 2^d ordre fait donc gagner en expressivité au prix d'une certitude qui n'est plus absolue (qualifiée d'épidictique) et de l'acceptation d'un système provenant du choix nécessaire d'un moule pour les témoins existentiels. Il permet notamment de capturer l'arithmétique de manière logique, où les individus sont les propositions logiques et leur égalité l'équivalence linéaire ([Gir5]).

Conclusion

L'exigence de justifications internes a donc permit d'étendre la logique : en s'intéressant au comportement des preuves et non seulement à la prouvabilité, nous pouvons formuler des questions logiques générales qui utilisent les objets logiques. De plus, cela isole l'incertitude logique au niveau de l'adéquation entre l'acceptation et l'usage des preuves. Enfin, la notion de test permet de justifier des règles en considérant des objets plus généraux que les preuves.

La syntaxe transcendantale est un projet internalisant ces observations pour aboutir à une boîte à outils ouverte : elle permet de définir des concepts logiques automatiquement compatibles avec des objets quelconques. Cette approche tranche avec les systèmes axiomatiques n'offrant que des logiques contingentes car invérifiables. Le choix nécessaire d'un format vient du 2^d ordre où les réponses doivent embarquer un système permettant de les comprendre.

Références

[Abr1] On Transcendental syntax : a Kantian program for logic ? - V. M. Abrusci, P. Pistone, 2014 `https://www.academia.edu/10495057/On_Trascendental_syntax_a_Kantian_program_for_logic`
[Eng1] A gentle introduction to Girard's Transcendental syntax (v7) - Boris Eng, 2022 : `https://arxiv.org/pdf/2012.04752.pdf`
[FM1] MULTIPLICATION 101 - But what even IS 4*8 - Flammable Maths, 2019 : `https://www.youtube.com/watch?v=Ynvydo_fKWo`
[Gir1] On the meaning of logical rules I : syntax vs. semantics - J.Y. Girard, 1998 : `https://girard.perso.math.cnrs.fr/meaning1.pdf`
[Gir2] Le fantôme de la transparence - J.Y. Girard, 2016 : `https://www.editions-allia.com/fr/livre/749/le-fantome-de-la-transparence`
[Gir3] Un tract anti-système - J.Y. Girard, 2020 : `https://girard.perso.math.cnrs.fr/systeme.pdf`
[Gir4] Transcendental syntax III : equality - J.Y. Girard, 2022 : `https://girard.perso.math.cnrs.fr/trsy3.pdf`
[Gir5] Transcendental syntax IV : logic without systems - J.Y. Girard, 2022 : `https://girard.perso.math.cnrs.fr/trsy4.pdf`
[Gir6] Deux ou trois choses que je sais d'elle : la logique - J.Y. Girard, 2022 : `https://www.youtube.com/watch?v=KoPkIChIs54`
[Law1] A Universal Approach to Self-Referential Paradoxes, Incompleteness and Fixed Points - Noson S. Yanofsky, 2003 : `https://arxiv.org/pdf/math/0305282.pdf`
[NL1] `https://ncatlab.org/nlab/show/computational%20trilogy`
[NL2] `https://ncatlab.org/nlab/show/Type+Two+Theory+of+Effectivity`