

L'INTERPRETATION "DIALECTICA"

Th. LUCAS

Bibliographie.

- [1] GOEDEL K. "Ueber eine bisher noch nicht benützte Erweiterung
des finiten Standpunktes"
Dialectica, 12, (1958), pp. 280-287.
- [2] SHOENFIELD J.R. "Mathematical Logic"
Addison - Wesley, Reading, Mass., etc... 1967.
- [3] TROELSTRA A.S. "Aspects of Constructive Mathematics"
in "Handbook of Mathematical Logic", Ed. BARWISE J., North-
Holland, Amsterdam, etc... 1977.

Le programme de Hilbert demandait de prouver la consistance de l'arithmétique par des moyens finitistes.

"Par des moyens finitistes" peut signifier

- (1) par la seule considération de signes, de combinaisons de signes, d'assertions élémentaires sur ces combinaisons ("a finitary proof deals only with concrete objects" [2, p. 214])
- (2) par des raisonnements constructifs ("the proof deals with these objects in a constructive fashion" [2, p. 214]).

Par ailleurs, le deuxième théorème d'incomplétude de Gödel montre que si l'arithmétique A est consistante, alors

$$A \not\vdash \text{Con } A.$$

On admet généralement que :

- (1) les objets concrets des démonstrations finitistes et leurs combinaisons sont tous "représentés" dans A et que Con_A "représente" l'assertion que A est consistante,
- (2) les raisonnements constructifs sont "représentés" dans A .

On interprète donc le deuxième théorème d'incomplétude comme affirmant qu'il n'y a pas moyen de démontrer par des moyens finitistes la consistance de A .

Dans [1], Gödel propose de conserver (2) mais d'élargir (1) en admettant la considération de concepts abstraits (que seront les fonctionnelles) :

"Da die finite Mathematik als die der anschaulichen Evidenz definiert ist, so bedeutet das [...], dass man für den Widerspruchsfreiheitsbeweis der Zahlentheorie gewisse, abstrakte Begriffe braucht. Dabei sind unter abstrakten (oder nichtanschaulichen) Begriffen solche zu verstehen, die wesentlich von zweiter oder höherer Stufe sind, das heißt, die nicht Eigenschaften oder Relation konkreter Objekte (z.B. von Zeichenkombinationen) beinhalten, sondern sich auf Denkgebilde (z.B. Beweise, sinnvolle Aussagen usw.) beziehen, wobei in den Beweisen Einsichten über die letzteren gebraucht

werden, die sich nicht aus den kombinatorischen (raumzeitlichen) Eigenschaften der sie darstellenden Zeichenkombinationen, sondern nur aus deren Sinn ergeben".

Quelques remarques préliminaires constitueront le premier paragraphe de cet exposé. Dans le second, nous esquisserons un calcul de fonctionnelles. Dans le troisième nous présenterons "l'interprétation Dialectica".

I: Quelques remarques préliminaires.

Rappelons que pour prouver de façon finitiste la consistance de l'arithmétique, il suffit de prouver celle de l'arithmétique intuitionniste. On décrit en effet de façon effective une transformation $\ast$ qui à une formule φ du langage de l'arithmétique associe une formule $\varphi^\ast$ de telle sorte que $A \vdash_C \varphi$ entraîne $A \vdash_I \varphi^\ast$. Si l'arithmétique classique était inconsistante, on y démontrerait " $0 = 1$ " et de là : $A \vdash_I (0 = 1)^\ast$, puis $A \vdash_I 0 = 1$, car $(0 = 1)^\ast$ est $(0 = 1)$. (Voir l'exposé page 1 ci-dessus).

L'idée de Gödel est proche de la procédure classique de "Skolemisation" d'une théorie. Dans cette procédure, (à titre d'exemple), à une formule du type

$$\forall x \exists y \forall z \exists t \quad Q(x, y, z, t) \quad (1)$$

dans un langage L , on associe dans le langage L enrichi de symboles de fonctions une formule du type

$$\forall x \forall z \quad Q(x, gx, z, hxz) \quad (2)$$

de sorte que prouver (1) dans une théorie T revient à prouver (2) dans une théorie $T^\ast$ contenant T et des axiomes sur ces symboles de fonctions. En termes plus attentifs à l'aspect constructif, il s'agira d'associer constructivement aux formules telles que (1) des fonctions, et cela de telle sorte que si $A \vdash_I (1)$, on puisse "construire" g et h . Pour formaliser quelque peu, nous présenterons la construction de g et h dans un langage de fonctionnelles LF (voir § 2) et le résultat de Gödel prendra la forme suivante :

"A chaque formule $P(\bar{z})$ de A , on associe une formule sans quantificateurs $P_D(\bar{x}, \bar{y}, \bar{z})$.

