

Sémantique formelle

Cours de Licence de Sciences du Langage (L3)

Alain Lecomte – Professeur, Université Paris 8

Leçon 3 – Types et λ -termes

3 Sémantique des constituants syntaxiques

3.1 Récursivité et compositionnalité

Les leçons 1 et 2 ont permis de jeter les fondements d'une sémantique vériconditionnelle, c'est-à-dire basée sur les conditions de vérité des phrases. Nous avons vu que nous pouvions associer à *au moins certaines* phrases des représentations sous forme de formules logiques (pour l'instant essentiellement du premier ordre) et ensuite, à partir de ces formules, calculer la valeur de vérité de ces phrases par rapport à un modèle. Notre association de formules logiques à des phrases a été relativement intuitive : nous n'avons pas suivi un algorithme particulier. Pour nous convaincre de la justesse de la traduction, nous nous sommes appuyés simplement sur l'intuition selon laquelle les conditions de vérité des formules étaient bien les mêmes que celles des phrases. Ce faisant, nous avons considéré seulement deux sortes d'entités en leur associant des types d'objet bien déterminés : d'un côté, les *noms propres* (représentés par des *constantes*¹) et les *variables*, supposés dénoter des éléments du domaine d'interprétation noté D (ce qu'on appelle des « entités individuelles »), et de l'autre, les *phrases complètes*, supposées dénoter le « vrai » ou le « faux », ces deux valeurs constituant l'ensemble des valeurs de vérité (*truth-values*) $\{0, 1\}$.

Se pose maintenant le problème de l'association systématique (et pas seulement « intuitive ») d'une représentation sémantique à une phrase. On ne peut évidemment pas supposer que nous ayons une simple liste de représentations sémantiques associées de manière biunivoque à toutes les phrases possibles. Une telle liste serait infinie. D'après la thèse chomskyenne de la créativité du langage, nous pouvons toujours produire une phrase nouvelle, qui n'a peut-être jamais encore été produite et dont pourtant la signification est accessible. Si nous voulons rendre compte de cette propriété du langage, nous sommes obligés, comme cela a été noté par le philosophe et logicien Donald Davidson², de définir l'association d'une représentation sémantique à une phrase au moyen d'une *procédure récursive*.

En se mettant au plus haut niveau de généralité, une procédure récursive n'est rien d'autre qu'un mécanisme *fini* (autrement dit s'exprimant au moyen d'un nombre fini de symboles) permettant d'engendrer ou de reconnaître les éléments d'un ensemble *infini*. Un exemple de procédure récursive est celui qu'on utilise en logique pour calculer la valeur de vérité d'une formule de logique propositionnelle. Une telle formule peut être d'une longueur et d'une complexité arbitraires, or nous sommes toujours capables de l'évaluer, par rapport à une assignation donnée de valeurs de vérité aux variables propositionnelles qu'elle contient. Rappelons pour mémoire la définition récursive de la vérité pour une formule de logique propositionnelle :

Définition (Rappel) : Soit φ une assignation de valeurs de vérité à des variables propositionnelles $p_1, p_2, \dots, p_n$. Soit A une formule qui contient ces variables propositionnelles. La valeur de vérité $\varphi^*(A)$ de cette formule, par rapport à l'assignation φ , est calculée de manière que :

- Si $A = p_i$, où $i \in \{1, 2, \dots, n\}$, $\varphi^*(A) = \varphi(A)$,

¹ Dans la suite, le statut des noms propres sera révisé.

² Cf. Donald Davidson, 1967, 'Truth and Meaning', *Synthese* 17 : 304—323, repris dans S. Davis et B. Gillon, 2004, *Semantics, a reader*, Oxford University Press

- Si $A = (B \wedge C)$, $\varphi^*(A) = 1$ si et seulement si $\varphi^*(B) = \varphi^*(C) = 1$
- Si $A = (B \vee C)$, $\varphi^*(A) = 0$ si et seulement si $\varphi^*(B) = \varphi^*(C) = 0$
- Si $A = (B \Rightarrow C)$, $\varphi^*(A) = 0$ si et seulement si $\varphi^*(B) = 1$ et $\varphi^*(C) = 0$
- Si $A = (\neg B)$, $\varphi^*(A) = 1$ si et seulement si $\varphi^*(B) = 0$

Une telle définition pourrait suffire si les entités du langage n'étaient que des propositions, c'est-à-dire les contenus de phrases complètes. Or, nous savons qu'il n'en est rien : les entités du langage contiennent aussi des *noms*, des *verbes*, des *adjectifs*, des *prépositions*, des *articles* etc. D'où la nécessité d'adapter ce genre de définition au langage.

Comment passer du niveau des plus petits composants syntaxiques de la phrase (articles, noms etc.) à celui de la phrase ? Ici intervient une autre thèse bien connue, il s'agit du *principe de compositionnalité*, attribué généralement à G. Frege (bien qu'il ne semble pas qu'il l'ait lui-même exprimé explicitement) mais surtout développé par le philosophe et logicien américain Richard Montague³.

Principe de compositionnalité : *la signification d'une expression est fonction des significations de ses constituants et de la manière dont ceux-ci sont combinés dans l'expression.*

On doit noter que la deuxième partie de cette formulation (« et de la manière etc. ») est importante, c'est ce qui fait que *Pierre aime Marie* n'a pas le même sens que *Marie aime Pierre* ! D'autre part, le mot « fonction » joue un rôle fondamental dans l'énoncé de ce principe. On pourrait en effet l'entendre d'une manière vague (selon l'usage du mot « fonction » dans le langage ordinaire), mais depuis Montague, on l'entend au contraire d'une manière très précise, autrement dit dans le sens *mathématique* du terme. On va même plus loin : les significations elles-mêmes sont traduites soit par des objets soit par des fonctions. La signification d'une expression composée est ainsi une fonction de fonctions (et qui donne comme résultat une fonction).

