# No-counterexample interpretation et spécification des théorèmes de l'arithmétique

Denis Bonnay

# Le contenu computationnel des preuves : No-counterexample interpretation et spécification des théorèmes de l'arithmétique

mémoire sous la direction de Jean-Louis Krivine
par Denis Bonnay

septembre 2002
D.E.A. logique et fondements de l'informatique
Université Paris VII

# Contents


**1 La preuve de consistance d'Ackermann** **4**
- 1.1 L'epsilon-calcul et l'arithmétique epsilon . . . . . 4
- 1.2 La méthode de substitution . . . . . 5
- 1.3 La preuve de terminaison . . . . . 7
- 1.4 Le bornage . . . . . 10
  - 1.4.1 Le codage des ordinaux . . . . . 10
  - 1.4.2 Le bornage en fonction du nombre de substitutions . . . . 11
  - 1.4.3 Le bornage du nombre de substitutions . . . . . 12

**2 La No-counterexample interpretation** **15**
- 2.1 La notion d'interprétation . . . . . 15
- 2.2 Vérification de la condition $\beta$) . . . . . 17
- 2.3 Modularité de l'interprétation . . . . . 19
- 2.4 Retour sur les lemmes . . . . . 22

**3 Réalisabilité classique et arithmétique** **25**
- 3.1 Le lambda-c calcul et les règles de typage . . . . . 25
- 3.2 La réalisabilité . . . . . 25
- 3.3 L'arithmétique du second ordre . . . . . 27
- 3.4 La spécification des théorèmes de l'arithmétique . . . . . 30
  - 3.4.1 Pour les énoncés $\Pi_2$ . . . . . 30
  - 3.4.2 Pour les énoncés $\Sigma_2$ . . . . . 31
  - 3.4.3 Dans le cas général . . . . . 33
- 3.5 Remarque finale sur les deux approches . . . . . 35

Que nous apporte la preuve d'un théorème, en plus du simple fait de savoir que le théorème est une conséquence de certains axiomes ? Il s'agit de voir quel contenu purement calculatoire peut être extrait de preuves qui, lorsqu'elles utilisent toutes les ressources de la logique classique ne vont pas forcément exhiber pour nous les objets dont elles parlent. Si l'on applique ce programme à l'arithmétique, on aimerait que les preuves nous fournissent des fonctions calculables. Mais l'interprétation naïve selon laquelle la preuve d'un énoncé de la forme $\forall_1 \exists y_1 ... \forall x_k \exists y_k A(x_1 y_1 ... x_k y_k)$ devrait nous fournir des fonctions calculables $\psi_1(x_1)...\psi_k(x_1...x_k)$ telles que $A(x_1\psi_1(x_1)...x_k\psi_k(x_1...x_k))$ soit vrai, ne vaut pas, comme il est facile de le montrer. Considérons le prédicat récursif primitif $Halt(x_1, x_2, y)$ qui est vrai si et seulement si la machine de Turing de numéro $x_1$ avec l'entrée $x_2$ s'arrête au bout de $y$ pas. On a bien :

$\vdash_{AP} \forall x_1 \forall x_2 \exists y(Halt(x_1, x_2, y) \vee \forall z \sim Halt(x_1, x_2, z))$

ou encore

$\vdash_{AP} \forall x_1 \forall x_2 \exists y \forall z(Halt(x_1, x_2, y) \vee \sim Halt(x_1, x_2, z))$

Mais il n'y a pas de fonction récursive $\psi$ telle que $Halt(x_1, x_2, \psi(x_1, x_2) \vee \forall z \sim Halt(x_1, x_2, z)$. L'interprétation naïve est donc fausse au moins à partir de $\Pi_3$ et donc de $\Sigma_2$.

Nous allons présenter la solution de Kreisel à ce problème, qui est connue sous le nom de No-counterexample interpretation. L'idée de la démonstration de Kreisel est que les preuves de non-contradiction de l'arithmétique, dont le but, dans l'esprit du programme Hilbertien, était de montrer que le détour par des moyens de preuves non-finitistes pour prouver des énoncés purement numériques était acceptable, peuvent en fait être également utilisées pour donner un sens finitiste aux énoncés complexes. Nous commencerons donc par donner la preuve de consistance d'Ackermann sur laquelle s'appuie toute la démonstration de Kreisel [8] et [9].

Nous présenterons enfin un des prolongements contemporains de ce programme via la réalisabilité classique. Le programme d'extraction du contenu computationnel des preuves est repris, mais les méthodes changent. On ne cherche plus à se ramener à des fonctions récursives, comme le faisait Kreisel ou comme le font les interprétations fonctionnelles[1] dans l'esprit de l'interprétation *Dialectica* de Gödel (voir [4] et [2] pour une exposition des prolongements). Le paradigme de l'interprétation est celui des programmes : l'idée derrière les travaux de Krivine [12] et [14] est d'interpréter directement les axiomes comme des instructions de programmation, au sens où par exemple l'extension de l'isomorphisme de Curry-Howard repose sur l'interprétation de l'axiome de Pierce comme un *exception-handler*. On présente ici les résultat de spécification pour des théorèmes de l'arithmétique classique du second ordre (où l'axiome de compréhension découle de la règle pour l'élimination des quantificateurs uni-

[1] Dans ces interprétations, c'est d'ailleurs d'abord un système intuitionniste qui est interprété dans une théorie fonctionnelle; on cherche ensuite à étendre l'interprétation au-delà de l'arithmétique de Peano en ajoutant des principes de récurrence dans la théorie fonctionnelle (par exemple le principe de bar-récursion pour $AP^2 + AC$). Ensuite l'interprétation proprement dite repose sur le préalable de la traduction par la double négation du système classique dans le système intuitionniste.

versels du second ordre), en exhibant le lien avec l'interprétation de Kreisel et en insistant sur la question de la modularité de l'interprétation.

# 1 La preuve de consistance d'Ackermann

## 1.1 L'epsilon-calcul et l'arithmétique epsilon

L'$\epsilon$-calcul est un langage sans quantificateur dans lequel les quantificateurs sont remplacés par des $\epsilon$-termes que l'on peut voir comme des fonctions de choix. Si $A(x, y_1...y_n)$ est une formule avec $x, y_1...y_n$ comme variables libres, $\epsilon_x A(x, y_1...y_n)$ est un terme avec $y_1...y_n$ comme variable libres, qu'il faut voir comme une fonction donnant pour des $b_1...b_n$ quelconques un témoin pour $A(x, b_1...b_n)$ s'il y en a un. Les $\epsilon$-termes sont gouvernés par un schéma d'axiome, l'axiome du transfini de Hilbert :

$A(t) \rightarrow A(\epsilon_x Ax)$

On peut alors définir les quantificateurs de la manière suivante :

$\exists x A(x) \equiv A(\epsilon_x Ax)$

$\forall x A(x) \equiv A(\epsilon_x \sim Ax)$

et retrouver à partir de là les règles habituelles pour $\forall, \exists$ dans le système choisi.

On rappelle ici le système à la Hilbert utilisé par Ackermann, qu'on désignera par $\mathrm{PA}_\epsilon$. A quelques variantes près, il s'agit du système $\mathrm{Z}_\mu$ de Hilbert et Bernays [6]. Les termes et les formules du langage sont formés à partir de variables, de 0, des fonctions $d, ', +, \times$, du signe d'égalité et des symboles logiques $\rightarrow$ et $\sim$. La seule règle utilisée est le *modus ponens*. Il y a trois groupes de schémas d'axiomes :

1) Schémas d'axiomes pour le niveau propositionnel

A,B désignent des formules sans variables libres

I.1 $A \rightarrow (B \rightarrow A)$

I.2 $(A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C))$

I.3 $(\sim A \rightarrow \sim B) \rightarrow (B \rightarrow A)$

2) Schémas d'axiomes de l'arithmétique

a,b,c désignent des termes sans variables libres

II.01 $a = a$

II.02 $a' = b' \rightarrow a = b$

II.03 $a \neq 0 \rightarrow d(a') = a$

II.04 $a + 0 = a$

II.05 $a + b' = (a + b)'$

II.06 $a \times 0 = 0$

II.07 $a \times b' = (a \times b) + a$

II.08 $a = b \rightarrow a' = b'$

II.09 $a = b \rightarrow d(a) = d(b)$

II.10 $a = b \rightarrow a + c = b + c$

II.11 $a = b \rightarrow c + a = c + b$

II.12 $a = b \rightarrow a \times c = b \times c$

II.13 $a = b \to c \times a = c \times b$

3) Schémas d'axiomes pour les $\epsilon$-termes

a,b désignent des termes sans variable libre

III.1 $A(a) \to A(\epsilon_x Ax)$

III.2 $A(a) \to \epsilon_x Ax \neq a'$

III.3 $\sim A(\epsilon_x Ax) \to \epsilon_x Ax = 0$

III.4 $a = b \to \epsilon_x A(x, a) = \epsilon_x A(x, b)$

On remarque que toutes les formules d'une déduction sont des formules sans variables libres.

On aurait pu utiliser à la place du groupe I un autre système déductif, mais l'essentiel est ici que la démonstration puisse se faire avec un système de logique classique : les seules propriétés utilisées dans la démonstration de la consistance sont que ces règles préservent la vérité des formules sans quantificateurs et qu'interviennent seulement des formules closes. Les schémas d'axiomes du groupe II pourraient être remplacées n'importe quels autres schémas correspondant à des formules universelles vraies de l'arithmétique. En particulier, on pourrait enrichir le langage avec d'autres fonctions récursives et leurs définitions ou intégrer par exemple le schéma de récurrence primitive.

Les axiomes de la forme III.1-III.4 sont appelés formules critiques.Dans le cadre de l'arithmétique, les $\epsilon$-termes sont vus non seulement comme des témoins mais comme les plus petits témoins possibles. Le schéma d'induction complète est dérivable dans l'arithmétique-epsilon, en utilisant essentiellement III.1 et III.2. A partir de $A(0)$ et $A(t) \to A(t')$, l'idée est de montrer $A(\epsilon_x \sim Ax)$, en utilisant le fait que $\epsilon_x \sim Ax$ ne peut être différent de 0.

## 1.2 La méthode de substitution

Le but de l'article d'Ackermann est de fournir une preuve de la consistance de l'arithmétique-epsilon, montrant ainsi que le détour par le transfini via les $\epsilon$-termes n'introduit pas de contradiction. L'idée est de remplacer les $\epsilon$-termes d'une preuve par des entiers de manière à ce que tous les axiomes critiques soient vrais. Comme les axiomes obtenus à partir des schémas de I et II sont vrais, toutes les étapes de la démonstration deviennent alors des formules purement numériques vraies. Ceci montre qu'il n'y a pas de preuve de $0 = 1$, car cette formule est fausse. Le point qui nous intéresse dans cette démonstration est la possibilité de borner les valeurs susbstituées aux $\epsilon$-termes en fonction de constantes de la preuve, mais avant d'en arriver là, il faut donner la preuve de non-contradiction.

**Definition 1** *On désignera par catégorie (Grundtyp) d'un $\epsilon$-terme clos $\epsilon_x Ax$ le terme obtenu à partir de celui-ci en remplaçant tous ces sous-termes immédiats dans lesquels $x$ n'apparaît pas libre par des variables fraîches.*

Par exemple $\epsilon_x(0' + \epsilon_y(y = 0'') = \varepsilon_z(z' = x)$ appartient à la catégorie $\epsilon_x(w = \varepsilon_z(z' = x)$, tandis que $\epsilon_x(\epsilon_y(y = 0'') + x = x'')$ est à lui-même sa propre catégorie. Deux $\epsilon$-termes peuvent appartenir à la même catégorie. Le rang

(*Rang)* d'une catégorie est le nombre d'$\epsilon$-termes enchassées qu'elle contient. Le rang d'un $\epsilon$-terme est le rang de sa catégorie.

**Definition 2** *Une substitution (Gesammtersetzung) relativement à un ensemble de formules closes est l'assignation à chaque catégorie ayant k variables libres de l'ensemble des catégories des formules d'une fonction k-aire (aux catégories sans variables libres sont donc assignés des entiers).*

On appelle la fonction assignée à une catégorie son substituant. La solution d'un ensemble de formules closes relativement à une substitution pour l'ensemble des catégories de ses $\epsilon$-termes, est le résultat obtenu en remplaçant dans les formules les $\epsilon$-termes par les fonctions qui sont assignées aux catégories auxquelles ils appartiennent. On dira que le substituant d'une catégorie est nul s'il s'agit de 0 ou d'une fonction constante égale à 0.

On dira qu'une substitution pour un ensemble de catégories a la propriété P dans le cas suivant. Si à $\epsilon_x A(x, w_1...w_n)$ une des catégories est assignée la fonction $f(w_1...w_n)$, alors, pour des entiers quelconques $b_1...b_n$, soit $f(b_1...b_n) = 0$, soit $A(f(b_1...b_n), b_1...b_n)$ est le cas et $f(b_1...b_n)$ est le plus petit entier m tel que $A(m, b_1...b_n)$.

Comme les $\epsilon$-termes d'un axiome de la forme III.4 appartiennent à la même catégorie, une substitution rend forcément tous les axiomes de cette forme vraie. De plus, si une substitution a la propriété P, les axiomes de la forme III.2 et III.3 sont également rendus vrais.

Relativement à une preuve vue comme ensemble de formules, on se donne une énumération de ces catégories telle que si un $\epsilon$-terme $b$ avec $x$ comme variable libre est un sous-terme d'un autre $\epsilon$-terme $\epsilon_x Ax$, la catégorie de $b$ précède la catégorie de $\epsilon_x Ax$ (*). On définit alors une suite de substitutions $S_0...S_n...$ de la manière suivante :

1) $S_0$ assigne à toutes les catégories des substituants nuls.

2) Si on est arrivé à la subsitution $S_n$ qui a la propriété P, on regarde le premier axiome de la forme III.1 qui est rendu faux par $S_n$. Disons qu'il s'agit de $A(a, b) \rightarrow A(\epsilon_x A(x, b), b)$. On définit alors $S_{n+1}$ ainsi. Si la formule est fausse, cela signifie que $A(a, b)$ est vrai et $A(\epsilon_x A(x, b), b)$ faux. On modifie alors la valeur du subsituant $f(x)$ de la catégorie de $\epsilon_x A(x, b)$ pour la valeur m, où m est la valeur donnée par $S_n$ au terme $b$, en la remplaçant par le plus petit $n$ tel que $A(n, m)$. Toutes les catégories qui suivent la catégorie de $\epsilon_x A(x, b)$ reçoivent des subsituants nuls. Les autres substituants sont inchangés. On vérifie que $S_{n+1}$ a encore la propriété P. Soit $\epsilon_x B(x, w_1...w_n)$ une catégorie. Si elle suit la catégorie dont on a modifié le substituant, son nouveau substituant est nul et donc la condition est trivialement remplie. Si elle la précède, ses substituants, ainsi que tous ceux d'autres $\epsilon$-termes de B $B(x, w_1...w_n)$ à cause de (*) sont inchangés, donc la propriété est héritée de $S_n$. Quant à $\epsilon_x A(x, y)$ elle-même, la condition est vérifiée lorsque $y$ reçoit $b$, par construction et par (*), et pour les autres elle est vérifiée comme dans le cas précédent.

Lorsqu'on applique ces substitutions aux formules d'une preuve, parmi les axiomes utilisés, seuls ceux de la forme III.1 sont susceptibles d'être faux. Si la

suite de substitutions s'arrête pour un certain $n$, c'est donc que tous les axiomes du résultat sont vrais, et toutes les formules du résultat également. Il suffit donc de montrer que la suite de substitutions termine. Intuitivement, il semble bien qu'on corrige peu à peu la substitution initiale. Néanmoins, il serait erroné de penser que chaque substitution corrige une fois pour toutes un axiome de sorte que la suite des substitutions finisse par épuiser la suite des axiomes. Par exemple, en modifiant le substituant de $\epsilon_x A(x,y)$ pour rendre vrai $A(a,b) \rightarrow A(\epsilon_x A(x,b),b)$, il se peut que je rende un autre axiome $B(c,d) \rightarrow B(\epsilon_x B(x,d),d)$ qui était vrai en modifiant la valeur de $d$.

## 1.3 La preuve de terminaison

**Definition 3** *Si une preuve a $m$ catégories et qu'une substitution $S$ annule tous les substituants à partir de la $n$-ième catégorie, on attribue à $S$ le nombre caractéristique $m - s$.*

**Definition 4** *On ordonne les $\epsilon$-termes clos de la preuve, de manière à ce qu'un sous-terme précède le terme dans lequel il apparaît. Soit $a_0...a_k$ une telle suite de termes, on définit l'ordre (Reduktionsgrad) d'une substitution $S$ par $o(S) = 2^k f(o) + 2^{k-1} f(1) + ... + 2^0 f(k)$ où $f(i) = 1$ si $S$ donne à $a_i$ la valeur 0 et $f(i) = 0$ sinon.*

**Definition 5** *Le degré d'une substitution $S$, noté $d(S)$, est son ordre relativement, non plus aux formules de la preuve, mais à l'ensemble de formules $A(0,b)...A(k,b)$ où $A(a,b) \rightarrow A(\epsilon_x A(x,b),b)$ est le premier axiome rendu faux par $S$ et où $k$ est la valeur que $S$ donne à $a$. Si $S$ est la substitution finale, on fixe $d(S) = 0$.*

**Definition 6** *L'indice, noté $i(S)$, d'une substitution $S$ est alors la paire (o,d) où $o$ est l'ordre et $d$ le degré de $S$. On ordonne les indices selon l'ordre lexicographique.*

**Definition 7** *On dira qu'une substitution $T$ est (strictement) progressive sur une substitution $S$ si chaque fois qu'un substituant de $S$ prend une valeur non nulle, $T$ lui donne la même valeur (et $S$ n'est pas progressive sur $T$).*

**Theorem 8** *Soient $S$ et $T$ deux substitutions fournissant des substituants pour toutes les catégories d'un ensemble de formules closes, si $T$ est progressive sur $S$, soit 1) $o(T) < o(S)$, soit 2) les $\epsilon$-termes des formules reçoivent tous les mêmes valeurs par $S$ et $T$.*

Sous l'hypothèse de progressivité, on montre non 2) implique 1). Soit $a_i$ le premier $\epsilon$-terme dans l'énumération sur lequel S et T diffèrent; comme T est progressive sur S et que T et S s'accordent sur tous les $\epsilon$-termes précédents, cela veut dire que S annule $a_i$. Par définition de $o$, $o(T) < o(S)$.