A une démonstration de $P(\bar{z})$ dans l'arithmétique intuitionniste on peut associer effectivement une suite $\bar{X}$ de termes de LF (dont les variables sont dans $\bar{z}$) de telle sorte que $P_D(\bar{x}, \bar{y}, \bar{z})$ soit valide".

La consistance de l'arithmétique en découle comme suit : si $A \vdash_1 (0 = 1)$, alors $(0 = 1)_D$, qui s'avère être $0 = 1$ est valide. Indépendamment de nos remarques initiales, on retrouve l'aspect non finitiste de cette démonstration de consistance dans l'examen de la validité de la formule sans quantificateurs $P_D(\bar{x}, \bar{y}, \bar{z})$ qui exige l'évaluation des termes de $\bar{X}$: des articles de Tait, Howard font apparaître des concepts non finitistes ou une induction transfinie jusqu'à ε_0 dans la preuve que ces évaluations s'achèvent pour donner un nombre entier (Voir aussi note à la fin de cet article).

II. Les fonctionnelles.

On définit les types inductivement :

1) 0 est un type,

2) si σ et τ sont des types, $(\sigma \rightarrow \tau)$ est un type.

Les types de forme $(\sigma_1 \rightarrow (\sigma_2 \rightarrow \dots \rightarrow (\sigma_n \rightarrow \tau) \dots))$ sont notés $(\sigma_1, \dots, \sigma_n) \rightarrow \tau$.

Une fonctionnelle de type 0 est un nombre entier.

Une fonctionnelle de type $(\sigma \rightarrow \tau)$ est une application de l'ensemble des fonctionnelles de type σ dans l'ensemble des fonctionnelles de type τ .

Dans une perspective constructiviste, on sera amené à restreindre cette hiérarchie de fonctionnelles : l'article de Gödel parle de fonctionnelle "calculables" ("berechenbar") : une fonctionnelle calculable de type $(\sigma \rightarrow \tau)$ est une application calculable de l'ensemble des fonctionnelles calculables de type σ dans l'ensemble des fonctionnelles calculables de type τ . Quoi qu'il en soit de l'interprétation de cette notion de calculabilité, ces hiérarchies de fonctionnelles doivent contenir les entiers, la fonction successeur et être stables pour les opérations d'application,

d'abstraction et de récursion; à noter cependant que si la notion d'égalité entre fonctionnelles obéit aux règles habituelles de l'égalité (y compris la compatibilité avec l'abstraction et la récursion), elle ne doit pas nécessairement être extensionnelle ($F = G$ dès que $\forall x \ F(x) = G(x)$). Nous adopterons ici le point de vue que cette notion est suffisamment claire pour, d'une part, donner un sens précis à la notion de validité (voir ci-dessous), d'autre part donner tout son sens à l'entreprise de Gödel. (On peut aussi adopter une perspective plus formelle et développer un calcul de fonctionnelles avant d'en préciser les éventuels modèles : cfr [3]).

Pour parler des fonctionnelles, nous introduisons un langage pour fonctionnelles LF.

Les symboles de LF sont

- (1) pour chaque type σ , les variables de type σ ,
- (2) les constantes 0, de type 0, S, de type $(0 \rightarrow 0)$,
 R_σ , de type $(\sigma, 0 \rightarrow (\sigma \rightarrow \sigma), 0) \rightarrow \sigma$,
- (3) les symboles λ et ap ,
- (4) le symbole $=$,
- (5) les symboles $\neg, \wedge, \vee, \rightarrow, \leftrightarrow, \forall, \exists$.

Les termes sont définis par les clauses:

- (1) variables et constantes de type σ sont des termes de type σ ,
- (2) si X est un terme de type σ et Y un terme de type $\sigma \rightarrow \tau$,
 $\text{ap } X Y$ est un terme de type τ ,
- (3) si X est un terme de type τ et x une variable de type σ ,
 $\lambda x X$ est un terme de type $(\sigma \rightarrow \tau)$.

