Apprentissage syntaxico-sémantique du langage naturel

Isabelle Tellier

LIFL et Université Charles de Gaulle-lille3 (UFR IDIST)

59 653 Villeneuve d'Ascq Cedex, FRANCE

tel : 03-20-41-61-78 ; fax : 03-20-41-61-71

Email : tellier@univ-lille3.fr

RESUME **:**

Dans cet article, nous nous intéressons à l'apprentissage formel du langage naturel par exemples positifs, en tenant compte simultanément d'informations syntaxiques et sémantiques.

Le formalisme grammatical utilisé est celui des grammaires catégorielles et la représentation des connaissances est assurée par une logique formelle. Le principe de compositionalité est traduit par un isomorphisme d'arbres permettant d'articuler précisément les deux niveaux d'analyse.

Nous proposons dans ce cadre un algorithme d'apprentissage et nous simulons son comportement sur un exemple simple. Son efficacité et sa pertinence cognitive sont discutées.

MOTS CLES :

langage naturel, inférence grammaticale, grammaires catégorielles, liens syntaxe/sémantique, représentation des connaissances, logique, modèles d'apprentissage

1. Introduction

L'apprentissage du langage est un processus cognitif humain universel qui pose de redoutables problèmes de modélisation. En effet, les psycholinguistes ont depuis longtemps remarqué que les enfants acquièrent leur langue maternelle en présence d'exemples positifs seuls ([Wexler & Culicover 80]). Pourtant, les langues naturelles appartiennent au moins à la classe des langages context-free ou algébriques ([Schieber 85]) et cette classe n'est pas apprenable par exemples positifs dans les principaux modèles théoriques de l'apprentissage (ceux de [Gold 67] et [Valiant 84]).

Pour résoudre cet apparent paradoxe, plusieurs solutions ont été proposées. Dans la première, issue des intuitions de Chomsky ([Chomsky 65, 68]), on suppose que les langues humaines ont des propriétés spécifiques dont l'esprit humain a une connaissance innée. Ainsi, les grammaires context-sensitive deviennent apprenables par exemple positifs si l'apprenant connaît une borne sur le nombre de leurs règles ([Shinohara 90]). Suivant une autre démarche, on peut faciliter l'inférence grammaticale grâce au choix et à la présentation des exemples ([Sakakibara 92], [Li & Vitanyi 93]). Cette approche formalise l'aide apportée par un professeur ([Denis & Gilleron 97]).

Dans un ensemble de travaux plus anciens, enfin, l'apprentissage des langues naturelles n'est pas réduit à la seule inférence grammaticale, et des informations sémantiques y jouent un rôle fondamental ([Hamburger & Wexler 75], [Anderson 77], [Langley 82], [Hill 83]). Notre travail se rattache à cette famille. Cette solution permet d'ailleurs une réinterprétation des précédentes. En effet, sans doute les langues naturelles ont-elles des propriétés structurelles (difficiles à caractériser) dues au fait qu'on peut leur associer une sémantique. De même, dans l'apprentissage naturel, la pertinence sémantique ou pragmatique des exemples joue-t-elle évidemment un grand rôle dans leur présentation.

Notre modèle prend donc en compte le fait que les langues naturelles *servent à véhiculer du sens*. Mais pour exploiter une telle propriété d'un point de vue informatique, il est nécessaire de disposer d'une théorie qui articule précisément la syntaxe et la sémantique. L'articulation la plus forte est décrite par le principe de compositionalité dû à Frege, selon lequel *le sens d'une phrase ne dépend que du sens de ses constituants et de leur mode de combinaison*. Ce principe a acquis une formulation rigoureuse explicite dans les travaux du logicien Montague ([Montague 74], [Dowty 81]) et ceux de ses héritiers.

Dans cet article, nous allons tout d'abord exposer une version contemporaine du cadre syntaxico-sémantique proposé par Montague, fondée sur les grammaires catégorielles et la sémantique logique. Nous montrerons ensuite en quoi ce modèle est adapté à la modélisation du processus d'apprentissage d'une langue naturelle.

2. Analyse syntaxique par les grammaires catégorielles

Dans les grammaires catégorielles, chaque mot est associé à un ensemble fini de catégories qui explicitent ses potentialités combinatoires. La syntaxe repose donc entièrement sur les catégories affectées aux mots du vocabulaire. Cette forte lexicalisation est bien adaptée au langage naturel ([Oehrle & *alii* 88]). De plus, les arbres d'analyse syntaxique produits par ces grammaires mettent en évidence la structure fonctionnelle du langage et vont être exploités dans la phase d'analyse sémantique.

2.1 Définition générale des grammaires catégorielles

Une grammaire catégorielle G est un quadruplet G=<V, C, f, S> avec :

- V est l'alphabet (ou le vocabulaire) fini de G ;

- C est l'ensemble fini des catégories élémentaires de G ;

A partir de C, on définit l'ensemble C' de toutes les catégories possibles de G comme la clôture récursive de C par les opérateurs notés / et \ (qui peuvent être considérés comme des traits de fraction orientés). C' est ainsi le plus petit ensemble vérifiant :

* $C \subseteq C'$;

* si $X \in C'$ et $Y \in C'$ alors $X/Y \in C'$ et $Y\backslash X \in C'$;

- f est une fonction : V—> P(C') où P(C') est l'ensemble des sous-ensembles finis de C', qui associe à chaque élément v de V l'ensemble fini $f(v) \subseteq C'$ de ses catégories ;

- $S \in C$ est la catégorie axiomatique de G.