**Theorem 9** *Si $S_j$ est progressive sur $S_i$, alors soit 1) $i(S_j) < i(S_i)$, soit 2) $S_{j+1}$ est progressive sur $S_{i+1}$ et elles sont obtenues de la même manière à partir de $S_j$ et $S_i$.*

Si $o(S_j) < o(S_i)$, 1) est le cas. Sinon, par le théorème précédent, les $\epsilon$-termes reçoivent les mêmes valeurs par les deux substitutions. Par conséquent, le premier axiome $A(a,b) \rightarrow A(\epsilon_x A(x,b),b)$ qu'elles rendent faux est le même. On considère alors l'ensemble de formules $A(0,b)...A(k,b)$. $S_j$ est toujours progressive sur $S_i$ relativement à cet ensemble. On réapplique le théorème précédent. Si $d(S_j) < d(S_i)$, on a $i(S_j) < i(S_i)$; sinon $\epsilon_x A(x,b)$ recevra la même valeur dans $S_j$ et $S_i$.

**Definition 10** *On définit la notion de m–série (m-Reihe) de substitutions. Une 1-série est constituée d'une seule substitution. Une m+1-série est une suite $S_i S_{i+1}...S_{i+t}$ $(t \geq 0)$ de substitutions successives telles que les nombres caractéristiques de $S_{i+1}...S_{i+t}$ sont $\leq m$ et ceux de $S_i, S_{i+t+1}$ sont $\geq m+1$ (ou bien $S_{i+t}$ est la substitution finale).*

On remarque qu'une m+1-série est constituée d'au moins une m-série.

**Definition 11** *L'indice d'une m-série est un ordinal défini de la manière suivante. Une 1-série a l'indice $\omega o + d$ où $(o,d)$ est le couple d'entiers attribué à la substitution qui la constitue. Une m+1 série est constituée d'une suite de m-séries successives d'indices $a_1...a_n$, elle a pour indice $\omega^{a_1} + ... + \omega^{a_n}$.*

**Theorem 12** *Soient $S_1...S_j, T_{j+1}...T_{j+l+1}$ deux m-séries consécutives avec $T_{j+1}$ de nombre caractéristique $m$ et $a_1...a_j, b_{j+1}...b_{j+l+1}$ les indices des 1-séries qui les constituent. 1) $T_{j+1}$ est strictement progressive sur $S_1$ et 2) On peut trouver un entier $p$ tel que $a_p > b_{j+p}$, et pour $1 \leq i < p$, $a_p = b_{j+p}$, et $S_i$ et $T_{j+1}$ ont le même nombre caractéristique pour $1 < i \leq p$.*

1) Soit $g$ le nombre de catégories. On considère la suite de ces $g$ catégories $\epsilon_1...\epsilon_{g-(m+1)}, \epsilon_{g-m}...\epsilon_g$. On sait déjà que $S_1$ et $T_{j+1}$ annulent les substituants de $\epsilon_{g-m}...\epsilon_g$. Maintenant, comme $S_2...S_j$ ont des nombres caractéristiques $< m$, elles ne touchent pas aux substituants de $\epsilon_1...\epsilon_{g-(m+1)}$. Comme $T_{j+1}$ est de nombre caractéristique $m$, elle est obtenue à partir de $S_j$ en rendant positive une valeur nulle de $\epsilon_{g-(m+1)}$. Donc $T_{j+1}$ est bien strictement progressive sur $S_1$.

2) Si $a_1 > b_{j+1}$, il suffit de prendre p=1. $a_1 < b_{j+1}$ car $T_{j+1}$ est bien strictement progressive sur $S_1$. On se place maintenant dans la configuration $a_1 = b_{j+1}$. On va supposer qu'il n'y a pas de p tel que $a_p \neq b_{j+p}$ et en dériver une contradiction.

Premier cas, la première m-série est strictement plus longue. On remarque que $T_{j+l+1}$ ne peut être la substitution finale, car par hypothèse, elle a le même indice $(o,d)$ que $S_{l+1}$, de sorte qu'on aurait alors $d(\ S_{l+1}) = 0$ de sorte que $S_{l+1}$ aurait déjà dû être la substitution finale. On montre facilement par induction que $T_{j+l+1}$ est progressive sur $S_{l+1}$. Pour 1, cela découle de 1). Pour $k+1$ $(k \leq l)$, cela découle de l'hypothèse de récurrence et du fait que $T_{j+k}$ et $S_k$ ont même indice, de sorte que le passage à $T_{j+k+1}$ et $S_{k+1}$ correspond à la même modification. Comme $T_{j+l+1}$ est progressive sur $S_{l+1}$, on peut appliquer le théorème 9 et en déduire que $T_{j+l+2}$ est progressive sur $S_{l+2}$. Or ceci est impossible car $S_{l+2}$ a un nombre caractéristique $< m$ tandis que comme $T_{j+l+2}$

suit une m-série, elle a un nombre caractéristique $\geq m$. On peut donc trouver un $p$ tel que $a_p \neq b_{j+p}$.

Deuxième cas : la deuxième m-série est au moins aussi longue que la première. $S_j$ et $T_{2j+1}$ ont même indice par hypothèse. Donc $T_{j+1}$ et $T_{2(j+1)}$ reçoivent le même nouveau substituant positif. Mais c'est impossible car $T_{2j+1}$ a un nombre caractéristique $< m$, ce qui veut dire qu'il a préservé le nouveau substituant positif de $T_{j+1}$.

On a donc démontré l'existence d'un premier $p$ tel que $a_p \neq b_{j+p}$. Et on doit bien avoir alors $a_p > b_{j+p}$ à cause du fait démontré à l'instant que $T_{j+k}$ est progressive sur $S_k$ pour $k \leq p$. Et $S_i$ et $T_{j+1}$ ont bien le même nombre caractéristique pour $1 < i \leq p$, étant donné que les résultats des substitutions sont les mêmes (par progressivité et égalité des indices pour $i < p$). On appellera $S_p$ et $T_{j+1+p}$ les substitutions déterminées par les deux m-séries consécutives.

**Theorem 13** *On se donne les mêmes conditions que dans le théorème précédent ainsi qu'un $p$ tel que $1 \leq p \leq m$ et les indices $a_1...a_s$, $b_1...b_t$ des p-séries $\Sigma_1...\Sigma_s$, $\Upsilon_1...\Upsilon_t$ qui composent les deux m-séries. Il existe alors un entier $q$ tel que $a_q$ et $b_q$ sont les indices des deux p-séries qui contiennent les substitutions déterminées. De plus, $a_q > b_q$ et $a_{q'} = b_{q'}$ pour $q' < q$.*

La démonstration se fait par récurrence sur $(m, p)$. Le théorème précédent nous donne le cas de base et les cas $m$ quelconque et $p = 1$. On suppose maintenant que le résultat vaut pour un $m$ fixé jusqu'à $p$ exclus ($p > 1$) et pour tous les $(m', p')$ où $m' < m$. On veut le résultat pour $(m, p)$. Le théorème précédent nous permet de déterminer les substitutions déterminées $S_p$ et $T_{j+1+p}$. Pour $i < p$, $S_i$ et $T_{j+1+i}$ ont les mêmes indice et nombre caractéristique, donc il existe un $q$ tel que $S_p$ appartient à $\Sigma_q$ et $T_{j+1+q}$ appartient à $\Upsilon_q$ tel que pour $j < q$, $a_j = b_j$. On décompose $\Sigma_q$ et $\Upsilon_q$ en $p-1$-séries $\Gamma_1...\Gamma_{s'}$, $\Delta_1...\Delta_{t'}$ d'indices $c_1...c_{s'}$, $d_1...d_{t'}$. On applique l'hypothèse de récurrence pour $(m, p-1)$. Le décrochage se fait nécessairement à l'intérieur de $\Sigma_q$ et $\Upsilon_q$, donc on trouve un $q'$ tel que $c_{q'} > d_{q'}$ et pour $j' < q'$, $c_{j'} = d_{j'}$. Donc $a_q = \omega^{c_1} + ... + \omega^{c_{q'-1}} + \omega^{c_{q'}} + ... + \omega^{c_{s'}}$ et $b_q = \omega^{c_1} + ... + \omega^{c_{q'-1}} + \omega^{d_{q'}} + ... + \omega^{d_{t'}}$. En appliquant l'hypothèse de récurrence avec $(p-1, p-1)$, on a que $c_1 > c_2 > ... > c_{s'}$ et $d_1 > d_2... > d_{t'}$. On peut alors conclure $a_q > b_q$.

**Corollary 14** *Deux m-séries consécutives $S_1...S_j, T_{j+1}...T_{j+l+1}$ avec $T_{j+1}$ de nombre caractéristique $m$ ont des indices strictement décroissants.*

**Theorem 15** *Pour toute preuve, la suite des substitutions termine sur un résultat qui rend vraie toutes les formules de la preuve.*

Soit g le nombre de catégories de la preuve. Comme il n'y a pas de chaîne infinie descendante d'ordinaux, il y a un nombre fini de g-séries de la forme $S_i...S_{i+t}$ avec $S_{i+t}$ non finale. Après cela viennent un nombre fini de g-1 séries et ainsi de suite jusqu'à la substitution finale qui est la dernière 1-série. La suite des substitutions termine donc, ce qui veut dire que tous les axiomes IIII.1 sont rendus vrais; comme tous les autres axiomes sont par ailleurs rendus vrais par

toutes les substitutions de la suite, toutes les formules de la preuve sont *in fine* rendues vraies.

## 1.4 Le bornage

La preuve de terminaison nous a montré qu'étant donnée une preuve dans l'arithmétique-epsilon, on pouvait trouver des substituants, sous la forme de fonctions récursives (et même de fonctions toujours égales à 0 sauf un nombre fini de fois) aux catégories de manière à ce que dans le résultat de la substitution toutes les formules soient vraies. Dans le cas de la preuve d'une formule existentielle, ce qui correspond dans le langage epsilon à une formule de la forme $A(\epsilon_x Ax)$, on obtient donc une substitution qui résout le terme $\epsilon_x Ax$ en un entier $n$ tel que $A(n)$ soit vrai. Ackermann montre à la suite de la preuve de consistance proprement dite qu'il est possible d'exhiber une fonction récursive qui donne une borne pour les valeurs qui sont substituées aux $\epsilon$-termes des formules. Indépendamment de son intérêt intrinsèque (en particulier, elle fait apparaître "ce qui compte" dans la preuve, c'est-à-dire les arguments de cette fonction), l'existence et la nature de cette fonction sont au coeur de la NCI de Kreisel. On expliquera donc en détail la construction de cette fonction, en adaptant l'exposition à l'utilisation de ce résultat par Kreisel, et on donnera ensuite les étapes de la preuve de ce que cette fonction borne effectivement les valeurs des $\epsilon$-termes, en omettant la partie la plus fastidieuse et la moins intéressante de la démonstration (Sätze 6 à 17 du paragraphe 7 de [1]).

On peut poser le problème de la manière suivante : les valeurs possibles pour les $\epsilon$-termes augmentent clairement avec le nombre de substitutions que comporte la preuve. Une fois identifiées les variables pertinentes de la preuve, trouver une fonction de ces variables et du nombre de ces substitutions qui borne les valeurs possibles, et une autre qui borne à partir de ces seules variables le nombre de substitutions. Si la première partie de la tâche est facile à mener, la seconde demandera davantage de travail. Avant cela, il nous faut définir un codage des types d'ordres des ordinaux $< \epsilon_0$ sur les entiers qui va permettre de manipuler au sein de l'arithmétique habituelle les indices de suite. Si cette opération peut s'expliquer par le souci finitiste d'Ackermann, ce codage n'en reste pas moins facultatif, au sens où l'on pourrait après tout manipuler directement les ordinaux. Mais l'utilisation par Kreisel du résultat de bornage repose de manière essentielle sur la possibilité d'effectuer le calcul des bornes dans l'arithmétique-epsilon, de sorte qu'il est nécessaire de présenter précisément les choses.

### 1.4.1 Le codage des ordinaux

Au vu de la preuve de terminaison, les ordinaux qu'on a besoin d'utiliser sont tous des ordinaux de type $m$ pour un $m$ donné au sens de la définition suivante :

**Definition 16** *Les ordinaux de la forme $\omega a + b$ sont de type* 1.

*Si $\alpha_1, \alpha_2, ...\alpha_i$ sont des ordinaux de type $n$ tels que $\alpha_1 > \alpha_2 ... > \alpha_i$, $\omega^{\alpha_1} + \omega^{\alpha 2} ... + \omega^{\alpha_i}$ est un ordinal de type $n + 1$.*

On cherche alors, pour chaque $m$, des relations d'ordre $<_m$ et une bijection $f(\beta, m)$ qui associe à un ordinal $\beta$ de type $m$ un entier telles que $f$ soit un isomorphisme entre les ordinaux de type $m$ avec l'ordre habituel sur les ordinaux et les entiers équipés de $<_m$ .

**Definition 17** $f(\omega a + b, 1) = 2^a(2b+1) - 1$

$2^a(2b+1) - 1 <_1 2^c(2d+1) - 1$ *si et seulement si* $(a, b) < (c, d)$ *avec l'ordre lexicographique*

$f(\omega^{\alpha_1} + \omega^{\alpha_2} ... + \omega^{\alpha_i}, m+1) = 2^{a_1} + 2^{a_2} ... + 2^{a_i}$ *où* $f(\alpha_j, m) = a_j$.

$2^{a_1} + 2^{a_2} ... + 2^{a_i} <_{m+1} 2^{b_1} + 2^{b_2} ... + 2^{b_j}$ *où* $a_1 >_m a_2 >_m ... >_m a_i$ *et* $b_1 >_m b_2 >_m ... >_m b_j$ *si et seulement si il existe un* $k \leq i, j$ *tel que* $a_1 = b_1, ...a_{k-1} = b_{k-1}$ *et* $a_k <_m b_k$ *ou si* $j > i$ *et* $a_k = b_k$ *pour tout* $k \leq i$.

Que $f$ et les $<_m$ réalisent l'isomorphisme souhaité se montrerait facilement par récurrence sur $m$ en utilisant simplement leur définition et l'unicité à l'ordre fixé des décompositions proposées des entiers.

On se donne pour la suite deux fonctions récursives primitives $\nu(x)$ et $\theta(x)$ telles que $\nu(2^a(2b+1) - 1) = a$ et $\theta(2^a(2b+1) - 1) = b$.

On peut alors définir une classe de fonctions spécifiques, les fonctions primitives récursives d'ordre fini, qui seront utilisées pour calculer la limite supérieure des valeurs des $\epsilon$-termes clos. Ces fonctions sont construites de manière analogue aux fonctions récursives primitives, moyennant une libéralisation du schéma de récurrence. Soit $n$ un entier quelconque, si $g, h, \phi$ sont des fonctions déjà définies et que $\phi$ satisfait la propriété $\phi(m) <_n m$ pour tout $m$, on peut définir une nouvelle fonction $f$ de la manière suivante :

$f(0, a) = g(0)$

$f(m, a) = h(a, m, f(\phi(m), a)$

On peut remarquer que le schéma de récurrence primitive d'ordre fini ne fait dépendre la valeur d'une fonction pour son argument que d'un nombre fini de valeurs précédemment calculées, ce qui le distingue d'une définition par récurrence sur les ordinaux faisant intervenir pour le calcul à la limite l'ensemble infini des valeurs précédentes, et explique son acceptabilité du point de vue finitiste qui est celui d'Ackermann. Nous allons voir ensuite que les fonctions récursives primitives d'ordre fini sont en fait des fonctions récursives.

### 1.4.2 Le bornage en fonction du nombre de substitutions

Etant donné l'ensemble de formules constituées par une preuve, on désignera à partir de maintenant par $m$ le degré maximal d'un terme dans la preuve, $e$ le nombre d'$\epsilon$-termes et $g$ le nombre de catégories.

On définit par récurrence une fonction $\phi(m, a)$ qui donne la valeur maximale des termes pour une substitution en fonction de la valeur maximale $a$ d'un substituant pour les $\epsilon$-termes clos (il importe ici de ne pas confondre valeur maximale d'un $\epsilon$-terme clos qui dépend directement des substituants aux catégories et valeur maximale d'un terme qui dépend ensuite des opérations éventuellement appliquées aux $\epsilon$-termes clos) :

**Definition 18** $\phi(0,a) = a$
$\phi(m+1,a) = [\phi(m,a)]^2 + 1$

Il faut ici se souvenir de ce que les seuls symboles de fonction sont $', d, +$ et $\times$. L'idée est qu'en un coup, on peut au plus soit ajouter un, si la valeur maximale des $\epsilon$-termes clos est $\leq 1$, avec $'$, soit multiplier le terme le plus grand par lui-même sinon.

On définit ensuite par récurrence une fonction $\omega(m,n)$ qui borne la valeur maximale des $\epsilon$-termes clos au bout de $n$ substitutions.

**Definition 19** $\omega(m,0) = \phi(m,0)$
$\omega(m,n+1) = \phi(m,\omega(m,n))$

Le cas de base découle de ce que la substitution initiale annule tous les substituants. L'étape de récurrence est justifiée par le fait que la valeur maximale d'un substituant pour $S_{n+1}$ est, telle que celle-ci a été définie, inférieure ou égale à la valeur maximale d'un terme à l'étape $S_n$.

### 1.4.3 Le bornage du nombre de substitutions

Reste donc à évaluer le nombre de substitutions. Cette évaluation repose essentiellement sur la définition par récurrence croisée de deux fonctions : la première $\tau(c,p,n,a)$ calcule une borne supérieure pour l'indice d'une $p+1$-section qui commence par une substitution $S_n$ et une $p$-section d'indice $a$; la seconde $\kappa(c,p,n,a)$ calcule une borne supérieure au sens de $<_p$ pour une $p$-série commençant en $S_n$ suivant une $p$-série d'indice $a$.

On suppose donnée une fonction récursive $\eta(a,p)$ telle que $\eta(2^{a_1} + ... + 2^{a_i}, p) = a_1$ où les $a_j$ constituent une suite strictement décroissante.

On commence par donner deux définitions préliminaires, pour une fonction $\psi(m,n,e)$ qui donne une borne pour le degré d'une substitution $S_n$ et pour une fonction $\lambda(a,p)$ qui calcule le nombre de substitutions dont est composée une $p$-série d'indice $a$.

**Definition 20** $\psi(m,n,e) = 2^{(\omega(m,n)+1)e}$

En effet, le degré est l'ordre relativement à un ensemble de formules B(0,$z_1$...$z_n$), ...B(z,$z_1$...$z_n$). Or celles-ci comporte au plus $(z+1)e$ termes, et $z \leq \omega(m,n+1)$ puisque $z$ est la valeur d'un terme de $S_n$.

