Une algèbre de l'ontologie finitaire
ou
un langage fonctionnellement complet pour la théorie finitaire des types

FRANÇOIS LEPAGE
Université de Montréal

La théorie des types simples est un instrument extrêmement puissant pour décrire des univers sémantiques, c'est-à-dire pour construire des modèles satisfaisant des ensembles d'énoncés pour des langages contenant des opérateurs et des foncteurs de tous ordres (comme cela semble être le cas pour la portion des langues naturelles pour laquelle il est possible de fournir une sémantique explicite). L'idée de base est très simple. Partant d'ensembles d'objets correspondant à un type de base, on définit l'univers des objets comme étant l'ensemble des fonctions définissables par induction sur ces ensembles de base. Par exemple, si au point de départ nous avons l'ensemble $\{0,1\}$ – les deux valeurs de vérité – et un ensemble quelconque d'individus E, l'ensemble $\{0,1\}^E$ des fonctions de E dans $\{0,1\}$ peut être interprété comme l'ensemble des propriétés d'individus. Ainsi, la propriété « être bleu » est la fonction qui à chaque individu associe soit 1 si l'individu est bleu, soit 0 si l'individu n'est pas bleu. Un autre exemple serait celui d'une propriété de propriété, soit un élément de $\{0,1\}^{(\{0,1\}^E)}$ dont un élément serait « être une couleur ». On voit la richesse potentielle de cet instrument. Formellement, pour cet exemple particulier, l'ensemble T des types est le plus petit ensemble tel que

(i) $t \in T$ (le type des valeurs de vérité),
(ii) $e \in T$ (le type des individus),
(iii) si $\sigma, \tau \in T$, alors $<\sigma,\tau> \in T$

L'ensemble des objets pour chaque type sera alors

(i) $D_t = \{0,1\}$ (ensemble des valeurs de vérité)
(ii) $D_e = E$ (ensemble des individus)
(iii)$D_{<\sigma,\tau>} = D_\tau^{D_\sigma}$ (ensemble des fonctions de D_σ dans D_τ).

Curieusement, il existe un langage très simple qui permet de formuler un système complet pour cet univers sémantique, c'est-à-dire un système dont les théorèmes sont exactement les énoncés valides (au sens général de Henkin, voir Henkin 1950, Tarski 1923 et Montague 1974). Ce langage, qui ne contient que très peu de primitifs, est le suivant.

Soit A_L l'alphabet de L tel que $L = \{\lambda, \equiv, (,), \{x_{\alpha_i}\}_{\alpha \in T, i \in \omega}\}$

$\{x_{\alpha_i}\}_{i \in \omega} = Var_\alpha$ est l'ensemble dénombrable des variables du type α.

L'ensemble des expressions bien formées EBF_α pour chaque type α du langage L est défini de la manière suivante :

(i) $\{ x_{\alpha_i} \}_{i \in \omega} \subseteq \mathrm{EBF}_\alpha$

(ii) si $A,B \in \mathrm{EBF}_\alpha$, alors $(A \equiv B) \in \mathrm{EBF}_t$

(iii) si $A \in \mathrm{EBF}_\alpha$ et $x_\beta \in Var_\beta$, alors $\lambda x_\beta A_\alpha \in \mathrm{EBF}_{<\beta,\alpha>}$

(iv) si $A \in \mathrm{EBF}_{<\beta,\alpha>}$ et $B \in \mathrm{EBF}_\beta$, alors $(AB) \in \mathrm{EBF}_\alpha$.

L'idée est très simple. On a des variables de chaque type en nombre infini dénombrable, on peut construire des énoncés d'identité, on peut construire les fonctions d'un type dans un autre et on peut appliquer une fonction sur n'importe quel argument du bon type. Explicitement et formellement, la sémantique est la suivante.

Une assignation de valeur aux variables est une fonction

$$\chi : \bigcup_i Var_{\alpha_i} \to \bigcup_i D_{\alpha_i} .$$

avec $\chi(x_\alpha) \in D_\alpha$

On définit la dénotation d'une EBF de type α de la manière suivante :

(i) $\|x_\alpha\|_\chi = \chi(x_\alpha)$

(ii) $\|(A \equiv B)\|_\chi = 1$ ssi $\|A\|_\chi = \|B\|_\chi$ et
$= 0$ autrement

(iii) $\|\lambda x_\beta A_\alpha\|_\chi$ est cette fonction $h : D_\beta \to D_\alpha$ telle que
$h(a) = \|A_\alpha\|_{\chi^{x_\alpha}_a}$ où $\chi^{x_\alpha}_a$ est comme $\chi^{x_\alpha}_a$ sauf que$(x_\alpha) = a$.