Dans ce formalisme, le langage reconnu par G est l'ensemble des concaténations finies de mots du vocabulaire pour lesquelles il existe une affectation de catégories qui peut être « réduite » à la catégorie axiomatique S. Plusieurs types de grammaires catégorielles ont été proposées, différant par les règles de réduction qui y sont admises. Les deux plus connus sont notés respectivement AB et L.

2.2 Grammaires catégorielles de type AB ([Bar Hillel 53])

Une grammaire catégorielle de type AB (pour Ajdukiewicz-Bar Hillel) est une grammaire catégorielle dans laquelle les règles de réduction admises pour toutes catégories X et Y dans C' sont :

- R1 : $X/Y \ . \ Y \longrightarrow X$

- R'1 : $Y \ . \ Y\backslash X \longrightarrow X$

La notation fractionnaire des catégories de C' trouve sa raison d'être dans ces règles. Les opérateurs / et \ sont traitées comme des fractions orientées ; la concaténation de catégories étant dans ce cas assimilée à une multiplication. Le caractère en général non commutatif des mots d'une phrase dans les langages naturels justifie la nécessité de distinguer une fraction « à gauche » notée \ et une fraction « à droite » notée /. Le langage L(G) reconnu par G est alors défini par :

$L(G)=\{w \in V^*; \exists n \in N \ \forall i \in \{1,..., n\} \ w_i \in V, w=w_1...w_n$ et $\exists C_i \in f(w_i), C_1...C_n \text{—}^*\text{—>} S\}$.

Exemple :

Définissons une grammaire catégorielle de type AB reconnaissant un petit sous-ensemble du français construit sur le vocabulaire V={un, chaque, chat, homme, Jean, Marie, Paul, dort, aime, est ...}. L'ensemble des catégories de base nécessaire se réduit à C={S, T, NC}, où T désigne la catégorie des termes, affectée aux noms propres, et NC celle des noms communs. Les verbes intransitifs reçoivent la catégorie T\S, les verbes transitifs (T\S)/T et deux catégories sont associées aux déterminants « un » et « chaque » : (S/(T\S))/NC pour l'introduction des groupes nominaux en position sujet et ((S/T)\S)/NC pour ceux qui sont en position de COD. Cette grammaire reconnaît des phrases comme « un chat dort » ou « Jean est Paul », comme l'illustre la figure 1.

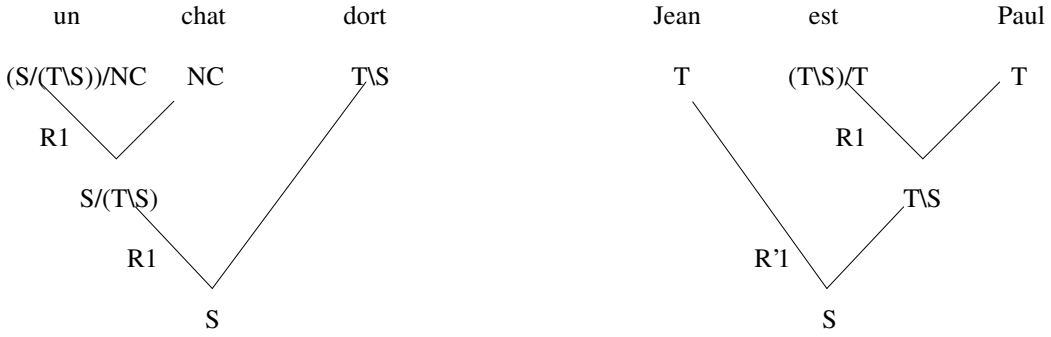


figure 1 : arbres d'analyse syntaxique

La classe des langages reconnus par les grammaires catégorielles de type AB est la classe des langages algébriques ([Bar Hillel & *alii* 60]). Mais les règles R1 et R'1 ne permettent pas des réductions que la notation fractionnaire des catégories rend pourtant intuitives comme :

- X/Y . Y/Z —> X/Z (combinaison de catégories fractionnaires) ;

- (Y\X)/Z —> Y\(X/Z) (associativité sur les opérateurs / et \).

Le système de Lambek-Gentzen a été défini pour remédier à ces limites. Il est à la base des grammaires catégorielles de type L.

2.3 Les grammaires catégorielles de type L ([Lambek 58])

Une grammaire catégorielle de type L (pour Lambek) est une grammaire catégorielle dans laquelle les règles de réduction admises sont tous les séquents valides du système de Lambek-Gentzen défini par :

- une infinité d'axiomes : pour tout X dans C', le séquent X —> X (1) est valide ;

- deux couples de règles d'inférence définis à la manière de Gentzen (i.e. si le ou les séquent(s) du dessus sont valides, alors celui du dessous l'est aussi) :

$$*\quad \frac{T \,.\, X \longrightarrow Y}{T \longrightarrow Y/X} \quad (2) \qquad\qquad \frac{X \,.\, T \longrightarrow Y}{T \longrightarrow X\backslash Y} \quad (2')$$

$$*\quad \frac{T \longrightarrow X \qquad U \,.\, Y \,.\, V \longrightarrow Z}{U \,.\, Y/X \,.\, T \,.\, V \longrightarrow Z} \quad (3) \qquad\qquad \frac{T \longrightarrow X \qquad U \,.\, Y \,.\, V \longrightarrow Z}{U \,.\, T \,.\, X\backslash Y \,.\, V \longrightarrow Z} \quad (3')$$

où $X \in C'$, $Y \in C'$ et U, V et T sont des suites quelconques de catégories de C' (T suite non vide).