**Definition 21** $\lambda(a,1) = 1$
$\lambda(2^{a_1} + ... + 2^{a_m}, p+1) = \lambda(a_1,p) + ... + \lambda(a_m,p)$

La définition est cette fois immédiatement claire.
On définit ensuite $\kappa(c,p,n,a)$ par cas

**Definition 22** *a)* $\kappa(c,p,n,0) = 0$. *Si* $a \neq 0$, *on distingue*
*b) si* $p = 1$, $\theta(a) \neq 0$, $\kappa(c,p,n,a) = 2^{\nu(a)}(2 \times \vartheta(a) - 1) - 1$.
*c) si* $p = 1$, $\theta(a) = 0$, $\kappa(c,p,n,a) = 2^{\nu(a)-1}(2 \times c + 1) - 1$

*d) si $p > 1$ et $a$ est pair, $\kappa(c, p, n, a) = a - 1$*
*e) si $p > 1$ et $a$ est impair, et $a = 2^{a_1} - 1$,*
$\kappa(c, p, n, a) = \tau(c, p - 1, n, \kappa(c, p - 1, n, a_1))$
*f) si $p > 1$ et $a$ est impair, et $a = 2^{a_1} + 2^{a_2} + ... + 2^{a_l}$ où les $a_i$ sont rangés par ordre décroissant*
$\kappa(c, p, n, a) = 2^{a_1} + \kappa(c, p, n + \lambda(a_1, p - 1), 2^{a_2} + ... + 2^{a_l} - 1)$

Vérifions informellement que $\kappa$ calcule bien ce qu'elle doit calculer. Le cas a) est évident. Les définitions pour b) et c) majorent bien la 1-série suivant une 1-série d'indice $a$. Pour le cas b), ceci découle directement du fait que $\kappa(c, p, n, a)$ est le $<_1$ prédécesseur immédiat de $a$ par construction. Pour le cas c), la valeur donnée à c sera $\psi(m, n, e)$, en effet le degré de la substitution suivante sera $\leq \psi(m, n, e)$ comme la valeur maximale possible pour un terme n'a pas augmenté d'une substitution à l'autre. d) sert juste à assurer que $\kappa(c, p, n, a) <_p a$ pour le bon fonctionnement de la récurrence; en pratique cela n'intervient pas, car la seule substitution d'indice 0 est la substitution initiale, et que par conséquent la seule $p$-série d'indice 0 est la 1-série qu'elle constitue, de sorte que les indices des autres sections sont toujours impaires. e) correspond au cas où la $p$-série P est composée d'une unique $p - 1$-série M. Par conséquent, la $p$-série suivant P commence par une $p - 1$-série qui succède immédiatement à M, de sorte qu'on peut l'évaluer à l'aide de la fonction $\tau$ à laquelle on fournit comme majorant de l'indice de M $\kappa(c, p - 1, n, a)$. f) correspond à la situation où P est composée de $l$ $p - 1$-séries, et fait avancer le calcul en majorant au moyen de l'indice de la première de ces séries et de $\kappa$ appliquée à la somme des indices des $p - 1$-séries restantes, de sorte que la valeur de l'argument décroît comme il convient, tout en tenant compte du décalage de $S_n$ à $S_{n+\lambda(a_1,p-1)}$.

**Definition 23** $\tau(c, p, n, 0) = 0$
$\tau(c, p, n, a) = 2^a + \tau(c, p, n + \lambda(a, p), \kappa(c, p, n + \lambda(a, p), a))$

Le cas de base est évident. Pour l'étape de récurrence, $\tau$ doit calculer l'indice $i$ d'une $p + 1$-série P commençant par une $p$-série M d'indice $a$. On a $i = 2^a + i'$ ou $i'$ doit être l'indice correspondant au reste des $p$-séries qui composent P, vues comme une $p + 1$-série. C'est précisément ce que calcul $\tau$ si on l'applique à un majorant de l'indice de la $p$-série qui suit M, ce qui nous est précisément donné par $\kappa(c, p, n + \lambda(a, p), a)$, en tenant compte du décalage de $S_n$ à $S_{n+\lambda(a_1,p)}$. On remarque que la définition de $\tau$ fait intervenir essentiellement une récurrence d'ordre supérieur; on n'a en effet aucune raison d'avoir $\kappa(c, p, n + \lambda(a, p), a) <_0 a$. Par contre, pour la bonne définition de $\tau$, il faut maintenant vérifier que $\kappa(c, p, n, a) <_p a$ quand $a \neq 0$.

La démonstration se fait par récurrence sur $p$. On a vu que les cas a), b) et d) ne posaient pas de problème. Pour le cas c), il faut en fait voir la variable c comme un paramètre : $\kappa$ est bien définie pour des valeurs convenables de c, en particulier lorsque $c = \psi(m, n, e)$. Supposons que $\kappa(c, p, n, a) <_p a$ quand $a \neq 0$ et montrons la propriété pour $p + 1$. Par l'hypothèse de récurrence, $\tau$ est calculable pour des $p' \leq p$ et donc $\kappa$ pour des $p' \leq p + 1$. On démontre

ensuite un lemme par récurrence transfinie imbriquée dans la précédente que $\eta(\tau(c,p,n,a),p) = a$. Le cas $a = 0$ est immédiat. Supposons maintenant que la propriété tient pour des $a' <_p a$.

(1) $\kappa(c,p,n+\lambda(a,p),a) <_p a$ par hypothèse de récurrence

(2) $\eta(\tau(c,p,n+\lambda(a,p),\kappa(c,p,n+\lambda(a,p),a),p) = \kappa(c,p,n+\lambda(a,p),a)$ puisque (1) permet d'appliquer l'hypothèse de récurrence transfinie.

(3) $\eta(\tau(c,p,n+\lambda(a,p),\kappa(c,p,n+\lambda(a,p),a),p) <_p a$ par (1) +(2)

(4) si $\eta(b,p) <_p a$, alors $\eta(2^a + b,p) = a$ par définition de $\eta$.

(5) $\eta(2^a + \tau(c,p,n+\lambda(a,p),\kappa(c,p,n+\lambda(a,p),a),p) = a$ en appliquant (4) à (3) avec $b = \tau(c,p,n+\lambda(a,p),\kappa(c,p,n+\lambda(a,p),a)$

(6) $\eta(\tau(c,p+1,n,a),p+1) = a$ par définition de $\tau$.

Examinons maintenant le cas e). On veut $\kappa(c,p+1,n,a) <_{p+1} a$ où $a$ est $2^{a_1} - 1$.

(1) $\kappa(c,p+1,n,a) = \tau(c,n,\kappa(c,p,n,a_1))$ par définition de $\kappa$

(2) $\eta(\tau(c,p,n,\kappa(c,p,n,a_1),p) = \kappa(c,p,n,a_1)$ par le lemme

(3) $\kappa(c,p,n,a_1) <_p a_1$ par l'hypothèse de récurrence

(4) $\eta(a,p) = a_1$ par hypothèse sur $a$.

(5) $\eta(\kappa(c,p+1,n,a),p) <_p \eta(a,p)$ par (1), (2), (3) et (4)

(6) si $\eta(a,p) <_p \eta(b,p)$, alors $a <_{p+1} b$ par définition de $\eta$.

(7) $\kappa(c,p+1,n,a) <_{p+1} a$ en appliquant (6) à (5).

Reste le cas f).

? $\kappa(c,p+1,n,a) <_{p+1} 2^{a_1} + 2^{a_2} + ... + 2^{a_l} - 1$

? $2^{a_1} + \kappa(c,p+1,n+\lambda(a_1,p),2^{a_2} + ... + 2^{a_l} - 1) < 2^{a_1} + 2^{a_2} + ... + 2^{a_l} - 1$ par définition de $\kappa$

? $\kappa(c,p+1,n+m,2^{a_l} - 1) < 2^{a_l}$ par application répétée de la définition de $\kappa$, pour un certain $m$, ce qui nous ramène au cas e).

On se contente pour finir du schéma de la démonstration en omettant de démontrer deux propositions. Les démonstrations omises se font presque toutes sur le principe d'une induction sur $p$ où interviennent simultanément $\tau$ et $\kappa$, mais plusieurs étapes sont nécessaires avant de pouvoir démontrer les propriétés de $\lambda$ et $\tau$ dont on a besoin. On définit un prédicat T à quatre arguments qui s'appliquera à $(\psi(m,n,e),p,n,a)$ si $a$ est l'indice d'une $p$-série commençant par $S_n$ pour une preuve dont les constantes sont $m$ et $e$. On montre alors premièrement que le majorant donné par $\tau$ se comporte bien avec $\lambda$ et deuxièmement que $\tau$ et $\lambda$ ensemble se comportent bien vis-à-vis de $\leq_p$ .

**Proposition 24** *Si* $T(\psi(m,n,e),p+1,n,a)$, *alors*
$\lambda(a,p+1) \leq \lambda(\tau(\psi(m,n,e),p,n,\eta(a,p)),p+1)$

**Proposition 25** *Si* $a \leq_p b$ *et* $T(\psi(m,n,e),n,a)$, *alors*
$\lambda(\tau(\psi(m,n,e),p,n,a),p+1) \leq \lambda(\tau(\psi(m,n,e),p,n,b),p+1)$

Supposons donc qu'on connaisse les indices $a_1...a_t$ des $t$ $g$-séries qui composent l'unique $g+1$-séries que constitue la suite complète des substitutions, la proposition [ ] nous dit que

$\lambda(2^{a_1} + ... + 2^{a_t},g+1) \leq \lambda(\tau(\psi(m,n,e),g,1,a_1),g+1)$

L'évaluation du nombre de substitutions ne dépend alors plus de $t$.La seconde proposition nous dit qu'il suffit de trouver un majorant $b$ au sens de $\leq_g$ de $a_1$ pour avoir

$\lambda(2^{a_1} + ... + 2^{a_t}, g+1) \leq \lambda(\tau(\psi(m,n,e), g, 1, b), g+1)$

On définit enfin une fonction $\rho(n,e)$.

**Definition 26** $\rho(1,e) = 2^{e+1} - 1$

$\rho(n+1,e) = 2^{\rho(n,e)} - 1$

On voit que $\rho(1,e)$ majore au sens de $<_1$ l'indice d'une 1-série initiale, puisque $e$ est le nombre d'$\epsilon$-termes clos de la preuve et que si $\rho(n,e)$ majore au sens de $<_n$ l'indice d'une $n$-série initiale, alors $\rho(n+1,e)$ majore au sens de $<_{n+1}$ l'indice d'une $n+1$-série initiale. Donc $a_1 <_g \rho(g,e)$. On en déduit

$\lambda(2^{a_1} + ... + 2^{a_t}, g+1) \leq \lambda(\tau(\psi(m,n,e), g, 1, \rho(g,e)), g+1)$ où la seconde expression ne dépend plus des indices des séries de la suite des substitutions tels qu'on pourrait les calculer en appliquant effectivement la méthode de substitution. D'où le théorème final.

**Theorem 27** *Soit une preuve de l'arithmétique-epsilon comportant e $\epsilon$-termes correspondant à g catégories distinctes et telle que le degré des termes ne dépasse pas m, la valeur maximale des $\epsilon$-termes clos dans la substitution finale est* $born(m,e,g) = \omega(m, \lambda(\tau(\psi(m,n,e), g, 1, \rho(g,e)), g+1))$.

# 2 La No-counterexample interpretation

## 2.1 La notion d'interprétation

Kreisel définit une notion générale d'interprétation qui convient à un système déductif quelconque pour l'arithmétique. La spécificité de la démarche est d'exiger que l'interprétation du système se fasse à l'intérieur de celui-ci afin de contrôler que l'interprétation elle-même est bien finitiste. On se donne un codage récursif des formules du système.

**Definition 28** *Une formule $A[x_1...x_n, f_1...f_m]$ sans variables liées, et dont les variables libres sont parmi les variables d'individus $x_1...x_n$ et les variables de fonctions $f_1...f_m$ est vérifiable si, pour tous numéraux $a_1...a_n$ et toutes fonctions récursives $\phi_1...\phi_m$, $A[x_i := a_i, f_j := g_j]$ est vraie.*

**Definition 29** *On appelle alors interprétation d'un système $\Sigma$ une fonction calculable $f(n,a)$ telle que :*

*$\alpha$) $f(n,a)$ est le numéro d'une formule $A_n$ dont toutes les variables sont libres lorsque a est le numéro d'une formule A de $\Sigma$.*

*$\beta$) Si on a une preuve de A dans $\Sigma$, on peut trouver à partir de la preuve un n tel que $A_n$ est vérifiable.*

*$\gamma$) Si on a une preuve de $\sim A$ dans $\Sigma$, pour chaque n, on trouve une instanciation des variables libres de $A_n$ qui rend $A_n$ fausse.*

*$\delta$) Si on a une preuve de $A \rightarrow B$ dans $\Sigma$, on trouve une fonction calculable $g(n)$ telle que si $A_n$ est vérifiable $B_{g(n)}$ l'est aussi.*

Le but est bien de capturer le "sens finitiste" des preuves non-constructives, c'est-à-dire de passer de preuves d'énoncés quantifiés à des formules ayant un contenu numérique immédiat. Par exemple, si $\forall x Ax$ est une formule universelle, $Ax$ sera une bonne manière d'interpréter une preuve quelconque de $\forall x Ax$, car à partir de cette preuve, on peut tirer une preuve de $An$ pour n'importe quel numéral $n$. Mais on a vu dans l'introduction que l'interprétation ne saurait être aussi directe. Inversement, il est toujours possible de trivialiser la notion d'interprétation. Par exemple, si $e(a)$ est le code de $\sim A$ quand $a$ est le code de $A$ et si $Prov_\Sigma(x, y)$ est un prédicat de prouvabilité pour $\Sigma$, on peut montrer facilement que, sous hypothèse de la consistance de $\Sigma$, l'association à $A$ de $\sim Prov_\Sigma(x, e(a))$ constitue une interprétation pour $\Sigma$. On veut au contraire que l'interprétation d'une formule conserve autant que possible la *signification* de celle-ci.

La NCI est donnée pour une extension de $\mathrm{PA}_\epsilon$ dans laquelle on ajoute des symboles de variables de fonctions, que l'on ne quantifie pas. Présentons d'abord le principe de la NCI, avant de voir en quoi elle peut constituer une interprétation de $\mathrm{PA}_\epsilon$. Soit $A$ une formule en forme prénexe

$\forall x_1 \exists y_1 ... \forall x_n \exists y_n A'(x_1...x_n, y_1...y_n)$

où $A'$ est sans quantificateur. L'interprétation naïve, dont on a vu qu'elle ne marchait pas, consistait à chercher des fonctions calculables $\phi_1...\phi_n$ telles que

$A'(x_1...x_n, \phi_1(x_1)...\phi_n(x_1...x_n))$

soit vérifiable. Dans les termes de Kreisel, ceci constituerait une *Erfüllung* de $A$. Considérons ce que serait une *Erfüllung* de $\sim A$. $\sim A$ elle-même est équivalente à :

$\exists x_1 \forall y_1 ... \exists x_n \forall y_n \sim A'(x_1...x_n, y_1...y_n)$

Donc l'*Erfüllung* serait donnée par des fonctions calculables $\psi_1...\psi_n$ (où $\psi_1$ est une fonction 0-aire) telles que

$\sim A'(\psi_1...\psi_n(y_1...y_{n-1}), y_1...y_n)$

soit vérifiable. Ceci constituerait un contre-exemple à $A$ au sens où les fonctions donnent pour toute instanciation $b_1...b_n$ des $y_1...y_n$ des valeurs $\psi_1...\psi_n(b_1...b_{n-1})$ qui montrent que les $b_1...b_n$ ne sont pas de bons témoins pour les existentiels de la formule de départ. Faute de pouvoir obtenir les fonctions calculables $\phi_1...\phi_n$, l'idée est de montrer qu'il n'y a pas d'*Erfüllung* de $\sim A$, autrement dit pas de contre-exemple à $A$. Pour chaque $\psi_1...\psi_n$, on veut donc trouver des $b_1...b_n$ tels que $A'(\psi_1...\psi_n(b_1...b_{n-1}), b_1...b_n)$. Ceci revient à exhiber des fonctionnels $\chi_1 \cdots \chi_n$ dont les variables libres sont parmi les variables de fonctions $f_1...f_n$ telles que

$A'(f_1...f_n(\chi_1 \cdots \chi_{n-1}), \chi_1 \cdots \chi_n)$

soit vérifiable.

Si on arrive à passer d'une preuve de $A$ à l'existence d'une suite de tels fonctionnels, ceci nous donnera bien une interprétation, moyennant une énumération adéquate des fonctionnels en question ($A_m$ sera alors la formule $A'(f_1...f_n(\chi_1 \cdots \chi_{n-1}), \chi_1 \cdots \chi_n)$ où $\chi_1 \cdots \chi_n$ est la $m$-ième suite de fonctionnels) et, *last but not least*, la vérification des conditions $\gamma$) et $\delta$). L'interprétation se fera donc dans un formalisme comportant des variables libres de fonctions.

## 2.2 Vérification de la condition $\beta$)

Voyons comment la preuve d'Ackermann nous permet de vérifier relativement facilement la condition $\beta$). On commence par donner l'idée générale qui est de se servir de symboles de variables de fonctions pour paramétrer la preuve d'Ackermann et le bornage qui en résulte. On part de

$\vdash_{PA_\epsilon} \forall x_1 \exists y_1 ... \forall x_n \exists y_n A'(x_1...x_n, y_1...y_n)$

Il en découle

$\vdash_{PA'_\epsilon} \exists y_1 ... \exists y_n A'(f_1...f_n(y_1...y_{n-1}), y_1...y_n)$

où $PA'_\epsilon$ ne diffère de $PA_\epsilon$ que par l'adjonction aux langages de symboles de variables libres de fonctions comme $f_1...f_n$. En définissant une suite adéquate d'$\epsilon$-termes

$e_{y_n} = \epsilon_{y_n} A'(f_1...f_n(y_1...y_{n-1}), y_1...y_n)$

$e_{y_{k-1}} = \epsilon_{y_{k-1}} A'(f_1...f_n(y_1...y_{k-1} e_{k...} e_{n-1}), y_1...y_{k-1} e_{k...} e_n)$

on obtient une formule équivalente à la précédente où les $\epsilon$-termes ont remplacé les quantificateurs existentiels.

