(2015) B.I.A.A. no **101** BULLETIN D'INFORMATIQUE APPROFONDIE ET APPLICATIONS
ISSN 0291-5413 (Université d'Aix-Marseille)

# Contradiction ou indécidabilité : il faut choisir !

Patrick ISOARDI[1]

**Résumé. –** À propos du théorème d'incomplétude de Gödel.

## 1. Formaliser les mathématiques : le rêve de Hilbert

"Je suis un menteur" .

L'individu qui considère que dans le monde qui l'entoure tout ne peut être que vrai ou faux, ne peut pas statuer sur la vérité de cette phrase. En effet :

- s'il considère que la phrase "Je suis un menteur" est vraie, alors celui qui l'énonce est un menteur car c'est ce qu'affirme la phrase. Les menteurs ne prêchant que le faux, cette phrase est fausse ; il y a contradiction avec l'hypothèse de départ.

- s'il considère que la phrase "Je suis un menteur" est fausse, celui qui l'énonce fait partie des menteurs car ce qu'il dit est faux. Mais disant qu'il est un menteur, c'est donc vrai. Cela contredit l'hypothèse sur les menteurs.

La phrase du menteur semble ne pouvoir être ni vraie ni fausse. C'est un paradoxe.

Le langage naturel autorise ce genre de discours, mais le langage rigoureux des Mathématiques ne devrait permettre d'exprimer que des vérités non contradictoires. Certains mathématiciens et logiciens du début du XX siècle, notamment Russel, Whitehead ou Hilbert ont durement travaillé pour ne plus employer le langage naturel dans les mathématiques. Ils voulaient les formaliser en proposant un système de notation rigoureux permettant d'écrire sans ambiguïté les idées élémentaires et des méthodes de raisonnement qui à partir de ces idées élémentaires démontreront exactement et complètement tous les théorèmes de Mathématiques que le système ainsi formalisé peut produire.

Il s'agissait là d'un objectif similaire à celui de Leibniz qui quelques temps plus tôt voulait réduire le raisonnement à du calcul et les paradoxes à des erreurs de syntaxe : " les chimères ne pourront être écrites en ces caractères ! " C'est en fait cet objectif qui s'est révélé chimérique lorsqu'**en 1931 Gödel démontra que si un système formalisé contient les opérations arithmétiques élémentaires et qu'on le veut non contradictoire alors il existe des vérités qu'il ne peut pas démontrer.**

Ainsi le paradoxe ne réside pas dans le langage utilisé pour écrire la phrase du menteur, mais dans l'incapacité de déterminer par le raisonnement si elle est vraie ou fausse.

L'étude qui suit précise l'environnement dans lequel Gödel a travaillé. Elle devrait permettre d'appréhender les idées essentielles de la démonstration de son théorème d'incomplétude.

1. Ndr : d'après le numéro 73 du bulletin de 2006.

1.1. **Système formel.** C'est le cadre dans lequel Hilbert espérait que les mathématiques pourraient être produites, en partant des idées élémentaires sur lesquelles on appliquerait de façon mécanique des règles définissant les processus mentaux du raisonnement.

Un système formel se définit par la construction suivante :

- un ensemble fini de symboles ; c'est le vocabulaire avec lequel on peut écrire des mots et des phrases,
- des règles de formation avec lesquels on écrit des *formules* qui sont des phrases syntaxiquement correctes ; c'est la grammaire du système,
- un sous ensemble de formules : les *axiomes*,
- on définit une liste finie de relations entre les formules : les *règles d'inférence*.

Les deux premiers points définissent un langage et les deux derniers la notion de démonstration.

Une *démonstration* est une suite finie de formules dont chacune est soit un axiome soit une conséquence immédiate de l'application d'une règle d'inférence sur la formule précédente.

Un *théorème* du système est une formule dont il existe une démonstration.

A partir des axiomes le système ainsi formalisé produira mécaniquement et complètement tous les théorèmes [2].

1.2. **L'indécidabilité : un grain de sable dans les systèmes mécaniques.** Programmons un ordinateur pour qu'il applique successivement toutes les règles d'inférences à tous les axiomes et aux théorèmes générés. Notons au préalable que ce processus peut être infini. Imaginons à présent que nous voulions savoir si une certaine formule F est un théorème. Quatre cas peuvent se présenter à l'ordinateur :

1- Il liste F et pas non F $\Rightarrow$ La formule F est un théorème.

2- Il liste non F et pas F $\Rightarrow$ La formule non F est un théorème.

3- Il liste F et non F $\Rightarrow$ Le système est *contradictoire* ou inconsistant, c'est à dire qu'il démontre à la fois une chose et son contraire. Ce que Hilbert voulait éviter pour les mathématiques.

4- Il ne liste ni F ni non F :

- Si l'ordinateur s'arrête , on dit que la formule F est *indécidable* dans le sens où il est impossible de démontrer formellement F ou nonF par une démonstration déductive des axiomes. F indécidable $\Leftrightarrow$ F et nonF ne sont pas démontrables. On substitue ici la propriété : " *est démontrable* " à celle de " est un théorème " ; cela est dû à l'apparition de l'indécidabilité que l'on n'imaginait pas avec un " bon " système formel qui devrait toujours se trouver dans un des deux premiers cas.