Le langage L(G) reconnu par G est alors le suivant :

$L(G)=\{w \in V^*;\ \exists n \in N\ \forall i \in \{1,..., n\}\ w_i \in V,\ w=w_1...w_n \text{ et } \exists C_i \in f(w_i),\ C_1...C_n \longrightarrow S \text{ est valide}\}$.

Exemple :

Les règles R1 et R'1 sont, bien sûr, des séquents valides du système de Lambek (conséquence immédiate des règles (3) et (3') avec des axiomes de type (1) au-dessus). Mais d'autres règles le sont aussi, parmi lesquelles les plus utiles, pour toute catégorie X, Y et Z dans C', s'écrivent ([Moortgat 88]) :

- R2 : X/Y . Y/Z —> X/Z R' 2 : Z\Y . Y\X—> Z\X
- R3 : (Y\X)/Z —> Y\(X/Z) R' 3 : Y\(X/Z)—> (Y\X)/Z
- R4 : X —> Y/(X\Y) R' 4 : X—> (Y/X)\Y

Démontrons par exemple que la règle R4 est un séquent valide :

$$\dfrac{\dfrac{X \longrightarrow X \ (1) \qquad Y \longrightarrow Y \ (1)}{X \,.\, X\backslash Y \longrightarrow Y \ \ (3')}}{X \longrightarrow Y/(X\backslash Y) \ \ (2)}$$

La figure 2 montre deux arbres d'analyse syntaxique pour la phrase « Chaque homme aime une femme ». Si l'affectation des catégories aux mots est la même que précédemment, alors l'usage soit de la règle R2 soit de la règle R2' est nécessaire. Ces deux analyses seront toutes les deux utiles dans la phase d'analyse sémantique car la phrase est ambiguë (comme on le verra en 3.2).

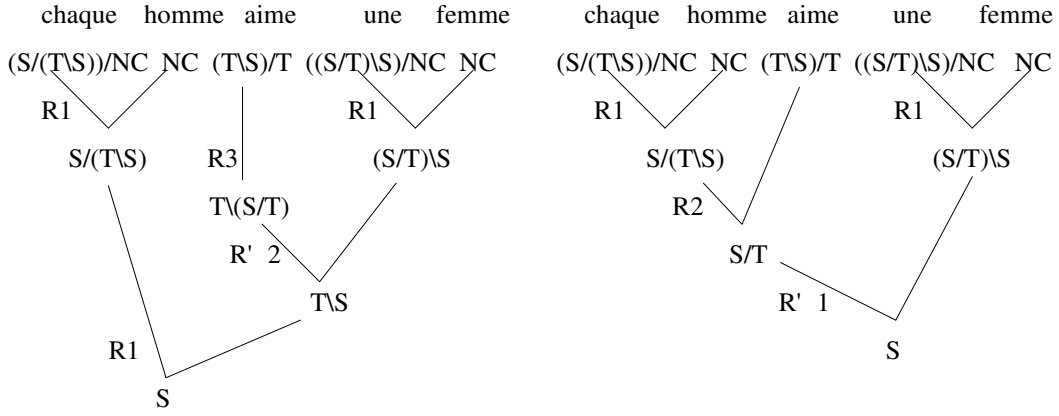


figure 2 : deux arbres d'analyse pour la phrase « chaque homme aime une femme »

Le système de Lambek-Gentzen et les grammaires catégorielles de types L ont donné lieu à de nombreuses études. Les principaux résultats les concernant sont les suivants :

- Le système de Lambek-Gentzen est décidable. En effet, dans chaque règle d'inférence il y a un opérateur / ou \ en moins au dessus de la ligne qu'en dessous. Pour décider si un séquent donné est valide ou non, il suffit donc d'appliquer systématiquement toutes les règles d'inférence possibles en chaînage arrière, en éliminant ainsi un opérateur à chaque étape. Après un nombre d'étapes borné par le nombre initial d'opérateurs dans le séquent, on vérifie si les séquents restants sont ou non des axiomes.

- Il est impossible des définir un nombre fini de séquents valides (par exemples les règles R1 à R'4) équivalent au système de Lambek-Gentzen complet ([Zielonka 81]). La définition sous forme de règles d'inférences sur des séquents est donc la plus puissante possible.

- La classe des langages reconnus par les grammaires catégorielles de type L est la classe des langages algébriques ([Pentus 92]), qui demeure une approximation intéressante pour décrire les phénomènes linguistiques. Le problème de l'appartenance d'une phrase à un tel langage a une complexité polynomiale relativement à la taille de cette phrase (mais malheureusement pas relativement à la taille de la grammaire) ([Finkel & Tellier 96]).

Le principal intérêt du système de Lambek est qu'il permet de limiter le nombre de catégories associées à chaque mot du vocabulaire (ce nombre est un paramètre important dans le calcul de la complexité de l'analyse syntaxique). Par exemple, la figure 2 illustre qu'en associant une seule catégorie

à « aime » et en utilisant des règles valides de Lambek, on obtient deux analyses syntaxiques pour la phrase. Pour obtenir ces deux analyses mais en se limitant à l'usage des règles R1 et R'1, on serait obligé d'associer à « aime » deux nouvelles catégories complexes et ad hoc : (T\S)/((S/T)\S) et (S/(T\S))\(S/T).