3.2 Prédicats et fonctions

3.2.1 Prédicats unaires

Pour bien comprendre cette intervention de la notion de fonction en sémantique, nous allons partir de la distinction opérée par Frege entre expressions *saturées* et expressions *non saturées*⁴. Partons d'une phrase extrêmement simple comme *Marie dort*. Cette phrase a pour contenu une proposition et possède, en principe, une valeur de vérité, elle est donc une expression « saturée ». De même le constituant *Marie* possède, selon nos hypothèses formulées au début de ce paragraphe, une dénotation fixe : il dénote un objet individuel dans notre domaine d'interprétation, c'est aussi une expression saturée. Mais considérons *dort* : *dort* ne renvoie ni à un objet individuel, ni à une valeur de vérité. On ne peut obtenir à partir de lui une dénotation fixe sous forme de valeur de vérité que si on le complète (autrement dit on le « sature ») au moyen d'un objet individuel, qui deviendra son sujet, de manière à obtenir *Marie dort*, par exemple (ou *Paul dort*, ou *Nathalie dort* etc.). Autrement dit le constituant verbal *dort* est *insaturé*. Son contenu est donc celui d'une *fonction*. A *dort*, on associe ainsi la fonction :

³ Voir en particulier R. Montague, 1970a, 'English as a formal language', R. Montague, 1970b, 'Universal grammar', R. Montague, 1973, 'The proper treatment of quantification in ordinary English', tous ces articles reproduits dans R. Montague, 1974, *Formal Philosophy, Selected Papers of Richard Montague*, édité par R. Thomason, Yale University Press

⁴ G. Frege, 1904, 'Qu'est-ce qu'une fonction', repris dans G. Frege, 1971, *Ecrits logiques et philosophiques*, ed. du Seuil, pp. 160—168

$$\mathbf{dort} : x \mapsto \mathbf{dort}(x)$$

qui est une fonction de D dans $\{0, 1\}$, définie par : $\mathbf{dort}(x) = 1$ si x dort et $= 0$ sinon.

Cette association n'est pas contradictoire avec celle que nous avons faite dans la leçon 1, entre *prédicats unaires* et *parties de D* . Pourquoi ?

Nous savons que si X est un sous-ensemble (une partie) d'un ensemble D , alors nous pouvons définir une fonction particulière $\mathbf{1}_X$ de D dans $\{0, 1\}$, qu'on appelle la fonction caractéristique de X , en posant :

$$\text{pour tout } x \in D, \mathbf{1}_X(x) = 1 \text{ si } x \in X, = 0 \text{ sinon}$$

Remarquons que cette fonction est totale⁵, c'est-à-dire définie pour *tout* x de D .

Réciproquement, si f est une fonction totale de D dans $\{0, 1\}$, nous pouvons toujours associer à f un sous-ensemble de D , X_f défini par : $X_f = \{x ; f(x) = 1\}$.

On peut bien sûr démontrer que l'ensemble associé à la fonction caractéristique de X par ce procédé est bien X (on retombe sur nos pieds !) et que la fonction caractéristique de X_f est bien f (on retombe aussi sur nos pieds !). Ainsi établit-on une correspondance biunivoque (une « bijection ») entre les sous-ensembles d'un ensemble D et les fonctions totales de D dans $\{0, 1\}$.

Ceci nous permet de dire que nous avons en réalité fait deux fois la même chose, en associant dans la leçon 1, un sous-ensemble de D à un prédicat unaire, et dans cette leçon-ci, une fonction de D dans $\{0, 1\}$ à un tel prédicat unaire.

3.2.2 Prédicats binaires et n -aires

De la même manière, nous pouvons associer à tout prédicat binaire une fonction. Il y a deux manières de voir une telle fonction (dont on peut démontrer qu'elles sont équivalentes).

- Première manière :

Soit par exemple le verbe *aimer*, dans *Pierre aime Marie*. Il apparaît comme doublement insaturé, puisqu'il lui faut *deux* objets individuels pour donner une signification saturée sous la forme d'une proposition. On peut donc associer à *aimer* la fonction totale **aimer** de $D \times D$ dans $\{0, 1\}$ qui, à tout (x, y) de $D \times D$ associe 1 si x aime y (= « $\mathbf{aimer}(x, y)$ reçoit la valeur vraie ») et 0 sinon.

- Deuxième manière :

On procède par étapes. On remarque que *aime Marie* est comme *dort*, autrement dit c'est un prédicat unaire qu'on représente par une fonction totale **aime**_{Marie} de D dans $\{0, 1\}$. Mais si on étudie le syntagme verbal *aime Marie*, on voit que *aime* y figure comme une expression insaturée, donc une fonction, qui, à tout individu x de D associe la fonction **aime** _{x} de D dans $\{0, 1\}$. On associe donc à *aimer* la fonction totale **aimer** de D dans l'ensemble des fonctions totales de D dans $\{0, 1\}$ qui associe à tout individu x , la fonction **aime** _{x} qui à tout individu y associe 1 si x aime y et 0 sinon.

Notation : on notera $\{D \rightarrow \{0, 1\}\}$ l'ensemble des fonctions totales de D dans $\{0, 1\}$. Cet ensemble est souvent noté aussi : $\{0, 1\}^D$. Ainsi au verbe transitif *aimer* peut être associée une fonction **aimer** de D dans $\{0, 1\}^D$, ou de D dans $\{D \rightarrow \{0, 1\}\}$ (donc un élément de $\{D \rightarrow \{D \rightarrow \{0, 1\}\}\}$).

On peut généraliser cette démarche à un prédicat n -aire quelconque, par exemple à un verbe ditransitif comme *donner*. A un prédicat n -aire P , on peut associer :

- ou bien une fonction de D^n dans $\{0, 1\}$
- ou bien une fonction $D \rightarrow \{D \rightarrow \{D \rightarrow \dots D \rightarrow \{0, 1\} \dots\}\}$ (avec n fois D)

⁵ On appelle aussi souvent *application* une fonction totale

3.3 La nécessité de passer en ordre supérieur

Dans la leçon 2, nous avons étudié seulement la logique des prédicats du premier ordre. « Premier ordre » signifie que les variables ne peuvent désigner que des individus, c'est-à-dire des éléments du domaine d'interprétation D , à l'exclusion de tout autre chose (par exemple des sous-ensembles de D). En corollaire, les quantificateurs ne portent que sur des variables *individuelles*. On ne peut pas exprimer une quantification portant sur des ensembles, par exemple on ne peut pas dire : « pour tout ensemble d'individus A non vide », ni « il existe un ensemble A non vide tel que etc. », non plus d'ailleurs que « pour tout prédicat P etc. », ni « il existe un prédicat P tel que... ». Or, on peut très bien imaginer des phrases du langage ordinaire qui utilisent de telles expressions, par exemple :

- *cette voiture a toutes les propriétés que possède un objet de luxe*
- *il y a une propriété commune à tous les habitants de la planète Terre*
- etc.

en mathématiques :

- *tout ensemble qui contient 0 et qui contient le successeur de chacun de ses éléments est identique à l'ensemble des entiers naturels*
- *tout ensemble de nombres entiers qui est majoré possède un plus grand élément*
- etc.