Les termes de forme $\text{ap } X Y$ seront notés $(X Y)$ et $X_1 X_2 \dots X_n$ abrégiera $((\dots((X_1 X_2) X_3) \dots X_n))$. Le symbole λ est considéré comme liant x dans $\lambda x X$ et $\lambda x_1 \dots \lambda x_n X$ est noté $\lambda x_1 \dots x_n X$.

Les formules atomiques de type σ de LF sont les formules de la forme $X = Y$ où X et Y sont des termes de type σ .

Les formules s'obtiennent à partir des formules atomiques au moyen des connecteurs logiques de (5). Les formules de type 0 s'obtiennent à partir des formules atomiques de type 0 par les connecteurs propositionnels de (5) et les quantificateurs en des variables de type 0. Une 0-formule est une formule de type 0 sans quantificateurs.

Modulo les remarques du début de ce paragraphe, la notion de validité est claire. La constante 0 est interprétée par le nombre naturel 0, S par la fonction successeur, ap par l'application de fonctionnelle à fonctionnelle, λ par l'abstraction, R par l'opération de récursion et = par l'égalité entre fonctionnelles. Il suffit pour ce qui suivra de définir la satisfaction d'une 0-formule pour une valuation donnée et sa validité, comme la satisfaction pour toutes les valuations, " $\models P$ " signifiera que la 0-formule P est valide.

Voici maintenant quelques indications sur le pouvoir d'expression de LF. Pour faciliter la lecture, on ne mentionnera pas le type des variables et termes, lorsqu'il peut être reconstitué par le contexte.

Il résulte immédiatement de nos définitions que l'explicitation de variables s'exprime dans LF, en ce sens que pour tout terme X de type 0 et pour toutes variables $x_1, \dots, x_n$, il y a un terme Z de LF dont les variables libres sont parmi celles de X, mais différentes de $x_1, \dots, x_n$ et tel que $\models Z \ x_1 \dots x_n = X$.
(Prendre $Z = \lambda x_1 \dots x_n X$)

En utilisant les symboles R_σ , on montrera que les "schémas de récursion" s'expriment dans LF; plus précisément, pour tous termes X, de type 0 et Y, pour toutes variables $x_1, \dots, x_n, z$ il y a un terme Z dont les variables libres sont parmi celles de X, mais différentes de $x_1, \dots, x_n$ et tel que :

$$\models Z \ 0 \ x_1 \dots x_n = X$$

$$\models Z \ (S \ z) \ x_1 \dots x_n = Y \ (Zz) \ x_1 \dots x_n .$$

Ceci nous permet d'"exprimer" l'addition, la multiplication, la fonction prédécesseur, la soustraction partout définie, l'égalité entre termes de type 0 (en ce sens qu'il y a un terme E avec $\models Exy = 0 \leftrightarrow x = y$), les tableaux de vérité des connecteurs propositionnels. On en déduira que pour toute 0-formule P, il y a un terme X_P de type 0 sans autres variables libres que celles de P tel que $\models P \leftrightarrow X_P = 0$.

On en déduira aussi que les "définitions par cas" s'expriment dans LF, en ce sens que pour toute 0-formule P, pour tous termes X et Y de type 0 et pour toutes variables $x_1, \dots, x_n$, il y a un terme Z dont les variables libres sont parmi celles de P, de X ou de Y, mais différentes de $x_1, \dots, x_n$ et tel que

$$\models P \rightarrow Zx_1 \dots x_n = X$$

$$\models \neg P \rightarrow Zx_1 \dots x_n = Y .$$

On peut enfin utiliser nos remarques sur l'expressibilité de l'addition et de la multiplication pour identifier les formules du langage de A tel que celui utilisé par exemple p. 2 à des formules de type 0 de LF.

III. L'interprétation "Dialectica"

Dans les définitions qui suivent, $\bar{x}, \bar{y}, \dots$ désignent des suites de variables de LF, $\bar{X}, \bar{Y}, \dots$ des suites de termes de LF. Si $\bar{X} = (X_1, \dots, X_m)$ et $\bar{Y} = (Y_1, \dots, Y_n)$, $(\bar{X}\bar{Y})$ est la suite $(X_1Y_1 \dots Y_n, \dots, X_mY_1 \dots Y_n)$. Si $\bar{x} = (x_1, \dots, x_m)$, $\forall \bar{x}$ est $\forall x_1 \dots \forall x_m$ et $\exists \bar{x}$ est $\exists x_1 \dots \exists x_m$.