(iv) $\|(A_{<\beta,\alpha>} B_\beta)\|_\chi = \|A_{<\beta,\alpha>}\|_\chi(\|B_\beta\|_\chi)$

Le présent texte propose d'apporter une réponse à une question très simple dont j'avoue ne pas savoir si elle a déjà été posée et, le cas échéant, si elle a reçue une réponse satisfaisante. Cette question est la suivante : dans un univers dont les objets sont soit des *individus* de types de base, soit des valeurs de vérité, soit des fonctions appartenant à la hiérarchie définissable à partir des individus et des valeurs de vérité, est-il possible – en supposant que l'on ait un nom pour chaque individu de chaque type de base, un nom pour le vrai, un nom pour le faux – avec comme seule ressource l'abstracteur λ, son inverse, l'application fonctionnelle, et l'identité, de définir un *nom canonique* pour chaque objet de l'univers, c'est-à-dire que pour tout $a \in D_\alpha$, il existe une expression E_α telle que pour tout χ, $\|E_\alpha\|_\chi = a$.

Supposons, pour simplifier, que, outre les valeurs de vérités, il n'y ait qu'un seul type d'individus de base et que E désigne l'ensemble des individus comme ci-dessus. Si la cardinalité de E est infinie, la réponse à la question est trivialement non pour la simple raison que 2^E, par exemple, est dans ce cas non dénombrable et que le nombre d'expressions du langage est dénombrable. Mais qu'en est-il quand E est fini ?

Une version très élémentaire de ce problème a été posé et résolu par Post dans sa thèse en 1920. La question était la suivante : est-il possible de trouver un nom pour chaque fonction de vérité n-aire, en utilisant que $\wedge$ et $\neg$. La réponse bien connue est oui et elle consiste à introduire la notion de forme normale disjonctive après avoir défini $\vee$ en terme de $\neg$ et $\wedge$. Comme, par ailleurs, on peut définir $\neg$ en λ-calcul par (voir Montague 1974 et Andrews 1963).

$$\neg =_{\text{déf}} \lambda x(x \equiv \mathrm{F})$$

où x est une variable du type t des valeurs de vérité et F le nom du faux et

$$\wedge =_{\text{déf}} \lambda x \lambda y(\lambda z((zx)y) \equiv \lambda z((z\mathrm{T})\mathrm{T})$$

où T est un nom du vrai, x et y sont des variables du type valeur de vérité et z une variable du type $<t,<t,t>>$, c'est-à-dire du type fonction prenant pour argument des valeurs de vérité et pour valeurs des fonctions prenant pour argument des valeurs de vérité et pour valeurs des valeurs de vérité. zxy est donc du type t. En introduisant le « $\vee$ » par les lois de De Morgan et en écrivant $(x \wedge y)$ pour $\wedge xy$ et $(x \vee y)$ pour $\vee xy$, le nom canonique de « $\supset$ », par exemple, est

$$\lambda x \lambda y[(x \wedge y) \vee (\neg x \wedge y) \vee (\neg x \wedge \neg y)].$$

La technique est simple : il suffit de décrire toutes et rien que les lignes de la table où l'énoncé est vrai en posant x (ou y) quand x (ou y) est vrai et $\neg x$ (ou $\neg y$) quand x (ou y) est faux.
On peut généraliser.
Il s'agit de donner un nom canonique à $f : 2^n \rightarrow 2$ (où 2 = {0,1})
Soit $\mathbf{1}(f)$ l'ensemble des lignes de la table de vérité où f est vrai. Alors

$$\lambda p_0 \ldots \lambda p_{n-1} \bigvee_{j \in \mathbf{1}(f)} \left(\bigwedge_{k<n} l_{jk} \right)$$

où l_{jk} est p_k ou $\neg p_k$ selon que p_k est vrai ou faux sur la ligne j, est un nom de f.

