Argumentum. Journal of the Seminar of Discursive Logic, Argumentation Theory and Rhetoric 18 (1): 45-65, 2020

Pierre JORAY
Université de Rennes (France)

Un système de déduction naturelle pour la Protothétique de Leśniewski

A Natural Deduction System for Leśniewski's Prothothetic

Abstract: Stanisław Leśniewski's system called Protothetic is one of the most stimulating systems of propositional logic. Including quantification and a special rule for introducing definitions among its theorems, it is a powerful means for the study of many questions of logic and metalogic. In this paper we present how to obtain a version of Protothetic in the style of Jaśkowski-Fitch's natural deduction systems. Unlike Leśniewski's original system, which requires a fairly laborious learning process, the use of this new version, called PND, is accessible to anyone familiar with the well-known methods of natural deduction for propositional logic. We show that PND contains classical propositional logic and opens the possibility of developments of Protothetic in intensional logic.

Keywords: logic, natural deduction, Protothetic, Leśniewski

1. Préambule

Denis Miéville a consacré une grande part de sa carrière académique et de sa vie intellectuelle à l'œuvre géniale du logicien Stanisław Leśniewski. Initiés en 1984 par une thèse de doctorat qui reste à mes yeux l'un des textes majeurs sur le sujet, ses travaux n'ont eu de cesse de faire connaître et développer l'originalité des outils formels et l'esprit dynamique des logiques du maître polonais[1]. Denis Miéville était de ceux qui ont le talent de transmettre et de partager leurs passions intellectuelles. Je lui dois l'essentiel de ma formation de logicien et ce qui restera pour moi les

[1] Cf. Miéville (1984, 2001, 2004 et 2009).

meilleurs moments de mon parcours de chercheur. Pour lui rendre hommage, je voudrais ici lui dédier les premiers résultats d'un projet que nous avions formé à l'Université de Neuchâtel, il y déjà de nombreuses années : reconstruire les logiques de Leśniewski sous la forme de systèmes de déduction naturelle. Les résultats présentés ici constituent la première étape de ce projet : un système de déduction naturelle pour la Protothétique, c'est-à-dire le calcul des propositions étendu de Leśniewski. J'aurais évidemment aimé pouvoir partager avec Denis Miéville cette recherche et discuter les moyens de l'élargir à l'Ontologie et à la Méréologie ; sa disparition subite en aura malheureusement décidé autrement.

2. La Protothétique

Comme le savent les lecteurs de Denis Miéville, la Protothétique est une logique propositionnelle *étendue*[2]. Dit autrement, c'est une logique où la généralité n'est pas exprimée par des variables libres, mais par des expressions universellement quantifiées. Par exemple, plutôt que d'exprimer le principe de non-contradiction par une formule telle que $\sim (p \wedge \sim p)$, ou par le schéma qui en donne la forme $\sim (\alpha \wedge \sim \alpha)$, Leśniewski l'exprimait par une formule tout à fait explicite du langage-objet : $(\forall p) \sim (p \wedge \sim p)$. Mais là ne s'arrête pas le caractère étendu de la Protothétique. Dans l'expression qui précède seules les propositions sont représentées par des variables. Leśniewski voulait également disposer de variables de connecteurs ou de foncteurs. Ces variables devaient également être quantifiables, afin de pouvoir exprimer des généralités portant sur ces connecteurs ou foncteurs, comme par exemple dans :

$$\sim (\forall fpq)\big(f(pq) \supset f(qp)\big)$$

qui exprime que les connecteurs binaires ne sont pas tous commutatifs. Cependant, l'ambition de Leśniewski était plus large encore. Il refusait de borner son langage aux seuls connecteurs unaires et binaires, comme dans le calcul classique, et même de le limiter à une quelconque liste finie de familles de foncteurs, toujours arbitraire à ses yeux. Ainsi, il fallait pouvoir accéder à n'importe quel type de fonction concevable dans un langage propositionnel : connecteurs unaires, binaires, *n*-aires, mais aussi foncteurs prenant ces connecteurs pour arguments, et fournissant à leur tour les arguments de foncteurs de types supérieurs et ainsi de suite, sans limite de

[2] Sur la Protothétique, cf. Miéville (1984 et 2001), Srzednicki et Stachniak (1998). Les œuvres de Leśniewski ont été traduites en anglais (Leśniewski 1992), et pour une part en français (Leśniewski 1989).

complexité. Pour maîtriser cette hiérarchie ouverte et se prémunir de tout paradoxe, Leśniewski développa au début des années 20 la notion de *catégorie sémantique*, par laquelle tout foncteur est caractérisé par trois critères : 1) le nombre de ses arguments ; 2) les catégories respectives de ceux-ci ; 3) la catégorie du résultat de l'application du foncteur à ses arguments. La conjonction, par exemple, relève ainsi de la catégorie des foncteurs à deux arguments propositionnels et produisant une proposition. Dans la notation commode introduite par K. Ajdukiewicz (1935), cette catégorie reçoit l'index *S/SS*. La négation reçoit de même l'index *S/S* (celle des foncteurs formateurs de proposition à un seul argument propositionnel). Un index plus complexe comme (*S/SS*)/(*S/SS*) désigne la catégorie des foncteurs formateurs de connecteur binaire à partir d'un connecteur binaire comme unique argument. La potentialité des catégories que visait Leśniewski dans sa Protothétique est aisément générée par les deux clauses suivantes :

(1) S est une catégorie (la catégorie de base des expressions propositionnelles) ;

(2) si $\alpha, \beta_1, \cdots, \beta_n$ sont des catégories, alors $\alpha/\beta_1 \cdots \beta_n$ est une catégorie.

Bien entendu, aucun langage ne peut exprimer de façon actuelle des foncteurs pour l'infinité des catégories sémantiques générées par ces clauses. La Protothétique fut alors conçue comme un système développemental, c'est-à-dire un système ouvert, donnant un accès potentiel à l'expression de foncteurs de toute catégorie sémantique possible. Dans un système développemental les définitions sont exprimées par des thèses du langage-objet et ces thèses sont inscrites sur la base d'une règle ou directive dédiée[3]. Cela signifie non seulement que les définitions sont rédigées exclusivement avec les moyens linguistiques du langage-objet, mais aussi que les constantes définies intègrent officiellement le langage formel. Un système développemental n'est donc pas entièrement déterminé dès le départ, mais permet la construction réglée et potentielle de différents *développements*, générés par l'inscription successive de définitions et de thèses. Dans les systèmes originaux de Leśniewski, l'état initial du système est constitué par l'inscription des axiomes et la description d'une série de directives, rassemblant des règles d'inférence classiques et des directives développementales, dont en particulier celle de définition. Une conséquence évidente de cette manière inhabituelle de concevoir le formalisme logique est que, en dehors des axiomes, il n'y a de thèse dans ces systèmes que relativement à un développement donné. Cette remarque vaut également pour la notion d'*expression bien formée* (désormais *ebf*), car le langage se

[3]Sur le statut des définitions chez Leśniewski, cf. Joray (2006).

trouve enrichi à chaque fois qu'une définition se trouve inscrite. Dans une syntaxe standard, la notion d'ebf est déterminée dès le départ. Leśniewski ne pouvait s'en contenter. Il lui fallait une syntaxe ouverte et contextuelle, permettant de déterminer dans tout développement si une suite de signes donnée constitue une ebf de telle ou telle catégorie. A titre d'illustration considérons un développement incluant la thèse suivante :

$(\forall p) \sim (p \wedge \sim p)$.