2. Un tel système formel est déductif et mécanique au sens où il n'est pas nécessaire de faire appel à l'intelligence. On ne sait pas forcément de quoi on parle ni si ce qu'on dit est vrai. Ici la question n'est pas de savoir si une formule F est vraie mais si elle est bien la conséquence de l'application de règles d'inférences sur les axiomes ; dans ce cas F est un théorème et la suite des formules obtenues par l'application précédente est sa démonstration. Lorsqu'il n'y a pas de vérité intuitive le système formel est souvent qualifié de syntaxique ou encore typographique.

Bien que F soit indécidable, si elle est vraie (au sens où F détermine une propriété que l'on peut vérifier par ailleurs), nous dirons que le système est *incomplet* . Un système formel complet permettrait de démontrer tout ce qui est vrai. C'est ce que Hilbert recherchait.

- Si l'ordinateur ne s'arrête pas, on ne peut pas dire que F est indécidable mais que l'ordinateur n'a pas encore trouvé une démonstration pour F ou nonF, son travail n'étant pas encore terminé. Par exemple il n'est toujours pas démontré que tout nombre pair est la somme de 2 nombres premiers ; pourtant cela semble vrai aussi loin qu'on est allé avec un ordinateur. C'est une conjecture, celle de Goldbach : une formule (qui semble) vraie qui n'est pas (encore) démontrée.

Après cette analyse quelques questions se posent , notamment :

- Comment faire la distinction entre une conjecture et une formule qui n'est pas démontrable ?

- Comment peut-on savoir qu'une formule est vraie si on ne peut pas la démontrer ?

1.3. **Système mécanique et vérité.** La question précédente revient en fait sur la comparaison entre l'homme et la machine. Plus précisement : si un humain est capable de savoir que la phrase qu'il donne à un ordinateur est vraie, pourquoi l'ordinateur ne serait-il pas capable de découvrir la vérité ?

Voici la phrase : "L'ordinateur ne répondra jamais vrai à cette phrase". Que fait l'ordinateur ?

- S'il répond vrai, il affirme que "L'ordinateur ne répondra jamais vrai à cette phrase". Or ce n'est pas le cas, puisqu'il vient justement de répondre vrai à la phrase. Si l'ordinateur ne se contredit pas, il ne peut donc pas répondre vrai.

- S'il répond faux, il affirme que "L'ordinateur ne répondra jamais vrai à cette phrase" est une affirmation fausse. Or l'affirmation n'est pas fausse puisque l'ordinateur vient justement de ne pas répondre vrai. Si l'ordinateur ne se contredit pas, il ne peut donc pas répondre faux. L'ordinateur ne peut répondre ni vrai ni faux à la question. Elle est pour lui indécidable.

Et nous, pouvons nous répondre à la question ?... La phrase dit : "L'ordinateur ne répondra jamais vrai à cette phrase". Nous venons de voir qu'en effet, l'ordinateur ne peut pas répondre vrai. Cette phrase est donc une vérité que l'on vient de vérifier " par ailleurs " en dehors de l'ordinateur et que celui-ci ne pourra jamais découvrir [3]. Ce n'est pas une conjecture, mais bien une certitude ; une vérité qu'un système mécanique ne pourra jamais démontrer. Gödel l'exprimera par une formule arithmétique dans un système formel, rendant celui-ci incomplet.

1.4. **Mécanique, intelligence.** Cette différence entre " l'homme qui sait " et " la machine qui ne peut pas trouver la vérité " a été mise en évidence dans un système formel [4] par

3. Observons toutefois que l'indécidabilité de l'ordinateur est là justement parce que la phrase soulève un paradoxe. Si un système mécanique arrivait à donner une valeur de vérité à un paradoxe, cela serait inquiétant ! En cette circonstance, l'ordinateur réagit exactement de la manière attendue : quand on lui demande de démontrer un paradoxe, il n'y arrive pas, et c'est heureux !

4. Symboles : M, I, U. Axiomes : MI. Règles : (1) xI → xIU (2) Mx→ Mxx (3) xIII → xU (4) xUU → x. La lettre "x" ne sert qu'à remplacer n'importe quelle chaîne.

Douglas Hofstadter dans son livre : "Gödel Escher Bach" (page 38) où il pose la question : La formule MU est-elle un théorème du système ?

Jouons un peu à l'intérieur de ce système formel. Par exemple l'application de la règle 1 sur l'axiome MI produit MIU alors que les règles 3 et 4 ne peuvent s'appliquer sur cet axiome. La démonstration produite par l'application sur l'axiome MI de la suite des règles 223 génère MI MII MIIII MIU.

Après quelques amusements de cette nature, la question initiale revient : ce mode mécanique de dérivation produira t-il MU un jour ?

Cette énigme n'est pourtant qu'un problème d'arithmétique. Si on se place à l'extérieur du système l'observation des règles et de l'axiome MI montre que le nombre de I ne peut pas être un multiple de 3 et la règle 3 ne produira donc jamais xU. MU n'est donc pas un théorème du système formel, mais celui-ci ne pourra le démontrer alors qu'un simple raisonnement arithmétique donne la solution.