$\vdash_{PA'_\epsilon} A'(f_1...f_n(e_1...e_{n-1}), e_1...e_n)$

Supposons un instant qu'on ait choisi d'instancier les $x_i$ par des fonctions récursives déterminées $\psi_1...\psi_n$, de sorte qu'on ait

$\vdash_{PA''_\epsilon} A'(\psi_1...\psi_n(e'_1...e'_{n-1}), e'_1...e'_n)$

où $e'_j = e_j[\overrightarrow{f} := \overrightarrow{\psi}]$ et $PA''_\epsilon$ ne diffère de $PA_\epsilon$ que par l'adjonction des symboles de fonctions $\psi_1...\psi_n$. La méthode de substitution va s'appliquer à $PA''_\epsilon$ comme à $PA_\epsilon$, de sorte qu'à partir de la preuve précédente, on obtient des numéraux $m_1...m_n$ pour lesquels

$A'(\psi_1...\psi_n(m_1....m_{n-1}), m_1...m_n)$ (*)

soit vrai. De plus, on peut donner à l'avance une limite supérieure $m$ pour les $m_1...m_n$ en utilisant une fonction analogue à $born(m, e, g)$. $born(m, e, g)$ ne convient pas directement, car on a enrichi le langage de $PA_\epsilon$ avec de nouveaux symboles de fonctions, de sorte que la fonction $\phi(m, a)$ doit être remplacée par une fonction $\phi'(m, a)$ qui majore les valeurs des termes en tenant compte des nouvelles fonctions.

Il apparaît ainsi que $born(m, e, g)$ dépend des fonctions qu'on substitue aux symboles de variables de fonctions. L'idée de Kreisel est alors de capturer précisément cette dépendance en remplaçant $born(m, e, g)$ par une fonctionnelle $born'[f_1...f_n]$ (les paramètres $m, e, g$ étant fournis par la preuve) comportant comme variables libres de fonctions les $f_1...f_n$. Pour coller exactement avec la notion d'interprétation qui a été définie, on pourra alors exprimer l'interprétation d'une formule au moyen d'un schéma de minimisation.

La démonstration de $\beta$) est pour ainsi dire faite. Mais pour faire les choses convenablement, il faut définir la classe de fonctionnels. Avant cela, on commence par définir des avatars $\phi'_{\overrightarrow{x}}$ et $\omega'_{m, \overrightarrow{x}}$ des fonctions $\phi$ et $\omega$ avec les mêmes significations intuitives. Le paramètre $m$ correspond toujours au degré maximal des termes. On désigne par $g_j$, $j \leq l$, les $l$ symboles de fonctions - constantes ou variables - $k_j$-aires qui apparaissent dans la preuve.

**Definition 30** $\phi'_{\overrightarrow{x}}[g_j, a, 1] = max_{j \leq l, a_1...a_{k_j} \leq a}(g_j(a_1...a_{k_j}))$

$\phi'_{\overrightarrow{x}}[g_j, a, n+1] = \phi'_{\overrightarrow{x}}[g_j, \phi'_{\overrightarrow{x}}[g_j, a, n], 1],$
$\omega'_{m,\overrightarrow{x}}(0) = \phi'_{\overrightarrow{x}}[g_j, 0, m]$
$\omega'_{m,\overrightarrow{x}}(n+1) = \phi'_{\overrightarrow{x}}[g_j, \omega'_{m,\overrightarrow{x}}(n), m]$

$max_{j\leq l, a_1 ... a_{k_j} \leq a}(g_j(a_1 ... a_{k_j}))$ est représentable dans $\mathrm{PA}_\epsilon$ par $\epsilon_x(\forall \overrightarrow{y} \leq a, g_1(\overrightarrow{y}) \leq x \wedge ... \wedge g_l(\overrightarrow{y}) \leq x)$, donc $\phi'_{\overrightarrow{x}}$ définie par récurrence à partir de cette fonction également, et $\omega'_{m,\overrightarrow{x}}$ du même coup.

On modifie la définition de $\psi$ en $\psi'$en remplaçant $\omega$ par $\omega'_{m,\overrightarrow{x}}$

**Definition 31** $\psi'_m(n, e) = 2^{(\omega'_{m,\overrightarrow{x}}(n)+1)e}$

**Definition 32** *L'ensemble des fonctionnelles primitives récursives d'ordre fini contenant les variables d'individus $a_i$ et les variables de fonctions $f_j$ est le plus petit ensemble E tel que*

*a) les fonctions primitives récursives d'ordre fini de variables $a_i$ sont dans E*

*b) si $g(x, y)$ est une fonction primitive récursive d'ordre fini et $\chi \in E$, $g(a_i, \chi) \in E$.*

*c) si $\chi \in E$, $\omega'_{m,\overrightarrow{x}}(\chi)$ est dans E.*

*d) si $\chi_1 ... \chi_n \in E$ et $f_j$ est n-aire, $f_j$ appliquée n-éléments parmi $\{\chi_1 ... \chi_n, a_1 ... a_n\}$est dans E.*

*e) si $\chi \in E$ et $A(x, ...a_i..., ...f_j...)$ est une formule sans quantificateur dont les variables libres sont $x$ et des variables parmi les $a_i, f_j$ alors $\epsilon_x x \leq \chi \wedge A(x, ...a_i..., ...f_j...) \in E$.*

La définition de Kreisel [9] est corrigée conformément à Kreisel [10] : l'ajout d'une clause assurant la clôture par minimisation bornée est nécessaire pour rejoindre la définition précise d'une interprétation. Afin d'éviter dans un premier temps les codages qui compliqueraient l'établissement du résultat (en obligeant à tenir compte des fonctions de codage pour l'accroissement des valeurs des termes), on a en outre modifié la définition de manière à y faire entrer des fonctionnelles à plusieurs variables de fonctions d'arités quelconques. On peut maintenant énoncer le théorème.

**Theorem 33** *Si on a une preuve de $\forall x_1 \exists y_1 ... \forall x_n \exists y_n A'(x_1 ... x_n, y_1 ... y_n)$ dans $PA_\epsilon$, alors il existe des fonctionnelles récursives d'ordre fini $\eta_1 ... \eta_n$ dont les variables libres sont parmi $f_1 ... f_n$ telles que $A'(f_1 ... f_n(\eta_1 ... \eta_{n-1}), \eta_1 ... \eta_n)$ est vérifiable.*

D'une part, on observe que les fonctionnelles qui correspondent aux fonctions du théorème de bornage sont de l'espèce voulue $\omega'_{m,\overrightarrow{x}}(n)$ est une fonctionnelle récursive primitive, donc $\psi'_m(n, e)$ aussi. $\lambda$ et $\rho$ sont des fonctions primitives récursives, $\tau$ et $\kappa$ sont des fonctions primitives récursives d'ordre fini, donc $\lambda(\tau(\psi'_m(n, e), g, 1, \rho(g, e)), g + 1)$ est une fonctionnelle primitive récursive d'ordre fini, donc $\omega'_{m,\overrightarrow{x}}(\lambda(\tau(\psi'_m(n, e), g, 1, \rho(g, e)), g + 1))$ également. On note $born'_{m,n,e}[f_1 ... f_n]$cette dernière fonctionnelle paramétrée par $m, n, e$.

D'autre part, on constate que le théorème de bornage reste valable (il suffit pour cela de vérifier qu'avec les changements effectués sur $\omega, \phi$ et $\psi$ la démonstration reste correcte). Etant donnée une preuve de l'arithmétique-epsilon dans le langage augmentée des fonctions récursives $\overrightarrow{\psi}$, comportant $e$ $\epsilon$-termes correspondant à $g$ catégories distinctes et telle que le degré des termes ne dépasse pas $m$, la valeur maximale des $\epsilon$-termes clos dans la substitution finale est $born'_{m,n,e}[f_1 := \psi_1 ... f_n := \psi_n]$.

Soient $\psi_1 ... \psi_n$ des fonctions récursives susceptibles d'instancier les variables de fonctions $f_1 ... f_n$, on obtient à partir d'une preuve $\pi$ de $A$ une preuve $\pi'(m, n, e)$ de $A'(\psi_1 ... \psi_n(e'_1 ... e'_{n-1}), e'_1 ... e'_n)$. On peut choisir les $\pi'$ de manière à ce que les triplets $(m, n, e)$ soient les mêmes, indépendamment du choix des $\psi_1 ... \psi_n$ (ils dépendent seulement des constantes de la preuve $\pi$). On se donne un codage $\alpha_n$ des n-uplets et des fonctions $p_n^1 ... p_n^n$ de décodage, tout cela récursif primitif. On note $\beta$ fonctionnelle $\alpha_n(born'_{m,n,e}, ..., born'_{m,n,e})$. On définit alors $\eta_i = p_n^i(\epsilon_z(z \leq \beta \wedge A'(f_1 ... f_n(p_n^1(z) ... p_n^n(z)), p_n^1(z) ... p_n^n(z)))$. Les $\eta_i$ sont bien à leur tour des fonctionnelles récursives primitives d'ordre fini . Le théorème 15 et le théorème de bornage adapté nous garantissent l'existence de $m_1 ... m_n \leq born'_{m,n,e}[f_1 := \psi_1 ... f_n := \psi_n]$ pour lesquels $A'(\psi_1 ... \psi_n(m_1 .... m_{n-1}), m_1 ... m_n)$ est vraie.

Par conséquent, $A'(f_1 ... f_n(\eta_1 ... \eta_{n-1}), \eta_1 ... \eta_n)$ est vérifiable.

## 2.3 Modularité de l'interprétation

Les conditions $\gamma$) et $\delta$) reviennent à faire peser des conditions de modularité sur l'interprétation proposée. On veut pouvoir induire de ce qu'il existe une interprétation pour $\sim$A qu'il n'existe pas d'interprétation pour A; et on veut passer d'une interprétation pour A à une interprétation pour B lorsqu'on possède une preuve de $A \rightarrow B$. Ces conditions supplémentaires sont non triviales (voir [7]); elles distinguent la NCI de Kreisel des versions que l'on obtient via traduction de Gödel et interprétation fonctionnelle (c'est-à-dire les versions tirées de [4]).

Voyons d'abord où se loge la difficulté dans la preuve de $\gamma$). L'idée de la preuve est simple : réappliquer la méthode de substitution à une variante de $\sim A'$ contenant les fonctionnelles à réfuter ce qui nous fournira parmi les substituants finaux les fonctions récursives recherchées. Mais la réalisation de l'idée est un peu délicate. Supposons qu'on ait une preuve dans $PA_\epsilon$ de $\sim A$ où $A$ est $\forall x_1 \exists y_1 ... \forall x_n \exists y_n A'(x_1 ... x_n, y_1 ... y_n)$ et que les fonctionnelles à réfuter $\eta_1 ... \eta_n$ dont les variables soient représentables dans une extension de $PA_\epsilon$ par des $\epsilon$-termes $t_1 ... t_n$ (la notion de représentation est la même que pour les fonctions, cela signifie que l'on peut prouver $\eta_i = t_i$).

On définit une suite d'$\epsilon$-termes $e_1 ... e_n$ par :

$e_1 = \epsilon_{x_1} \forall y_1 ... \exists x_n \forall y_n \sim A'(x_1 ... x_n, y_1 ... y_n)$

$e_{k+1} = \epsilon_{x_{k+1}} \forall y_{k+1} ... \exists x_n \forall y_n \sim A'(e_1 ... e_k(a_1 ... a_{k-1}), x_{k+1} ... x_n, a_1 ... a_k, y_{k+1} ... y_n)$

Il est clair qu'à partir d'une preuve de $\sim A$ on peut obtenir une preuve de $\sim A'(e_1 ... e_n(a_1 ... a_{n-1}), a_1 ... a_n)$, puisque cette dernière formule est équivalent à

$\exists x_1...\exists x_n A'(x_1...x_n, a_1...a_n)$. Si on remplace les $a_i$ par des $t'_i$ avec $t'_i \equiv t_i[f_1 := e_1...f_n := e_n]$, on obtient une preuve de $\sim A'(e_1...e_n(t'_1...t'_{n-1}), t'_1...t'_n)$. En appliquant la méthode de substitution (qui reste valable quelles que soient les fonctions calculables avec lesquelles on a enrichi le langage - maintenant on ne se préoccupe plus de bornage), on obtient bien des fonctions $g_1...g_n$ à substituer aux $e_1...e_n$ telles que le résultat de $\sim A'(e_1...e_n(t'_1...t'_{n-1}), t'_1...t'_n)$ soit une formule vraie de l'arithmétique.

Pourtant ceci ne nous donne pas le résultat recherché; en effet, le fait que $t_n$ représente $\eta_n$ ne garantit absolument pas que l'on ait $t_n(e_1..e_n) = \eta_n(g_1...g_n)$. Par exemple, le terme $\epsilon_x(x = y + 3)$ est une fonction de $y$ qui représente la fonction récursive $y+3$, mais si $\epsilon_x(x = y+3)$ apparaît dans une preuve, tout ce que nous dit le théorème 15, c'est que toutes les formules dans lequel il apparaît donnent des formules vraies, mais il se peut très bien que ce soit le cas sans que pour autant le substituant final de $\epsilon_x(x = y + 3)$ soit la fonction récursive $+\mathbf{3}$, et on peut même affirmer que ce ne sera pas le cas, car les fonctions qui jouent le rôle de substituants sont toujours nulles sauf pour un nombre fini de valeur. Par contre, comme $\epsilon_x(x = y+3)$ représente $+\mathbf{3}$, il est possible de forcer le résultat pour certaines valeurs. Il suffit d'ajouter à l'ensemble de formules de la preuve des formules comme $\epsilon_x(x = t + 3) = +\mathbf{3}(t)$ (qu'on dérive de $\forall z(\epsilon_x(x = y+3) = +\mathbf{3}(y))$ ce qui force $\epsilon_x(x = y+3)$ à se comporter comme $+\mathbf{3}$ pour la valeur finale de $t$. Ceci est le principe de la solution à notre problème, même si la situation est un peu compliquée par le fait qu'on ait affaire à des termes représentant des fonctionnelles et pas simplement des fonctions.

On se donne deux lemmes qu'on démontrera plus tard afin de montrer le théorème voulu.

**Lemma 34** *Soit $\eta[a_1...a_n, f_1...f_n]$ une fonctionnelle récursive primitive d'ordre fini représentée par un terme $t[a_1...a_n, f_1...f_n]$ et $u_i[x_1...x_{m_i}]$ un terme quelconque substituable à $f_i$ d'arité $m_i$, il existe un terme $R[a_1...a_n]$ tel que pour tout terme $v[x_1...x_{m_i}]$, on peut prouver dans $PA_\epsilon$ :*

$$\left[\bigwedge_{i=0}^{n} \forall \overrightarrow{x}\,(\overrightarrow{x} \leq R[a_1...a_n] \rightarrow u_i[\overrightarrow{x}] = v_i[\overrightarrow{x}])\right] \rightarrow t[a_1...a_n, u_1...u_n] = t[a_1...a_n, v_1...v_n]$$

**Lemma 35** *Si $\varphi(n)$ est une fonction primitive récursive d'ordre fini, on peut trouver un terme $t(n)$ tel que l'on peut prouver dans $PA_\epsilon$ que $t(n)$ satisfait les relations récursives qui définissent $\varphi(n)$.*

**Lemma 36** *On peut trouver des fonctions récursives $h'_1...h'_n$ paramétrées par des termes $u^1_i, v^1_i$ sans variables libres et des termes $u^j_i, v^j_i$ contenant $N$ comme seule variable libre tels que les égalités suivantes soient prouvables dans $PA_\epsilon$ :*

$e_1 = h'_1(u^1_1...u^1_n, v^1_1...v^1_n)$

$(y_1 \leq N \wedge ... \wedge y_k \leq N) \rightarrow e_{k+1} = h'_{k+1}(y_1...y_k, u^1_1...u^{k+1}_n, v^1_1...v^{k+1}_n)$

On a maintenant les outils pour démontrer

**Theorem 37** *Si on a une preuve de $\sim A$ dans $PA_\epsilon$, alors pour toutes fonctionnelles primitives récursives d'ordre fini $\eta_1...\eta_n$ dont les variables sont parmi $f_1...f_n$, on trouve des fonctions récursives $g_1...g_n$ telles que*

$\sim A'(g_1...g_n(\overline{\eta}_1...\overline{\eta}_{n-1}), \overline{\eta}_1...\overline{\eta}_n)$ *où* $\overline{\eta}_i = \eta_i[f_1 := g_1, ..., f_n := g_n]$

La démonstration commence comme indiqué plus haut. Comme par le lemme 35, les fonctions récursives primitives d'ordre fini sont représentables, il suit directement de la définition des fonctionnelles primitives récursives d'ordre fini que celles-ci aussi sont représentables. Soient donc $t_1...t_n$ les termes qui les représentent. A partir d'une preuve de $\sim A$, on obtient une preuve de

$\sim A'(e_1...e_n(t'_1...t'_{n-1}), t'_1...t'_n)$ (0)

où les $e_i$ et les $t'_i$ sont définis comme précédemment.

On considère les $h'_i$ du lemme 36 et on pose $\eta'_i = \eta_i[f_1 := g_1, ..., f_n := g_n]$. Le lemme 34 nous dit que si les $\epsilon$-termes $e_i$ et les fonctions récursives $h'_i$ sont suffisamment égaux (au sens de $R$ indépendant des paramètres de $h'_i$), les termes représentant les fonctionnelles les confondent, ce qui implique que les $t'_i$ et les $\eta'_i$ donnent à leur tour le même résultat (puisque le terme $t_i$ auquel on a donné comme argument les $h'_i$ représente $\eta'_i$). Formellement, le lemme 34 nous donne un terme $R$ tel que

$$\left[\bigwedge_{i=0}^{n} \forall \overrightarrow{x}(\overrightarrow{x} \leq R \rightarrow e_i[\overrightarrow{x}] = h'_i[\overrightarrow{x}])\right] \rightarrow \left[\bigwedge_{i=0}^{n} t'_i = \eta'_i\right] \quad (1)$$

Les $t'_i$ ne comportent pas de variable d'individu libre : la seule variable libre d'individu $f_1$ est remplacé par le terme clos $e_1$; par conséquent, vu la manière dont $R$ est construit, il ne comporte pas de variable libre. Et les $t'_i$ n'ont plus de variable de fonction libre. Cette remarque est importante car la méthode de substitution s'applique seulement à des formules closes. L'introduction de variables de fonctions libres dans le système sert à la formulation des résultats, mais les preuves qui servent à leur établissement se font elles sans recourir à de telles variables.