Cette méthode règle la question pour les fonctions de type $<t,<t,<\ldots<t,t>\ldots>>>$. Qu'en est-il pour les types quelconques. Restreignons-nous pour l'instant à la hiérarchie des fonctions définissables à partir du seul type t, c'est-à-dire à partir de l'ensemble {0,1} (on appelle ces types les types propositionnels). Pour ces types on peut généraliser l'approche ci-dessus (van Benthem 1995). Pour ce faire, nous aurons besoin des des notions formelles suivantes. Soit f une fonction de type $\sigma = <\sigma_0,<\sigma_1,<\ldots<\sigma_{n-1},t>\ldots>>>$ (tous les types ont cette forme, le cas dégénéré étant t).
On appelle $\vec{f} = < f_{\sigma_0}, \ldots, f_{\sigma_{n-1}} >$ un projecteur de f (parce que $f(f_{\sigma_0})\ldots(f_{\sigma_{n-1}}) \in \{0,1\}$
Supposons par hypothèse d'induction que nous ayons un nom canonique $(g_i)^c$ pour

chaque g_i d'un type inférieur à σ et que pour chaque variable $X_{\sigma j}$, $[X_{\sigma j}(\vec{g})^c]$ soit $[...[X_{\sigma j}(g_1)^c]...(g_k)^c]$, c'est-à-dire l'expression syntaxique de type t construite en appliquant $X_{\sigma j}$ au projecteur formé des noms canoniques $(\vec{g})^c = <(g_1)^c,..., (g_k)^c>$.

Si $\vec{p}_i = <p_{i1},...,p_{in}>$ est un projecteur de f, alors si nous définissons $\delta_{p_{ij}}$

$$\delta_{p_{ij}}([x_{\sigma j}(\vec{g})^c]) = [x_{\sigma j}(\vec{g})^c] \text{ si } p_{ij}(\vec{g}) = 1$$
$$\delta_{p_{ij}}([x_{\sigma j}(\vec{g})^c]) = \neg[x_{\sigma j}(\vec{g})^c] \text{ si } p_{ij}(\vec{g}) = 0$$

alors

Proposition

$$\lambda x_{\alpha_0}\ldots\lambda x_{\alpha_{n-1}}[(\bigvee_{\vec{p}_i\in 1(f)}(\bigwedge_{\vec{g}\in Pr(\sigma_0)}\delta_{p_{i_0}}([x_{\alpha_0}(\vec{g})^c]) \wedge ... \wedge \bigwedge_{\vec{g}\in Pr(\alpha_{n-1})}\delta_{p_{i_{n-1}}}([x_{\alpha_{n-1}}(\vec{g})^c]))]$$

est un nom canonique de f. ($\bigvee$ et $\bigwedge$ sont respectivement la disjonction et la conjonction généralisées).

Cette méthode de construction de noms d'objets n'est cependant pas directement exportable au cas général où, en plus des valeurs de vérité, nous avons des individus.

Cette manière de procéder utilise en effet le fait essentiel que nous sommes en présence de fonctions de vérité : toute fonction de la théorie propositionnelle des types est entièrement caractérisée par les valeurs qu'elle attribue à ses différents projecteurs, soit des 0 ou des 1. C'est l'existence d'une algèbre – ici l'algèbre de Boole – sur {0,1} et le fait que cette algèbre peut s'exprimer dans le langage qui nous permet de reconstruire une formule qui désigne canoniquement une fonction donnée. Que se passe-t-il si la valeur d'une fonction n'est pas une valeur de vérité mais un individu ? Prenons un exemple très simple. Soit $f: A \rightarrow B$ une fonction arbitraire d'un ensemble fini A dans un ensemble fini B.

Supposons que que nous ayons un nom pour chaque objet de A et pour chaque objet de B. La question est : est-il possible de construire un nom canonique pour f utilisant l'abstraction, l'application fonctionnelle et l'identité ? La réponse est clairement non. Si $f(a) = b$ et si f^c est un nom canonique de f, alors f^c est $\lambda x_{ee} Y_e$ (les formules dont le connecteur principal est ≡ sont de type t) et $([\lambda x_e Y_e a^c]^c) = b^c$ (où a^c et b^c sont des noms canoniques de a et de b). Une valeur possible de Y est b^c. Mais alors f est la fonction constante telle que $f(x) = b$ pour tout x. Comme Y est de type e et que f^c est de type $<e,e>$, la seule possibilité est que Y soit $\lambda x_e Z_e$. f^c est alors $\lambda x_{ee} \lambda x_e Z_e$ et $([\lambda x_{ee} \lambda x_e Z_e a^c]^c)^c = b^c$. Encore une fois, une valeur possible de Z est b^c. Mais alors f est la fonction constante telle que $f(x) = b$ pour tout x. On ne peut régresser ainsi indéfiniment, le nombre de λ apparaissant dans f^c étant fini. Donc, à moins que f ne soit une fonction constante, il n'y a pas de formule utilisant l'abstraction, l'application fonctionnelle et l'identité dont la valeur est f.