De cette thèse, la règle d'inférence dite de *substitution* doit nous permet dans la Protothétique d'inférer de nouvelles thèses obtenues en remplaçant les occurrences de p par une ebf de la même catégorie sémantique. On pourrait alors vouloir appliquer à la thèse donnée, la substitution

$p \,/\, (\forall qr)(q \vee r)$

et donc inscrire la thèse supplémentaire suivante :

$\sim \big((\forall qr)(q \vee r) \wedge \sim (\forall qr)(q \vee r)\big)$.

Cependant, la règle de substitution ne sera respectée ici que si ce qui est substitué est une ebf de la catégorie *S dans le développement en question*. En particulier, il faut que la constante $\vee$, de catégorie *S/SS*, ait été préalablement définie. En plus de garantir la validité d'une inférence, la règle intervient comme un test : une suite de signes étant avancée, elle doit tester sa bonne formation dans le développement actuel du système. Les directives et règles de Leśniewski présentent ainsi une complexité supérieure à celles des systèmes standards : elles doivent non seulement assurer la préservation du vrai, mais aussi la bonne formation linguistique des nouvelles inscriptions. Comme Denis Miéville l'a exposé, Leśniewski parvint à élaborer des solutions formelles pleinement rigoureuses répondant à ces requisits et évitant toute confusion ou ambiguïté dans les développements de ses systèmes. Le résultat est brillant et la Protothétique est sans doute la logique propositionnelle la plus achevée et la plus complète jamais conçue.

3. La Protothétique en déduction naturelle

Mon projet consiste à donner de la Protothétique une version en déduction naturelle dans le style des logiques de S. Jaśkowski (1934) et de F. B. Fitch (1952)[4]. La difficulté est alors la suivante : comment se passer des axiomes et reporter une fois de plus sur les règles et directives les rôles qu'ils jouent dans le système original, à savoir caractériser les constantes

[4] Pour une présentation pédagogique et en français de la déduction naturelle de Fitch, cf. Grize (1969).

primitives et fournir les premières ebf à partir desquelles peuvent s'appliquer les tests de bonne formation ? La solution que j'apporte ici présente forcément une certaine complexité. Mais le pari que je relève est qu'une fois cette complexité posée, on obtient un système qui peut être exposé par des voies suggestives qui le rendent accessible à quiconque maîtrise les règles usuelles de la déduction naturelle. Il peut sembler paradoxal que la complexité théorique soit nécessaire à la facilité d'usage, mais il faut garder à l'esprit qu'il en va de même de la déduction naturelle classique : bien que d'une grande simplicité d'usage, la complexité de sa métathéorie dépasse à bien des égards celle des systèmes axiomatiques.

De façon très générale, le système que je présente ici (nommé PND – de l'anglais *Protothetic in Natural Deduction*) est constitué d'un appareillage formel permettant la construction explicite de suites ordonnées finies d'inscriptions (dites *développements*) dont chaque élément est inscrit sur la base des développements qui précèdent et en vertu

(a) soit d'une directive d'inscription de définition,

(b) soit d'une preuve explicite construite à l'aide des règles d'inférence du système.

Chaque inscription dans un développement est dite *thèse* de ce développement. Une thèse sera dite *définition* du développement si son inscription se fait selon (a) et *théorème* du développement si son inscription se fait selon (b). Comme un même système peut recevoir des développements différents et même divergents (en vertu des définitions posées), ce qui caractérise le système *lui-même* est son état initial (ou développement 0), à savoir un langage primitif, la directive de définition et une liste de règles d'inférence.

3.1 Le langage de PND

Comme la Protothétique originale, PND est une logique doublement étendue, c'est-à-dire quantifiée et potentiellement ouverte à n'importe quelle catégorie sémantique conçue à partir de l'unique catégorie de base *S*. Toujours comme la Protothétique, PND s'appuie sur un langage primitif extrêmement modeste, comprenant, outre le quantificateur universel, une seule constante primitive : le biconditionnel $\equiv$. Les deux catégories sémantiques présentes à l'état initial sont donc : *S* et *S*/*SS*. Pour reprendre un moyen commode imaginé par Denis Miéville (2001, 58), ces caractéristiques initiales seront posées par le biais d'une *bibliothèque catégorielle*. Une bibliothèque catégorielle est constituée de la liste des catégories disponibles à tel état de développement, chacune (sauf la

catégorie de base) étant associée à un *contexte* et à la liste des constantes disponibles. La bibliothèque initiale de PND est la suivante :

$B_0[S; S/SS : (--) : \equiv]$.

Pour rendre efficiente l'idée de bibliothèque, il convient de poser dès le stade initial certaines conventions d'application et certains éléments terminologiques. Pour la clarté de l'exposé, je me contenterai ici de le faire sur la base d'une simple illustration. L'inscription suivante servira d'exemple de ce qui sera considéré comme une ebf de catégorie *S* dès le stade initial :

Exemple 1 : $\lfloor pq \rfloor\lceil \equiv \big(p \equiv (pq)\big) \rceil$

D'abord spécifions que toute inscription dans PND sera une suite (finie ordonnée) de symboles issus d'un certain vocabulaire général, caractérisé par les quatre familles de symboles suivants :

1. les lettres latines minuscules (avec ou sans indices), dites *variables* ;
2. les quatre symboles spéciaux $\lfloor, \rfloor, \lceil, \rceil$, dits *délimitateurs* ;
3. les symboles usuels de parenthèses (hormis les délimitateurs), dits *parenthèses* ;
4. les symboles qui ne tombent pas sous les clauses 1-3 et qui ne sont pas des signes usuels de ponctuation, dits *constantes*.

En second lieu, on notera que le langage visé s'appuie, comme dans la notation polonaise de Łukasiewicz, sur une écriture préfixée : les foncteurs sont inscrits en tête de leurs arguments. Cependant, contrairement à la notation polonaise, les parenthèses sont conservées. Ainsi dans l'exemple 1, la partie d'inscription $\equiv \big(p \equiv (pq)\big)$ serait notée $p \equiv (p \equiv q)$ en écriture usuelle et $\equiv p \equiv pq$ en écriture polonaise. Dans l'écriture lesniewskienne adoptée pour PND, lorsqu'il y a application d'un foncteur à ses arguments, le symbole (ou expression) de foncteur est toujours bien repérable par le fait qu'il précède immédiatement son ou ses arguments insérés entre deux parenthèses symétriques d'une certaine forme. On dit que le foncteur précède immédiatement son *contexte*. Pour éviter toute ambiguïté, les foncteurs de catégories différentes seront associés à des contextes différents (soit par la forme des parenthèses symétriques, soit par le nombre des arguments présents entre les parenthèses). Dans l'exemple 1, le seul contexte apparaissant est de la forme$(--)$. Comme l'indique la bibliothèque initiale B_0, il s'agit du contexte propre à la catégorie *S/SS*. La bibliothèque indique aussi que le biconditionnel $\equiv$ est de cette catégorie. Cela signifie que pour l'appliquer, il faut le mettre, comme dans l'exemple 1, devant le contexte $(--)$. L'index catégorielle *S/SS* nous indique qu'une inscription

comme $\equiv (pq)$ est de catégorie *S*, et que les deux arguments p et q sont de catégorie *S*.