Ainsi les deux exemples précédents montrent simplement qu'un système mécanique ne peut pas toujours étudier de l'intérieur avec ses propres règles, la vérité d'une phrase ou la théorèmeté d'une formule. En revanche, on peut le faire de l'extérieur par des raisonnements métamathématiques.

Une nouvelle question se pose : si on inclut le raisonnement (ici l'arithmétique) à l'intérieur du système formel, peut-on alors déterminer la nature théorèmatique d'une formule ? Gödel montrera que la réponse reste négative ; le raisonnement ne permet pas toujours de déterminer la vérité.

1.5. **Vérité et démonstration.** Essayons d'être plus précis sur la relation entre vérité et démontrabilité ou théorèmeté.

La phrase : "x est pair" exprime la propriété arithmétique de parité que le sujet x peut avoir. On ne sait pas si cette propriété est vraie ou fausse, mais lorsqu'on remplace x par 5 par exemple, on obtient la phrase : "5 est pair" qui est fausse. Selon le même procédé, "14 est pair" est une phrase vraie. Ainsi, lorsqu'on remplace x par un entier, on pourra diviser celui-ci par 2 et comparer le reste de cette division à 0 pour démontrer ou réfuter l'affirmation. La suite des opérations de division et de comparaison constitue un procédé mécanique de décision (algorithme) qui donne toujours au bout d'un temps fini, un résultat vrai (si le reste est 0) ou faux (si le reste est 1). Toutefois on observera que le résultat n'est pas un théorème, mais une valeur de vérité.

On retrouve les deux manières classiques de résoudre les problèmes :

- la méthode algorithmique remise en vigueur avec l'essor de l'informatique, qui prescrit une suite d'opérations arithmétiques à effectuer, terminée par un test qui répond au bout d'un temps fini par vrai ou faux lorsque la propriété est décidable.

- la méthode déductive des systèmes formels où il s'agit de combiner un ensemble d'axiomes au moyen d'inférences logiques pour démontrer des théorèmes (qui sont alors considérés comme des vérités).

Quelle relation unit les 2 méthodes ?

Si $P(x)$ signifie "x à la propriété P" et que $\forall n$ il existe un algorithme qui détermine au bout d'un temps fini la valeur de vérité de la formule $P(n) \Rightarrow$ la propriété $P(x)$ est dite décidable. De plus :

| si P(x) est décidable et que P(n) est vraie $\Rightarrow$ P(n) est démontrable dans un système formel. (1) |
|---|

Dans le cas où P(n) est faux $\Rightarrow$ P(n) est réfutable. Mais la réfutation n'est pas naturelle dans les systèmes formels.

Sautons dans un nouveau niveau d'abstraction où le problème de l'infini va devenir de plus en plus pressant mais dont la réflexion nous aidera par la suite.

Considérons la formule existentielle : "$\exists$x P(x)" et la formule universelle : "$\forall$ P(x)" dans lesquelles P(x) est une propriété décidable. Nous remarquerons que P(x) représente une infinité de formules alors que les 2 autres sont chacune d'elles une simple formule contenant une variable et qui sera vraie ou fausse.

Par exemple si P(x) : "x est pair" alors "$\exists$x P(x)" est vraie car P(2) est vraie et "$\forall$x P(x)" est fausse car P(1) est faux. Plus généralement :

- si on est sûr de trouver un n tel que P(n) est vrai alors "$\exists$x P(x)" est vrai et un système formel trouvera dans la suite P(0), P(1), P(2), ..... le théorème P(n). Ce qui démontre la formule "$\exists$x P(x)".

$$\boxed{\text{“}\exists x\ P(x)\text{” vrai} \Rightarrow \text{“}\exists x\ P(x)\text{” démontrable. (2)}}$$

Mais si on ne sait pas que ce n existe, tant qu'on ne l'a pas trouvé P(0), P(1), P(2),.... sont faux et "$\exists$x P(x)" l'est aussi et peut le rester indéfiniment, sans pouvoir le démontrer voire l'affirmer car peut-être qu'un jour on trouvera ce n.

- de la même façon, si on est sûr de trouver un n tel que P(n) est fausse alors "$\forall$x P(x)" est fausse. Tant qu'on n'a pas trouvé ce n, P(0) , P(1) , .... sont vrais ; "$\forall$ P(x)" reste vraie sans pouvoir le démontrer car la recherche peut être infinie.

Ainsi : "$\forall$x P(x)" est réfutable mais si elle est vraie, l'étude cas par cas ne peut pas le démontrer de manière mécanique. En effet si "$\forall$x P(x)" est vraie alors P(0), P(1), P(2), ..... sont des théorèmes mais leur obtention est sans fin. Ce serait le cas de la conjecture de Goldbach qui pourrait être vraie, mais qui en même temps ne peut être démontrée à partir des axiomes de l'arithmétique.