A chaque formule P de LF nous associons une formule P^D de forme $\exists \bar{x} \forall \bar{y} P_D$ où P_D est une formule sans quantificateurs de LF et dont les variables libres sont parmi $\bar{x}$, $\bar{y}$ et celles de P . Les définitions entraîneront que si P est une formule de type 0 (en particulier une formule du langage de l'arithmétique), P_D est une 0-formule.

(1) Si P est atomique, $P^D = P_D = P$. En particulier, $\perp^D = \perp_D = \perp$.

(2) Si $P = P_1 \wedge P_2$ et si $P_1^D = \exists \bar{x} \forall \bar{y} P_{1D}$ et

$P_2^D = \exists \bar{x}' \forall \bar{y}' P_{2D}$, alors $P^D = \exists \bar{x} \bar{x}' \forall \bar{y} \bar{y}' (P_{1D} \wedge P_{2D})$

(Il est sous-entendu qu'on évite les confusions de variables et que P^D n'est défini qu'à un changement de variables liées près).

(3) Si $P = P_1 \vee P_2$ et si $P_1^D = \exists \bar{x} \forall \bar{y} P_{1D}$ et

$P_2^D = \exists \bar{x}' \forall \bar{y}' P_{2D}$, alors

$P^D = \exists z \bar{x} \bar{x}' \forall \bar{y} \bar{y}' ((z = 0 \rightarrow P_{1D}) \wedge (z \neq 0 \rightarrow P_{2D}))$

(où z est une variable de type 0).

(4) Si $P = \exists z Q$ et si $Q^D = \exists \bar{x} \forall \bar{y} Q_D$,

alors $P^D = \exists z \bar{x} \forall \bar{y} Q_D$.

(5) Si $P = \forall z Q$ et si $Q^D = \exists \bar{x} \forall \bar{y} Q_D(\bar{x}, z, \dots)$

alors $P^D = \exists \bar{x}' \forall z \bar{y} Q_D(\bar{x}'z, z, \dots)$

(6) Si $P = P_1 \rightarrow P_2$ et si $P_1^D = \exists \bar{x} \forall \bar{y} P_{1D}(\bar{x}, \bar{y}, \dots)$

et $P_2^D = \exists \bar{x}' \forall \bar{y}' P_{2D}(\bar{x}', \bar{y}', \dots)$,

alors $P^D = \exists \bar{x}'' \bar{y}'' \forall \bar{x} \bar{y}' (P_{1D}(\bar{x}, \bar{y}'' \bar{x} \bar{y}', \dots) \rightarrow P_{2D}(\bar{x}'' \bar{x}, \bar{y}', \dots))$.

Pour la justification de (5), voir le § 1.

Pour celle de (6), considérer les équivalences classiques :

$$\exists x \forall y P_{1D} \rightarrow \exists x' \forall y' P_{2D}$$

$$\sim \forall x \exists x' (\forall y P_{1D} \rightarrow \forall y' P_{2D})$$

$$\sim \forall x \exists x' \forall y' \exists y' (P_{1D} \rightarrow P_{2D})$$

et appliquer à cette dernière formule la procédure de Skolem appelée dans le § 1. (On peut aussi raisonner en termes de preuves).

THEOREME

Soit une formule $P(\bar{z})$ du langage de l'arithmétique. A une démonstration de $P(\bar{z})$ dans l'arithmétique intuitionniste, on peut associer une suite de termes $\bar{X}$ de LF dont les variables sont dans $\bar{z}$ de telle sorte que $\vdash P_0(\bar{X}, \bar{y}, \bar{z})$.

Démonstration. Elle se fait par induction sur la forme de cette démonstration. Un système se pliant bien aux nécessités de la démonstration est donné par les axiomes et règles qui suivent

(voir [1] et [3]) : $\frac{P \rightarrow Q}{Q}$; $\frac{P \rightarrow Q \quad Q \rightarrow R}{P \rightarrow R}$; $P \vee P \rightarrow P$;

$P \rightarrow P \vee Q$; $P \vee Q \rightarrow Q \vee P$; $\frac{P \rightarrow Q}{R \vee P \rightarrow R \vee Q}$;

$P \wedge Q \rightarrow P$; $P \wedge Q \rightarrow Q \wedge P$; $\frac{P \wedge Q \rightarrow R}{P \rightarrow (Q \rightarrow R)}$; $\frac{P \rightarrow (Q \rightarrow R)}{P \wedge Q \rightarrow R}$;

$I \rightarrow P$; $\frac{P \rightarrow Q}{P \rightarrow \forall x Q}$; $\forall x Q(x, \dots) \rightarrow Q(t, \dots)$; $\frac{Q \rightarrow R}{\exists x Q \rightarrow R}$;

$Q(t, \dots) \rightarrow \exists x Q(x, \dots)$ (ces quatre derniers axiomes et règles avec les restrictions usuelles; les axiomes de l'égalité et ceux de Peano mais où l'axiome d'induction est remplacé par la règle

$\frac{P(Q, \dots) \quad P(x, \dots) \rightarrow P(Sx, \dots)}{P(z, \dots)}$.