En troisième lieu, soulignons que la quantification universelle est exprimée à l'aide des quatre délimitateurs, par la forme $\lfloor \cdots \rfloor \lceil - \rceil$. Une inscription quantifiée comme l'exemple 1 est dite *généralisation*. La partie $\lfloor pq \rfloor$ en est le *quantificateur* et les occurrences de variables qui y figurent en sont les *lieurs*. La partie $\lceil \equiv (p \equiv (pq)) \rceil$ est le *sous-quantificateur* de cette généralisation et l'inscription qui y figure en est l'*essence*. Il est attendu d'une généralisation qu'elle soit de catégorie *S* et que son essence soit également de catégorie *S*. Par contre, les lieurs peuvent être de n'importe quelles catégories disponibles dans la bibliothèque en cours. C'est ce qu'on voit dans l'exemple 2, ci-dessous :

Exemple 2 : $\lfloor qf \rfloor \lceil \equiv (p\ f(pq)) \rceil$

Dans cet exemple, les lieurs q et f du quantificateur sont respectivement des catégories *S* et *S/SS*. La catégorie de *q* est déterminée par le fait qu'une variable équiforme est dans l'essence en seconde position dans un contexte $(--)$ et celle de *f* par le fait qu'une variable équiforme est dans l'essence en position de foncteur devant ce même contexte. La distinction entre occurrences de variables *libres* et *liées*, celle entre expressions *ouvertes* et *fermées*, ainsi que la notion de *fermeture* d'une expression s'appliquent exactement comme dans les langages standards. L'exemple 2 est une formule ouverte puisqu'elle contient deux occurrences libres de la variable *p*. En ajoutant cette variable aux lieurs de la généralisation, on obtiendrait la fermeture de l'expression. On notera encore que, puisque la catégorie des variables ou des expressions est déterminée par leurs positions dans ou devant tel ou tel contexte, on exigera (sous peine d'indétermination catégorielle) : (1) que tout lieur d'une généralisation lie au moins une occurrence de variable dans l'essence de la généralisation ; (2) que les occurrences équiformes d'une variable dans une même ebf soient toutes de la même catégorie.

3.2 Les définitions dans PND

Comme suggéré plus haut, la bibliothèque d'un développement s'élargira chaque fois que sera posée une définition. Pour voir quels sont dans PND les mécanismes de cet élargissement, il convient maintenant de formuler la directive de définition. Par soucis de clarté, je présente ici une version simplifiée, suffisante pour comprendre la mécanique essentielle de PND :

Directive de définition (version simplifiée)
L'expression T peut être inscrite comme nouvelle thèse d'un développement *d* si T est la fermeture d'une expression T' telle que

1. T' est une expression biconditionnelle,
   $\equiv (- -)$
2. le premier argument de T' (dit *definiendum*) est soit une constante seule, soit une constante appliquée à un contexte,
   $\equiv (\$ -)$ ou $\equiv (\$\langle - \cdots - \rangle -)$
3. si le *definiendum* présente un contexte, toutes les places d'argument de ce contexte sont occupées par des variables, différentes les unes des autres,
   $\equiv (\$\langle v_1 \cdots v_n \rangle -)$
4. la constante du *definiendum* ne figure pas dans la bibliothèque du développement *d*,
5. le second argument de T' (dit *definiens*) est une ebf de catégorie *S* dans le développement *d*,
   $\equiv (\$\langle v_1 \cdots v_n \rangle \; Ebf \;)$
6. le *definiendum* et le *definiens* ont les mêmes variables libres (aucune, si le *definiendum* n'a pas de contexte),
   $\equiv (\$\langle v_1 \cdots v_n \rangle \; Ebf_{[v_1 \cdots v_n]})$
7. si le *definiendum* a un contexte et que celui-ci figure dans la bibliothèque du développement *d*, alors chaque variable libre de T' est telle qu'elle a dans le *definiens* la même catégorie que celle qui est attendue pour la place que cette variable occupe dans le contexte du *definiendum*.
8. si le *definiendum* a un contexte et que celui-ci ne figure pas dans la bibliothèque du développement *d*, alors : l'expression T' modifiée par le remplacement de la constante du *definiendum* par une variable qui n'apparaît pas dans T' et par le remplacement des parenthèses du contexte par celles d'un contexte qui figure dans la bibliothèque de *d* n'est pas une ebf du développement *d*.

L'inscription de T comme nouvelle thèse conduit au développement *d*+1. La bibliothèque B_{d+1} est obtenue à partir de B_d de la façon suivante : (a) si le *definiendum* n'a pas de contexte, la constante définie est ajoutée à la catégorie *S* ; (b) si le contexte du *definiendum* figure dans B_d, alors la constante définie est ajoutée à la catégorie à laquelle ce contexte est associé ; (c) si le contexte du *definiendum* ne figure pas dans B_d, alors la catégorie de la constante définie (déterminée sur la base des occurrences des variables libres dans le *definiens*) est ajoutée, en lui associant le contexte choisi dans le *definiendum* et la constante définie elle-même.

Cette directive appelle deux commentaires. D'abord, il faut souligner que la thèse T inscrite comme définition est la *fermeture* de l'expression T' qui répond aux 8 conditions. Elle a donc l'une des deux formes schématiques suivantes :

$\equiv (\$\, Ebf)$ ou $\lfloor v_1 \cdots v_n \rfloor\lceil \equiv \left(\$\langle v_1 \cdots v_n \rangle\, Ebf_{[v_1 \cdots v_n]}\right)\rceil$.

Ensuite, il convient d'indiquer que les clauses 7 et 8 sont celles qui imposent que la définition d'une constante se fasse avec un choix de contexte cohérent avec ce que le développement contient préalablement (utilisation du contexte disponible si la catégorie existe déjà, choix convenable d'un contexte nouveau sinon). Voyons maintenant quelques exemples d'utilisation de cette directive. Partant de la bibliothèque initiale, j'indiquerai à chaque étape comment la bibliothèque du développement obtenu se trouve élargie par la définition inscrite.

Etat initial $B_0[S; S/SS : (--) : \equiv]$

T1 : $\equiv (\top \lfloor p \rfloor\lceil \equiv (p\ p)\rceil)$ *Définition*

$B_1[S: \top; S/SS : (--) : \equiv]$

T2 : $\equiv (\bot\ \lfloor q \rfloor\lceil q \rceil)$ *Définition*

$B_2[S: \top, \bot; S/SS : (--) : \equiv]$

Les thèses T1 et T2 définissent deux constantes d'une catégorie existante : la catégorie *S*. T1 introduit une constante pour le vrai, T2 une constante pour le faux. Ces constantes sont alors ajoutées en bibliothèque à la catégorie *S*. Voyons maintenant la définition d'une constante pour une catégorie nouvelle : *S/S*.