D'autre part, les paragraphes précédents nous ont montré qu'il existe dans la sémantique d'une langue une multitude de *types d'objet* qui diffèrent des seules entités individuelles et valeurs de vérité, par exemple des fonctions (pour représenter les verbes ou les adjectifs). Il faudra donc enrichir la logique si nous voulons être capables de représenter de telles phrases, et ainsi passer au second ordre, voire à des ordres supérieurs. Mais il faudra le faire avec prudence car sinon... on obtient des systèmes trop compliqués.

3.4 Types sémantiques

Nous allons donc introduire un peu d'ordre dans les types d'objet considérés. En particulier, nous avons constaté que, *à partir d'objets de type a et de type b* , on peut toujours désormais fabriquer des objets du type *des fonctions qui à partir d'un objet de type a forment un objet de type b* .

Par exemple, un prédicat unaire forme un objet du type « valeur de vérité » à partir d'un objet du type « entité individuelle », on dira que son type est *fonctionnel*, c'est en l'occurrence le type des fonctions de « entité individuelle » vers « valeur de vérité ». De même, un prédicat binaire forme un objet du type des fonctions de « entité individuelle » vers « valeur de vérité » à partir d'un objet de type « entité individuelle » etc.

Utilisons donc maintenant des symboles pour désigner ces types. Nous avons actuellement deux types de départ (dits « types primitifs »), celui des entités individuelles que nous noterons **e**, et celui des valeurs de vérité que nous noterons **t** (pour *truth-value*). Ensuite, nous avons une règle générale qui nous dit que si nous avons un type a et un type b , alors il est toujours possible de définir le type, que nous noterons $\langle a, b \rangle$, des fonctions des objets de type a vers les objets de type b .

En résumé :

Définition [types primitifs] : les types :

- **t** : type des propositions
- **e** : type des entités individuelles

Remarque : ceci n'est pas exclusif de l'ajout ultérieur d'autres types primitifs. Comme on peut s'en douter, la prise en compte de types primitifs dépend de ce que nous sommes prêts à accepter comme objets de nos discours, autrement dit dépend de notre ontologie implicite. Par exemple, si nous considérons que notre discours parle non seulement d'objets individuels mais aussi d'événements, voire de *situations*, alors nous pourrions introduire des types spécifiques pour cela.

Définition [types (en général)] :

- un type primitif est un type
- si a et b sont des types alors $\langle a, b \rangle$ est un type.

Exemples :

- Sont des types sémantiques :
 $e, t, \langle e, t \rangle, \langle e, \langle e, t \rangle \rangle, \langle \langle e, t \rangle, t \rangle, \langle \langle \langle e, t \rangle, t \rangle, \langle e, t \rangle \rangle$ etc. (une infinité potentielle)
- Etant donné ce que nous avons vu plus haut, un verbe intransitif comme *dormir* est de type $\langle e, t \rangle$, un verbe transitif comme *aimer* est de type $\langle e, \langle e, t \rangle \rangle$, un verbe ditransitif comme *donner* est de type $\langle e, \langle e, \langle e, t \rangle \rangle \rangle$ etc.

3.5 Catégories syntaxiques et types sémantiques

3.5.1 Expressions catégorisées d'une langue et types sémantiques

Pour schématiser, nous posons désormais qu'une langue L est un ensemble d'expressions sur un vocabulaire V munies d'une catégorie syntaxique (appartenant à un ensemble Cat) et d'un type sémantique.

Nous entendons par *expression sur un vocabulaire V* une suite finie d'éléments de V (on note souvent V^* l'ensemble des suites finies d'éléments de V). Par exemple :

- *Pierre*
- *Marie*
- *Pierre apprécie Marie*
- *apprécie Marie*
- *Pierre apprécie*
- *apprécie*

sont des expressions sur $V = \{Pierre, Marie, apprécie\}$

(remarquer qu'à ce niveau de généralité, *Pierre Pierre Pierre* ou *apprécie apprécie Marie* en sont aussi... !). Bien sûr toutes ces expressions ne sont pas dotées d'une catégorie syntaxique, par exemple *Pierre Pierre Pierre* n'en serait pas dotée, non plus d'ailleurs que *Pierre apprécie* (sauf en relation avec certains contextes de dialogue, mais alors l'expression doit être prise comme une exclamative, cf. « Pierre apprécie ! »). Il est donc important de préciser que nous ne retenons dans la définition d'une langue que les expressions qui sont *dotées d'une catégorie syntaxique* (ce qui suppose évidemment une grammaire sous-jacente qui assigne ces catégories aux expressions). On notera aussi que cela permet de faire des distinctions intéressantes : une même expression peut être catégorisée différemment selon les contextes, cela donne autant d'éléments de la langue différents (même s'ils se prononcent de la même façon).

Ainsi, (*rit*, V1) et (*rit*, SV) sont deux expressions catégorisées différentes, (*manger des grenouilles*, SVinf), (*manger des grenouilles*, SN), et (*manger des grenouilles*, N) sont également des expressions catégorisées différentes (cf. « ils sont souvent allés en Bresse pour manger des grenouilles », « manger des grenouilles dégoûte les anglais », « le manger des grenouilles a été étudié par les biologistes »).

Si une expression possède une catégorie syntaxique, elle possèdera aussi un type sémantique. De plus, deux expressions ayant la même catégorie syntaxique auront le même type sémantique⁶. Cela nous autorisera à parler d'une fonction τ de l'ensemble des catégories syntaxiques *Cat* dans l'ensemble des types sémantiques *Typ*.

A titre d'hypothèse, nous allons définir un premier ensemble de types qui pourra être étendu par la suite.