Nous examinerons quelques cas et, pour alléger la notation, supposerons autant qu'il est possible avoir à faire à des suites de variables réduites à une seule variable.

(a) Le cas des axiomes arithmétiques autres que l'induction est trivial.

(b) Le cas de la règle $\frac{P \rightarrow Q}{Q}$

Soient $P^D = \exists x \forall y P_D(x, y)$ et $Q^D = \exists x' \forall y' Q_D(x', y')$.

La formule $(P \rightarrow Q)^D$ sera de forme

$$\exists x''y'' \forall xy' (P_D(x, y''xy') \rightarrow Q_D(x''x, y'))$$

L'hypothèse d'induction nous donne des termes X, X'', Y'' (dont les variables sont parmi les variables libres de P, Q et P respectivement) tels que

$$\models P_D(X, y) \quad (1)$$

$$\text{et} \quad \models P_D(x, Y''xy') \rightarrow Q_D(X''x, y') \quad (2).$$

On en tire successivement

$$\models P_D(X, Y''xy') \quad (3), \text{ par (1),}$$

$$\models P_D(X, Y''xy') \rightarrow Q_D(X''x, y') \quad (4), \text{ par (2),}$$

$$\models Q_D(X''x, y') \quad (5), \text{ par (3) et (4).}$$

Le terme $X''x$ est le terme recherché.

(c) Le cas de l'axiome $P \vee P \rightarrow P$.

Ecrivons P^D sous les formes $\exists x \forall y P_D(x, y)$,

$$\exists x' \forall y' P_D(x', y') \text{ et } \exists x'' \forall y'' P_D(x'', y'') \quad (6).$$

La formule $(P \vee P)^D$ s'écrit sous la forme

$$\exists z \ x \ x' \forall y \ y' ((z = 0 \rightarrow P_D(x, y)) \wedge (z \neq 0 \rightarrow P_D(x', y'))).$$

Notons-la ici

$$\exists z \ x \ x' \forall y \ y' (Q(z, x, x', y, y')).$$

La formule $(P \vee P \rightarrow P)^D$ s'écrit alors en considérant (6) :

$$\begin{aligned} &\exists y''' \ y'' \ x''' \ \forall z \ x \ x' (Q(z, x, x', y''' \ zxx'y'', y'' \ zxx'y'') \\ &\rightarrow P_D(x''' \ zxx', y'')), \end{aligned}$$

c.à.d.

$$\exists y''' \ y'' \ x''' \ \forall z \ x \ x'$$

$$((z=0 \rightarrow P_D(x, y''' \ zxx'y'')) \wedge (z \neq 0 \rightarrow P_D(x', y'' \ zxx'y'')))$$

$$\rightarrow P_D(x''' \ zxx', y''))$$

Nous devons donc trouver des termes Y''' , Y^{IV} , X''' tels que

$$\models ((z=0 \rightarrow P_D(x, Y'''zxx'y'')) \wedge (z \neq 0 \rightarrow P_D(x', Y^{IV} zxx'y'')) \rightarrow P_D(X'''zxx', y'')).$$

Il suffit pour cela de poser $Y^{IV} = Y'''$ et de déterminer X''' et Y''' de telle sorte que

$$\models Y''' zxx'y'' = y''$$

et $\models X''' zxx' = x$,

ce qui peut se faire dans LF (en respectant les contraintes sur les variables).

(d) Le cas de l'axiome $(P \wedge P \rightarrow P)$.