T3 : $\lfloor p \rfloor\lceil \equiv (\sim (p) \equiv (p\ \bot))\rceil$ *Définition*

$B_3[S: \top, \bot; S/SS : (--) : \equiv; S/S : (-) : \sim]$

Il s'agit de la négation propositionnelle, la catégorie, le contexte associé et la constante sont ajoutés en bibliothèque. Notons que le contexte choisi est différent de celui de *S/SS* ; les parenthèses sont les mêmes, mais le nombre de place d'argument est différent. Déterminer que la nouvelle catégorie de la constante définie est *S/S* se fait par analyse de l'inscription : la variable p apparaît en première position d'un contexte $(--)$ et est donc de catégorie *S*. Comme le *definiendum* doit dans son entier être de catégorie *S*, que la nouvelle constante ne s'applique qu'au seul argument p, elle est bien de catégorie *S/S*.

T4 : $\lfloor pq \rfloor\lceil \equiv \left(\mathrm{W}(pq)\ \sim \left(\equiv (pq)\right)\right)\rceil$ *Définition*

$B_4[S: \top, \bot; S/SS : (--) : \equiv, \mathrm{W}; S/S : (-) : \sim]$

La définition T4 introduit la disjonction exclusive. Comme il s'agit d'une constante de la catégorie *S/SS* déjà à disposition, le choix du contexte nous

est imposé par la bibliothèque en cours et la constante définie vient s'ajouter en bibliothèque. Voyons pour terminer un exemple de définition qu'autoriserait la version non simplifiée de la directive de définition :

$$\text{T5} : \lfloor fpq \rfloor \left\lceil \equiv \left(Dual\langle f \rangle (pq) \sim \left(f\left(\sim (p) \sim (q) \right) \right) \right) \right\rceil \textit{Définition}$$

$$B_5[S: \top, \bot; S/SS : (--) : \equiv, \mathrm{W}; S/S : (-) : \sim; (S/SS)/(S/SS) : \langle - \rangle : Dual]$$

La constante définie n'est pas ici inscrite devant un, mais devant deux contextes successifs. Le résultat de l'application de la constante au premier contexte, exprimé par $Dual\langle f \rangle$, est à comprendre comme un foncteur complexe (dit *paramétré*) s'appliquant ensuite au second contexte $(--)$. Conformément à la bibliothèque en cours, l'inscription $Dual\langle f \rangle$ est donc de catégorie *S/SS*. Comme l'unique argument *f* apparaît en position de foncteur *S/SS* dans le *definiens*, il suit alors que la constante '*Dual*' est de la catégorie (*S/SS*)/(*S/SS*). Cette nouvelle catégorie est donc inscrite dans la bibliothèque, accompagnée du contexte $\langle - \rangle$ et de la nouvelle constante. Les catégories qui ne présentent pas un unique *S* au numérateur de leur index sont introduites de cette manière. Elles sont nommées *catégories de foncteurs paramétrés*. Une directive de définition complète doit en tenir compte, mais comme je ne fais pas ici usage de foncteurs paramétrés, je me suis contenté d'en présenter une version simplifiée[5].

3.3 Preuves et théorèmes dans PND

Il convient maintenant d'examiner comment inscrire des théorèmes dans un développement de PND. L'idée est la suivante : un certain développement *d* étant déjà construit, on avancera une nouvelle inscription et celle-ci ne recevra le statut de nouvelle thèse que lorsque sera exhibée une preuve dans *d* dont l'inscription avancée sera la dernière ligne. Il convient donc de caractériser ce qu'est une *preuve* dans un développement et, pour ce faire, on s'appuiera sur la caractérisation de la notion plus générale de *déduction* dans un développement de PND.

Déduction
Une *déduction* dans un développement *d* est une suite finie ordonnée d'ebf de catégorie *S* de *d*, telle que chacune des ebf est soit une thèse de *d*, soit une hypothèse (c'est-à-dire une ebf quelconque de catégorie *S* de *d*), soit une inscription obtenue par l'application d'une règle d'inférence de PND à des ebf qui précèdent dans la déduction.

[5] Pour la version non simplifiée, cf. Miéville (2001, 76-77).

Preuve
Une *preuve* dans un développement *d* est une déduction dans *d* dont la dernière ebf est fermée et ne dépend d'aucune hypothèse.

Pour rendre ces deux notions efficientes, il me faut maintenant exposer les règles d'inférence de PND. Elles sont au nombre de six (une règle structurelle, trois relatives au biconditionnel et deux relatives au quantificateur universel). Il faut encore préciser que lorsqu'une hypothèse *h* est inscrite dans une déduction, toutes les ebf qui sont inscrites à sa suite (y compris elle-même) sont dites *dépendre* de cette hypothèse *h* ; cette dépendance a une portée qui s'étend jusqu'à la première ebf (non comprise) inscrite en vertu d'une règle qui *décharge* l'hypothèse *h*. Pour clarifier, à l'usage, les rapports de dépendance dans une déduction, on utilisera à la manière de Fitch (1952) des barres verticales afin de visualiser la portée de la dépendance aux hypothèses dans les déductions. Ainsi, à chaque inscription d'une hypothèse, on commencera une nouvelle barre qui ne se terminera qu'une fois l'hypothèse déchargée. Une preuve se reconnaîtra alors par le fait que sa dernière ebf est fermée et qu'elle ne sera sous le coup d'aucune barre de dépendance.

3.4 Règles d'inférence de PND

Dans cette section, chaque règle de PND sera donnée par une caractérisation précise, puis par une version schématique reprenant le graphisme visuel de Fitch. On notera que dans les versions précises, il faudra toujours comprendre les notions de *déduction*, d'*ebf* et d'*appartenance à une catégorie* comme relatives au développement où la règle se trouve appliquée.

Règle *reit*
Dans une déduction une ebf *E* inscrite au rang *m* peut être répétée à un rang *n* supérieur à *m*, pour autant qu'aucune hypothèse dont dépend *E* au rang *m* ne soit déchargée jusqu'au rang *n*.

Schématiquement :

m. | *E*
| | *hyp*
| |
n. | | *E* *m*, reit

Règle $\equiv e$

Dans une déduction, si une ebf E se trouve au rang l et une ebf de la forme $\equiv (EF)$ ou de la forme $\equiv (FE)$ se trouve au rang m, alors l'ebf F peut être inscrite à un rang n supérieur à l et à m, pour autant que les trois inscriptions des rangs l, m et n dépendent des mêmes hypothèses.

Schématiquement :

$l.$	E	
$m.$	$\equiv (EF)$ ou $\equiv (FE)$	
	⋯	
$n.$	F	$l, m, \equiv e$

Règle $\equiv i$

Dans une déduction, **si** toutes les conditions suivantes sont remplies :

1. une hypothèse E est inscrite au rang l,
2. une ebf F est inscrite deux fois aux rangs m et m+1 supérieurs à l,
3. l'ebf F en m+1 est inscrite comme hypothèse,
4. une ebf E (équiforme à la première hypothèse en l) est inscrite au rang n supérieur à m+1,
5. les ebf des rangs l et m dépendent des mêmes hypothèses,
6. les ebf des rangs m+1 et n dépendent des mêmes hypothèses,
7. les hypothèses dont dépendent les ebf des rangs m+1 et n sont celles dont dépendent celles des rangs l et m, auxquelles s'ajoute uniquement l'hypothèse F en m+1,
8. aucune ebf entre les rangs l et m (y compris) ne se trouve répétée par *reit* au-delà du rang m,

alors

les deux hypothèses E au rang l et F au rang m sont déchargées au-delà du rang n et on peut inscrire au rang n+1 l'expression qui leurs correspond de la forme $\equiv (EF)$.