3.5.2 Un exemple de fonction τ :

SN	e
N	<e, t>
Vt	<e, <e, t>>
Vi	<e, t>
SV	<e, t>
A	<<e,t>, <e, t>>
S	t
Det def	<<e, t>, e>

Exemples :

- un nom (N), comme *philosophe*, a le type sémantique **<e, t>**,
- un déterminant défini (Det def) comme *le* a le type sémantique **<<e, t>, e>**
- un syntagme nominal (SN) comme *le philosophe* a le type sémantique **e**
- un verbe transitif (Vt) comme *regarde* a le type sémantique **<e, <e, t>>**
- un nom propre (SN) comme *Orion* a le type sémantique **e**
- un syntagme verbal (SV) comme *regarde Orion* a le type sémantique **<e, t>**
- une phrase (S) comme *le philosophe regarde Orion* a le type sémantique **t**

3.6 Réduction d'une suite de types

Ces exemples nous montrent immédiatement que ce système de types fonctionne d'après une règle selon laquelle dès qu'on met ensemble (par la grammaire) une expression de type a et une expression de type <a, b>, on obtient une expression de type b. Y. Bar-Hillel, dans les années cinquante, avait, à la suite du logicien polonais Ajduckiewicz, développé un système similaire (« une notation quasi-arithmétique pour les catégories sémantiques »). Au lieu

d'écrire <a, b>, on peut en effet utiliser une notation fractionnaire : $\frac{b}{a}$ et dire que la règle ci-dessus se ramène à une simple simplification de fraction. La démarche qui aboutit ainsi au fait que l'expression *le philosophe regarde Orion* possède bien le type **t** se ramène ainsi au calcul suivant :

$$\frac{\frac{e}{t} \frac{t}{e} \frac{e}{e}}{e} e = \frac{e}{t} \frac{t}{e} \frac{t}{e} = e \frac{t}{e} = t$$

⁶ Si besoin est, on subdivisera la catégorie syntaxique en plusieurs sous catégories, comme c'est le cas dans le paragraphe suivant, où la catégorie V est subdivisée en Vt : verbes transitifs et Vi : verbes intransitifs. De même on peut faire une distinction entre les déterminants définis et les indéfinis.

Ou bien, avec notre notation $\langle a, b \rangle$ au lieu de $\frac{b}{a}$:

$$\frac{\frac{\langle \langle e, t \rangle, e \rangle}{e} \quad \langle e, t \rangle}{t} \quad \frac{\langle e, \langle e, t \rangle \rangle}{\langle e, t \rangle} \quad e$$

Evidemment, dans cet exemple, il y a plusieurs réductions possibles. Dans celle qui est exécutée ci-dessus, le *e* le plus à droite, qui correspond à *Orion* se simplifie avec le dénominateur de l'avant dernière fraction, mais on pourrait avoir aussi :

$$\frac{\frac{\langle \langle e, t \rangle, e \rangle}{e} \quad \langle e, t \rangle}{\langle e, t \rangle} \quad \frac{\langle e, \langle e, t \rangle \rangle}{e} \quad t$$

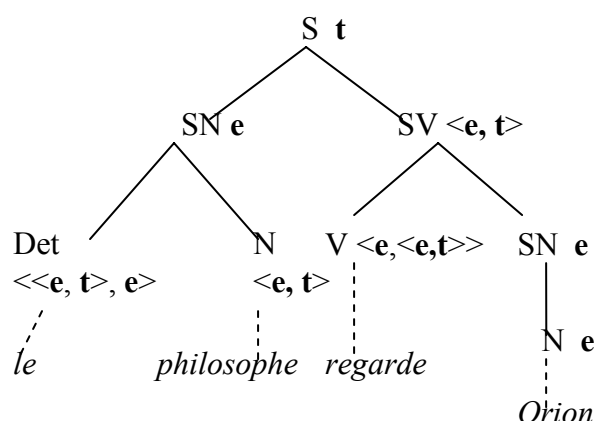
Si, dans $\langle e, \langle e, t \rangle \rangle$, le premier *e* correspond à la place du complément d'objet direct, la première réduction correspond bien à l'interprétation souhaitée où *Orion* prend la place de ce complément, mais dans la deuxième réduction, c'est *le philosophe* qui prend cette place, donnant plutôt une lecture équivalente à *Orion regarde le philosophe* ! L'obtention d'une réduction conforme à l'interprétation souhaitée ne peut donc être indépendante d'un ordre *syntactique* dans lequel s'effectuent les opérations. Cet ordre syntaxique sera soit guidé par une grammaire de constituants, soit par une « grammaire catégorielle ». Nous reviendrons dans une annexe à la solution « grammaire catégorielle ». Pour l'instant, nous retenons la solution « grammaire de constituants ». Nous admettons que pour les exemples traités ici, la syntaxe est réglée par une petite grammaire telle que :

$$\begin{array}{ll} S \rightarrow SN \, SV & Det \rightarrow le \\ SN \rightarrow Det \, N' & N \rightarrow philosophe \\ SN \rightarrow Np & Np \rightarrow Orion \\ N' \rightarrow N & A \rightarrow américain \\ \\ SV \rightarrow V \, SN & V \rightarrow regarde \\ N' \rightarrow N \, A & \end{array}$$

Nous posons comme principe d'association entre syntaxe et sémantique, le premier principe suivant :

Principe d'Application : si *A* et *B* sont deux constituants syntaxiques, si le type sémantique de l'un est $\langle \mu, \nu \rangle$ et le type sémantique de l'autre est μ et s'il existe une règle $X \rightarrow A \, B$ ou une règle $X \rightarrow B \, A$, alors le type sémantique du constituant *X* obtenu par cette règle est ν .

D'après ce principe et étant donnée la grammaire ci-dessus, sur notre exemple, les réductions se font nécessairement de la manière suivante (en utilisant l'arbre syntaxique pour marquer les résultats des réductions aux différentes étapes) :



Plus loin, ce principe sera enrichi de manière à produire effectivement les représentations sémantiques attendues.

Exercices :

1- Quels types donner aux mots des phrases suivantes pour obtenir la réduction de la suite des types à **t** ?