De plus, nous pensons que les grammaires catégorielles sont bien adaptées à la formalisation de l'apprentissage car il est plus naturel d'apprendre les propriétés combinatoires des mots que d'apprendre des règles abstraites.

3. De la syntaxe à la sémantique

L'idée clé due à Montague ([Montague 74]) est la définition d'un isomorphisme qui s'applique sur les arbres d'analyse syntaxique et produit des arbres sémantiques. Ce mécanisme, traduction formelle du principe de compositionalité, permet d'associer automatiquement à une phrase syntaxiquement correcte une (ou plusieurs dans le cas de phrases ambiguës) formule(s) logique(s). Nous montrons ici qu'il s'adapte très facilement aux grammaires catégorielles (Montague utilisait un autre formalisme).

3.1 La représentation sémantique

La représentation sémantique que nous utilisons est une légère extension de la logique des prédicats du premier ordre, inspirée de la « logique intensionnelle » définie par Montague ([Dowty 81], [Chambreuil 89]). Elle sera notée IL.

- IL est un langage typé : l'ensemble I de tous ses types possibles contient :

  * des types élémentaires : $e \in I$ (type des « entités ») et $t \in I$ (type des « valeurs de vérité ») ;

  * pour tous les types $u \in I$ et $v \in I$, $<u,v> \in I$ ($<u,v>$ est le type des fonctions prenant un argument de type u et renvoyant un résultat de type v).

- sémantique de IL (modèle associé): à chaque type $w \in I$ correspond un ensemble de dénotation D_w

  * $D_e = E$ où E est l'ensemble dénombrable de toutes les entités du modèle ;

  * $D_t = \{0,1\}$;

  * $D_{<u,v>} = D_v^{Du}$: l'ensemble de dénotation d'un type composé est un ensemble de fonctions.

IL inclut aussi les quantificateurs usuels $\exists$ et $\forall$ et les expressions lambda. Ainsi, si x est une variable de type u et Φ est une expression de type v, alors l'expression $\lambda x.\Phi$ est une expression de type $<u,v>$ et la substitution logique n'est possible qu'entre types identiques (ainsi l'expression précédente ne peut s'appliquer qu'à une expression de type u).

Exemple :

dort'(Jean') est une formule logique de IL de type t, c'est-à-dire une proposition, construite avec :

- dort', de type $<e,t>$, qui est une fonction de E dans $\{0,1\}$ ou encore un prédicat à un argument ;

- Jean', de type e, qui est un élément de E.

3.2 L'isomorphisme d'arbre ([Rozier & Tellier 92])

L'isomorphisme qui s'applique aux arbres produits par l'analyse syntaxique est défini par :

- la traduction des catégories de la grammaire en types logiques (fonction $k : C' \longrightarrow I$) :

  * catégories élémentaires : k(S)=t et pour celles utilisées dans l'exemple k(T)=e et k(NC)=<e,t>;

  * catégories de C' : pour tout $X \in C'$ et $Y \in C'$, $k(X/Y)=k(Y\backslash X)=<k(Y),k(X)>$.

- la traduction du vocabulaire (fonction $q : V \times C' \longrightarrow IL$) : à chaque couple (v,U) où $v \in V$ et $U \in f(v) \subseteq C'$ est une de ses catégories est associée une formule de IL de type $k(U) \in I$ notée q(v,U).

Les traductions utilisées par la suite sont les suivantes :

* q(un, (S/(T\S))/NC)=q(un, ((S/T)\S)/NC)=λPλQ∃x[P(x)∧Q(x)] (le type de cette traduction logique justifie les catégories syntaxiques complexes attribuées aux déterminants) ;

q(chaque, (S/(T\S))/NC)=q(chaque, ((S/T)\S)/NC)=λPλQ∀x[P(x)→Q(x)] ;

où x et y sont des variables de type e, P et Q des variables de type <e,t>.

* au verbe « être » sont associées plusieurs catégories syntaxiques différentes suivant que son attribut est un nom propre, un adjectif ou un nom commun. Le cas le plus simple, dont nous nous contenterons ici, est celui illustré par le deuxième arbre de la figure 1, où l'attribut est un nom propre :

q(est,(T\S)/T)=λxλy[y=x] où x et y sont des variables de type e.

* tout autre mot m est traduit par une constante logique notée m'.

- traduction du système de Lambek-Gentzen :

* les axiomes X —> X se traduisent par la fonction identité ;

* les règles du système de Lambek se traduisent par des règles de combinaisons entre formules :

$$\frac{t \,.\, x \longrightarrow y}{t \longrightarrow \lambda x[y]} \quad (2) \qquad\qquad \frac{x \,.\, t \longrightarrow y}{t \longrightarrow \lambda x[y]} \quad (2')$$

$$\frac{t \longrightarrow x \qquad u \,.\, f(x) \,.\, v \longrightarrow z}{u \,.\, f \,.\, t \,.\, v \longrightarrow z} \quad (3) \qquad\qquad \frac{t \longrightarrow x \qquad u \,.\, f(x) \,.\, v \longrightarrow z}{u \,.\, t \,.\, f \,.\, v \longrightarrow z} \quad (3')$$

Ces règles préservent la correspondance définie par k entre les catégories grammaticales et les types logiques. Cette propriété assure, par exemple, que les phrases syntaxiquement correctes se traduiront toujours par des propositions logiques (car k(S)=t : type des valeurs de vérité).