On pose alors $R' = \overset{n}{\underset{i=1}{max}}(R, t'_i)$

Si on remplace $N$ par $R'$ dans le lemme 36, on des preuves dans $\mathrm{PA}_\epsilon$ des formules

$(y_1 \leq R' \wedge ... \wedge y_k \leq R') \rightarrow e_{k+1}(y_1...y_k) =$
$h'_{k+1}(y_1...y_k, u_1^1...u_n^{k+1}, v_1^1...v_n^{k+1})$ (2)

Comme $R_i \leq R$, (2) nous donne les membres de la conjonction qui est l'antécédent de (1) d'où

$\bigwedge_{i=0}^{n} t'_i = \eta'_i$ (3)

Dans (3) les $\eta'_i$ contiennent les $u_i^1, v_i^1$ et les $u_i^{m+1}, v_i^{m+1}$ comme paramètres

Comme $t'_i \leq R'$, (3) nous donne les égalités

$e_{k+1}(t'_1...t'_k) = h'_{k+1}(\eta'_1...\eta'_k, u_1^1...u_n^{k+1}, v_1^1...v_n^{k+1})$ (4)

On applique alors la méthode de substitution à l'ensemble de formules constituées par les preuves de (0), (3) et (4). Le résultat de (0) en particulier est une formule vraie. Mais on sait en plus que les substituants des $e_i$ coïncident avec des fonctions récursives pour les valeurs des fonctionnelles, et que la valeur pour ces $e_i$ des termes représentant les fonctionnelles est égale à la valeur des fonctionnelles pour ces fonctions récursives. Par conséquent, si on appelle $g_1...g_n$ les substituants des $e_1...e_n$, (0) nous donne bien :

$\sim A'(g_1...g_n(\overline{\eta}_1...\overline{\eta}_{n-1}), \overline{\eta}_1...\overline{\eta}_n)$

**Theorem 38** *Soient deux formules* $A \equiv \forall x_1 \exists y_1 ... \forall x_n \exists y_n A'(x_1...x_n, y_1...y_n)$
*et* $B \equiv \forall x_1 \exists y_1 ... \forall x_m \exists y_m B'(x_1...x_m, y_1...y_m)$,
*une preuve* $\pi$ *dans* $PA_\epsilon$ *de* $A \to B$
*et* $\eta_1...\eta_n$ *des fonctionnelles dont les variables libres sont parmi* $f_1...f_n$ *telles que* $A'(f_1...f_n(\eta_1...\eta_{n-1}), \eta_1...\eta_n)$ *est vérifiable,*
*on peut trouver des fonctionnelles* $\xi_1...\xi_m$ *dont les variables libres sont parmi* $g_1...g_m$ *telles que* $B'(g_1...g_m(\xi_1...\xi_{m-1}), \xi_1...\xi_m)$ *est vérifiable.*

La démonstration suit de près celle du théorème précédent, on garde les notations utilisées pour les termes convoqués dans sa démonstration.

Soient $\psi_1...\psi_m$ des fonctions récursives susceptibles d'instancier les $g_1...g_m$.

D'une part, on a vu que $\sim A \to \sim A'(e_1...e_n(t'_1...t'_{n-1}), t'_1...t'_n)$ était prouvable dans $PA_\epsilon$, donc par contrapposition, on a aussi une preuve de $A'(e_1...e_n(t'_1...t'_{n-1}), t'_1...t'_n) \to A$. D'autre part, comme dans la démonstration du théorème 33 $B \to B'(\psi_1...\psi_m(\epsilon_1...\epsilon_{m-1}), \epsilon_1...\epsilon_m)$ pour certains $\epsilon$-termes $\epsilon_1...\epsilon_m$. Combiné avec la preuve $\pi$ qui est donnée par l'hypothèse, ceci nous donne une preuve de :

$A'(e_1...e_n(t'_1...t'_{n-1}), t'_1...t'_n) \to B'(\psi_1...\psi_m(\epsilon_1...\epsilon_{m-1}), \epsilon_1...\epsilon_m)$ (0)

On peut ajouter aux formules de la preuve de (0) les formules des preuves de (3) et (4) du théorème précédent. Ceci nous assure que dans la substitution finale, le résultat de $A'(e_1...e_n(t'_1...t'_{n-1}), t'_1...t'_n)$ est une formule vraie. Par conséquent, le résultat de $B'(\psi_1...\psi_m(\epsilon_1...\epsilon_{m-1}), \epsilon_1...\epsilon_m)$ est également une formule vraie. Les valeurs des $\epsilon_1...\epsilon_m$ sont alors bornées par $born'_{m,n,e}[f_1 := \psi_1...f_n := \psi_n]$ et $m, n, e$ ne dépendent pas du choix de $\psi_1...\psi_m$ (moyennant le choix d'une preuve standard pour $B \to B'(\psi_1...\psi_m(\epsilon_1...\epsilon_{m-1}), \epsilon_1...\epsilon_m)$). Il suit que les $\xi_i$ définis par $\xi_i = p^i_n(\epsilon_z(z \leq \beta \wedge B'(g_1...g_m(p^1_m(z)...p^m_m(z)), p^1_m(z)...p^m_m(z)))$ où $\beta$ est la fonctionnelle $\alpha_m(born'_{m,n,e}, ..., born'_{m,n,e})$ conviennent.

## 2.4 Retour sur les lemmes

La démonstration du lemme 34 se fait simplement par induction sur la forme de la fonctionnelle. On indique comment construire $R$ pour chaque étape :

a) max $(a_i)$
b) max $(a_i, \chi, R_\chi)$
c) max $(t, R\chi)$ où $t$ représente $\omega'_{m,\overrightarrow{x}}(\chi)$ appliquée aux $u_i$.
d) max $(a_i, R_{\chi_j})$
e) max $(R\chi)$

La démonstration du lemme 35 se trouve dans Kreisel [9]. Intuitivement, la validité du lemme est claire : les valeurs des fonctions primitives récursives d'ordre fini sont calculables en un nombre fini d'etapes; si toutes les fonctions calculables sont récursives (thèse de Church), comme toutes les fonctions récursives sont représentables dans $AP_\epsilon$ (voir [5] pour une démonstration de ceci) celles-ci le sont également. On remarque que la notion de représentation ici utilisée est spécifique à l'arithmétique epsilon : les fonctions sont représentées par des $\epsilon$-termes et pas par des formules.

Nous donnons enfin la démonstration du lemme 36, un peu difficile à suivre quoiqu'assez simple, en suivant scrupuleusement celle de Kreisel [8].

On peut trouver des fonctions récursives $h'_1...h'_n$ paramétrées par des termes $u_i^1, v_i^1$ sans variables libres et des termes $u_i^{m+1}, v_i^{m+1}$ contenant $N$ comme seule variable libre tels que

$e_1 = h'_1(u_1^1...u_n^1, v_1^1...v_n^1)$

$(y_1 \leq N \wedge ... \wedge y_k \leq N) \rightarrow e_{k+1} = h'_{k+1}(y_1...y_k, u_{k+1}^{k+1}...u_n^{k+1}, v_{k+1}^{k+1}...v_n^{k+1})$

On commence par définir les termes primitifs récursifs (toutes les minimisations et les quantifications sont bornées) en jeu. Les $a_j^i, b_j^i$ sont des variables libres

$h'_1(a_1^1...a_n^1, b_1^1...b_n^1) = \epsilon_{x_1} \forall y_1 ... \exists x_n \forall y_n$

$\left[y_1 \leq b_1^1 \wedge ... \wedge y_n \leq b_n^1 \rightarrow x_1 \leq a_1^1 \wedge ... \wedge x_n \leq a_n^1 \sim A'(x_1...x_n, y_1...y_n)\right]$

$h_{r+1}(y_1...y_r, c_1...c_r, a_{r+1}^{r+1}...a_n^{r+1}, b_{r+1}^{r+1}...b_n^{r+1}) = \epsilon_{x_{r+1}} \forall y_{r+1} ... \exists x_n \forall y_n$

$\left[y_{r+1} \leq b_{r+1}^{r+1} \wedge ... \wedge y_n \leq b_n^{r+1} \rightarrow x_{r+1} \leq a_{r+1}^{r+1} \wedge ... \wedge x_n \leq a_n^{r+1} \sim A'(c_1...c_r x_{r+1}...x_n, y_1...y_n)\right]$

$h'_{r+1}(y_1...y_r, a_1^1...a_n^{r+1}, b_1^1...b_n^{r+1}) = h_{r+1}(y_1...y_r, h'_1...h'_r, a_{r+1}^{r+1}...a_n^{r+1}, b_{r+1}^{r+1}...b_n^{r+1})$

Afin de surmonter la complexité de la syntaxe et des doubles indices, on commence par donner la méthode pour trouver les $u_j^i, v_j^i$ à partir d'un exemple. On considère la formule $\exists x \forall y B(a, x, y)$. $a$ est une variable libre qui sert de paramètre et $B$ peut comporter d'autres variables liées. On considère les termes :

$\epsilon_z \forall v \left[v \leq N \rightarrow \epsilon_x \left(\forall y B(v, x, y)\right) \leq z\right]$ qu'on note $u$

$\epsilon_z \forall v, w \left[v \leq N \wedge w \leq u \rightarrow \epsilon_y \left(\sim B(v, w, y)\right) \leq z\right]$ qu'on note $v$

Intuitivement $u$ et $v$ sont deux des $u_j^i, v_j^i$ qu'on cherche, c'est-à-dire des paramètres qui, relativement à un $N$ quelconque permettent de borner les quantifications. C'est ce que veulent dire les deux propositions suivantes :

**Proposition 39** $a \leq N \rightarrow \epsilon_x \left(\forall y B(a, x, y)\right) = \epsilon_x \forall y \left[y \leq v \rightarrow x \leq u \wedge B(a, x, y)\right]$

**Proposition 40** $a \leq N \rightarrow \exists x \forall y B(a, x, y) \longleftrightarrow \exists x \forall y \left[y \leq v \rightarrow x \leq u \wedge B(a, x, y)\right]$

En toute rigueur, on aurait besoin pour le lemme d'établir que ces propositions sont prouvables dans $\mathrm{PA}_\epsilon$. On se contente ici de les prouver informellement, la formalisation, fastidieuse, ne posant pas de problème particulier.

$\epsilon_x \left(\forall y B(a, x, y)\right)$ est noté $\lambda(a)$

$\epsilon_x \forall y \left[y \leq v \rightarrow x \leq u \wedge B(a, x, y)\right]$ est noté $\sigma(a)$

$\epsilon_y \sim B(a, c, y)$ est noté $\rho(a, c)$

Premier cas : $\exists x \forall y B(a, x, y)$ est vrai pour tout $a \leq N$. $\forall y B(a, \lambda(a), y)$ est donc vrai aussi. Par ailleurs, quand $a$ parcourt [0;N], il existe un maximum $x$ des $\lambda(a)$ et $x \leq u$. Donc 40 est vrai. Mais est-ce que $\epsilon_x \forall y \left[y \leq v \rightarrow x \leq u \wedge B(a, x, y)\right]$ pourrait être strictement inférieur à cet $x$ ? Soit un $c$ qui pour un certain $a$ est strictement inférieur à $\lambda(a)$, il y a donc un $y \in \{\rho(a, c) / a \leq N, c \leq \lambda(a)\}$ tel que $\sim B(a, c, y)$, c'est–à-dire un $y \leq v$ puisque $v$ majore cet ensemble. Autrement dit, $c$ ne peut être $\epsilon_x \forall y \left[y \leq v \rightarrow x \leq u \wedge B(a, x, y)\right]$. On a ainsi montré 39.

Deuxième cas : il existe un $a \leq N$ tel que $\forall x \exists y B(a, x, y)$. On a ainsi $\forall x \sim B(a, x, \rho(a, x))$. Pour $x \leq u$, $\rho(a, x) \leq v$ par définition de $v$. Il n'y a donc pas de $x \leq u$ tel que $\forall y \leq v \left(B(a, x, y)\right)$. Les deux membres de l'équivalence de 40 sont donc faux et les deux $\epsilon$-termes de 39 nuls.

On montre maintenant comment utiliser la méthode de l'exemple pour obtenir les $u_i^1, v_i^1$.

On commence par appliquer la méthode avec $B(a,x,y) \equiv \exists x_2 \forall y_2 ... \exists x_n \forall y_n \sim A'(x, x_2 ... x_n, y, y_2 ... y_n)$ pour avoir $u_1^1 \equiv u, v_1^1 \equiv v$. La proposition 39 nous donne alors

$\epsilon_{x_1} (\forall y_1 ... \exists x_n \forall y_n \sim A'(x_1 ... x_n, y_1 ... y_n))$

$= \epsilon_{x_1} \forall y_1 \left[ y_1 \leq v_1^1 \rightarrow x_1 \leq u_1^1 \wedge \exists x_2 \forall y_2 ... \exists x_n \forall y_n \sim A'(x_1, ... x_n, y_1 ... y_n) \right]$

Et on continue : on procède de même avec

$B(x_1, y_1, x_2, y_2) \equiv \exists x_2 \forall y_2 \left[ y_1 \leq v_1^1 \rightarrow x_1 \leq u_1^1 \wedge \exists x_3 \forall y_3 ... \exists x_n \forall y_n \sim A'(x_1, ... x_n, y_1 ... y_n) \right]$

pour obtenir $u_2^1 \equiv u, v_2^1 \equiv v$. Il faut préciser les $N$ qui bornent les $x_1, y_1$ qui jouent ici le rôle des paramètres $a$. On exige $x_1 \leq u_1^1$ et $y_1 \leq v_1^1$.

On doit alors prouver

$\epsilon_{x_1} (\forall y_1 ... \exists x_n \forall y_n \sim A'(x_1 ... x_n, y_1 ... y_n))$

$= \epsilon_{x_1} \forall y_1 \exists x_2 \forall_2 \left[ y_1 \leq v_1^1 \wedge y_2 \leq v_2^1 \rightarrow x_1 \leq u_1^1 \wedge x_2 \leq u_2^1 \wedge \exists x_2 \forall y_2 ... \exists x_n \forall y_n \sim A'(x_1, ... x_n, y_1 ... y_n) \right]$

Pour cela, il suffit d'avoir

$\epsilon_{x_1} \forall y_1 \left[ y_1 \leq v_1^1 \rightarrow x_1 \leq u_1^1 \wedge \exists x_2 \forall y_2 ... \exists x_n \forall y_n \sim A'(x_1, ... x_n, y_1 ... y_n) \right]$

$= \epsilon_{x_1} \forall y_1 \exists x_2 \forall_2 \left[ y_1 \leq v_1^1 \wedge y_2 \leq v_2^1 \rightarrow x_1 \leq u_1^1 \wedge x_2 \leq u_2^1 \wedge \exists x_2 \forall y_2 ... \exists x_n \forall y_n \sim A'(x_1, ... x_n, y_1 ... y_n) \right]$

On peut alors se ramener à montrer dans $\mathrm{PA}_\epsilon$ l'équivalence

$\exists x_2 \forall y_2 \left[ y_1 \leq v_1^1 \rightarrow x_1 \leq u_1^1 \wedge \exists x_3 \forall y_3 ... \exists x_n \forall y_n \sim A'(x_1, ... x_n, y_1 ... y_n) \right]$

$\longleftrightarrow \exists x_2 \forall y_2 \left[ y_1 \leq v_1^1 \wedge y_2 \leq v_2^1 \rightarrow x_1 \leq u_1^1 \wedge x_2 \leq u_2^1 \wedge \exists x_2 \forall y_2 ... \exists x_n \forall y_n \sim A'(x_1, ... x_n, y_1 ... y_n) \right]$

On raisonne par cas : si $y_1 > v_1^1$, les deux formules sont vraies. Si $y_1 \leq v_1^1$ et $x_1 > u_1^1$ la première formule est fausse, et la seconde aussi (prendre $y_2 = 0$ pour s'assurer que l'antécédent est vrai). Si $y_1 \leq v_1^1$ et $x_1 \leq u_1^1$, l'antécédent du lemme 40 est vrai ce qui nous donne

$\exists x_2 \forall y_2 \exists x_3 \forall y_3 ... \exists x_n \forall y_n \sim A'(x_1, ... x_n, y_1 ... y_n)$

$\longleftrightarrow \exists x_2 \forall y_2 \left[ y_2 \leq v_2^1 \rightarrow x_2 \leq u_2^1 \wedge \exists x_2 \forall y_2 ... \exists x_n \forall y_n \sim A'(x_1, ... x_n, y_1 ... y_n) \right]$

et l'équivalence du même coup.

On obtient les autres $u_i^1, v_i^1$ en répétant la procédure. On a ainsi un $h'_1(u_1^1 ... u_n^1, v_1^1 ... v_n^1)$ prouvablement égal à $e_1$.

On montre maintenant comment obtenir $h'_{r+1}(y_1 ... y_r, u_1^1 ... u_n^{r+1}, v_1^1 ... v_n^{r+1})$ tel que

$(y_1 \leq N \wedge ... \wedge y_k \leq N) \rightarrow e_{r+1} = h'_{r+1}(y_1 ... y_r, u_1^1 ... u_n^{r+1}, v_1^1 ... v_n^{r+1})$

en supposant qu'on a su le faire pour $h'_1 ... h'_r$.

On considère le terme $t \equiv \epsilon_{x_{r+1}} \forall y_{r+1} ... \exists x_n \forall y_n \sim A'(c_1 ... c_r x_{r+1} ... x_n, y_1 ... y_n)$ dont les variables sont $c_1 ... c_r, y_1 ... y_r$

On sait trouver des termes $u_{r+1} ... u_n$, $v_{r+1} ... v_r$ contenant les variables libres $c_1 ... c_r$ et $N$ tels que l'on peut prouver :

$(y_1 \leq N \wedge ... \wedge y_k \leq N) \rightarrow t = \epsilon_{x_{r+1}} \forall y_{r+1} ... \exists x_n \forall y_n$

$[y_{r+1} \leq v_{r+1} \wedge ... \wedge y_n \leq v_n \rightarrow x_{r+1} \leq u_{r+1} \wedge ... \wedge x_n \leq a_n \sim A'(c_1 ... c_r, x_{r+1} ... x_n, y_1 ... y_n)]$

On pose alors, pour $r+1 \leq j \leq n$, $u_j^{r+1} = u_j [c_1 := h'_1 ... c_r := h'_r]$ et de même $v_j^{r+1} = v_j [c_1 := h'_1 ... c_r := h'_r]$.

Dans l'égalité précédente, on remplace dans $t$ les $c_j$ par les $e_j$ ce qui nous donne $e_{r+1}$, et dans le membre de droite les $c_j$ par les $h'_j$. Cette substitution est licite car pour $y_1 \leq N \wedge ... \wedge y_k \leq N$, on sait que $e_j = h'_j$. Ceci nous donne alors :

$(y_1 \leq N \wedge ... \wedge y_k \leq N) \rightarrow e_{r+1} = h'_{r+1}(y_1 ... y_r, u_1^1 ... u_n^{r+1}, v_1^1 ... v_n^{r+1})$.