- a- *Le linguiste américain lit Alice au Pays des Merveilles*
(Alice au Pays des Merveilles est vu comme un seul mot, et c'est un nom propre)
- b- *Le linguiste américain lit souvent Chomsky*
- c- *Le linguiste américain pense que Marie a lu Chomsky*
- d- *Le linguiste parle à la fille avec un pull bleu*
- e- *Quel livre Marie a lu ?*
- f- *Quel livre Chomsky pense que Marie a lu ?*

Est-ce que, dans tous les cas, la règle de réduction postulée ci-dessus (« principe d'application ») est suffisante ? sinon, quel principe faut-il ajouter ?

3.7 Types et interprétations

On doit maintenant se poser la question : du point de vue de la sémantique dénotationnelle, que représente un type sémantique ? Nous avons commencé de répondre à cette question au paragraphe 3.2. Nous pouvons maintenant être plus exhaustifs. La réponse tient dans le tableau suivant :

t	$\{0, 1\}$
e	D
<e, t>	$\{0, 1\}^D$
<e, <e, t>>	$(\{0, 1\}^D)^D = \{0, 1\}^{D \times D}$
<<e, t>, e>	$D^{\{0, 1\}^D}$

Commentaires :

- **t** est interprété comme l'ensemble des valeurs de vérité ($t = \text{truth value}$), autrement dit toute expression de type **t** a une valeur qui est soit 1 (vrai) soit 0 (faux),
- **e** est interprété comme l'ensemble des individus du domaine, autrement dit toute expression de type **e** a pour valeur un individu de cet ensemble, noté D ,
- **<e, t>** est interprété comme l'ensemble des fonctions totales de D dans $\{0, 1\}$, autrement dit toute expression de type **<e, t>** a pour valeur une fonction totale qui, à tout individu associe une valeur de vérité, une telle fonction est toujours associée à un prédicat unaire, c'est-à-dire *une propriété*. De ce point de vue, notre affectation de

type signifie simplement que *tout nom commun est conçu sémantiquement comme une propriété* (au nom commun *philosophe*, correspond la propriété *être_un_philosophe*),

- $\langle e, \langle e, t \rangle \rangle$ est interprété comme l'ensemble des fonctions totales de D dans l'ensemble des fonctions totales de D dans $\{0, 1\}$, autrement dit toute expression de type $\langle e, \langle e, t \rangle \rangle$ a pour valeur une fonction totale qui, à un individu du domaine, associe une fonction totale qui, à un individu du domaine, associe une valeur de vérité. Cette fois, il faut donc *deux* individus pour donner une valeur de vérité. On sait que l'ensemble de fonctions totales en question est *isomorphe* à l'ensemble des fonctions totales de $D \times D$ dans $\{0, 1\}$ (intuitivement, cela revient au même de procéder en deux temps : d'abord donner le premier individu, puis le deuxième, ou en un seul temps : donner d'un coup les deux individus sous forme d'un couple).
- $\langle \langle e, t \rangle, e \rangle$ est interprété comme l'ensemble des fonctions totales de l'ensemble des fonctions totales de D dans $\{0, 1\}$ dans D . Autrement dit, toute expression de type $\langle \langle e, t \rangle, e \rangle$ a pour valeur une fonction totale qui, étant donnée une fonction totale de D dans $\{0, 1\}$ (*une propriété*), lui associe un individu. C'est le type que nous avons donné provisoirement à un déterminant (seulement les définis). Considérons en effet le déterminant défini *le*. A partir du nom *philosophe*, représenté par la propriété *être_un_philosophe*, *le* permet d'obtenir un individu bien particulier. On peut donc associer à *le* cette fonction particulière qui, étant donné l'extension d'une propriété, va sélectionner un individu de cette extension (une fonction *de choix* en quelque sorte). Nous verrons cependant plus loin que cette assignation de type ne convient pas aux autres déterminants comme *un*, *chaque*, *quelque*, *tout* etc.

Nous pouvons évidemment résumer le tableau et les commentaires précédents par une définition récursive très simple :

Définition [dénotation des types sémantiques] : étant donné un domaine (supposé non vide) D et une fonction d'interprétation I , les types sémantiques sont interprétés à partir du domaine D de la manière suivante :

- $I(t) = \{0, 1\}$
- $I(e) = D$
- pour tous types a et b , $I(a \rightarrow b) = I(b)^{I(a)}$

3.8 Notation des fonctions : le λ -calcul

Nous savons désormais que les verbes sont interprétés comme des fonctions. Il nous faut un système de notation pour exprimer ces fonctions, de telle sorte notamment que lorsque nous devons appliquer la fonction **observe** à deux arguments **Galilée** et **Orion**, nous puissions savoir *dans quel ordre* nous devons faire ces applications.

Comme l'observa Frege, si nous partons d'une phrase comme :

(1) *Galilée observe Orion*

nous obtenons par « abstraction sur la première place » un schéma :

(2) *_ observe Orion*

que nous pouvons voir comme « une fonction » pouvant s'appliquer à l'objet « Galilée ». En poursuivant le même processus, c'est-à-dire en abstrayant sur la deuxième place, nous pouvons obtenir le schéma :

(3) $_ observe _$

qui peut être vu à son tour comme une fonction pouvant s'appliquer à l'objet « Orion » pour donner le schéma (2). Mais rendus à ce stade, nous constatons que rien ne distingue les deux places à partir desquelles a eu lieu un processus d'abstraction, en particulier rien n'indique dans quel ordre ces deux places doivent être remplies : si nous repartons dans l'autre sens, autrement dit si nous essayons de reconstruire (1) à partir du schéma (3) et des deux objets « Galilée » et « Orion », rien dans (3) ne nous indique quelle place doit être remplie par « Galilée », ni quelle place doit être remplie par « Orion ».

Church et Curry, dans les années trente, ont inventé le λ -calcul, qui permet de répondre à ce type de question. Le schéma (2) sera réécrit sous la forme de la fonction :

(4) $\lambda x (x \text{ observe Orion})$

Par abstraction sur la deuxième place, on obtient alors :

(5) $\lambda y \lambda x (x \text{ observe } y)$

qui porte la marque de l'ordre dans lequel on a effectué les abstractions. De cette manière, partant de (5), si nous appliquons d'abord (5) à « Orion », nous obtiendrons nécessairement (4) et si nous appliquons (4) à « Galilée », nous obtiendrons nécessairement (1).