L'arbre de démonstration de la règle R4, donné en 2.3, se traduit donc par :

$$\cfrac{\cfrac{x \longrightarrow x \qquad y=f(x) \longrightarrow y}{x \,.\, f \longrightarrow f(x)} \quad (3')}{x \longrightarrow \lambda f[f(x)]} \quad (2)$$

En traduisant les arbres de démonstration des règles Ri (1^2i^24), on retrouve ainsi les règles, notées Wi (1^2i^24), définissant les combinaisons les plus usuelles entre formules logiques ([Moortgat 88]) :

- W1 : f . x —> f(x) W' 1 : x . f—> f(x)
- W2 : f . g —> λv[f(g(v))] W' 2 : g . f—> λv[f(g(v))]
- W3 : f —> λvλw[(f(w))(v)] W' 3 : f—> λvλw[(f(w))(v)]
- W4 : x —> λf[f(x)] W' 4 : x—> λf[f(x)]

Exemples :

Les figures 3 et 4 illustrent l'application de l'isomorphisme d'arbres aux exemples de la figure 1.

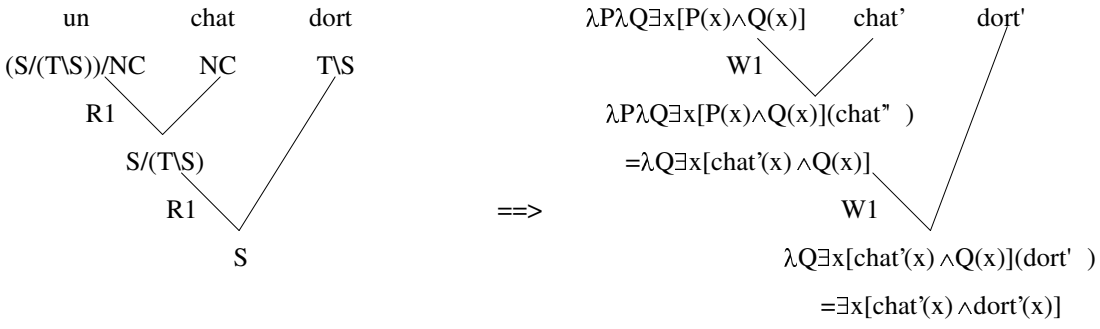


figure 3 : premier exemple de « traduction » sémantique

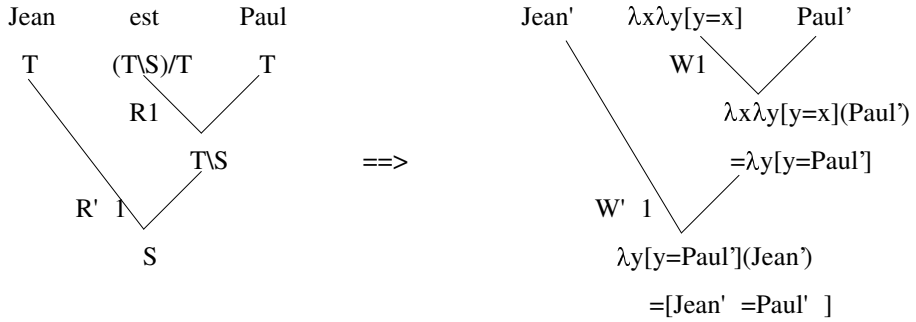


figure 4 : deuxième exemple de « traduction » sémantique

La racine de l'arbre sémantique est donc la traduction logique de la phrase de départ.

Ce mécanisme rend compte des ambiguïtés dues à la portée des quantificateurs. Les deux arbres d'analyse syntaxique de la phrase « chaque homme aime une femme » donnés en figure 2 se traduisent respectivement par deux formules logiques différentes, exprimant les deux sens possibles de la phrase :

- $\forall x[homme'(x) \rightarrow \exists y[femme'(y) \wedge aime'(y)(x)]]$;

- $\exists y[femme'(y) \wedge \forall x[homme'(x) \rightarrow aime'(y)(x)]]$.

4. Apprentissage syntaxico-sémantique

Maintenant que le cadre syntaxico-sémantique est donné, il reste à fixer les conditions de l'apprentissage.

4.1 Connaissances innées et concepts à apprendre

L'apprentissage naturel d'une langue a lieu en présence de phrases syntaxiquement correctes et d'indications sur le sens qu'elles véhiculent. De même, notre algorithme dispose en entrées de couples de données constitués d'une suite de mots et d'une formule logique.

Les connaissances innées admises dans notre modèle sont minimales : elles se limitent au système de Lambek-Gentzen et aux règles de traduction associées. Contrairement aux précédents travaux faisant intervenir de la sémantique, le sens des mots isolés (par exemple le fait que « Jean » se traduise en « Jean' » et « dort » en « dort' ») n'est pas supposé connu *a priori*. Cela revient à dire qu'on ne suppose aucune connaissance initiale sur l'association entre les mots et les concepts.

Comment, finalement, la cible est-elle représentée ? Le formalisme adopté permet, grâce aux fonctions f et q, de rattacher entièrement la syntaxe et la sémantique du langage aux mots du vocabulaire. Ainsi, la cible est une liste de triplets de la forme (v,U,w) où $v \in V$, $U \in f(v) \subseteq C'$ et $w = q(v,U) \in IL$.