Il est difficile d'énoncer d'autres relations entre vrai et démontrable, mais celles-ci ont permis à Gödel d'écrire dans un système formel non contradictoire contenant les opérations arithmétiques élémentaires une formule vraie qui ne sera jamais démontrable rendant ce système incomplet.

1.6. **Métalangage et arithmétique.** Revenons à la théorèmeté ou démontrabilité de la formule F. Comme nous l'avons vu précédemment c'est une propriété universelle manifestement extérieure au système formel. L'étude des propriétés de ce type se fait habituellement au moyen d'un métalangage différent de celui du système formel étudié. Généralement, il faut créer un second système formel qui traite du premier. L'inquiétude est de continuer cette construction à l'infini. Par contre si la phrase " F est démontrable " est elle-même une formule du système formel initial, celui-ci contient des formules ayant pour sujet d'autres formules du système ; il contient donc son métalangage et il peut s'introspecter notamment au sujet de la nature théorèmatique de ses formules. Existe t-il un tel système formel capable de s'introspecter ?

Vérifier que F est démontrable dans un système formel revient à chercher s'il existe au moins une démonstration dont la suite de formules constituée par l'application de règles d'inférences sur un axiome ou un théorème se termine par la formule F. Tant qu'on ne l'a pas trouvé, il faut continuer car rien ne nous dit qu'il n'existe pas et si tel est le cas dans ce mode mécanique de recherche sans fin, nous ne saurons jamais que F n'est pas démontrable. Par contre si on se donne la formule F et une suite finie D de formules, on peut toujours facilement vérifier en un nombre fini d'étapes si D est une démonstration de F. Appelons DEM(F,D) la phrase : "D est une démonstration de F. Comme pour la propriété de parité, quand on connaît F et D, un algorithme peut fournir en un temps fini le résultat vrai, faux. La décision se fait par des décalages sur les formules de D et des comparaisons pour vérifier que les règles d'inférences utilisées sont bien celles du système formel. Décalages et comparaisons sont des opérations arithmétiques.

Gödel montre que l'arithmétique est un métalangage pour l'étude des systèmes formels. Cela évite la régression infinie des métalangages où les symboles utilisés par le langage et le métalangage ne sont pas les mêmes.

1.7. **Arithmétique formelle.** L'arithmétique s'intéresse aux opérations (+, x) sur les entiers naturels et à leurs propriétés. Pour que l'arithmétique soit un métalangage pour l'étude des systèmes formels, il faut formaliser les opérations arithmétiques. Ces opérations sont des fonctions qui combinent des valeurs de N et donnent un résultat dans N ; par exemple z = f(x,y) . Ici z n'est pas une valeur de vérité mais un entier, or une formule définit une relation entre deux ou plusieurs objets et cette relation se voit attribuée une valeur de vérité. Comment passer de la fonction arithmétique f à la relation ? En remplaçant x,y,z par un triplet d'entiers k,m,n on pourra toujours vérifier si l'égalité n = f(k,m) est vraie ou fausse. Ainsi :

| |
|---|
| toute fonction arithmétique f à 2 variables peut être *exprimée* dans un système formel par une formule F(x,y,z) qui définit une propriété décidable du nombre z = f(x,y). (3) |

Dans le même souci de formalisation, un nombre dont les entiers font partie est un concept dans lequel nous devons distinguer sa valeur n qui intervient dans les calculs arithmétiques de son écriture formelle, celle qui interviendra dans les formules que nous noterons **n** [5]. Il existe plusieurs écritures formelles d'un même nombre : 10 en décimal , X en romain, 3130 en ASCII , ssssssssss0 généralement utilisée dans un système formel [6].

Ainsi formalisée, l'arithmétique peut être incluse dans un système formel.

5. si le gras n'apparaît pas, le lecteur saura faire la différence entre l'écriture qui apparaît dans les formules et la valeur utilisée dans les calculs.

6. L'ensemble des entiers naturels N est un ensemble ordonné infini, qui possède cette propriété remarquable que ses éléments peuvent être obtenus en se donnant le premier 0 et la fonction s (successeur) qui à tout entier associe le suivant : s(0)=1, s(1)=ss(0)=2, s(2)=sss(0)=3,. . .. Plus généralement un entier quelconque n s'écrira s. . .(n fois s). . ..s0

## 2. Le Théorème d'incomplétude de Gödel

Si un système formel contenant les opérations arithmétiques élémentaires n'est pas contradictoire alors il est incomplet.

Disons au préalable que l'astuce de Gödel consiste à faire en sorte que tout " raisonnement " à l'intérieur du système formel se ramène à des calculs arithmétiques. C'est pour cette raison que le système évoqué doit nécessairement formaliser les opérations arithmétiques de base.

Pour appréhender l'ensemble de la démarche, énumérons les étapes de la démonstration :

1 - Gödel montre comment construire dans son système une formule arithmétique G qui exprime la phrase métamathématique : " G n'est pas démontrable ".

2 - Il montre alors que G est démontrable si sa négation non G est démontrable. Si le système produit G, il produit nonG et réciproquement ⇒ il est contradictoire.

3 - Or si comme il se doit l'arithmétique n'est pas contradictoire, par négation de l'implication précédente G et non G ne sont pas des théorèmes ⇒ G est indécidable.