Remarquons que l'utilisation du schéma (5) plutôt que celle du schéma (6) suivant :

(6) $\lambda x \lambda y (x \text{ observe } y)$

est adaptée au cas où nous avons une grammaire qui fabrique un SV à partir d'un verbe et d'un SN (objet direct) avant de fabriquer un S par combinaison d'un SN (sujet) et du SV.

Le λ -calcul contient une définition des λ -termes :

Définition [λ -termes]

- les variables et les constantes sont des λ -termes,
- si x est une variable et si M est un λ -terme, alors $\lambda x M$ est un λ -terme,
- si M et N sont des λ -termes, alors $(M N)$ est un λ -terme (l'application de M à N , qui sera aussi noté parfois $M(N)$)

Il permet d'obtenir un calcul sur les fonctions, permettant de définir rigoureusement leur *application* à un argument, leurs *compositions* etc.

Définition [β -réduction]

- $(\lambda x M N) \triangleright M[N/x]$ (résultat de la substitution de N à x dans M partout où x figure)

Avec ces définitions, nous pouvons calculer (en admettant que **observe**, Orion et Galilée sont des constantes) :

$(\lambda y \lambda x ((\text{observe } y) x) \text{ Orion}) \triangleright \lambda x ((\text{observe Orion}) x)$

$(\lambda x ((\text{observe Orion}) x) \text{ Galilée}) \triangleright ((\text{observe Orion}) \text{ Galilée})$

NB : le fait d'écrire $((\text{observe } y) x)$ ou $(\text{observe } x y)$ ou **observe**(x, y) ou $(x \text{ observe } y)$ est inessentiel. L'écriture conforme à la définition des lambda-termes est la première, les autres

en sont des réécritures qu'on peut utiliser en fonction de ses goûts propres (c'est ce qu'on appelle du « sucre syntaxique » !). Noter toutefois que l'ordre des variables n'est pas toujours le même. Dans $((\mathbf{observe} \ y) \ x)$, on a mis l'objet direct (y) le plus près du verbe (c'est une structure qui correspond bien à l'arbre syntaxique usuel), alors que dans $(\mathbf{observe} \ x \ y)$ et dans $\mathbf{observe}(x, y)$, l'objet direct est, conformément « à l'habitude » en logique des prédicats, en deuxième position par rapport au sujet.

Ainsi désormais, de même qu'un nom propre est représenté (provisoirement) par une **constante** et une phrase complète par une **proposition** exprimée par une formule de logique, un verbe transitif sera représenté par un **λ -terme** $\lambda y \lambda x ((v \ x) \ y)$ (ou $\lambda y \lambda x \ v(x, y)$ etc.) où v est une constante du lambda calcul (NB : ce qui suppose implicitement que systématiquement, la place de l'objet direct est remplie avant celle du sujet), par exemple $\lambda y \lambda x ((\mathbf{aimer} \ y) \ x)$.

Par conformité avec ce que nous avons vu dans la leçon 1, nous désignerons toujours par $[[\psi]]^{M,g}$ l'interprétation d'une expression ψ par rapport à une structure M et une assignation g , ou simplement par $[[\psi]]$ cette interprétation quand il n'y a pas d'ambiguïté. Avec cette notation, si x est une variable individuelle, et si ϕ est une proposition, $[[\lambda x. \phi]]^{M,g}$ sera **la** fonction h qui, à tout élément x_0 de D associe 1 si $[[\phi[x_0/x]]]^{M,g} = 1$ et 0 sinon. Nous allons maintenant étendre le champ de la fonction d'interprétation $[[\cdot]]^{M,g}$.

3.9 Abstractions d'ordre supérieur

3.9.1 Abstraction d'une propriété

Le processus d'abstraction n'est pas limité aux variables individuelles ($x, y, z, \dots$). A condition de respecter certaines règles (que nous verrons plus loin, **λ -calcul typé**), on peut procéder à des abstractions sur des variables d'ordre supérieur, par exemple des *variables de prédicat*. On aurait pu ainsi continuer le processus qui a conduit à (5) en abstrayant sur une variable mise à la place de la seule constante qui reste, **observe**. On aurait obtenu :

$$(7) \quad \lambda P \lambda y \lambda x (x \ P \ y)$$

Ou, en respectant la définition des lambda-termes :

$$(8) \quad \lambda P \lambda y \lambda x ((P \ y) \ x)$$

On aurait pu alors aussi bien calculer :

$$\begin{aligned} &(((\lambda P \lambda y \lambda x ((P \ y) \ x) \ \mathbf{observe}) \ \text{Orion}) \ \text{Galilée}) \triangleright \\ &((\lambda y \lambda x ((\mathbf{observe} \ y) \ x) \ \text{Orion}) \ \text{Galilée}) \triangleright \\ &(\lambda x ((\mathbf{observe} \ \text{Orion}) \ x) \ \text{Galilée}) \triangleright \\ &((\mathbf{observe} \ \text{Orion}) \ \text{Galilée}) \end{aligned}$$

Evidemment, ceci n'a pas un grand intérêt (juste celui de définir une sorte de schéma général pour tous les verbes transitifs), mais on verra une application intéressante de cette possibilité dans l'exercice suivant :

Exercice : démontrer qu'en attribuant au pronom réflexif (*se* en français, *him(her)self* en anglais) la représentation sémantique $\lambda P. \lambda u. ((P \ u) \ u)$, on transforme un verbe transitif en un verbe réflexif. Quel est alors le type sémantique du pronom réflexif ?

Au point où nous en sommes, une autre application concerne le déterminant défini. Comme dit au 2.6, celui-ci est de type sémantique $\langle\langle e, t \rangle, e\rangle$, ce qui correspond à une fonction de l'ensemble des fonctions de D dans $\{0, 1\}$ dans D , autrement dit à une fonction qui à toute propriété associe un individu. Mais on ne peut avoir de tels objets que si nous avons, en plus des variables individuelles, des variables de prédicat. En effet, une telle fonction s'écrit :

$$h : X \mapsto x \text{ tel que } [[X(x)]] = 1 \text{ et } x \text{ est } \textit{unique} \text{ à être tel que } [[X(x)]] = 1 \\ \text{où } X \text{ est une variable de prédicat}$$

Il s'ensuit qu'implicitement nous sortons du domaine de la logique des prédicats du premier ordre pour être dans une logique des prédicats *d'ordre supérieur* quelconque !