Exemple :

La grammaire donnée en exemple en 2.2 et traduite en 3.2 est représentée par l'ensemble suivant :

{ (un, (S/(T\S))/NC, $\lambda P \lambda Q \exists x[P(x) \wedge Q(x)]$), (un, ((S/T)\S)/NC, $\lambda P \lambda Q \exists x[P(x) \wedge Q(x)]$),

(chaque, (S/(T\S))/NC, $\lambda P \lambda Q \forall x[P(x) \rightarrow Q(x)]$), (chaque, ((S/T)\S)/NC, $\lambda P \lambda Q \forall x[P(x) \rightarrow Q(x)]$),

(chat, NC, chat'), (homme, NC, homme'), (Jean, T, Jean'), (Marie, T, Marie'), (Paul, T, Paul'),

(dort, T\S, dort'), (aime, (T\S)/T, aime'), (est, (T\S)/T, $\lambda x \lambda y[y=x]$) }.

La figure 5 résume ces choix.

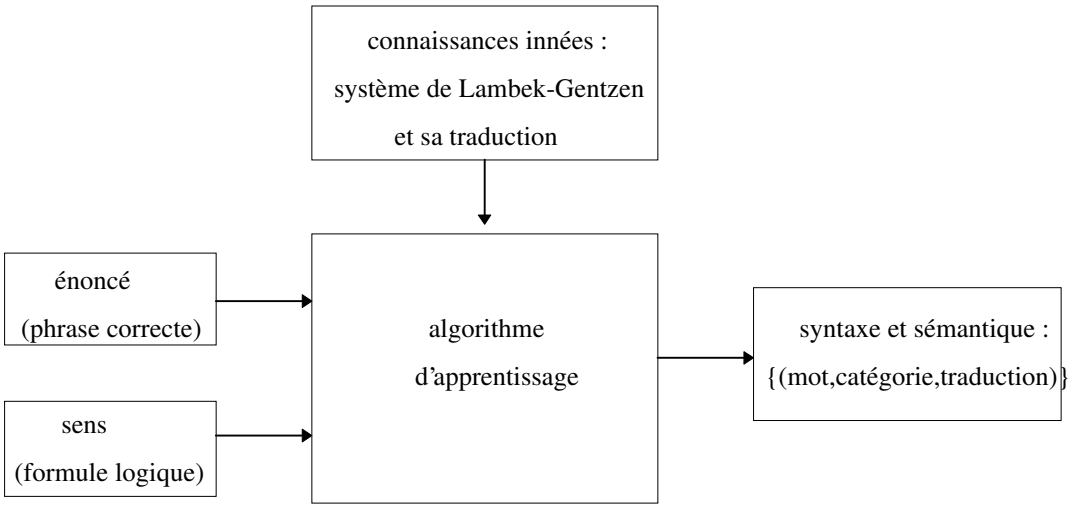


figure 5 : le cadre de l'apprentissage

4.2 L'algorithme d'apprentissage

Le stratégie d'apprentissage que nous proposons est la suivante :

- hypothèse courante := ∅ ;
- Pour chaque couple <p, l> où p est une phrase correcte et l une formule logique qui la traduit faire :
  - Pour chaque mot de p faire :
    * s'il appartient à l'hypothèse courante alors lui affecter la(les) catégorie(s) de cette hypothèse courante ;
    * sinon, lui affecter toutes les catégories possibles qui permettent la réduction à S de la suite de catégories ;
  - Pour chaque arbre d'analyse syntaxique possible de p faire :
    - Pour chaque couple (mot,catégorie) de l'arbre faire :
      * si elle existe dans l'hypothèse courante mettre q(mot,catégorie) dans l'arbre de traduction ;
      * sinon, chercher les valeurs les plus simples possibles qui permettent de retrouver l à la racine de l'arbre de traduction ;
  - Mettre à jour l'hypothèse courante.

4.3 Simulation de l'algorithme sur un exemple simple

Supposons que le premier couple de données fourni en entrée de l'algorithme (quand l'hypothèse courante est ∅) soit : <Jean dort, dort'(Jean')>.

- Les deux mots de la phrase d'entrée sont inconnus. Les deux façons possibles de leur affecter une catégorie syntaxique sont :

  * affecter S/A à « Jean » et A à « dort », A étant une catégorie quelconque de C' ;

  * affecter B à « Jean » et B\S à « dort », B étant une catégorie quelconque de C'.

- La première phase de l'algorithme produit donc deux arbres d'analyse syntaxique possibles.

* la traduction sémantique du premier arbre est donné dans la figure 6 (les entrées de l'algorithme sont présentées dans des rectangles).

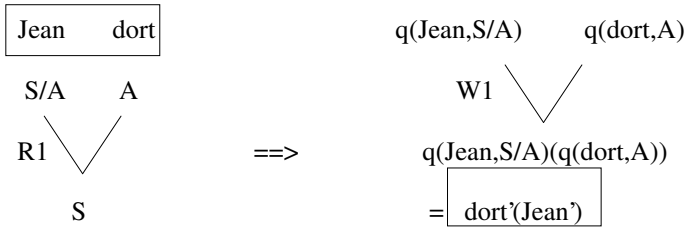


figure 6 traduction sémantique de la première hypothèse syntaxique

L'identification entre la traduction sémantique calculée et la formule logique donnée en entrée permet d'inférer : q(Jean,S/A)=dort' et q(dort,A)=Jean'.