4 - Gödel montre que pourtant G est vraie ; vraie au sens ou G dit qu'un entier possède une certaine propriété arithmétique que l'on peut définir et vérifier par ailleurs.

5 - Puisque la formule G est vraie mais indécidable, le système est incomplet.

A travers l'arithmétique c'est bien évidemment toutes les Mathématiques qu'il devenait impossible de formaliser sans contradiction. Nous voilà maintenant en mesure d'approfondir les grandes lignes de la démonstration du théorème d'incomplétude.

### 2.1. La numérotation de Gödel.

Le fait d'assigner des nombres à des formules réduit la logique du raisonnement à des calculs arithmétiques.

- La première étape assigne à chaque symbole du système formel un nombre non nul différent : n1, n2,...., nk.

- Puis Gödel affecte à chaque formule un nombre qui est le produit des premiers nombres premiers élevés à la puissance du nombre représentant les symboles qui y figurent : $2^{n1} \times 3^{n2} \times \ldots\ldots \times p^{nk}$

où 2, 3,..., p est la suite ordonnée des nombres premiers.

- Selon un procédé identique, Gödel affecte un nombre à chaque suite de formules (démonstration) : $2^{m1} \times 3^{m2} \times \ldots\ldots \times q^{ms}$ où 2, 3,..., q est la suite ordonnée des nombres premiers et m1, m2,. . ..ms sont les nombres de Gödel des formules de la suite.

Du fait de l'unicité de la décomposition d'un nombre en produit de facteurs premiers, cette méthode établit un codage bi-univoque entre les expressions du système formel et un sous-ensemble de N. Le codage est ainsi possible dans tout système contenant l'arithmétique et c'est justement parce qu'il est possible qu'on rencontrera là aussi l'indécidabilité.

Inversement un nombre étant donné, on peut déterminer si c'est un nombre de Gödel[7] et si le nombre est bien de Gödel, on peut retrouver exactement l'expression qu'il code :

- un symbole,

7. Ce n'est pas toujours le cas. 8800= $2^5 \times 5^2 \times 11^1$ n'est pas un nombre de Gödel car la suite des nombres premiers n'est pas respectée ; il manque les nombres premiers 3 et 7.

- une formule,
- une démonstration

Gödel plonge ainsi le système formel initial dans un codage numérique. L'important est de comprendre que tout est nombre, dans cette approche gödelienne des mathématiques. Les nombres peuvent être considérés comme les atomes au sens où tout le reste peut être construit à partir d'eux. Pour exemple appliquons la numérotation de Gödel au système formel du paragraphe § 1.4.

A chaque symbole du système, on fait correspondre un nombre : M → 1 I → 2 U → 3

Ainsi, pour une formule on a un nombre de Gödel :

l'axiome MI → $2^1 \times 3^2 = 18$ , la formule MU → $2^1 \times 3^3 = 54$,...

La succession des règles 223 sur l'axiome génère les formules MI, MII, MIIII, MIU dont les nombres de Gödel respectifs sont :

$$18, \quad 2^1 \times 3^2 \times 5^2 = 450, \quad 2^1 \times 3^2 \times 5^2 \times 7^2 \times 11^2 = 2668050, \quad 2^1 \times 3^2 \times 5^3 = 2250$$

Le nombre de Gödel de cette démonstration est donc :

$$2^{18} \times 3^{450} \times 5^{2668050} \times 7^{2250}$$

Inversement, ce nombre étant donné, la décomposition en produit de nombres premiers donnera la suite unique des nombres de Gödel des formules qui, décomposés eux-même en produit de nombres premiers, donneront par décodage des symboles l'écriture typographique de ces formules. De même que l'ensemble de règles typographiques engendre des théorèmes, le codage de Gödel engendre des nombres entiers ; le passage de la typographie au nombre est biunivoque.

La question initiale : "MU est-elle un théorème" ? devient : " 54 est-il un nombre que peut produire le système formel" ?

2.2. **L'arithmétique : langage universel pour l'étude des systèmes formels.** L'étape suivante de Gödel est une utilisation ingénieuse de l'arithmétique pour vérifier des propriétés métamathématiques. Puisque toutes les formules ou suites de formules du système formel se trouvent codées par un nombre (de Gödel), il sera possible d'exprimer une propriété d'une formule du système par une simple formule arithmétique sur les codes ; les relations métamathématiques se réduisent ainsi à des opérations arithmétiques. Par exemple : la formule " MIIII contient la formule MII" est une relation métamathématique entre deux formules du système. Cela revient à dire que 450 (nombre de Gödel de MII) est un diviseur de 2668050 (nombre de Gödel de MIIII). Un simple calcul arithmétique permettra de le démontrer ou de le réfuter. Certains énoncés métamathématiques concernant des propriétés des formules du système formel pourront être exprimés et démontrés à l'intérieur du système lui-même.

Plus généralement, considérons une propriété arithmétique P(x) : "x à la propriété P". Le sujet x est destiné à être remplacé par un entier quelconque n pour vérifier dans quels cas P(n) est vraie et affirmer en conséquence que P(**n**) est un théorème.