La leçon est claire : il faut enrichir le langage et l'ontologie en nous inspirant du cas où les objets sont des valeurs de vérité en introduisant une sorte d'*algèbre d'individus*.

Soit E un ensemble fini d'individus. Une algèbre d'individus sur E est un quintuplet $<E, \oplus, \otimes, \blacklozenge, \spadesuit>$ où

(1) $\blacklozenge$ et $\spadesuit$ sont deux nouveaux individus distincts entre eux et de tous les éléments de E

(2) $\oplus$, $\otimes$ sont les opérations partielles suivantes sur E

$x \otimes y = \blacklozenge$ si $x = y$

$x \otimes y = \spadesuit$ si $x \neq y$

$x \oplus \blacklozenge = \blacklozenge \oplus x = x$ si $x \neq \spadesuit$ et $x \neq \blacklozenge$

$x \oplus \spadesuit = \spadesuit \oplus x = \blacklozenge$ si $x \neq \blacklozenge$ et $x \neq \spadesuit$

$\blacklozenge \oplus \spadesuit = \spadesuit \oplus \blacklozenge = \spadesuit$

$\spadesuit \oplus \spadesuit = \spadesuit$

$\blacklozenge \oplus \blacklozenge = \blacklozenge$

Dans les autres cas les opérations ne sont pas définies.

Soit $f{:}A \rightarrow B$. Supposons que

$a_0^c,\ldots,a_n^c$ sont les noms canoniques respectifs de $a_0,\ldots,a_n$, que

$(f(a_0))^c,\ldots,(f(a_n))^c$ sont les noms canoniques respectifs de $f(a_0),\ldots,f(a_n)$, que

$\|X \oplus^c Y\|_\chi = \|X\|_\chi \oplus \|Y\|_\chi$ et que $\|X \otimes^c Y\|_\chi = \|X\|_\chi \otimes \|Y\|_\chi$

alors

$\lambda x(((f(a_0))^c \oplus^c (x \otimes^c a_0^c)) \oplus^c \ldots \oplus^c ((f(a_n))^c \oplus^c (x \otimes^c a_n^c)))$

est un nom canonique de f.

Par exemple, soit $f : \{a,b\} \to \{c,d\}$ avec $f(a) = b$ et $f(c) = d$.

On vérifie que

$\|(\lambda x(((f(a))^c \oplus^c (x \otimes^c a^c)) \oplus^c ((f(c))^c \oplus^c (x \otimes^c b^c)))\|_\chi(a) =$

$\|(b^c \oplus^c (x \otimes^c a^c)) \oplus^c (d^c \oplus^c (x \otimes^c b^c))\|_{\chi'}$ = (où χ' est χ_a^x)

$\|b^c \oplus^c (x \otimes^c a^c)\|_{\chi'} \oplus \|d^c \oplus^c (x \otimes^c b^c)\|_\chi =$

$(\|b^c\|_{\chi'} \oplus \|x \otimes^c a^c\|_{\chi'}) \oplus (\|d^c\|_\chi \oplus \|(x \otimes^c b^c)\|_\chi) =$

$(b \oplus (\|x\|_{\chi'} \otimes \|a^c\|_{\chi'})) \oplus (d \oplus (\|x\|_{\chi'} \otimes \|b^c\|_{\chi'})) =$

$(b \oplus (a \otimes a)) \oplus (d \oplus (a \otimes b))$

$(b \oplus \blacklozenge) \oplus (d \oplus \spadesuit) =$

$(b \oplus \blacklozenge) =$

b

Remarquons qu'on peut trivialement appliquer cette méthode au cas particulier des fonctions de vérité en prenant soin de distinguer $\spadesuit_t$ de $\spadesuit_e$ et $\blacklozenge_t$ de $\blacklozenge_e$ de même que l'on peut généraliser à un nombre quelconque fini de types de base.