* la traduction sémantique du deuxième arbre est donné dans la figure 7

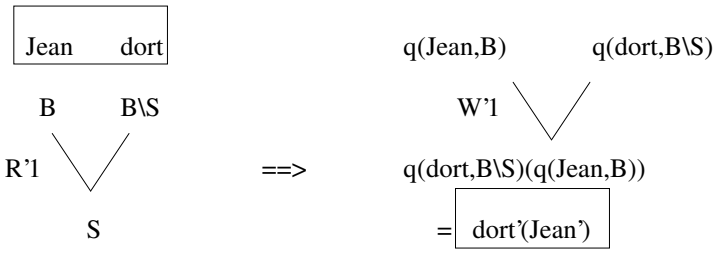


figure 7 : traduction sémantique de la deuxième hypothèse syntaxique

Cette fois, l'identification permet d'obtenir : q(dort,B\S)=dort' et q(Jean,B)=Jean'.

A ce stade, il n'y a aucune raison de préférer une solution à une autre. L'hypothèse courante crée s'écrit donc : H = {(Jean, S/A, dort'), (dort, A, Jean')} OU {(Jean, B, Jean'), (dort, B\S, dort')}

Supposons à présent que le deuxième couple de données d'entrée soit : <Marie dort, dort'(Marie')>.

- Cette fois, « dort » est présent dans l'hypothèse courante et « Marie » est inconnu :

* si « dort » est de catégorie A, alors « Marie » doit recevoir S/A ;

* si « dort » est de catégorie B\S, alors on peut affecter à « Marie » soit B soit S/(B\S). Or le séquent B —> S/(B\S) est démontrable dans le système de Lambek-Gentzen (voir règle R4). Seul B, qui est la catégorie la plus générale (permettant d'engendrer les éventuelles autres), sera donc pris en compte.

- de nouveau, nous devons traduire deux arbres syntaxiques possibles.

* le premier (similaire à celui de la figure 6) aboutit à : q(Marie,S/A)(Jean')=dort'(Marie') qui est une équation sans solution. Le premier ensemble de triplets de l'hypothèse courante est donc discrédité et abandonné. Remarquons que l'hypothèse portant sur « Jean » dans cet ensemble est également éliminée : en effet, bien qu'elle n'ait pas servie dans le traitement des nouvelles entrées, elle est solidaire de l'hypothèse sur « dort » qui a été jugée fausse. Il en aurait été de même si le deuxième couple de données d'entrée avait été : <Jean ronfle, ronfle'(Jean')>, qui aurait permis de discréditer la première hypothèse sur « Jean » et donc, en même temps, celle sur « dort ».

* le deuxième arbre (similaire à celui de la figure 7) aboutit à : dort'(q(Marie,B))=dort'(Marie') et permet donc d'inférer q(Marie,B)=Marie'.

La nouvelle hypothèse courante est donc : H'= { (Jean,B,Jean'), (Marie,B,Marie'), (dort,B\S,dort') }.

Notons que, sans faire intervenir la sémantique, nous n'aurions eu aucun moyen de départager nos deux hypothèses de départ. La seule justification pour donner au mot « dort » une catégorie fractionnaire est que son correspondant sémantique joue un rôle de fonction, de prédicat.

En annexe, est illustré comment, à partir de H', la donnée <un homme dort, ∃x[homme'(x)∧dort'(x)]> sera traitée. On obtient dans un premier temps 11 hypothèses syntaxiques pour le couple de mots inconnus « un » et « homme », réduites à 6 en éliminant les doublons et en se limitant aux catégories les plus générales. La phase sémantique permet d'éliminer 3 d'entre elles et produit finalement 4 hypothèses syntaxico-sémantiques, parmi lesquelles il est facile de voir qu'une seule résistera à la donnée d'un nouvel exemple utilisant de nouveau soit « un » en position sujet soit « homme ».

4.4 Traitement du polymorphisme

Comment notre algorithme d'apprentissage sera-t-il capable d'affecter plusieurs catégories différentes à un même mot, comme cela est nécessaire, par exemple, pour les déterminants ? L'affectation multiple - aussi appelée traditionnellement polymorphisme - nécessite d'ajouter de nouvelles informations dans l'hypothèse courante sans pour autant en éliminer aucune déjà présente.

La solution que nous préconisons est un mécanisme spécifique qui devient actif quand, en présence d'une nouvelle donnée ne contenant aucun mot inconnu, l'hypothèse courante se révèle incapable de proposer une analyse syntaxique correcte. Cette situation va se produire si, après « Jean dort », « Marie dort », « un homme dort » et « Marie aime Jean » (qui permet d'apprendre la catégorie et la traduction du mot « aime »), on propose la donnée : « Marie aime un homme ». En effet, la catégorie apprise jusqu'à présent pour « un » ne rend compte que de son usage en position sujet.