Peut-on remplacer x par F ? Notre système formel ne traite que de propriétés arithmétiques c'est à dire des propriétés sur les entiers. Le sujet x ne peut être qu'un entier ;

par exemple P(**4**) : “**4** est pair” [8]. La formule F est une suite de symboles et non pas un entier. En associant le nombre de Gödel f à la formule F, P(**f**) : “**f** est pair” [9] devient décidable dans le système. Elle exprime : “F est pair”. Même si cela n’a pas grand sens, en arithmétique la formule P(**f**) est correcte.

Pour procéder à cette substitution de façon automatique dans le corps d’une démonstration par exemple, Gödel définit une fonction sub(p,f) de la façon suivante :

- si p n’est pas un nombre de Gödel ⇒ sub(p,f)=0,
- si p est le nombre de Gödel d’une formule ou d’une propriété P de sujet x, on substitue **f** aux x. On obtient la formule P(f) dont on calcule le nombre de Gödel z et on pose sub(p,f)=z.

Aucun doute sub(p,f) est calculable et arithmétique. Si on considère un triplet d’entiers k,m,n on peut calculer sub(k,m) pour savoir s’il est égal ou non à n. Selon l’encadré (3) on peut exprimer dans le système la substitution par une propriété décidable notée SUB(x,y,z) telle que :

| SUB(**p,f,z**) construit la formule P(**f**) de nombre de Gödel z=sub(p,f). (4) |
|---|

Si P est une propriété arithmétique alors la substitution transforme une phrase métamathématique en une formule arithmétique sur les entiers.

2.3. **Autoréférence.** Si on veut reconduire le paradoxe du menteur G dit : “G est menteur” dans le système formel, il faut que la phrase G détermine une propriété P et se l’applique à elle-même. En effet, si G dit : “F est menteur” :

- si G dit vrai ⇒ ce que dit F est faux,
- si G dit faux ⇒ ce que dit F est vrai.

Il n’y a pas de paradoxe. L’autoréférence semble nécessaire au paradoxe. Appliquons-la à la propriété arithmétique de parité P(x) : “x est pair” de nombre de Gödel p. Cela donne “P(x) est pair” ou encore ““x est pair” est pair” qui n’a pas grand sens. Par contre, comme nous l’avons vu précédemment “**p** est pair” en a un car p est un entier ; c’est vrai ou faux. Ainsi on peut exprimer dans l’arithmétique formelle P(**p**) : “**p** est pair” mais disons tout de suite que cela ne crée pas l’autoréférence recherchée à cause justement de la rigueur du calcul arithmétique. En effet, substituer p à x calcule pour la formule “ **p** est pair ” un nouveau nombre de Gödel qui ne peut pas être p. Plus précisement pour que P(x) se réfère à elle-même, il faut remplacer les x de P(x) par **p**. On obtient par substitution SUB(**p,p,z**) la formule P(**p**) :“**p** est pair” de nombre de Gödel z = sub(p,p). Pour avoir l’autoréférence il faudrait que z = p c’est à dire p = sub(p,p). C’est impossible car P(x) ne se réduit pas au seul sujet x. On ne peut donc pas exprimer une autoréférence directe.

Oublions un temps l’autoréférence. Dans la phrase du menteur, remplaçons G par sa citation (la phrase entre-guillemets). On obtient G : ““G est menteur” est menteur” ou encore G : “Z est menteur” si on appelle Z la citation de G. Attention, cette nouvelle

8. 4 est l’écriture formelle du nombre “ quatre ”. Cette écriture pourrait être ssss0 selon la note 6. La formule P(**4**) : “ **4** est pair ” s’écrira alors dans le système formel "∃e ss0.e=ssss0"

9. **f** est l’écriture formelle du nombre de Gödel f de la formule F

phrase Z : "G est menteur" n'a pas le même sens que la phrase initiale. Z est ici une phrase et non pas une personne comme G l'est dans la phrase du menteur ; nous y reviendrons.

Pour l'heure on peut considérer que :

- G donne la propriété " est menteur " à une phrase Z,
- Z donne la même propriété à la personne G.

Ainsi dans un premier temps G ne parle pas d'elle même mais d'une phrase Z. Comme il s'avère que Z cite G ⇒ G parlera d'elle-même. De plus si on ne fait pas de différence entre personne et phrase, on arrive indirectement à l'autoréférence à partir d'une seule phrase génératrice G(x) qui attribue la propriété " est menteur " à un sujet x. Quelle est cette phrase génératrice G(x) ? Comment obtient-on la phrase autoreproductrice G ?

G :""G est menteur" est menteur" est obtenue en substituant G au sujet x de G(x) qui construit Z que l'on qualifie de menteur.

Dans un système formel si k est le nombre de Gödel de G(x), cette dernière phrase G serait obtenue par la formule programme : "SUB(**k,k,z**) et P(**z**)". En effet SUB(**k,k,z**) substitue **k** au sujet x de G(x) et construit la formule Z=G(**k**) de nombre de Gödel z = sub(k,k). P(**z**) affirme que Z à la propriété P. Ce programme dit donc :"G(**k**) à la propriété P ".