La généralisation à la théorie des types finis est immédiate.

Proposition : Soit une fonction f de type $<\alpha_0, <\alpha_1, <\ldots<\alpha_n,t>\ldots>>>$ ou encore de type $<\alpha_0, <\alpha_1, <\ldots<\alpha_n,e>\ldots>>>$. L'expression

$$\lambda x_{\alpha_0}\ldots\lambda x_{\alpha_n}\Big[\bigoplus^c_{\vec{a}_i \in P(f)}\Big[(f(\vec{a}_i))^c \oplus^c \Big[\bigoplus_{j=0}^{n}{}^c\Big[\bigoplus^c_{\vec{g}\in P(\alpha_j)}(x_{\alpha_j}(\vec{g})^c) \otimes^c (a^c_{i\alpha_j}(\vec{g})^c)\Big]\Big]\Big]\Big]$$

(où $\vec{a}_i = <a_{i\alpha_0}\ldots a_{i\alpha_n}>$) est un nom canonique de f.

Preuve

Remarquons tout d'abord la valeur de $\bigoplus_{j=0}^{n} = x_0 \otimes \ldots \otimes x_n$ où tous les x_j sont soit ♦, soit ♠, est ♠ s'il y a au moins un x_j qui est ♠ et est ♦ si, et seulement si, tous les x_j sont ♦.

Soit

$$\|[\lambda x_{\alpha_0}\ldots\lambda x_{\alpha_n}[\bigoplus^{c}_{\vec{a}_i \in P(f)}[f((\vec{a}_i)^c)\oplus^c[\bigoplus_{j=0}^{n}{}^{c}[\bigoplus^{c}_{\vec{g}\in P(\alpha_j)}(x_{\alpha_j}(\vec{g})^c)\otimes^c(a^c_{i\alpha_j}(\vec{g})^c)]]]]\|_\chi(\vec{a}_k)$$

D'après la définition de $\| \ \|_\chi$, cette valeur est celle

$\|[\bigoplus^{c}_{\vec{a}_i \in P(f)}[(f(\vec{a}_i))^c\oplus^c[\bigoplus_{j=0}^{n}{}^{c}[\bigoplus^{c}_{\vec{g}\in P(\alpha_j)}(x_{\alpha_j}(\vec{g})^c)\otimes^c(a^c_{i\alpha_j}(\vec{g})^c)]]]\|_{\chi'}$ où χ' est comme χ sauf que $\chi'(x_{\alpha_j}) = a_{k\alpha_j}$.

Deux cas sont possibles :

(i) $k = i$.

Dans ce cas on a $\|(x_{\alpha_j}(\vec{g})^c)\otimes^c(a^c_{i\alpha_j}(\vec{g})^c)\|_{\chi'} = \blacklozenge$ pour tout $\vec{g}\in P(\alpha_j)$ et pour tout j, $0 \leq j \leq n$. Donc,

$\|f((\vec{a}_k)^c)\oplus^c[\bigoplus_{j=0}^{n}{}^{c}[\bigoplus^{c}_{\vec{g}\in P(\alpha_j)}(x_{\alpha_j}(\vec{g})^c)\otimes^c(a^c_{i\alpha_j}(\vec{g})^c)]]]\|_{\chi'} = \|f((\vec{a}_k)^c)\|_{\chi'}\oplus \blacklozenge = f(\vec{a}_k)\oplus \blacklozenge = f(\vec{a}_k)$.

(ii) $k \neq i$

Dans ce cas on a $\|(x_{\alpha_j}(\vec{g})^c)\otimes(a^c_{i\alpha_j}(\vec{g})^c)\|_{\chi'} = \spadesuit$ pour certain(s) $\vec{g}\in P(\alpha_j)$ et certain(s) j, $0 \leq j \leq n$ et, pour tous les autres $\vec{g}\in P(\alpha_j)$ et j, $0 \leq j \leq n$, $\|(x_{\alpha_j}(\vec{g})^c)\otimes(a^c_{i\alpha_j}(\vec{g})^c)\|_{\chi'} = \blacklozenge$.