Soit par exemple la phrase :

(9) *la fille au pull bleu ciel dort*

cette phrase a une interprétation qui prend la valeur vraie si et seulement si :

(10) il existe une fille x telle que x a un pull bleu ciel et toute fille ayant un pull bleu ciel est x et x dort

soit en formule logique :

$$(11) \quad (\exists x)(\textit{fille}(x) \wedge a_pull_bleu(x) \wedge ((\forall y)(\textit{fille}(y) \wedge a_pull_bleu(y)) \Rightarrow (y = x))) \wedge \textit{dort}(x))$$

En raisonnant intuitivement (car nous n'avons pas tous les moyens de faire autrement), on peut penser que puisqu'il y a existence et unicité de x tel que x est une fille et x a un pull bleu ciel⁷, on peut utiliser une notation pour désigner cet unique x possédant cette propriété. Notons le : $tx.(\textit{fille}(x) \wedge a_pull_bleu(x))$ (soit en français : le x tel que x est une fille et x a un pull bleu ciel). Cette expression désigne un unique élément de D , et (11) peut se réécrire :

$$(12) \quad \textit{dort}(tx.(\textit{fille}(x) \wedge a_pull_bleu(x)))$$

Ceci dit, nous avons introduit un nouveau genre de fonctions, les fonctions des propriétés dans les individus, qui peuvent elles aussi s'exprimer par des λ -termes. Le déterminant défini est associé à une telle fonction, son λ -terme est :

$$(13) \quad \lambda P.tx.P(x)$$

Noter que *être une fille au pull bleu ciel* se traduit dans notre système par :

$$(14) \quad \lambda y.(\textit{fille}(y) \wedge a_pull_bleu(y))$$

et que, en appliquant (13) à (14), on obtient bien :

⁷ L'analyse du défini est en réalité plus difficile. Que se passe-t-il en effet s'il n'y a **pas** de fille au pull bleu ciel ? doit-on dire nécessairement que la phrase est fausse ? ou bien « qu'elle n'a pas de sens » ? en quel sens pourrait-on alors dire qu'elle « n'a pas de sens » ? mais pour l'instant, cette analyse nous suffira.

$$\begin{aligned}
& (\lambda P. \lambda x. P(x) \lambda y. (fille(y) \wedge a_pull_bleu(y))) \triangleright \\
& \lambda x. (\lambda y. (fille(y) \wedge a_pull_bleu(y)) x) \triangleright \\
& \lambda x. (fille(x) \wedge a_pull_bleu(x))
\end{aligned}$$

3.9.2 Les paradoxes de l'application à soi-même et l'utilité des types

Le λ -calcul possède donc une puissance plus grande que ce qu'on avait peut-être pu percevoir au premier abord... mais qui dit grande puissance dit aussi grands risques. On pourrait par exemple concevoir, pour une propriété, la propriété de s'appliquer à soi-même, et donc aussi la propriété de **ne pas** s'appliquer à soi-même ! Regardons comment cela s'exprimerait en λ -calcul. Une propriété P s'appliquant à elle-même signifierait que $P(P)$ prend la valeur vraie (par exemple, on peut dire que la propriété d'être une idée est en elle-même une idée, ou bien on pourrait dire que la propriété d'être drôle est en elle-même... drôle !), ou avec la notation du lambda-calcul, que $(P P)$ prend la valeur vraie. Ainsi, la propriété de s'appliquer à soi-même se noterait-elle : $\lambda P. (P P)$. par exemple, si on veut savoir si la propriété d'être drôle est en elle-même drôle, on évalue : $(\lambda P. (P P) \lambda Z. drôle(Z))$, qui donne, par β -réduction : $(\lambda Z. drôle(Z) \lambda Z. drôle(Z)) \triangleright drôle(\lambda Z. drôle(Z))$. La propriété de ne pas s'appliquer à soi-même se noterait, de manière similaire par : $\lambda P. \neg P(P)$ ou, en notation lambda : $\lambda P. (\neg (P P))$. Posons nous alors les questions :

1. est-ce que la propriété de s'appliquer à soi-même s'applique à elle-même ?
2. est-ce que la propriété de ne pas s'appliquer à soi-même s'applique à elle-même ?

Pour la question 1 : il s'agit d'étudier si $(\lambda P. (P P) \lambda P. (P P))$ est vrai. Mais :

$$(\lambda P. (P P) \lambda P. (P P)) \triangleright (\lambda P. (P P) \lambda P. (P P)) \triangleright (\lambda P. (P P) \lambda P. (P P)) \triangleright \dots$$

autrement dit ce lambda-terme ne se réduit jamais (on dit qu'il ne se « normalise » jamais), on obtient une suite infinie de lambda-termes qui sont toujours les mêmes.

Pour la question 2 : c'est encore pire. Supposons que $\lambda P. \neg P(P)$ s'applique à elle-même, alors, par définition, cela voudrait dire qu'elle ne s'applique pas à elle-même, et supposons qu'elle ne s'applique pas à elle-même, alors elle vérifierait justement la propriété de ne pas s'appliquer à elle-même, autrement dit $\lambda P. \neg P(P)$ s'appliquerait à elle, mais elle, c'est justement cette propriété là, donc elle s'appliquerait à elle-même ! Nous serions alors en pleine contradiction.

Un tel argument a été énoncé la première fois par le philosophe et logicien anglais Bertrand Russell en 1902. Il voulait montrer par là que le système de Frege qui a priori permettait de définir la notion de « s'appliquer à soi-même » en tant que propriété, était contradictoire.

Du constat de cette contradiction, ont résulté de nombreuses recherches, ayant conduit à des solutions différentes. En ce qui concerne notre objet, on voit qu'une solution consiste à interdire la possibilité qu'une fonction s'applique à elle-même. Pour cela, il faut justement revenir à notre idée initiale de *type* et faire en sorte que les seules fonctions qui puissent s'appliquer à un objet de type a soient celles qui sont de type $\langle a, b \rangle$ pour un b quelconque (mais on ne peut pas avoir une fonction de type $\langle a, b \rangle$ s'appliquant à une fonction de type $\langle a, b \rangle$, donc pas une fonction pouvant s'appliquer à elle-même)..