Pour qu'il y ait autoréférence, il faut que " SUB(**k,k,z**) et P(**z**)" = G(**k**). Ainsi la formule génératrice serait G(x) :"SUB(**x,x,y**) et P(**y**)". Vérifions que l'on atteint bien l'autoréférence.

En substituant **k** à x dans G(x) on obtient à la formule G(**k**) :"SUB(**k,k,y**) et P(**y**)" qui à l'image de G construit la formule Y de nombre de Gödel y=sub(k,k) et lui donne la propriété P. G(**k**) : "Y à la propriété P ".

En remplaçant x par **k** dans G(**k**), selon la définition de l'encadré (4), G(**k**) a pour nombre de Gödel sub(k,k). Or sub(k,k) est unique et nous venons de voir que y = sub(k,k) donc Y=G(**k**). Ainsi :

| |
|---|
| G(**k**) :"SUB(**k,k,y**) et P(**y**)" ⇔ G(**k**) :"G(**k**) à la propriété P" ; k nombre de Gödel de G(x). (5) |

G(k) est une formule autoreproductrice, amenant l'autoréférence. Elle pourrait exprimer la phrase G précédente si la propriété " est menteur " était une propriété arithmétique ; ce n'est pas le cas. Mais Gödel trouvera le moyen de projeter dans l'arithmétique une propriété analogue à celle-ci : la propriété métamathématique : "n'est pas démontrable".

2.4. **Paradoxe.** Revenons à nouveau à la phrase du menteur. Nous constatons que le caractère paradoxal n'est pas seulement dû à l'autoréférence il dépend aussi du choix de la propriété. Si G dit : "G est Crétois", ce n'est pas paradoxal, c'est vrai ou faux suivant la nationalité de G. Pour reproduire le paradoxe du menteur il faut une phrase qui énonce qu'elle ment, ou encore qu'elle est fausse.

"G est menteur" est un énoncé sur un objet du langage. Il attribue une propriété P à l'objet G ; c'est la personne G qui ment.

Si on remplace G par sa citation comme précédemment, on a un énoncé sur un énoncé du langage. On est dans le métalangage : ""G est menteur" est menteur". Ici c'est la citation qui ment. C'est peut-être du mauvais français, mais cela a un sens ; c'est faux de dire "G est menteur" ou encore G dit : "" G est menteur" est fausse". Donc G ne ment pas. On est effectivement en plein paradoxe !

On a attribué la même propriété de menteur à la personne G et à sa citation Z : "G est menteur". Ainsi par substitution, le double effet amène la contradiction. C'est une autoréférence contradictoire.

S'il y a supercherie d'identité entre ces deux niveaux de lecture, il n'y en a pas lorsque tout est nombre. En attribuant un entier à chaque formule, Gödel a exprimé dans le langage de l'arithmétique une formule autoréférente paradoxale syntaxiquement correcte.

2.5. **La formule autoréférente de Gödel.** Revenons à la question métamathématique qui nous préoccupe depuis le début : la formule F est-elle un théorème du système formel ? F est un théorème ou encore F est démontrable s'il existe une démonstration D qui aboutisse à F. Cela peut être exprimé dans le système par la formule à une variable : "∃**d** DEM(**f,d**)". **f** est l'écriture formelle du nombre de Gödel de F, **d** est l'écriture formelle du nombre de Gödel de la démonstration D. Comme nous l'avons vu précédemment, au paragraphe § 1.6, DEM(**f,d**) est une propriété arithmétique décidable [10] ; il suffit simplement de savoir si d et d+f sont premiers entre eux.

La formule autoréférente choisie par Gödel dit d'elle même qu'elle n'est pas démontrable ; G : "G n'est pas démontrable". Pour obtenir son écriture formelle, partons de la formule autoréférente de l'encadré (5) ; G(**k**) : "G(**k**) à la propriété P". Remplaçons :

- G(**k**) par G de nombre de Gödel g

- P est la propriété métamathématique : "n'est pas démontrable", exprimée dans le système par la propriété arithmétique P(x) : "non(∃**d** DEM(**x,d**))" On obtient en projection de la formule métamathématique G : " G n'est pas démontrable" dans un système formel, la formule arithmétique G : "non(∃**d** DEM(**g,d**))".

On vient de mélanger des propriétés purement arithmétiques et des formules contenant une référence numérique (le nombre de Gödel g) à la notation utilisée pour formuler ces propriétés arithmétiques. Plus précisement, la propriété "n'est pas démontrable" ne fait pas partie de l'arithmétique pure (celle qui ne fait référence à aucun système de notation). C'est parce que ces notations sont codées par les nombres de Gödel qu'on peut leur appliquer de l'arithmétique. Le code permet de projeter des énoncés métamathématiques en formules arithmétiques à l'intérieur du calcul lui-même. Dans la construction précédente, g est associé à une formule arithmétique du système. La question posée est de savoir si un nombre g possède une certaine propriété arithmétique P(x). Donc la formule G se trouvent bien à l'intérieur de l'arithmétique. Il n'y a pas supercherie !

Codage des formules et projection dans l'arithmétique constituent la clé du raisonnement élaboré par Gödel.