Donc,

$\|(f(\vec{a}_i))^c\oplus^c[\bigoplus_{j=0}^{n}{}^{c}[\bigoplus^{c}_{\vec{g}\in P(\alpha_j)}(x_{\alpha_j}(\vec{g})^c)\otimes^c(a^c_{i\alpha_j}(\vec{g})^c)]]]\|_{\chi'} = \|(f(\vec{a}_i))^c\|_{\chi'}\oplus \spadesuit = f(\vec{a}_i)\oplus \spadesuit = \blacklozenge$.

(i) et (ii) entraînent que

$\|[\lambda x_{\alpha_0}\ldots\lambda x_{\alpha_n}[\bigoplus^{c}_{\vec{a}_i \in P(f)}[(f(\vec{a}_i))^c)\oplus^c[\bigoplus_{j=0}^{n}{}^{c}[\bigoplus^{c}_{\vec{g}\in P(\alpha_j)}(x_{\alpha_j}(\vec{g})^c)\otimes^c(a^c_{i\alpha_j}(\vec{g})^c)]]]]\|_\chi(\vec{a}_k) = f(\vec{a}_k)\oplus \blacklozenge = f(\vec{a}_k)$.

Cette manière de procéder peut sembler *ad hoc*.Si l'idée d'effectuer des opérations sur les individus n'a rien de choquant, postuler l'« existence » d'individus comme ♠ et ♦est une hypothèse pour le moins bizarre. Remarquons cependant que, au niveau du langage objet, nous n'avons aucunement besoin de noms pour dénoter ces étranges individus : les noms canoniques ne contiennent que les symboles habituels du langage de la théorie des types plus $\oplus^c$ et $\otimes^c$. Cela suggère fortement que ces individus sont *éliminables* au profit d'une redéfinition adéquate des opérateurs sur les individus « normaux ». Voici une manière de le faire.

Soit l'opération ternaire partielle suivante sur E :

$\partial_\otimes : E^3 \to E$

avec

$\partial_\otimes(x,y,z) = x$ si $y = z$ et indéfinie si $y \neq z$.

et la famille de fonctions suivantes

$\partial_{\oplus_n}(x_0, ..., x_{n-1}) = a$ si pour tout i tel que $0 \leq i \leq n-1$, $x_i = a$ ou n'est pas défini et
$\partial_{\oplus_n}(x_0, ..., x_{n-1})$ n'est pas défini dans les autres cas.

Reprenant notre exemple de la fonction triviale f avec $f(a) = c$ et $f(b) = d$ et utilisant nos deux fonctions spéciales, on vérifie facilement que

$\lambda x(\partial^c_{\oplus_2}(\partial^c_\otimes((f(a))^c, a^c, x), \partial^c_\otimes((f(b))^c, b^c, x)))$ est un nom canonique de f.
La généralisation aux fonctions de type quelconque est immédiate.

Proposition : Soit une fonction f de type $<\alpha_0, <\alpha_1, <\ldots<\alpha_n,t>...>>>$ ou encore de type $<\alpha_0, <\alpha_1, <\ldots<\alpha_n,e>...>>>$. L'expression

$$\lambda x_{\alpha_0} \ldots \lambda x_{\alpha_{n-1}} [\partial^c_{\oplus_{\vec{a}_i \in P(f)}} [\partial^c_{\oplus_{j=0}} [\partial^c_{\oplus_{\vec{g} \in P(\alpha_j)}} (\partial^c_\otimes((f(\vec{a}_i))^c, a^c_{i_{\alpha_j}}(\vec{g})^c, x_{\alpha_j}(\vec{g})^c)]]]$$

est un nom canonique de f.

Bibliographie

Andrews, P. B., ‘A Reduction of the Axioms for the Theory of Propositional Types’, *Fundamenta Mathematicæ* LII, 1963, 345-350.

van Benthem, J., *Language in Action*, Amsterdam, North-Holland/Cambridge, MIT Press, 1995.

Henkin, L., ‘Completeness in the Theory of Types’, *The Journal of Symbolic Logic*, 15, 1950, 81-91.

Montague, R., ‘Universal Grammar’, in *Formal Philosophy*, New Haven, Yale University Press, 1974, 222-246.

Tarski, A., ‘Sur le terme primitif de la logistique’, *Fundamenta Mathematicæ* IV, 1923, 59-74.