On vérifiera que $(P \wedge P \rightarrow P)^D$ s'écrit sous la forme

$$\begin{aligned} \exists y''xx' \forall y' y'x'' (P_D(x'', y''x''yy') \\ \rightarrow P_D(xx'', y) \wedge P_D(x'x'', y')). \end{aligned}$$

Nous devons donc trouver des termes Y'' , X , X' tels que

$$\models P_D(x'', Y''x''yy') \rightarrow P_D(Xx'', y) \wedge P_D(X'x'', y').$$

On commencera par poser $X' = X$ et déterminer X par

$$\models Xx'' = x''.$$

Le problème revient donc à trouver Y'' de telle sorte que

$$\models P_D(x'', Y''x''yy') \rightarrow P_D(x'', y) \wedge P_D(x'', y') \quad (7).$$

Il suffit pour cela de déterminer Y'' au moyen d'une "définition par cas" :

$$\models P_D(x'', y) \rightarrow Y''x''yy' = y' \quad (8)$$

$$\models \neg P_D(x'', y) \rightarrow Y''x''yy' = y \quad (9);$$

en effet, (8) entraîne

$$\models P_D(x'', y) \rightarrow (P_D(x'', Y''x''yy') \rightarrow P_D(x'', y'))$$

et (9) entraîne

$$\models \neg P_D(x'', y) \rightarrow \neg P_D(x'', Y''x''yy'),$$

d'où (7) résultera en utilisant

$$\models P_D(x'', y) \vee \neg P_D(x'', y).$$

(e) Le cas de la règle $\frac{P(0) \quad P(z) \rightarrow P(Sz)}{P(x)}$

$(P(0))^D$ est de forme $\exists x \forall y P_D(x, y, 0)$,

$(P(z) \rightarrow P(Sz))^D$ est de forme

$$\exists yx' \forall xy' (P_D(x, yxy', z) \rightarrow P_D(x'x, y', Sz)),$$

de sorte que par hypothèse d'induction, il y a des termes X, Y, X' (z peut figurer dans ces deux derniers) tels que

$$\models P_D(X, Y, 0) \quad (10)$$

$$\text{et} \quad \models P_D(x, Yxy', z) \rightarrow P_D(X'x, y', Sz) \quad (11).$$

Il s'agit de trouver un terme X'' pour lequel

$$\models P_D(X'', y'', z) \quad (12).$$

On commence par "expliciter" la variable z de Y et X' en déterminant Y_1 et X'_1 par

$$\models Y_1 xy'z = Yxy'$$

$$\models X'_1 xz = X'x$$

de sorte que (11) s'écrit :

$$\models P_D(x, Y_1xy'z, z) \rightarrow P_D(X'_1xz, y', Sz) \quad (13).$$

On détermine un terme X''_1 par récursion:

$$\models X''_1 0 = X \quad (14)$$

$$\models X''_1 (Sz) = X'_1 (X''_1 z)z \quad (15)$$

(ou en posant directement $X''_1 = R X_1 X'_1$)

et on pose $X'' = X''_1 z$. On prouve alors (12) par récurrence sur la valeur de z . Si z a la valeur 0, (12) est assuré par (10) et (14).

Nous supposons maintenant que

$$P_D(X'', y'', z) \text{ est satisfait lorsque } z \text{ prend la valeur } n \text{ et } y'' \text{ une valeur quelconque} \quad (16).$$

D'ailleurs, (13) entraîne

$$\models P_D(X'', Y_1 X'' y'' z, z) \rightarrow P_D(X'_1 X'' z, y'', Sz).$$

En vertu de (16), l'antécédent de cette implication est satisfait lorsque z prend la valeur n .

Donc, toujours pour cette même valeur de z ,

$$P_D(X'_1 X'' z, y'', Sz),$$

à savoir $P_D(X'_1 (X''_1 z) z, y'', Sz),$

à savoir $P_D(X''_1 (Sz), y'', Sz)$ (en vertu de (15)),

est satisfait. En d'autres termes,

$$P_D(X'', y'', z)$$

est satisfait lorsque z prend la valeur $n + 1$, ce qu'il fallait démontrer.

N.B. Si le calcul des fonctionnelles est entièrement présenté de façon formelle (sans notion sémantique, comme dans l'article de M. Crabbé page 83) la démonstration précédente montre que $P_D(X'', y'', z)$ pour tout terme clos z seulement si tout terme fermé de type 0 se réduit à un numéral. Plus précisément, pour z clos et de type 0,

- 1) si l'on admet le point de vue sémantique, z désigne un naturel, et ceci justifie la preuve par induction.
 - 2) si l'on adopte le point de vue syntactique, il faut pour justifier la preuve par induction, démontrer au préalable que z se réduit à un numéral; cette démonstration excède l'arithmétique.
-