# 3 Réalisabilité classique et arithmétique

## 3.1 Le lambda-c calcul et les règles de typage

On définit l'ensemble $\Lambda_c$ des termes t et l'ensemble $\Pi$ des piles $\pi$ :

$t = cc, x, (t)t, \lambda x.t, k_\pi$

$\pi = \rho, t.\pi$

La réduction $\succ$ est alors définie sur $\Lambda_c \times \Pi$, l'ensemble des exécutables :

$(t)u \star \pi \succ t \star u \cdot \pi$ (push)

$\lambda x.t \star u \cdot \pi \succ t[u/x] \star \pi$ (pop)

$cc \star t \cdot \pi \succ t \star k_\pi \cdot \pi$ (store)

$k_\pi \star t \cdot \pi' \succ t \star \pi$ (restore)

Les règles de typage correspondent à la logique classique du second ordre. Les seuls symboles logiques utilisés sont $\rightarrow$ et $\forall$.

Soit $\Gamma$ un contexte de la forme $x_1 : A_1, ..., x_n : A_n$, les règles de typage sont :

1. $\Gamma \vdash x_i : A_i$ où $1 \leq i \leq n$
2. si $\Gamma \vdash t : A \rightarrow B$ et $\Gamma \vdash u : A$ alors $\Gamma \vdash tu : B$
3. si $\Gamma, x : A \vdash t : B$ alors $\Gamma \vdash \lambda x.t : A \rightarrow B$
4. si $\Gamma \vdash t : (A \rightarrow B) \rightarrow A$ alors $\Gamma \vdash cct : A$
5. si $\Gamma \vdash t : A$ alors $\Gamma \vdash t : \forall x A$ pour $x$ non libre dans $\Gamma$
6. si $\Gamma \vdash t : A$ alors $\Gamma \vdash t : \forall X A$ pour $X$ non libre dans $\Gamma$
7. si $\Gamma \vdash t : \forall x A$ alors $\Gamma \vdash t : A[\tau/x]$ pour tout terme $\tau$
8. si $\Gamma \vdash t : \forall X A$ alors $\Gamma \vdash t : A[\phi(x_1...x_n)/X x_1...x_n]$ pour toute formule $\phi(x_1...x_n)$ (axiome de compréhension pour la logique du second ordre)

Les autres connecteurs sont définis ensuite de la manière usuelle :

$\bot \equiv \forall X X$

$A \wedge B \equiv \forall X((A \rightarrow (B \rightarrow X)) \rightarrow X)$

$A \vee B \equiv \forall X(((A \rightarrow X) \rightarrow (B \rightarrow X)) \rightarrow X)$

$\exists X A \equiv \forall X(A \rightarrow \bot) \rightarrow \bot$

De plus $a = b \equiv \forall X(Xa \rightarrow Xb)$

## 3.2 La réalisabilité

L'idée de la réalisabilité est d'interpréter les formules F par des sous-ensembles $|F|$ de $\Lambda_c$. La détermination de $|F|$ se fait relativement à un modèle M de la logique classique du second ordre et à un ensemble $\amalg \sqsubseteq \Lambda_c \times \Pi$ de processus qui correspond aux processus observables. On montre des propriétés sur les termes qui sont dans $|F|$ (qui réalisent F) en choisissant des $\amalg$ particuliers. Ceci permet ensuite d'avoir ces propriétés pour les termes de type F, c'est-à-dire pour les termes qui codent des preuves de F moyennant un lemme d'adéquation qui nous assure que si $\vdash t : A$ alors $t \in |A|$ .

Un modèle $M$ est défini comme la donnée d'un ensemble M d'individus et d'une interprétation $f_M : M^k \mapsto M$ pour chaque symbole de fonction f k-aire du langage. Le domaine de variation des prédicats d'arité k est $P(\Pi)^{M^k}$. On exige que $\amalg$ soit clos par anti-réduction.

On définit la valeur de vérité d'une formule F par

$|A| = \|A\|^{\perp} = \{t \in \Lambda_c / \forall \pi_A \in \|A\|, t * \pi_A \in \Pi\}$

On définit maintenant $\|A\| \subseteq \Pi$ pour une formule $A$ à paramètre dans M par induction sur $A$.

Si $A$ est atomique $A \equiv Ra_1...a_k$ et $R \in P(\Pi)^{M^k}$, on pose $\|Ra_1...a_k\| = R(a_1...a_k)$

$\|A \to B\| = \{t.\pi / t \in |A|, \pi \in \|B\|\}$

$\|\forall x A\| = \bigcup_{a \in M} \|A[x := a]\|$

$\left\|\forall X^k A\right\| = \bigcup_{R \in P(\Pi)^{M^k}} \|A[X := R]\|$

Quelques valeurs sont distinguées. $\top = \| \emptyset \|^{\perp}$ est la plus grande valeur de vérité. $\perp = \| \Pi \|$, donc $|\perp|$ la plus petite valeur de vérité.

**Theorem 41** *Si $x_1 : F_1 ... x_n : F_n \vdash t : F$ et si $s_1 \in |F_1|, ..., s_n \in |F_n|$ alors $t[s_1/x_1 ... s_n/x_n] \in |F|$*

La preuve se fait par induction sur le typage. On examine les différents cas :

1. La dernière règle est : $\Gamma \vdash x_i : A_i$ où $1 \leq i \leq n$, c'est immédiat.

2. La dernière règle est : si $\Gamma \vdash t : A \to B$ et $\Gamma \vdash u : A$ alors $\Gamma \vdash tu : B$

$tu[\overrightarrow{s}/\overrightarrow{x}] = t[\overrightarrow{s}/\overrightarrow{x}]u[\overrightarrow{s}/\overrightarrow{x}]$ et l'hypothèse d'induction nous donne $t[\overrightarrow{s}/\overrightarrow{x}] \in A \to B$ et $u[\overrightarrow{s}/\overrightarrow{x}] \in |A|$. Donc $t[\overrightarrow{s}/\overrightarrow{x}] * u[\overrightarrow{s}/\overrightarrow{x}].\pi_B \in \Pi$ donc (clôture par anti-réduction de $\Pi$), $t[\overrightarrow{s}/\overrightarrow{x}]u[\overrightarrow{s}/\overrightarrow{x}] * \pi_B \in \Pi$, d'où $tu[\overrightarrow{s}/\overrightarrow{x}] \in |B|$

3. La dernière règle est : si $\Gamma, y : A \vdash t : B$ alors $\Gamma \vdash \lambda y.t : A \to B$.

On suppose que $y$ est une variable fraiche qui n'apparaît pas dans les $s_i$ et qui est différente des $x_i$. $(\lambda y.t)[\overrightarrow{s}/\overrightarrow{x}] = \lambda y.t[\overrightarrow{s}/\overrightarrow{x}]$. Soient $u \in |A|$, $\pi_B \in \|B\|$, on veut $\lambda y.t[\overrightarrow{s}/\overrightarrow{x}] * u.\pi_B \in \Pi$.

$\lambda y.t[\overrightarrow{s}/\overrightarrow{x}] * u.\pi_B \succ t[\overrightarrow{s}/\overrightarrow{x}, u/y] * \pi_B$ or l'hypothèse de récurrence nous donne précisément $[\overrightarrow{s}/\overrightarrow{x}, u/y] * \pi_B \in \Pi$, d'où le résultat.

4. La dernière règle est : si $\Gamma \vdash t : (A \to B) \to A$ alors $\Gamma \vdash cct : A$

On veut $cc \in |((A \to B) \to A) \to A|$. Supposons $u_{(A \to B) \to A} \in |(A \to B) \to A|$ et $\pi_A \in \|A\|$. On veut $cc * u_{(A \to B) \to A}.\pi_A \in \Pi$. $cc * u_{(A \to B) \to A}.\pi_A \succ u_{(A \to B) \to A} * k_{\pi_A}.\pi_A$. Ceci est dans $\Pi$ à condition que $k_{\pi_A} \in |A \to B|$. Soient $v_A \in |A|$ et $\pi_B \in \|B\|$, est-ce que $k_{\pi_A} * v_A.\pi_B \in \Pi$ ? oui, car $k_{\pi_A} * v_A.\pi_B \succ v_A * k_{\pi_A}$.

5. La dernière règle est : si $\Gamma \vdash t : A$ alors $\Gamma \vdash t : \forall x A$ pour $x$ non libre dans $\Gamma$.

L'hypothèse d'induction nous donne $t[\overrightarrow{s}/\overrightarrow{x}] \in |A[b/x]|$ pour tout $b$ donc $t[\overrightarrow{s}/\overrightarrow{x}] \in \bigcap_{b \in M} |A[b/x]| = |\forall x A|$

6. La dernière règle est : si $\Gamma \vdash t : A$ alors $\Gamma \vdash t : \forall X^k A$ pour $X^k$ non libre dans $\Gamma$.

L'hypothèse d'induction nous donne $t[\overrightarrow{s}/\overrightarrow{x}] \in \left|A[R^k/X^k]\right|$ pour tout $R^k$ donc $t[\overrightarrow{s}/\overrightarrow{x}] \in \bigcap_{R \in P(\Pi)^{M^k}} \left|A[R^k/X^k]\right| = \left|\forall X^k A\right|$

7. si $\Gamma \vdash t : \forall x A$ alors $\Gamma \vdash t : A[\tau/x]$ un terme $\tau$

Il suffit de voir que $|A[\tau/x]| \sqsupseteq |\forall x A|$ (considérer $A[b/x]$ où $b = \tau$) et d'appliquer l'hypothèse d'induction.

8. La dernière règle est : si $\Gamma \vdash t : \forall X A$ alors $\Gamma \vdash t : A[\phi(x_1...x_n)/Xx_1...x_n]$ pour une formule $\phi(x_1...x_n)$

Il suffit de voir que $|A[\phi/X]| \sqsupseteq |\forall X A|$ (considérer $A[R/\phi]$ où $R(a_1...a_n) = \|\phi(a_1...a_n)\|$ et d'appliquer l'hypothèse d'induction.

**Corollary 42** *Si $\vdash t : F$ alors $t \in |F|$*

On montre également un théorème simple sur l'égalité qui sera utile par la suite.

**Theorem 43** *Soient a,b deux entiers;*
$\|a = b\| = \|\top \to \bot\|$ *si* $a \neq b$.
$\|a = b\| = \|\forall X\,(X \to X)\,\|$ *si* $a = b$.

$\|a = b\| =_{def} \|\forall X(Xa \to Xb)\|$

Si $a \neq b$, on peut prendre $|Xa| = \Lambda_c$ et $\|Xb\| = \Pi$ d'où $\|a = b\| \supseteq \{t.\pi / t \in \Lambda_c, \pi \in \Pi\}$ et l'égalité suit trivialement de ce que les piles de $\|a = b\|$ ayant nécessairement la forme $t.\pi$, il n'existe par de surensemble de piles de cette forme.

Si $a = b$, on a nécessairement $\|Xa\| = \|Xb\|$, d'où $\|\forall X(Xa \to Xb)\| = \|\forall X\,(X \to X)\,\|$.

## 3.3 L'arithmétique du second ordre

**Definition 44** *Une formule F est réalisée s'il existe un lambda-c terme t sans continuation tel que $t \Vdash F$ pour tout choix de* $\Pi$.

On part d'une formulation standard de l'arithmétique de Peano du second ordre ($PA_2$), par exemple l'ensemble d'axiomes suivants formulés dans un langage L contenant $\{0, s, +\times\}$ :

1. axiomes pour le successeur
$s0 \neq 0$ et $\forall x, y\,(sx = sy \to x = y)$
2. axiomes pour l'addition
$\forall x(x + 0 = x)$
$\forall x, y(x + sy = s(x + y))$
3. axiomes pour la multiplication
$\forall x(x \times 0 = 0)$
$\forall x, y(x \times sy = x \times y + y)$
4. axiome d'induction
$\forall x Int(x)$ où $Int(x) \equiv \forall X\,[\forall y\,(Xy \to Xsy)\,, X0 \to Xx]$

Afin de pouvoir typer tous les théorèmes de $PA_2$, on voudrait que tous ces axiomes soient réalisés. En effet, si un axiome $A$ est réalisé, disons par un terme $t$, alors le typage $\Gamma \vdash t : A$ est compatible avec le lemme d'adéquation.

1. Soit $u$ un terme quelconque. On montre que $\lambda x.xu \in |s0 \neq 0| = |(\top \to \bot) \to \bot|$. Soit $t_{\top\to\bot} \in |\top \to \bot|$, pour toute pile $\pi$, $t_{\top\to\bot} * u.\pi \in \Pi$. Or $\lambda x.xu * t_{\top\to\bot}.\pi \succ t_{\top\to\bot} * u.\pi$. Par ailleurs, il est clair que $\lambda x.x \in |\forall x, y\,(sx = sy \to x = y)|$.

2 et 3. Il suit du théorème 43 que toutes les formules équationnelles vraies sont réalisées par l'identité.

Par contre, l'axiome d'induction n'est pas réalisé (on montre qu'il ne l'est pas même pour un modèle à deux éléments). Ceci conduit à envisager une relativisation des quantificateurs du premier ordre : on cherche à se dispenser de l'axiome d'induction en limitant les théorèmes aux éléments qui le satisfont. On devra utiliser pour cela :

4'. $\forall x_1 ... \forall x_k \, [Int(x_1) ... Int(x_k) \rightarrow Int(f x_1 ... x_k)]$ pour tout symbole de fonction $f$ du langage.

**Definition 45** *On définit $A^{int}$ par induction sur la forme de $A$*
*si $A$ est atomique, $A^{int} \equiv A$*
$(A \rightarrow B)^{int} \equiv A^{int} \rightarrow B^{int}$
$(\forall x A)^{int} \equiv \forall x (Int(x) \rightarrow A^{int})$
$(\forall X A)^{int} \equiv \forall X (A^{int})$

**Theorem 46** *Si $\vdash_{PA_2} A$, alors $\vdash_{1+2+3+4'} A^{int}$ pour $A$ une formule close.*

L'induction sur la longueur des preuves exige une formulation un peu plus générale.

**Theorem 47** *Si $\Gamma \vdash_{PA_2} A$, alors $\Gamma^{int} \cup \{Int(x_i)\}_{x_i \in vlib(A)} \vdash_{1+2+3+4'} A^{int}$.*

La démonstration par induction ne pose pas de problème particulier. On s'attarde seulement sur les règles où quelque chose se passe.

a) Si $\Gamma \vdash A$ alors $\Gamma \vdash t : \forall x A$ pour $x$ non libre dans $\Gamma$

L'hypothèse d'induction nous dit $\Gamma^{int} \cup \{Int(x_i)\}_{x_i \in vlib(A)} \vdash_{1+2+3+4'} A^{int}$.

On veut $\Gamma^{int} \cup \{Int(x_i)\}_{x_i \in vlib(\forall x A)} \vdash_{1+2+3+4'} \forall x (Int(x) \rightarrow A^{int})$

Il suffit d'appliquer une $\rightarrow_{intro}$ qui élimine $Int(x)$ des hypothèses.

b) si $\Gamma \vdash \forall x A$ alors $\Gamma \vdash t : A[\tau/x]$ pour tout terme $\tau$.

L'hypothèse d'induction nous donne $\Gamma^{int} \cup \{Int(x_i)\}_{x_i \in vlib(\forall x A)} \vdash_{1+2+3+4'} \forall x (Int(x) \rightarrow A^{int})$

L'idée est d'effectuer une $\forall_{élim}$ qui introduise $\tau$ puis une $\rightarrow_{élim}$ avec $Int(\tau)$. On montre facilement par induction sur la forme d'un terme qu'on peut avoir $Int(\tau)$ pour n'importe quel terme, en utilisant $Int(0)$, 4' et des hypothèses supplémentaires $Int(x_j)$ pour les variables libres $x_j$ de $\tau$. Donc on aura bien :

$\Gamma^{int} \cup \{Int(x_i)\}_{x_i \in vlib(A[\tau/x])} \vdash_{1+2+3+4'} A[\tau/x]^{int})$.

c) On vérifie enfin facilement que $\vdash \forall x Int(x)^{int}$

Reste alors à montrer

**Theorem 48** *$\forall x_1 ... \forall x_k \, [Int(x_1) ... Int(x_k) \rightarrow Int(f x_1 ... x_k)]$ est réalisé pour tout symbole de fonction $f$ du langage.*

**Lemma 49** *Soient $\xi, \eta, t_1 ... t_k$ des $\lambda$-termes du $\lambda$-calcul ordinaire, si $\eta$ n'est pas une application et si $\xi > \eta t_1 ... t_k$ où $>$ désigne la réduction de tête paresseuse, alors pour toute pile $\pi$, $\xi * \pi \succ \eta t_1 ... t_k * \pi$*

La démonstration se fait par induction sur la longueur de la réduction de tête. Si $\xi$ est $\eta t_1...t_k$, le résultat est immédiat. Si $\xi$ est $(\lambda yu)v\overrightarrow{w}$, $\xi >_1 u[y := v]\overrightarrow{w} > \eta t_1...t_k$. Par hypothèse d'induction, $u[y := v]\overrightarrow{w} * \pi \succ \eta t_1...t_k * \pi$. Comme $\eta$ n'est pas une application, des étapes de réduction doivent nécessairement avoir lieu qui empilent les $\overrightarrow{w}$ sur $\pi$. Donc $u[y := v]\overrightarrow{w} * \pi \succ u[y := v] * \overrightarrow{w}.\pi \succ \eta t_1...t_k * \pi$. D'où $\xi * \pi \succ u[y := v] * \overrightarrow{w}.\pi \succ \eta t_1...t_k * \pi$.

On ne redémontre pas le théorème suivant qui donne pour la définition de la réalisabilité classique un résultat du $\lambda$-calcul ordinaire.

**Theorem 50** *Soit n un entier et $\nu$ un $\lambda$-terme du $\lambda$-calcul ordinaire $\beta$-équivalent à l'entier de Church n, $\nu \Vdash Int[s^n 0]$*

Soit s un $\lambda$-terme déterminé représentant le successeur sur les entiers de Church. On définit $T = \lambda f \lambda n.(((n)\lambda g g \circ s)f)0$ (T est l'opérateur de stockage de [13]). On montre le lemme suivant

**Lemma 51** *si $\phi * s^n 0.\pi_X$ pour tout $\pi_X \in \|X\|$, alors $T\phi \Vdash Int(n) \to X$*

Soit $\nu \Vdash Int(n)$. Soit II un choix de bottom fixé et $\pi_X$ une pile quelconque dans $\|X\|$, on veut montrer

$T\phi * \nu.\pi_X \in$ II.