Schématiquement :

$l.$	E	*hyp*
	⋮	
$m.$	F	
m+1.	F	*hyp*
	⋮	
$n.$	E	
n+1.	$\equiv (EF)$	l-m, m+1-n, $\equiv i$

On notera que l'énoncé précis de la règle $\equiv i$ présente la dépendance de la seconde hypothèse comme emboîtée dans celle de la première. Le schéma les représente néanmoins comme des portées successives. Cette simplification est autorisée par la clause 8, qui interdit toute réitération du premier espace vers le second.

Règle $\equiv ass$
Dans une déduction, si une ebf de la forme $\equiv (E \equiv (FG))$ (respectivement $\equiv (\equiv (EF)\, G)$) est inscrite au rang *m*, alors l'ebf qui lui correspond de la forme $\equiv (\equiv (EF)\, G)$ (respectivement $\equiv \big(E \equiv (FG)\big)$) peut être inscrite au rang *n* supérieur à *m*, pour autant que les ebf inscrites aux rangs *m* et *n* dépendent exactement des mêmes hypothèses.

Schématiquement :

m.	$\equiv \big(E \equiv (FG)\big)$		*m.*	$\equiv (\equiv (EF)G)$	
	⋯	OU		⋯	
n.	$\equiv (\equiv (EF)G)$	$m, \equiv ass$	*n.*	$\equiv \big(E \equiv (FG)\big)$	$m, \equiv ass$

Règle $\lfloor\ \rfloor e$
Dans une déduction, si une généralisation *E* dont *v* est un lieur de catégorie *c* est inscrite au rang *m* et si *G* est une ebf de catégorie *c*, alors on peut inscrire au rang *n* supérieur à *m* l'expression *E*' obtenue à partir de *E* par effacement du lieur *v* et par remplacement de toutes les occurrences libres de *v* dans l'essence de *E* par l'ebf *G*, pour autant que toutes les occurrences de variables libres dans *G* restent libres dans l'expression *E*', que toutes les occurrence d'une même variable dans *E*' soient d'une même catégorie et que *E* au rang *m* et *E*' au rang *n* dépendent des mêmes hypothèses. Si l'expression *E*' est sans lieur après les modifications apportées à *E*, *E*' sera remplacée par son essence.

Schématiquement :

m.	$\lfloor v_1 \cdots v_i v \rfloor \lceil F \rceil$	ou $\lfloor v \rfloor \lceil F \rceil$		
	⋯			
n.	$\lfloor v_1 \cdots v_i \rfloor \lceil F_{[v/G]} \rceil$	ou $F_{[v/G]}$	$m, \lfloor\ \rfloor e$	

où - la variable *v* dans *F* et l'ebf *G* sont de la même catégorie sémantique,
- les occurrences de variables libres dans *G* restent libres dans le résultat de la substitution,

- les occurrences d'une même variable dans $F_{[v/G]}$ sont toutes de même catégorie.

Règle $\lfloor\ \rfloor i$
Dans une déduction, si une ebf *E* contenant au moins une occurrence libre de la variables *v* est inscrite au rang *m* et ne dépend d'aucune hypothèse dans laquelle *v* est en occurrence libre, alors on peut inscrire à un rang *n*, supérieur à *m*, soit l'expression *E*' obtenue par ajout de *v* aux lieurs de *E*, si *E* est une généralisation, soit la généralisation *F* dont *v* est l'unique lieur et dont *E* est l'essence, si *E* n'est pas une généralisation, pour autant que les inscriptions aux rangs *m* et *n* dépendent des mêmes hypothèses.

Schématiquement :

m. $\lfloor v_1 \cdots v_i \rfloor\lceil E \rceil$ ou E
⋮
n. $\lfloor v_1 \cdots v_i v \rfloor\lceil E \rceil$ ou $\lfloor v \rfloor\lceil E \rceil$ *m*, $\lfloor\ \rfloor i$

où - *E* contient au moins une occurrence libre de *v*,
- l'ebf en ligne *m* ne dépend d'aucune hypothèse contenant *v* libre.

3.5 Un développement de PND

Jusqu'ici cinq thèses ont été inscrites, qui sont toutes des définitions. La construction qui suit ajoute des thèses supplémentaires à cette base. Le développement obtenu permettra de montrer qu'il est possible de prouver la fermeture universelle de n'importe quelle tautologie du calcul des propositions standard dans PND. Si la notation n'est pas habituelle, les moyens de déduction sont absolument similaires à ceux, bien connus, des systèmes de Jaśkowski-Fitch.

T6 : $\lfloor p \rfloor\lceil \equiv (pp) \rceil$
Preuve :

1.	p	*hyp*
2.	p	1, *reit*
3.	p	*hyp*
4.	p	3, *reit*
5.	$\equiv (pp)$	1-2, 3-4, $\equiv i$
6.	$\lfloor p \rfloor\lceil \equiv (pp) \rceil$	5, $\lfloor\ \rfloor i$

Lorsque les preuves sont brèves et ne requièrent pas d'hypothèse, on se contentera, comme ci-dessous, d'une simple indication pour leurs constructions.

T7 : $\top$ Preuve : T1, T6, $\equiv e$
T8 : $\sim(\bot)$ Preuve : T3($\lfloor\ \rfloor e,\ p/\bot$),T6 ($\lfloor\ \rfloor e,\ p/\bot$), $\equiv e$

Lorsqu'une preuve ou une déduction présente une séquence remarquable, dont il pourrait être utile d'user à nouveau, on en dégagera un schéma utilisable dans les déductions ultérieures au titre de *règle dérivée*. La déduction suivante est même construite à cette seule fin :

Déduction :

1.	$\equiv (pq)$	*hyp*
2.	$\quad q$	*hyp*
3.	$\quad \equiv (pq)$	1, *reit*
4.	$\quad p$	2, 3, $\equiv e$
5.	$\quad p$	*hyp*
6.	$\quad \equiv (pq)$	1, reit
7.	$\quad q$	5, 6, $\equiv e$
8.	$\equiv (qp)$	2-4, 5-7, $\equiv i$

Règle $\equiv com$ (dérivée de 1-8)

n.	$\equiv (EF)$	
	...	
	$\equiv (FE)$	$n, \equiv com$

Le lecteur montrera sans difficulté qu'on peut dégager d'une manière similaire les deux autres règles dérivées suivantes :

Règle $\equiv syll$ (dérivée)

m.	$\equiv (EF)$	
n.	$\equiv (FG)$	
	...	
	$\equiv (EG)$	$m, n, \equiv syll$

Règle $\equiv conj$ (dérivée)

m.	E	
n.	F	
	...	
	$\equiv (EF)$	$m, n, \equiv conj$

T9 : $\lfloor pq \rfloor\lceil \equiv \big(\equiv (pq) \equiv (qp)\big)\rceil$
Preuve :

1.	$\equiv (pq)$	*hyp*
2.	$\equiv (qp)$	1, $\equiv com$
3.	$\equiv (qp)$	*hyp*
4.	$\equiv (pq)$	3, $\equiv com$
5.	$\equiv (\equiv (pq) \equiv (qp))$	1-2, 3-4, $\equiv i$
6.	$\lfloor pq \rfloor\lceil \equiv \big(\equiv (pq) \equiv (qp)\big)\rceil$	5, $\lfloor\ \rfloor i$

Je vais maintenant montrer que les deux règles caractéristiques de la négation propositionnelle classique sont accessibles dans le développement, sur la base de la définition posée en T3.