3.9.3 Le λ -calcul typé

Reprenons la définition des λ -termes.

Définition [λ -termes typés]

- les variables et les constantes de type τ sont des λ -termes de type τ ,
- si x est une variable de type μ et si M est un λ -terme de type ν , alors $\lambda x M$ est un λ -terme de type $\langle \mu, \nu \rangle$,
- si M est un λ -terme de type $\langle \mu, \nu \rangle$, et N est un λ -terme de type μ , alors $(M N)$ est un λ -terme de type ν

Désormais, nous pouvons, comme promis, étendre le champ de la fonction d'interprétation $[[\cdot]]^{M,g}$. Elle ne va s'appliquer évidemment qu'aux termes du λ -calcul typé. Nous aurons :

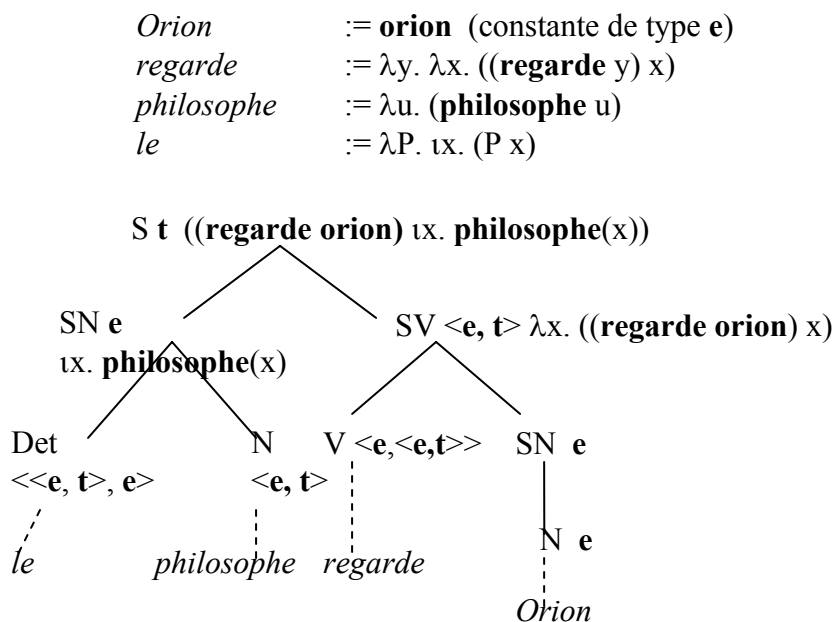
- 1- tout type primitif τ est interprété par un ensemble D_τ
par exemple, $D_e = D$ et $D_t = \{0, 1\}$
- 2- tout type fonctionnel $\langle \mu, \nu \rangle$ est interprété par l'ensemble $D_{\langle \mu, \nu \rangle} = \{D_\mu \rightarrow D_\nu\}$ (encore noté $D_\nu^{D_\mu}$)
- 3- si v est une variable de type μ et si ϕ est un λ -terme de type ν , alors, pour toute structure M et toute assignation g , $[[\lambda v. \phi]]^{M,g}$ est la fonction totale h de D_μ dans D_ν qui, à tout $a \in D_\mu$ associe $[[\phi[a/v]]]^{M,g}$.

3.10 Grammaire et sémantique

Nous avons maintenant ce qu'il faut pour obtenir une représentation sémantique d'une phrase reconnue par une grammaire G (au moins dans quelques cas simples). Nous complétons le principe d'application de la manière suivante :

Principe d'Application : si A et B sont deux constituants syntaxiques, si l'un possède la représentation sémantique $\lambda v. \phi$ où v est de type μ et ϕ de type ν , et l'autre une représentation sémantique α de type sémantique μ et s'il existe une règle $X \rightarrow A B$ ou une règle $X \rightarrow B A$, alors le constituant X obtenu par cette règle possède la représentation sémantique $(\lambda v. \phi \alpha)$ (qui se réduit à $\phi[\alpha/v]$) de type ν .

Dans le cas de l'exemple *le philosophe regarde Orion*, nous obtenons, avec :



Exercices :

- 1- Soit la phrase :

tout oiseau chante

que nous souhaitons traduire en :

$$(\forall x)(oiseau(x) \Rightarrow chante(x))$$

En admettant que *oiseau*, en tant que nom commun est de type sémantique $\langle e, t \rangle$ et qu'il lui correspond le λ -terme $\lambda x. oiseau(x)$, et que *chante*, en tant que verbe intransitif, est aussi de type sémantique $\langle e, t \rangle$ et qu'il lui correspond le λ -terme $\lambda u. chante(u)$, quel est le λ -terme associé à *tout oiseau* (obtenu par abstraction de *chante*) ? quel est son type sémantique ? Quel est le λ -terme associé à *tout* (obtenu par abstraction de *oiseau* à partir de *tout oiseau*) ? quel est son type sémantique ?

Même exercice avec la phrase :

Un oiseau chante

Quels sont les types sémantiques et les λ -termes associés de *un oiseau* et de *un* ?

2- Quel type sémantique et quel λ -terme pour la coordination *et* dans :

Marie chante et danse

3- Quels type sémantique et λ -terme pour la coordination *et* dans :

Un rossignol et un chardonneret chantent

Montrer qu'à chaque fois, on a bien une suite de types sémantiques qui se réduit à t .

Vérifier que la construction de la représentation sémantique finale de la phrase suit bien les étapes de cette réduction.

4- Tenter d'analyser de la même manière la phrase :

Le professeur de Paris 8 questionne un étudiant

Est-il facile d'obtenir la représentation sémantique souhaitée de cette phrase, en supposant qu'elle soit :

$$(\exists x)(etudiant(x) \wedge questionne(ty.(professeur(y) \wedge de_Paris8(y)), x))$$

5- Reprendre l'exercice du 2.5 et pour chacune des phrases donner la représentation sémantique obtenue en suivant les étapes de la réduction (et en attribuant bien sûr aux mots des phrases des représentations sémantiques sous forme de λ -termes convenables).