$T\phi * \nu.\pi_X \succ \lambda f \lambda n.(((n)\lambda g g \circ s)f)0 * \phi.\nu.\pi_X \succ ((\nu)\lambda g.g \circ s)\phi * 0.\pi_X$

Donc on se ramène à montrer :

$((\nu)\lambda g.g \circ s)\phi * 0.\pi_X \in$ II

Pour cela, on va interpréter judicieusement une variable de prédicat $P$ qui servira à instancier $Int(n)$.

on définit $\|Pk\| = \{s^{n-k}0.\pi_X / \pi_X \in \|X\|\}$ pour $0 \leq k \leq n$ et $\varnothing$ sinon.

si on avait

a) $\lambda g.g \circ s \Vdash \forall x(Px \to Psx)$

b) $\phi \Vdash P0$

on aurait gagné. En effet, $\nu \Vdash \forall x(Px \to Psx), P0 \to Pn$, d'où on tire $\nu * \lambda g.g \circ s.\phi.0.\pi_X \in$ II. Comme $((\nu)\lambda g.g \circ s)\phi * 0.\pi_X \succ \nu * \lambda g.g \circ s.\phi.0.\pi_X$, la clôture de II par anti-réduction nous donne le résultat.

a) $?\lambda g.g \circ s \Vdash \forall x(Px \to Psx)$

? $\lambda g.g \circ s * t_{Pk}.\pi_{Psk} \in$ II

si $k + 1 \succ n$,c'est terminé, car alors $|Psk| = \Lambda$, et donc $\lambda g.g \circ s.t_{Pk} \in |Psk|$

sinon $\pi_{Psk} \equiv s^{n-(k+1)}0.\pi_X$

? $\lambda g \lambda x g(s)x * t_{Pk}.s^{n-(k+1)}0.\pi_X \in$ II

? $\lambda x t_{Pk}(s)x * s^{n-(k+1)}0.\pi_X \in$ II

$?t_{Pk}.(s)s^{n-(k+1)}0.\pi_X \in$ II

Ce qui est vrai car $s^{n-k}0.\pi_X \in \|Pk\|$.

b) Ceci découle immédiatement de l'hypothèse. En effet, $\phi \Vdash P0$ si et seulement si $\phi * s^n 0.\pi_X \in$ II.

On montre maintenant le théorème 48. On se place dans le cas d'une fonction récursive $f$ unaire. On pose $\hat{k} =_{def} s^k 0$ et on utilise la notion de représentation habituelle pour les fonctions récursives dans le lambda-calcul : un $\lambda$-terme $\phi$

représente une fonction récursive totale f si, pour tout n, f(n)=p implique $\phi\hat{n} =_\beta \lambda f\lambda x(f)^p x\ p = f(n)$.

Soit $\phi$ un $\lambda$-terme représentant $f$ et $n$ un entier. $\phi\hat{n} =_\beta \lambda f\lambda x(f)^p x$ *où* $p = f(n)$ donc $\phi\hat{n} > \lambda f.\xi$ avec $\lambda f.\xi =_\beta \lambda f\lambda x(f)^p x$. Par le théorème 50, $\lambda f.\xi \Vdash Int[s^p 0]$. Soit $\pi_{Int[s^p 0]} \in \|Int[s^p 0]\|$, on a alors $\lambda f.\xi * \pi_{Int[s^p 0]} \in \Pi$.

Par le lemme 49, $\phi\hat{n} * \pi \succ \lambda f.\xi * \pi$, et comme $\phi\hat{n}$ est une application, ce que n'est pas $\lambda f.\xi$, la réduction commence par $\phi\hat{n} * \pi \succ \phi * \hat{n}.\pi$. Donc $\phi * \hat{n}.\pi \succ \lambda f.\xi * \pi$ et en particulier $\phi * \hat{n}.\pi_{Int[s^p 0]} \succ \lambda f.\xi * \pi_{Int[s^p 0]} \in \Pi$. Ceci correspond à l'hypothèse du lemme 51 pour $X = .Int[s^p 0]$. Donc $T\phi \Vdash Int[s^n 0] \to Int[s^p 0]$. Comme nous l'avons pour un $n$ quelconque, on a bien $T\phi \Vdash \forall x(Int[x] \to Int[f(x)])$, autrement dit la formule $\forall x(Int[x] \to Int[f(x)])$ est réalisée.

## 3.4 La spécification des théorèmes de l'arithmétique

### 3.4.1 Pour les énoncés $\Pi_2$

On montre que la preuve d'un énoncé $\Pi_2$ de l'arithmétique dans la logique classique du second ordre peut être vu comme un programme qui calcule la fonction associée à l'énoncé.

**Theorem 52** *Si* $\vdash \theta : [\forall x \exists y(f(x,y) = 0)]^{int}$, *alors* $\theta * \widehat{n}.Tt.\pi$ *s'évalue sur* $t * \widehat{p}\pi'$ *avec* $f(n,p) = 0$ *où* $t \equiv \lambda x\lambda y.yx$

Le lemme d'adéquation nous dit que $\theta \Vdash \forall x Int(x) \to [\exists y(f(x,y) = 0)]^{int}$. Donc pour un entier $n$ quelconque,

$\theta \Vdash Int(n) \to [\exists y(f(n,y) = 0)]^{int}$. On fixe $\Pi = \{t * \widehat{p}.\pi'/f(n,p) = 0, \pi' \in \Pi\}^{\succ^{-1}}$.

Le théorème 50 nous assure que $\widehat{n} \Vdash Int(n)$. Reste donc à montrer que $Tt.\pi \in \left\| [\exists y(f(n,y) = 0)]^{int} \right\|$.

$[\exists y(f(n,y) = 0)]^{int} \equiv \forall y\,(Int(y) \to (f(n,y) = 0 \to \bot)) \to \bot$ . Donc il faut avoir

$Tt \Vdash Int(m) \to (f(n,m) = 0 \to \bot)$ pour tout entier $m$.

Par le lemme 51, on se ramène à montrer

$t * \widehat{m}.\pi_{f(n,m)=0\to\bot} \in \Pi$ pour toute pile $\pi_{f(n,m)=0\to\bot} \in \|f(n,m) = 0 \to \bot\|$

Premier cas, $f(n,m) = 0$, dans ce cas, $t * \widehat{m}.\pi_{f(n,m)=0\to\bot} \in \Pi$ par définition de $\Pi$.

Deuxième cas, $f(n,m) \neq 0$. $\pi_{f(n,m)=0\to\bot}$ est de la forme $u.\rho$ avec $u \in |\top \to \bot|$ et $\rho \in \Pi$.

$t * \widehat{m}.u.\rho \succ u * \widehat{m}.\rho$. Or $u * \widehat{m}.\rho \in \Pi$ car $\widehat{m}.\rho$ est trivialement dans $\|\top \to \bot\|$. Par conséquent, on a bien $t * \widehat{m}.\pi_{f(n,m)=0\to\bot} \in \Pi$.

La comparaison avec ce que donnent les méthodes d'Ackermann et de Kreisel est déjà instructive pour ce cas.

Si $PA_\epsilon \vdash \forall x \exists y A(x,y)$, un contre-exemple serait un $a$ tel que $\forall y \sim A(a,y)$. La NCI donne donc une fonction $f$ telle que pour tout $a$, $A(a, f(a))$, de sorte que $f$ et le code $\theta$ d'une preuve jouent bien le même rôle. Néanmoins, on remarque

que la fonction $f$ de Kreisel écrase l'individualité des preuves : le calcul de $f(a)$ commence par le calcul d'une limite supérieure pour les valeurs de $f(a)$, cette limite étant la même pour des preuves de $\forall x \exists y A(x,y)$ qui ont les mêmes paramètres. En outre la limite en question n'a pas de rapport direct avec la valeur finale de $f(a)$. Au contraire, $\theta$ est à chaque fois un programme singulier qui calcule directement la valeur pour $f(a)$. En un sens, c'est d'ailleurs bien exactement ce qu'est la méthode de substitution : appliquée aux formules de la preuve, elle peut-être vue comme un algorithme pour calculer les valeurs de $f(a)$.

### 3.4.2 Pour les énoncés $\Sigma_2$

On va montrer en quoi le terme codant la preuve d'un énoncé $\Sigma_2$ constitue un fonctionnel analogue aux fonctionnels récursifs de Kreisel. La démonstration suit celle de [12].

On dira qu'un terme $t$ représente une fonction $f$ calculable à un argument si

$t * \hat{n}.\pi \succ \widehat{f(n)} * \pi$

Le rôle de $t$ peut être joué soit par un véritable $\lambda$-terme, soit par une constante $c_f$ équipée de la règle de réduction adéquate. Afin de résoudre les difficultés posées par la définition de la réduction dans le $\lambda_c$-calcul, on introduit une nouvelle constante $\zeta$ dont on définit ainsi le comportement face à une pile.

Si $t * \hat{n}.\pi \succ \hat{p} * \pi$, alors pour tout $\xi$, $\zeta * \xi.(t)\hat{n}.\pi' \succ \xi * \hat{p}.\pi'$

On pose $F[f] = (T)\lambda x \lambda y(((\zeta)y)(f)x)x$

**Theorem 53** *Si $\theta$ est le code d'une preuve de $[\exists x \forall y(\phi(x,y) = 0)]^{Int}$ dans l'arithmétique du second ordre et $\gamma$ une fonction récursive totale représentée par un terme t, alors, pour toute pile $\pi$, le programme $\lambda f \theta F[f]$ auquel on donne l'entrée $t.\pi$ aboutit à un état $\hat{n} * \pi'$ tel que $N \vDash \phi(n, \gamma(n)) = 0$.*

On commence par fixer $\text{II} : \text{II} = \{ \hat{n} * \pi' \ / \ N \vDash \phi(n, \gamma(n)) = 0, \pi' \in \Pi \}^{\succ^{-1}}$

Par le lemme d'adéquation, il suffit de montrer la propriété pour un $\theta$ qui réalise $[\exists x \forall y(\phi(x,y) = 0)]^{Int}$. On a

$\theta \Vdash \forall x[Int(x), \forall y(Int(y) \to \phi(x,y) = 0) \to \bot] \to \bot$

et $\lambda f \theta F[f] * t.\pi \succ \theta * F[f := t].\pi$

Donc, comme II est clos par anti-réduction, il suffit d'avoir

$F[f := t] \Vdash \forall x[Int(x), \forall y(Int(y) \to \phi(x,y) = 0) \to \bot]$

On va montrer, pour un $n$ quelconque :

$F[f := t] \Vdash Int(n), \forall y(Int(y) \to \phi(n,y) = 0) \to \bot$

Or les piles qui sont dans $\|\forall y(Int(y) \to \phi(n,y) = 0) \to \bot\|$ sont de la forme $\xi.\pi$ pour une pile $\pi$ quelconque et un $\xi$ tel que $\xi \Vdash \forall y(Int(y) \to \phi(n,y) = 0)$. Donc par le lemme 51, on se ramène à montrer :

$F[f := t] * \hat{n}.\xi.\pi \in \text{II}$ c'est-à-dire $\lambda x \lambda y(((\zeta)y)(t)x)x * \hat{n}.\xi.\pi \in \text{II}$

Par ailleurs,

$\lambda x \lambda y(((\zeta)y)(t)x)x * \hat{n}.\xi.\pi$

$\succ (((\zeta)\xi)(t)\hat{n})\hat{n} * \pi$

$\succ \zeta * \xi.(t)\hat{n}.\hat{n}.\pi$

$\succ \xi * \hat{p}.\hat{n}.\pi$ où $\gamma(n) = p$, puisque $t * \hat{n}.\pi \succ \widehat{f(n)} * \pi$.

Il suffit donc de montrer maintenant que $\xi * \hat{p}.\hat{n}.\pi$ est dans II. Comme $\xi \Vdash \forall y(Int(y) \rightarrow \phi(n, y) = 0)$, $\xi.\hat{p} \Vdash \phi(n, p) = 0$, cela revient à établir que $\hat{n}.\pi \in \|\phi(n, p) = 0\|$

Premier cas : $\phi(n, p) = 0$.

On a alors $\|\phi(n, p) = 0\| = \|\forall Z, Z \rightarrow Z\|$.

$\|\forall Z, Z \rightarrow Z\| \supseteq |\Pi| \rightarrow \|\Pi\|$, or on a automatiquement $\pi \in \|\Pi\|$ et, par définition de II, $\hat{p} \in |\Pi|$. Il suit que $\hat{n}.\pi \in \|\phi(n, p) = 0\|$.

Deuxième cas : $\phi(n, p) \neq 0$.

On a alors $||\phi(n, p) = 0|| = \top \rightarrow \bot$, ce qui nous donne immédiatement $\hat{n}.\pi \in \|\phi(n, p) = 0\|$

CQFD

On constate que dans la démonstration, la seule propriété de $F[t]$ qui est utilisée est

$F'[t] * \hat{n}.\xi\pi \succ \xi * \hat{p}.\hat{n}.\pi$ où $F'[t]$ est $F[t]$ moins le $T$ de l'application principale.

Autrement dit $F'[t]$ doit fournir en réponse à un entier $n$ un autre entier $p$ qui correspond à la tentative d'exhiber un contre-exemple et mettre en tête le terme $\xi$ qui assure le traitement par le programme de cette réponse. Intuitivement, $\xi$ va ensuite mettre en tête $\hat{n}$ si $n$ marche et proposer un autre $\hat{n}$ sinon. Mais il n'y a pas de raison de considérer que $F'[t]$ correspond à une stratégie de réfutation décidée à l'avance correspondant à une fonction récursive. C'est dans cet esprit qu'est formulé le théorème dans [11] et [12]. On ajoute aux termes du lambda-c calcul une constante $\kappa$ pour laquelle on fixe un comportement adéquat : $\kappa * \hat{n}.\xi.\pi \gg \xi * \hat{p}.[\hat{n}, \hat{p}].\pi$ ($[\hat{n}, \hat{p}]$ se laisse construire comme une constante ou comme une liste d'entiers); $\kappa$ peut être vu comme une instruction *input* demandant une réponse à $n$.. Le théorème s'énonce alors :

Si $\theta$ est le code d'une preuve de $[\exists x \forall y(\phi(x, y) = 0)]^{Int}$ dans l'arithmétique du second ordre, alors, pour toute pile $\pi$, le programme $\theta$ auquel on donne l'entrée $T\kappa.\pi$ aboutit à un état $[\hat{n}, \hat{p}] * \pi'$ tel que $N \vDash \phi(n, p) = 0$.

Remarquons néanmoins qu'au cours d'une réduction, $\kappa$ n'arrive qu'un nombre fini de fois en tête, de sorte que les valeurs données par l'instruction *input* peuvent toujours en droit être vues comme produites par une fonction récursive.

On peut alors voir le programme comme une stratégie gagnante pour $\exists loise$ dans un jeu P sémantique suivant : le jeu commence par le choix d'un entier $x$ par $\exists loise$, puis continue par un choix d'un entier $y$ par $\forall bélard$. $\exists loise$ peut alors soit s'arrêter et $\exists loise$ gagne si $\phi(x, y) = 0$ tandis qu'$\forall bélard$ perd dans le cas contraire. Mais $\exists loise$ peut également choisir de recommencer à proposer un $x$. Si le jeu dure infiniment longtemps, $\forall bélard$ gagne.

Une exécution du programme sur $T\kappa.\pi$ revient au fait de jouer une partie, l'état $\kappa * \hat{n}.\xi.\pi$ correspond au choix d'$\forall bélard$ et tous les états $\xi * \hat{p}.\hat{n}.\pi$ atteints lors de la réduction correspondent au fait d'atteindre une des positions finales.

La méthode de substitution peut elle aussi être interprétée comme fournissant une stratégie gagnant contre toute stratégie de l'adversaire. On commence par annuler tous les substituants des $\epsilon$-termes, ce qui revient à proposer

0; alors de deux choses l'une, soit tous les axiomes de la forme III.1 sont vrais, et 0 est gagnant, soit ce n'est pas le cas, l'algorithme continue alors à tourner en modifiant les substituants des catégories pour rendre vrais les axiomes de la forme III.1 - ce qui ne se traduit pas forcément à chaque pas par une modification de la valeur finale proposée. Ce que la preuve de terminaison établit, c'est qu'au bout d'un moment, l'algorithme rend vrai tous les axiomes critiques, ce qui implique que le substituant proposé pour l'$\epsilon$-terme de la formule finale donne une valeur gagnante. On pourrait penser que si la stratégie de l'adversaire n'est pas identifiée dès le départ à une fonction récursive, on ne peut pas donner à l'avance une limite au nombre de substitutions, mais en fait, on a vu dans le lemme 34 que la fonctionnelle qui fournit cette limite ne dépend que d'un nombre fini de valeurs des fonctions, ce qui veut dire qu'on n'a besoin que d'un nombre fini de tests sur les réponses de l'adversaire pour borner la valeur. Par contre, rien ne garantit que l'on s'arrête dès que l'on sort une valeur gagnante (il se peut que des axiomes soient encore faux dans la preuve), il en va d'ailleurs de même avec les programmes que constituent les $\lambda_c$-termes.

### 3.4.3 Dans le cas général

On dira qu'un terme $t$ représente une fonction $f$ calculable à k arguments si

$t * \widehat{n_1}...\widehat{n_k}.\pi \succ \widehat{f(n_1...n_k)} * \pi$.

Le rôle de $t$ peut être joué soit par un véritable $\lambda$-terme, soit une constante $c_f$ équipée de la règle de réduction adéquate. De manière analogue au $\zeta$ précédent, on définit des $\zeta_k$ qui permettent d'évaluer des fonctions à k arguments à l'intérieur d'une pile.

Si $t * \widehat{n_1}...\widehat{n_k}.\pi \succ \widehat{p} * \pi$, alors pour tout $\xi$, $\zeta_k * \xi.(t)\widehat{n_1}...\widehat{n_k}.\pi' \succ \xi * \hat{p}.\pi'$

On commence par définir une suite de termes par induction rétrograde :

$H_k = [x_1, ..., x_k]$ pour une représentation fixée des listes et pour un k donné

$H_j = (T)\lambda x_{j+1}\lambda y_{j+1}.(((\zeta_{j+1})y_{j+1})(f_{j+1})x_1...x_{j+1})H_{j+1}$ pour $0 \leq j < k$

On désigne alors par $F_k[f_1...f_k]$ le terme $H_0$. On remarque que le $F$ de la section précédente correspond bien à notre $H_1$ actuel.