Déduction :

1.	$\sim(\sim(p))$	*hyp*
2.	$\equiv(\sim(p)\ \perp)$	1, T3 $(\lfloor\ \rfloor e,\ p/\sim(p)), \equiv e$
3.	$\equiv(\sim(p)\ \equiv(p\ \perp))$	T3 $(\lfloor\ \rfloor e,\ p/p)$
4.	$\equiv(\equiv(p\ \perp)\sim(p))$	3, $\equiv com$
5.	$\equiv(\equiv(p\ \perp)\ \perp)$	4, 2, $\equiv syll$
6.	$\equiv(p\ \equiv(\perp\perp))$	5, $\equiv ass$
7.	p	6, T6 $(\lfloor\ \rfloor e,\ p/\perp), \equiv e$

Règle $\sim\sim e$ (dérivée de 1-7)

n. $\sim(\sim(E))$
…
E *n*, $\sim\sim e$

Pour la règle d'introduction (qui est de forme hypothétique), on la dérive par une démarche un peu différente. Soit E une ebf de catégorie *S*. Si on peut construire dans une déduction une séquence de la forme suivante :

m. E *hyp*
⋮
n. $\perp$

alors il sera toujours possible de la compléter par les lignes suivantes, d'où la règle dérivée d'introduction de la négation :

⋮

n+1.	$\perp$	*hyp*
n+2.	$\lfloor q\rfloor\lceil q\rceil$	T2, *n*+1, $\equiv e$
n+3.	E	*n*+2, $\lfloor\ \rfloor e,\ q/E$
n+4.	$\equiv(E\ \perp)$	*m*-*n*, *n*+1-*n*+3, $\equiv i$
n+5.	$\sim(E)$	T3 $(\lfloor\ \rfloor e,\ p/E), n$+4, $\equiv e$

Règle $\sim i$ (dérivée)

m. E *hyp*
⋮
n. $\perp$
$\sim(E)$ *m*-*n*, $\sim i$

T10 : $\lfloor p\rfloor\left\lceil\sim\left(\equiv\left(p\ \sim(p)\right)\right)\right\rceil$

Preuve :

1.	$\equiv(p\ \sim(p))$	*hyp*
2.	$\equiv(\sim(p)\ \equiv(p\ \perp))$	T3 $(\lfloor\ \rfloor e,\ p/p)$
3.	$\equiv(p\ \equiv(p\ \perp))$	1, 2, $\equiv syll$
4.	$\equiv(\equiv(pp)\ \perp)$	3, $\equiv ass$
5.	$\perp$	T6 $(\lfloor\ \rfloor e,\ p/p), 4, \equiv e$
6.	$\sim(\equiv(p\ \sim(p)))$	1-5, $\sim i$
7.	$\lfloor p\rfloor\lceil\sim(\equiv(p\ \sim(p)))\rceil$	6, $\lfloor\ \rfloor i$

La thèse suivante est la définition de la conjonction élaborée par Alfred Tarski dans sa thèse de doctorat (1923). Cette définition est loin d'être triviale et la suite du développement proposé aura pour but d'en montrer l'adéquation, par la dérivation des règles d'introduction et d'élimination caractéristiques de la conjonction[6].

T11 : $\lfloor pq \rfloor \lceil \equiv \left(\wedge (pq) \ \lfloor f \rfloor \lceil \equiv \left(p \equiv \left(\lfloor r \rfloor \lceil \equiv (p\, f(r)) \rceil \lfloor r \rfloor \lceil \equiv (q\, f(r)) \rceil \right) \right) \right) \rceil$ *Définition*

$B_{11}[S{:}\top, \bot; S/SS : (--) : \equiv, \mathrm{W}, \ \wedge ; S/S : (-) : \sim ; (S/SS)/(S/SS) : \langle - \rangle : Dual]$

Déduction :

1.	p	*hyp*	
2.	q	*hyp*	
3.	$\lfloor r \rfloor \lceil \equiv (p\, f(r)) \rceil$	*hyp*	
4.	p	1, *reit*	
5.	$f(r)$	3 $(\lfloor\ \rfloor e,\ r/r), 4,\ \equiv e$	**Règle** $\wedge i$ (dérivée de 1-18)
6.	q	2, *reit*	*m.* E
7.	$\equiv (q\, f(r))$	5, 6, $\equiv conj$	*n.* F
8.	$\lfloor r \rfloor \lceil \equiv (q\, f(r)) \rceil$	7, $\lfloor\ \rfloor i$	⋯
9.	$\lfloor r \rfloor \lceil \equiv (q\, f(r)) \rceil$	*hyp*	$\wedge (E\ F)$ *m, n,* $\wedge i$
10.	q	2, *reit*	
11.	$f(r)$	9 $(\lfloor\ \rfloor e,\ r/r), 10,\ \equiv e$	
12.	p	1, *reit*	
13.	$\equiv (p\, f(r))$	11, 12, $\equiv conj$	
14.	$\lfloor r \rfloor \lceil \equiv (p\, f(r)) \rceil$	13, $\lfloor\ \rfloor i$	
15.	$\equiv (\lfloor r \rfloor \lceil \equiv (p\, f(r)) \rceil \lfloor r \rfloor \lceil \equiv (q\, f(r)) \rceil)$		3-8, 9-14, $\equiv i$
16.	$\equiv (p\ \equiv (\lfloor r \rfloor \lceil \equiv (p\, f(r)) \rceil \lfloor r \rfloor \lceil \equiv (q\, f(r)) \rceil))$		1, 15, $\equiv conj$
17.	$\lfloor f \rfloor \lceil \equiv (p\ \equiv (\lfloor r \rfloor \lceil \equiv (p\, f(r)) \rceil \lfloor r \rfloor \lceil \equiv (q\, f(r)) \rceil))]$		16, $\lfloor\ \rfloor i$
18.	$\wedge (pq)$	T11 $(\lfloor\ \rfloor e,\ p/p,\ q/q), 17,\ \equiv e$	

T12 : $\lfloor p \rfloor \lceil \equiv \left(Ver(p) \equiv (pp) \right) \rceil$ *Définition* (Il s'agit du connecteur unaire dit *Verum*)

$B_{12}[S{:}\top, \bot; S/SS : (--) : \equiv, \mathrm{W}, \ \wedge ; S/S : (-) : \sim, Ver\ ; (S/SS)/(S/SS) : \langle - \rangle : Dual]$

T13 : $\lfloor p \rfloor \lceil Ver(p) \rceil$ Preuve : T12 $(\lfloor\ \rfloor e)$, T6 $(\lfloor\ \rfloor e), \equiv e + \lfloor\ \rfloor i$

T14 : $\lfloor p \rfloor \lceil \equiv \left(p\ \lfloor r \rfloor \lceil \equiv \left(p\ Ver(r) \right) \rceil \right) \rceil$ (Lemme pour la déduction à suivre)

[6] Sur l'analyse de la définition de Tarski et la nécessité de disposer, comme ici en T12 avec le *Verum*, d'un second connecteur unaire pour l'exploiter, cf. Joray (2011).