---

10. Il faut bien faire la distinction avec la formule précédente :"∃**d** DEM(**f,d**)"qui est arithmétique mais ne sera démontrable que si elle est vraie selon l'encadré (2), car si on ne sait pas que le **d** existe la recherche peut être sans fin.

2.6. **Une contradiction dans l'arithmétique.** Gödel montre alors que la formule arithmétique G est démontrable ⇔ non G est démontrable.

1 – si G est démontrable ⇒ il existe une démonstration de G et donc un nombre d tel que DEM(**g**,**d**) est vrai. Donc "∃**d** DEM(**g**,**d**)" vrai ⇒ la formule "∃**d** DEM(**g**,**d**)" est démontrable selon l'encadré (2) ⇒ non[non(∃**d** DEM(**g**,**d**))] est démontrable ⇒ nonG est démontrable. Ainsi, si G est démontrable, sa négation l'est aussi ; le système formel est contradictoire.

2 – si nonG est démontrable ⇒ "∃**d** DEM(**g**,**d**)" est démontrable. Cela signifie comme nous l'avons vu dans l'analyse des formules existentielles c.f. § 1.6 qu'en remplaçant d par 0,1,2,... on trouvera un nombre d tel que d et d+g soient premiers entre eux, c'est-à-dire une démonstration de G. Mais si on le pense, on ne peut pas le démontrer car la recherche de la démonstration (le nombre d) peut être sans fin. On est dans une situation inconfortable où nonG démontrable ⇒ G sera démontrable, mais on ne sait pas quand... La contradiction existe, sans pouvoir la démontrer. C'est ce que Gödel a appelé la w-inconsistance.

**Ainsi Gödel ne démontre pas comme on le dit souvent l'inconsistance absolue de l'arithmétique. Il dit simplement qu'un système formel contenant de l'arithmétique n'est pas parfait car on peut écrire des contradictions ou bien si comme il se doit l'arithmétique n'est pas contradictoire il existe des problèmes qui ne peuvent être résolus. En effet, par négation de l'implication précédente, G et non G ne sont pas démontrables ⇒ G est indécidable.**

2.7. **Incomplétude ou contradiction.** Il n'y a rien de remarquable à construire dans l'arithmétique une formule indécidable, sauf si c'est une vérité. Philosophiquement, c'est beaucoup plus frustrant d'avoir des vérités indémontrables que des problèmes non résolus. Et pourtant la formule arithmétique G est vraie ! Cette vérité est établie grâce à un raisonnement métamathématique et une projection dans le formalisme de l'arithmétique et non pas en la déduisant formellement des axiomes du système. Si on admet que l'arithmétique n'est pas contradictoire, on vient de voir que le système ne peut pas démontrer G. Donc la formule "G n'est pas démontrable" est bien vraie ; c'est le raisonnement métamathématique établi dans l'exemple de l'ordinateur, au début de l'exposé dans le paragraphe § 1.3. Cette phrase métamathématique est exprimée à l'intérieur de l'arithmétique et la raison d'être de la projection étant que les formules métamathématiques vraies sont projetées dans le système par des formules arithmétiques vraies : G est donc vraie.

**Au final Gödel a construit dans un système formalisant l'arithmétique non contradictoire une formule vraie et indécidable rendant ce système formel incomplet.**

**En allant un peu plus loin, Gödel montrera que même en posant comme axiome que G est vraie, on peut encore trouver une formule vraie qui n'est pas démontrable, signifiant qu'il est inutile d'ajouter des axiomes supplémentaires, le système reste incomplet. Ainsi la réalité restera toujours plus importante que l'ensemble des connaissances formalisables et on peut prétendre des choses vraies sans avoir à les justifier.**

## 3. Bibliographie

Cette étude s'appuie essentiellement sur les deux livres suivants :

1 - *Le théorème de Gödel* [GNNG89] de E. Nagel, J.-R. Newman, K. Gödel, J.-Y. Girard – Editions du Seuil. 1989

Ce livre comprend :

* traduit de l'anglais l'ouvrage de Ernest Nagel & James R. Newman de 1958, "la démonstration de Gödel",

* traduit de l'allemand l'article de Gödel de 1931, "Sur les propositions formellement indécidables des Principia Mathematica et des systèmes apparentés",

* un commentaire de Jean-Yves Girard, "le champ du signe ou la faillite du réductionnisme".

2 - *Gödel Escher Bach* [Hof85] : les brins d'une guirlande éternelle de D.Hofstadter – Interéditions. 1985

* Cette étude est aussi inspirée par de nombreux articles que l'on trouve sur Internet avec les mots clés suivants : Théories et systèmes formels - Propositions indécidables - Arithmétique et complétude - Logique et paradoxes - Autoréférence - Le programme de Hilbert - ...

## RÉFÉRENCES

[GNNG89] K. Gödel, E. Nagel, J. Newman, and J.-Y. Girard. *Le Théorème de Gödel*. Seuil, 1989.

[Hof85] D. Hofstadter. *Gödel Escher Bach. Les brins d'une guirlande éternelle*. Interéditions, 1985.