**Theorem 54** *Si $\theta$ est le code d'une preuve de*

$[\exists x_1 \forall y_1 ... \exists x_k \forall y_k (\phi(x_1...x_k, y_1...y_k) = 0)]^{Int}$ *dans l'arithmétique du second ordre et $\gamma_1(x_1)...\gamma_k(x_1...x_k)$ sont des fonctions récursives totales représentées par les $\lambda$-terme $t_1...t_k$, alors, pour toute pile $\pi$, $\lambda f_1...\lambda f_k \theta F_k[f_1...f_k] * t_1...t_k.\pi$ se réduit sur $[\hat{n}_1, ..., \hat{n}_k] * \pi'$ avec $N \vDash \phi(n_1...n_k, \gamma_1(n_1)...y_k(n_1...n_k)) = 0)$.*

On commence par fixer II : II =$\{$ $[\hat{n}_1, ..., \hat{n}_k]*\pi$ / $N \vDash \phi(n_1...n_k, \gamma_1(n_1)...y_k(n_1...n_k)) = 0\}$

Par le lemme d'adéquation, il suffit de montrer la propriété pour un $\theta$ qui réalise $[\exists x_1 \forall y_1 ... \exists x_k \forall y_k \phi(x_1...x_k, y_1...y_k) = 0]^{Int}$.

On pose

$A_i(x_1...x_i) \equiv [\exists x_{i+1} \forall y_{i+1} ... \exists x_k \forall y_k \phi(x_1...x_k, \gamma_1(x_1)...\gamma_i(x_1...x_i)y_{i+1}...y_k) = 0]^{Int}$, $A'_i(x_1...x_i) \equiv [\exists x_{i+1} \forall y_{i+1} ... \exists x_k \forall y_k \phi(x_1...x_k, \gamma_1(x_1)...\gamma_{i-1}(x_1...x_{i-1})y_i y_{i+1}...y_k) = 0]^{Int}$, $H'_j(n_1...n_j) \equiv H_j[x_1 := \hat{n}_1, ..., x_j := \hat{n}_j, f_1 := t_1, ..., f_k := t_k]$.

$\lambda f_1...\lambda f_k \theta F_k[f_1...f_k] * t_1...t_k.\pi \succ \theta H'_0 * \pi \succ \theta * H'_0.\pi$ donc comme $\theta \Vdash [\exists x_1 \forall y_1 ... \exists x_k \forall y_k \phi(x_1...x_k, y_1...y_k) = 0]^{Int}$, il s'agit de montrer $H'_0.\pi \in \|A_0\|$. La démonstration se fait par induction rétrograde en commençant par $H'_k(n_1...n_k).\pi \in \|\phi(n_1...n_k, \gamma_1(n_1)...y_k(n_1...n_k)) = 0)\|$. La situation est semblable à la fin de la démonstration précédente.

Premier cas : $\phi(n_1...n_k, \gamma_1(n_1)...y_k(n_1...n_k)) = 0$.

On a alors $\|\phi(n_1...n_k, \gamma_1(n_1)...y_k(n_1...n_k)) = 0\| = \|\forall Z, Z \to Z\|$.

$\|\forall Z, Z \to Z\| \supseteq |\Pi| \to \|\Pi\|$, or on a automatiquement $\pi \in \|\Pi\|$ et, par définition de II, $[\hat{n}_1, ..., \hat{n}_k] \in |\Pi|$. Il suit que $[\hat{n}_1, ..., \hat{n}_k].\pi \in \|[\hat{n}_1, ..., \hat{n}_k]\|$.

Deuxième cas : $\phi(n_1...n_k, \gamma_1(n_1)...y_k(n_1...n_k)) \neq 0$

On a alors $||\phi(n, p) = 0|| = \top \to \bot$, ce qui nous donne immédiatement $[\hat{n}_1, ..., \hat{n}_k].\pi \in \|\phi(n_1...n_k, \gamma_1(n_1)...y_k(n_1...n_k)) = 0\|$

On montre maintenant $H'_j(n_1...n_j).\pi \in \|A_j(n_1...n_j)\|$ sous l'hypothèse $H'_{j+1}(n_1...n_{j+1}).\pi \in \|A_{j+1}(n_1...n_{j+1})\|$.

On commence par remarquer que, pour un terme $\xi$ et une pile $\pi$ quelconques, on a :

$\lambda x_{j+1} \lambda y_{j+1}.(((\zeta_{j+1})y_{j+1})(t_{j+1})\hat{n}_1...\hat{n}_j x_{j+1})H'_{j+1}(n_1...n_j x_{j+1}) * \hat{n}_{j+1}.\xi.\pi$

$\succ (((\zeta_{j+1})\xi)(t_{j+1})\hat{n}_1...\hat{n}_j\hat{n}_{j+1})H'_{j+1}(n_1...n_j\hat{n}_{j+1}) * .\pi$

$\succ \zeta_{j+1} * \xi.(t_{j+1})\hat{n}_1...\hat{n}_j\hat{n}_{j+1}.H'_{j+1}(n_1...n_j\hat{n}_{j+1}).\pi$ puis comme $t_{j+1} * \hat{n}_1...\hat{n}_{j+1}.\pi \succ \hat{p}_{j+1}.\pi$ où $p_{j+1} = \gamma(n_1...n_{j+1})$

$\succ \xi * \hat{p}_{j+1}.H'_{j+1}(n_1...n_{j+1}).\pi \quad (*)$

On doit montrer $H'_j(n_1...n_j).\pi \in \|\forall x_{j+1}[Int(x_{j+1}), \forall y_{j+1}(Int(y_{j+1}) \to A_{j+1}(n_1...n_j x_{j+1})) \to \bot] \to \bot \|$, ce qui revient à $H'_j(n_1...n_j) \Vdash \forall x_{j+1}[Int(x_{j+1}), \forall y_{j+1}(Int(y_{j+1}) \to A_{j+1}(n_1...n_j x_{j+1})) \to \bot]$ ou encore $H'_j(n_1...n_j) \Vdash Int(n_{j+1}) \to [\forall y_{j+1}(Int(y_{j+1}) \to A_{j+1}(n_1...n_{j+1})) \to \bot]$ pour un $n_{j+1}$ quelconque.

$H'_j(n_1...n_j) \equiv (T)\lambda x_{j+1} \lambda y_{j+1}.(((\zeta_{j+1})y_{j+1})(t_{j+1})\hat{n}_1...\hat{n}_j x_{j+1})H'_{j+1}(n_1...n_j x_{j+1})$. Or les piles qui sont dans $\|\forall y_{j+1}(Int(y_{j+1}) \to A_{j+1}(n_1...n_{j+1})) \to \bot\|$ sont de la forme $\xi.\pi$ où $\pi$ une pile quelconque et $\xi$ un terme réalisant $\forall y_{j+1}(Int(y_{j+1}) \to A'_{j+1}(n_1...n_{j+1}))$. Donc, d'après le lemme 51 il suffit de montrer : $\lambda x_{j+1} \lambda y_{j+1}.(((\zeta_{j+1})y_{j+1})(t_{j+1})\hat{n}_1...\hat{n}_j x_{j+1}$ $\hat{n}_{j+1}.\xi.\pi \in$ II

Par (*) et la clôture de II par anti-réduction, il suffit pour cela d'avoir $\xi * \hat{p}_{j+1}.H'_{j+1}(n_1...n_{j+1}).\pi \in$ II. Mais, par définition, $\xi$ réalise $Int(p_{j+1}) \to A_{j+1}(n_1...n_{j+1})$, donc il suffit de vérifier que $\hat{p}_{j+1} \in |Int(n_{j+1})|$ et $H'_{j+1}(n_1...n_{j+1}).\pi \in \|A_{j+1}(n_1...n_{j+1})\|$, ce qui est immédiat pour $\hat{p}_{j+1}$ et découle de l'hypothèse d'induction pour $H'_{j+1}(n_1...n_{j+1}).\pi$.

CQFD.

On peut faire les mêmes remarques que pour le théorème précédent. Ici, la propriété utile du terme universel $F_k[t_1...t_k]$ et de ses sous termes est (*) de sorte que l'on peut faire jouer le rôle des $H_j$ par des constantes munies d'une règle de réduction appropriées; [11] dote ainsi des constantes $\kappa^j_{n_1p_1,...,n_jp_j}$ des règles de réduction suivantes

pour $0 \leq j \leq k-2$, $\kappa^j_{n_1p_1,...,n_jp_j}\widehat{n}_{j+1} * \xi.\pi \gg \xi * \widehat{p}_{j+1}.T\kappa^j_{n_1p_1,...,n_{j+1}p_{j+1}}$

pour $j = k-1$, $\kappa^j_{n_1p_1,...,n_{k-1}p_{k-1}}\widehat{n}_k * \xi.\pi \gg \xi * \widehat{p}_k.[\widehat{n}_1\widehat{p}_1, ..., \widehat{n}_k\widehat{p}_k]$

et montre le théorème :

Si $\theta$ est le code d'une preuve de

$[\exists x_1 \forall y_1 ... \exists x_k \forall y_k (\phi(x_1...x_k, y_1...y_k) = 0)]^{Int}$ dans l'arithmétique du second ordre alors, pour toute pile $\pi$, $\theta * T\kappa^0.\pi$ se réduit sur $[\widehat{n}_1\widehat{p}_1, ..., \widehat{n}_k\widehat{p}_k] * \pi'$ avec $N \vDash \phi(n_1...n_k, p_1...p_k) = 0$.

Là encore l'interprétation en termes de jeu est naturelle; le jeu est le même que précédemment; les quantificateurs existentiels correspondent aux coups d'$\exists$*loïse*, les universels à ceux d'$\forall$*bélard* et $\exists$*loïse* peut revenir en arrière à tout moment. Les constantes $\kappa^j_{n_1p_1,...,n_jp_j}$ correspondent toujours à des instructions *input* qui demandent à $\forall$*bélard* ce qu'il veut jouer en fonction de la position qui a été atteinte dans le jeu. Le déroulement du programme $\theta$ auquel on donne comme entrée $T\kappa^0.\pi$ où $\pi$ ne comporte pas de constantes $\kappa$, correspond bien à un déroulement du jeu selon les règles. En effet, si une constante $\kappa^{j+1}_{n_1p_1,...,n_{j+1}p_{j+1}}$ arrive en tête, c'est nécessairement que des constantes $\kappa^0...\kappa^j_{n_1p_1,...,n_jp_j}$ sont précédemment arrivés en tête, respectant la règle selon laquelle $\exists$*loïse* ne peut repartir que de positions qui ont déjà été atteintes; en outre, cela montre bien que tous les coups successifs du jeu sont joués, au sens où avant de terminer sur $[\widehat{n}_1\widehat{p}_1, ..., \widehat{n}_k\widehat{p}_k]$, le programme a nécessairement mis en tête au moins une fois chacun des $\kappa^i$.

## 3.5 Remarque finale sur les deux approches

L'analyse de Kreisel mettait l'accent sur la modularité de la notion d'interprétation. Le fait de vérifier immédiatement et de manière constructive les clauses $\gamma$) et $\delta$) est même ce qui constitue la spécificité de la démonstration de la NCI fondée sur l'$\epsilon$-calcul par rapport aux autres approches, qu'elles partent de la preuve de consistance Gentzen ou d'une interprétation fonctionnelle (voir par exemple [2]). Kohlenbach [7] a montré les clauses $\gamma$) et $\delta$) grâce au couple traduction de Gödel et interprétation fonctionnelle, mais au prix de lourdes manipulations, puisqu'il s'agit de montrer que ces clauses sont des théorèmes dans des extensions de $PA^\omega$ et de fournir ensuite une interprétation fonctionnelle pour ces théorèmes.

Il est naturel de poser les mêmes questions dans le cadre de l'interprétation par des $\lambda_c$-termes. La réponse sera différente selon le niveau auquel on se place.

Si par interprétation d'une formule, on entend seulement un $\lambda_c$-terme qui est le code d'une preuve, la modularité de l'interprétation est triviale. Pour l'analogue de la condition $\delta$), etant donnés deux termes $t$ et $u$ correspondant respectivement à une preuve de $A$ et à une preuve de $A \to B$, $(u)t$ donne bien sûr une interprétation pour $B$, moyennnant éventuellement les transformations liées à la mise sous forme prénexe. De même pour la condition $\gamma$), étant donné un terme $u$ codant une preuve de $\sim A$, on montre qu'il n'y a pas de termes $t$ codant une preuve de $A$. $(u)t$ serait alors typable avec $\perp$ . Par le lemme d'adéquation, on se ramène à prouver qu'il n'y a pas de $\lambda_c$-terme $v$ tel que $v \Vdash \forall X X$ pour tout choix de II. On considère en effet $\text{II} = \{c * \rho\}^{\succ^{-1}}$ et $\text{II}' = \{c' * \rho\}^{\succ^{-1}}$ où $c$ et $c'$ sont deux constantes fixées différentes et $\rho$ un fonds de pile. Soit $\pi$ une pile quelconque, on devrait avoir à la fois $v * \pi \succ c * \rho$ et $v * \pi \succ c' * \rho$ ce qui est impossible. Le fait que $\perp$ ne soit pas réalisé constitue une preuve que $\perp$ n'est pas dérivable dans $AP_2$, de sorte qu'on retrouve ce dont la NCI partait, à savoir une preuve de consistanc (même si les moyens utilisés ici n'ont rien de finitistes

bien sûr).

Reste qu'en un sens, un $\lambda_c$-terme qui se comporte comme le code d'une preuve, constitue tout aussi bien une interprétation que le code d'une preuve. Soit $A \equiv [\exists x_1 \forall y_1 ... \exists x_k \forall y_k (\phi(x_1...x_k, y_1...y_k) = 0)]^{Int}$ et $t$ un $\lambda_c$-terme tel que $t \Vdash A$ avec $\amalg_A = \{[\widehat{n}_1\widehat{p}_1, ..., \widehat{n}_k\widehat{p}_k] * \pi \ / \ N \vDash \phi(n_1...n_k, \gamma_1(n_1)...y_k(n_1...n_k)) = 0\}$. $t$ fournit une interprétation pour A, c'est-à-dire une stratégie gagnante dans le jeu associé à A. La question posée par $\delta$) est alors : étant donnés une preuve de $A \to B$ et un $\lambda_c$-terme qui réalise $A$ au sens de $\amalg_A$, peut-on trouver un terme qui réalise $B$ au sens de $\amalg_A$ ? On pourrait encore généraliser et demander : étant donnés deux $\lambda_c$-termes qui réalisent respectivement $A$ au sens de $\amalg_A$ et $A \to B$ au sens de $\amalg_{A \to B}$ peut-on trouver un terme qui réalise $B$ au sens de $\amalg_A$ ?

L'idée est que l'interprétation de $A \to B$ doit être une stratégie gagnante dans le jeu qui commence par le choix par $\exists$*loïse* de $\sim A$ ou de $B$ et qui se poursuite ensuite comme les jeux associés aux formes prénexes de $\sim A$ et $B$. Il semble alors nécessaire de fournir une spécification pour $A \to B$, qui n'est pas en forme prénexe; en fait on n'est pas obligé de résoudre le problème directement. On peut se contenter d'interpréter les connecteurs comme des quantificateurs. On peut supposer sans perte de généralité que $\sim A$ et $B$ sous forme prénexe sont caractérisés par les mêmes alternance de quantificateurs (au besoin, on ajoute des quantificateurs qui ne lient aucune variable). Soient $\sim A \equiv [\forall x_0 \exists x_1 \forall y_1 ... \exists x_k \forall y_k (\phi(x_0...x_k, y_1...y_k) = 0)$ et $B \equiv [\forall x_0 \exists y_1 \forall y_1 ... \forall x_k \exists y_k (\psi(x_0...x_k, y_1...y_k) = 0)]$, une preuve de $A \to B$ nous donne une preuve de $A \Rightarrow B \equiv \exists x_0 \forall x_0 ... \forall x_k \exists y_k (\vartheta(x_0...x_k, y_0...y_k) = 0)$ avec

$\vartheta(x_0...x_k, y_0...y_k) = 0$ si et seulement si

$y_0 = 0$ et $\phi(x_0...x_k, y_1...y_k) = 0$

$y_0 \neq 0$ et $\psi(x_0...x_k, y_1...y_k) = 0$

Le théorème de spécification nous dit bien alors que les réalisateurs de $A \Rightarrow B$ donnent une stratégie gagnante dans le jeu associé à $A \to B$.[11]

Pour revenir à la question posée, Coquand [3], dans lequel l'élimination des coupures est vue comme une interaction entre deux stratégies sur des jeux tout-à-fait analogues aux jeux ici considérés, fournit une réponse partielle. Un algorithme est en effet défini qui, étant données une formule $B$ existentielle et des stratégies gagnantes $\sigma$ et $\tau$ pour les formules $A \to B$ et $A$, fournit un témoin pour $B$. L'idée est d'organiser un *débat* qui fait jouer la stratégie $\sigma$ vue comme stratégie partielle sur $\sim A$ contre $\tau$ et de montrer que le débat termine, c'est-à-dire qu'au bout d'un moment $\sigma$ est obligé d'abandonner $\sim A$ et de jouer sur $B$.

# References


[1] W. Ackermann. Zur Widerspruchsfreiheit der reinen Zahlentheorie. Mathematische Annalen 117, pp. 162-194 (1940)

[2] J. Avigad et S. Feferman. Gödel's functional (Dialectica) Interpretation, pp. 337-405 in S.R. Buss (Ed.). Handbook of Proof Theory. Elsevier (1998)

[3] T. Coquand. A semantics of evidence for classical arithmetic. Journal of Symbolic Logic 60, pp. 325-337 (1995).

[4] K. Gödel. Uber eine bisher noch nicht benütze Erweiterung des finiten Standpunktes. Dialectica 12, pp. 280-287 (1958).

[5] D. Hilbert et P. Bernays. Grundlagen der Mathematik, vol. 1. Springer (1934).

[6] D. Hilbert et P. Bernays. Grundlagen der Mathematik, vol. 2. Springer (1939).

[7] U. Kohlenbach. On the no-counterexample interpretation. Journal of Symbolic Logic 64, pp. 1491-1511 (1999).

[8] G. Kreisel. On the interpretation of non-finitist proofs, part I. Journal of Symbolic Logic 16, pp. 241-267 (1951).

[9] G. Kreisel. On the interpretation of non-finitist proofs, part II. Journal of Symbolic Logic 17, pp. 43-58 (1952).

[10] G. Kreisel. Mathematical significance of consistency proofs. Journal of Symbolic Logic 23, pp. 155–182 (1958).

[11] J.-L Krivine. Countable choice and quote (preprint)

[12] J.-L. Krivine. Arithmetical theorems, call-by-value, objects (transparents)

[13] J.-L. Krivine. A general storage theorem for integers in call-by-name $\lambda$-calculus. Theoretical Computer Science 129, pp. 79-94 (1994)

[14] J.-L. Krivine. Typed lambda-calculus in classical Zermelo-Fraenkel set theory. Archives for Mathematical Logic 40 pp. 189-205 (2001)

[15] H. Wang. A survey of mathematical logic. Peking : Science Press (1963).