Preuve :

1.	p	*hyp*
2.	$Ver(r)$	T13 ($\lfloor\ \rfloor e,\ p/r$)
3.	$\equiv (p\ Ver(r))$	1, 2, $\equiv$ *conj*
4.	$\lfloor r\rfloor\lceil\equiv (p\ Ver(r))\rceil$	3, $\lfloor\ \rfloor i$
5.	$\lfloor r\rfloor\lceil\equiv (p\ Ver(r))\rceil$	*hyp*
6.	$Ver(r)$	T13 ($\lfloor\ \rfloor e,\ p/r$)
7.	p	5 ($\lfloor\ \rfloor e$), 6, $\equiv e$
8.	$\equiv (p\ \lfloor r\rfloor\lceil\equiv (p\ Ver(r))\rceil\ )$	1-4, 5-7, $\equiv i$
9.	$\lfloor p\rfloor\lceil\equiv (p\ \lfloor r\rfloor\lceil\equiv (p\ Ver(r))\rceil\ )\rceil$	8, $\lfloor\ \rfloor i$

Déduction :

1.	$\wedge (pq)$		*hyp*
2.	$\lfloor f\rfloor\lceil\equiv (p \equiv (\lfloor r\rfloor\lceil\equiv (p\ f(r))\rceil\lfloor r\rfloor\lceil\equiv (q\ f(r))\rceil))\rceil$		T11 ($\lfloor\ \rfloor e,\ p/p,\ q/q$), 1, $\equiv e$
3.	$\equiv (p \equiv (\lfloor r\rfloor\lceil\equiv (p\ Ver(r))\rceil\lfloor r\rfloor\lceil\equiv (q\ Ver(r))\rceil))$		2, $\lfloor\ \rfloor e,\ f/Ver$
4.	$\equiv (\equiv (p\ \lfloor r\rfloor\lceil\equiv (p\ Ver(r))\rceil\ )\ \lfloor r\rfloor\lceil\equiv (q\ Ver(r))\rceil\ )$		3, $\equiv ass$
5.	$\equiv (p\ \lfloor r\rfloor\lceil\equiv (p\ Ver(r))\rceil\ )$	T14 ($\lfloor\ \rfloor e,\ p/p$)	**Règle** $\wedge e_d$ (dérivée 1-8)
6.	$\lfloor r\rfloor\lceil\equiv (q\ Ver(r))\rceil$	4, 5, $\equiv e$	*n.* $\wedge$ $(E\ F)$
7.	$\equiv (q\ \lfloor r\rfloor\lceil\equiv (q\ Ver(r))\rceil\ )$	T14 ($\lfloor\ \rfloor e,\ p/q$)	⋯
8.	q	6, 7, $\equiv e$	F *n*, $\wedge e_d$

T15 : $\lfloor p\rfloor\lceil\equiv (\lfloor r\rfloor\lceil\equiv (p \sim (r))\rceil\ \bot)\rceil$ (Lemme pour la déduction à suivre)

Preuve :

1.	$\lfloor r\rfloor\lceil\equiv (p\ \sim (r))\rceil$	*hyp*
2.	$\equiv (p\ \sim (p))$	1, $\lfloor\ \rfloor e, r/p$
3.	$\sim (\equiv (p\ \sim (p))\ )$	T10 ($\lfloor\ \rfloor e,\ p/p$)
4.	$\equiv (\equiv (p\ \sim (p))\ \bot)$	T3 ($\lfloor\ \rfloor e,\ p/\equiv (p\ \sim (p))\ )$, 3, $\equiv e$
5.	$\bot$	2, 4, $\equiv e$
6.	$\sim (\lfloor r\rfloor\lceil\equiv (p\ \sim (r))\rceil)$	1-5, $\sim i$
7.	$\equiv (\lfloor r\rfloor\lceil\equiv (p\ \sim (r))\rceil\ \bot)$	T3 ($\lfloor\ \rfloor e,\ p/\lfloor r\rfloor\lceil\equiv (p\ \sim (r))\rceil$), 6, $\equiv e$
8.	$\lfloor p\rfloor\lceil\equiv (\lfloor r\rfloor\lceil\equiv (p\ \sim (r))\rceil\ \bot)\rceil$	8, $\lfloor\ \rfloor i$

Déduction :

1.	$\wedge (pq)$		*hyp*
2.	$\lfloor f\rfloor\lceil\equiv (p\ \equiv (\lfloor r\rfloor\lceil\equiv (p\ f(r))\rceil\lfloor r\rfloor\lceil\equiv (q\ f(r))\rceil))\rceil$		T11 ($\lfloor\ \rfloor e,\ p/p,\ q/q$), 1, $\equiv e$
3.	$\equiv (p\ \equiv (\lfloor r\rfloor\lceil\equiv (p\ \sim (r))\rceil\lfloor r\rfloor\lceil\equiv (q\ \sim (r))\rceil))$		2, $\lfloor\ \rfloor e,\ f/\sim$
4.	$\equiv (\lfloor r\rfloor\lceil\equiv (p\ \sim (r))\rceil\ \bot)$	T15 ($\lfloor\ \rfloor e,\ p/p$)	
5.	$\equiv (\lfloor r\rfloor\lceil\equiv (q\ \sim (r))\rceil\ \bot)$	T15 ($\lfloor\ \rfloor e,\ p/q$)	**Règle** $\wedge e_g$ (dérivée 1-8)
6.	$\equiv (\bot\ \lfloor r\rfloor\lceil\equiv (q\ \sim (r))\rceil)$	5, $\equiv com$	*n.* $\wedge$ $(E\ F)$
7.	$\equiv (\lfloor r\rfloor\lceil\equiv (p\ \sim (r))\rceil\ \lfloor r\rfloor\lceil\equiv (q\ \sim (r))\rceil)$	4, 6, $\equiv syll$	⋯
8.	p	3, 7, $\equiv e$	E *n*, $\wedge e_g$

4. Epilogue

Avec la négation et la conjonction, munies de leurs règles d'inférence classiques, le développement obtenu présente désormais un ensemble *adéquat* de connecteurs du calcul classique des propositions. Ce résultat montre que PND donne accès à l'ensemble des connecteurs classiques et aux moyens d'inférence qui leur sont propres. En guise d'exemple, l'accès au conditionnel se fait désormais par la définition très simple suivante :

$$\text{T16} : \lfloor pq \rfloor \left| \equiv \left(\supset (pq) \equiv \left(p \wedge (pq) \right) \right) \right| \quad \textit{Définition}$$

Le lecteur, maintenant initié aux développements de PND, montrera sans difficulté que la règle d'élimination classique (le *modus ponens*) en dérive très immédiatement. Quant à la règle d'introduction (de forme hypothétique), on la dérive en constatant dans la séquence suivante que, si on dispose des lignes *m*-*n*, alors il est toujours possible de poursuivre jusqu'en ligne *n*+5 :

m.	E	*hyp*
	⋮	
n.	F	
n+1.	$\wedge (EF)$	*m*, *n*, $\wedge i$
n+2.	$\wedge (EF)$	*hyp*
n+3.	E	*n*+2, $\wedge e_g$
n+4.	$\equiv (E \wedge (EF))$	*m*-*n*+1, *n*+2-*n*+3, $\equiv i$
n+5.	$\supset (EF)$	T16 $(\lfloor \rfloor e,\ p/E\ ,\ q/F), n+4,\ \equiv e$

Règle $\supset i$ (dérivée)

m.	E	*hyp*
	⋮	
n.	F	
	$\supset (E\ F)$	*m*-*n*, $\supset i$

Ces résultats montrent que PND présente une complétude relative au calcul classique en ce sens qu'est prouvable dans PND la fermeture de tout théorème du calcul classique – et donc toute tautologie du calcul bivalent[7]. A la traduction près des formules classiques par leurs fermetures universelles, PND contient donc tout le calcul classique. L'inverse n'est cependant pas le cas, car PND permet de prouver de nombreux théorèmes qui ne sont classiquement exprimables que dans la métathéorie. Pour prendre un exemple simple, déjà évoqué au début de cet article, on peut prouver dans PND une formule exprimant que les connecteurs binaires ne sont pas tous commutatifs. Ou, plus intéressant, on peut montrer un résultat

[7] On dispose en effet, en plus des règles sur le quantificateur, d'un ensemble de règles suffisant pour avoir un calcul classique complet à la Fitch.

que l'on trouve déjà dans la thèse de Tarski (1923), à savoir que, sous l'hypothèse suivante (dite *loi de susbtitution*) :

$$\lfloor pq \rfloor \lceil \equiv \left(\equiv (pq) \lfloor f \rfloor \lceil \equiv \left(f(p)\, f(q) \right) \rceil \right) \rceil ,$$

on peut déduire des formules exprimant qu'il y a exactement 2 constantes propositionnelles, 4 connecteurs unaires et 16 connecteurs binaires distincts (non équivalents). Tous ces résultats et de nombreux autres qui ont été mis en évidence dans la littérature lesniewskienne sont bien entendu déjà à disposition dans la Protothétique, mais PND permet désormais de les exprimer dans un système déductif clair et intuitif, proche de fait de la pratique déductive informelle qui était en cours chez les logiciens de Varsovie dès les années 20 et qui est aujourd'hui enseignée à la plupart des étudiants de logique, avec des systèmes de déduction naturelle dans l'esprit de Jaśkowski et Fitch.

En revanche, PND n'est pas un système aussi fort que la Protothétique, car la loi de substitution qui vient d'être évoquée, ainsi que celles qui assurent dans la Protothétique l'extensionnalité des foncteurs de toutes catégories n'y sont pas prouvables. A mes yeux, cette faiblesse de PND n'est pas un défaut, bien au contraire. D'abord, il faut noter qu'il serait aisé d'obtenir un système strictement équivalent à la Protothétique, en ajoutant à PND une directive d'extensionnalité telle qu'on la trouve chez Leśniewski[8]. Moins spécifique que la Protothétique car n'imposant pas le cadre de la stricte extensionnalité, PND est une logique plus générale et plus ouverte aux applications. Elle pourrait en particulier constituer un outil adéquat pour investiguer le champ peu exploré des logiques développementales intensionnelles.

[8] Sur cette directive et sur l'extensionnalité dans la Protothétique, cf. Miéville (1984, chap. III).

Références

AJDUKIEWICZ, Kazimierz. 1935. « Die Syntaktische Konnexität ». *Studia Philosophica* n°1 : 1-27. [Trad. anglaise dans S. McCall (ed). *Polish Logic 1920-1939*. Oxford : Clarendon. 1967 : 207-231 ; Trad. française par K. Gan-Krzywoszyńska dans *Philosophia Scientiæ* 12-2 (2007)].

FITCH, Frederic B. 1952. *Symbolic Logic: An Introduction*. New York: Ronald Press.

GRIZE, Jean-Blaise. 1969. *Logique moderne : Fascicule I*. Paris / La Haye : Mouton, Gauthier-Villars.

JAŚKOWSKI, Stanisław. 1934. « On the Rules of Suppositions in Formal Logic ». *Studia Logica* n°1 : 5-32. [trad. Anglaise dans S. McCall (ed). *Polish Logic 1920-1939*. Oxford : Clarendon. 1967 : 232-258].

JORAY, Pierre. 2006. « La définition dans les systèmes logiques de Łukasiewicz, Leśniewski et Tarski ». Dans *La philosophie en Pologne : 1918-1939,* R. Pouivet et M. Rebuschi (éds), 203-222. Paris : Vrin.

JORAY, Pierre. 2011. «Axiomatiques minimales et définitions : la thèse de Tarski sur le calcul biconditionnel ». *Travaux de Logique* n°20 (Universités de Neuchâtel et Rennes) : 57-83.

LEŚNIEWSKI, Stanisław. 1989. *Sur les fondements de la mathématique.* Trad. par G. Kalinowski, préface de D. Miéville. Paris : Hermès.

LEŚNIEWSKI, Stanisław. 1992. *Collected Works* (2 vols). Surma S. J., Srzednicki J. T., Barnett D. I. (eds). Warszawa : PWN / Dordrecht : Kluwer.

MIÉVILLE, Denis. 1984. *Un développement des systèmes logiques de Stanisław Leśniewski : Protothétique – Ontologie – Méréologie*. Berne : Peter Lang.

MIÉVILLE, Denis. 2001. *Introduction à l'œuvre de S. Leśniewski I. La Protothétique. Travaux de Logique* (hors série). Université de Neuchâtel.

MIÉVILLE, Denis. 2004. *Introduction à l'œuvre de S. Leśniewski II. L'Ontologie. Travaux de Logique* (hors série). Université de Neuchâtel.

MIÉVILLE, Denis. 2009. *Introduction à l'œuvre de S. Leśniewski VI. La métalangue d'une syntaxe inscriptionnelle. Travaux de Logique* (hors série). Université de Neuchâtel.

SRZEDNICKI, Jan T. J. et STACHNIAK Zbigniew. (eds). 1998. *Leśniewski's Systems : Protothetic.* Dordrecht : Kluwer.

TARSKI, Alfred. 1923. « Sur le terme primitif de la logistique ». *Fundamenta Mathematica* vol. 4 et 5. [Repris dans Tarski, A. *Logique, sémantique, métamathématique 1923-1944*. Tome 1. Paris : A. Colin. 1972 : 3-25].