Dans ce cas, il faudrait en théorie reprendre chaque mot de la phrase l'un après l'autre, faire « comme si » il était inconnu et chercher de nouveau ses catégories possibles. La complexité algorithmique de cette solution est malheureusement considérable. Pour la limiter, il faut tenir compte du fait que tous les mots du langage naturel ne sont pas susceptibles de recevoir plusieurs catégories différentes et que, même dans ce cas, ils vérifient certaines régularités. Deux heuristiques puissantes peuvent nous y aider :

- les mots grammaticaux (qui n'ont pas de sens par eux-mêmes et sont en nombre fini dans une langue comme les prépositions, les conjonctions, les déterminants...) sont beaucoup plus polymorphes que les mots lexicaux (qui ont un sens par eux-mêmes et constituent une liste non exhaustive comme les noms, les adjectifs, les verbes...). Or les mots grammaticaux, dans notre modèle, sont reconnaissables au fait que leur traduction sémantique contient des lambdas-expressions, alors que les mots lexicaux se traduisent presque toujours par une constante logique. Ils sont donc remis en cause en priorité.

- même quand un mot reçoit plusieurs catégories syntaxiques différentes, il y a de très fortes chances pour les traductions sémantiques correspondantes soient les mêmes. Cela est vrai pour les déterminants, mais aussi pour les mots lexicaux ambigus comme « ferme », « soupe » ou « garde », qui peuvent être des noms ou des formes verbales.

En tenant compte de ces heuristiques, la donnée « Marie aime un homme » se traite simplement. Une analyse plus fine sera nécessaire pour déterminer les éventuels cas pathologiques non pris en compte.

5. Evaluation et conclusion

Le modèle que nous proposons a quelques faiblesses. Ainsi, la complexité combinatoire de la recherche d'hypothèses syntaxiques est clairement exponentielle. En fait, l'algorithme est moins sensible à la complexité intrinsèque des données qu'à leur complexité *relativement à l'hypothèse courante*, qui peut se mesurer par le nombre de mots inconnus dans une nouvelle donnée. Nous envisageons donc de poser une borne *a priori* sur le nombre maximum de mots nouveaux acceptables dans une nouvelle phrase, quitte à mettre en attente la donnée correspondante jusqu'à ce que l'hypothèse courante ait suffisamment évolué pour la traiter. L'ordre de présentation des exemples est donc dans ce cas un paramètre fondamental pour assurer l'apprentissage : les exemples simples doivent précéder les exemples compliqués et les mots nouveaux doivent être introduits progressivement. Cette stratégie est raisonnable car les enfants, aussi, ont certainement plus de mal à comprendre une phrase quand leur connaissance de la langue est limitée.

Mais notre travail est aussi l'héritier de plusieurs approches qu'il renouvelle avantageusement.

Tout d'abord, des recherches récentes sur l'apprenabilité des grammaires catégorielles ([Buszkowski & Penn 90], [Adriaans 92], [Kanazawa 96], [Stabler 96]) illustrent leur bonne adaptation au processus d'apprentissage. Néanmoins, tous ces travaux se limitent aux grammaires de type AB, moins bien adaptées que les grammaires L à la formalisation des phénomènes linguistiques. De plus, tous (sauf [Adriaans 92]) fournissent en entrée de leur algorithme des phrases parenthèsées, dont on connaît déjà la structure syntaxique, ce qui n'est pas très satisfaisant. Adriaans, lui, a proposé un algorithme d'apprentissage prenant en compte aussi bien la syntaxe que la sémantique, mais en traitant ces deux niveaux successivement et indépendamment sans les articuler. Notre travail suggère au contraire que des informations sémantiques peuvent aider l'apprentissage de la syntaxe.

D'un autre côté, l'apprentissage syntaxico-sémantique a inspiré des études originales revendiquant une meilleure pertinence cognitive ([Hamburger & Wexler 75], [Anderson 77], [Langley 82], [Hill 83]). Pourtant, les représentations sémantiques utilisées dans ces articles et les liens qu'elles entretiennent avec la syntaxe ont un caractère *ad hoc* gênant ([Pinker 79]). L'usage de la logique, langage formel puissant et *a priori* indépendant de toute langue naturelle est incontestablement plus satisfaisant. Notre modèle, en fait, oblige à considérer que la logique est le « langage de la pensée » puisque nous supposons que les situations du monde perçues par l'apprenant sont traduites automatiquement en formules logiques *avant* tout traitement linguistique. Cela implique que pour apprendre la syntaxe d'une langue, il faut avoir précédemment acquis une structuration conceptuelle du monde. Remarquons d'ailleurs que seule la syntaxe de la logique est utilisée : nous n'utilisons pas les valeurs de vérité des entrées.

Ainsi, grâce à l'isomorphisme d'arbres, la structure fonctionnelle des formules logiques donne des indications sur la structure fonctionnelle des catégories syntaxiques, jouant un rôle comparable au parenthésage de la première famille de travaux. La nature des entrées de notre algorithme remédie donc aux principales critiques que l'on pouvait formuler à l'encontre des recherches précédentes.

Il semble très crédible de considérer que l'apprentissage d'une langue n'est possible que grâce à la reconnaissance de redondances à la fois syntaxiques et sémantiques : mis en présence de deux individus différents dans la même situation, l'apprenant qui entend respectivement « Jean dort. » et « Marie dort. » aura sans doute naturellement tendance à associer ce qu'il y a de commun entre les deux situations perçues avec ce qu'il y a de commun entre les deux énoncés. C'est aussi ce que fait notre algorithme. Plutôt que de chercher à reconstituer une syntaxe formelle, il « apprend à comprendre », c'est-à-dire à associer automatiquement à une phrase son interprétation sémantique (la syntaxe n'étant qu'un intermédiaire de cette traduction sémantique). Sa stratégie générale, son but et les connaissances innées limitées qu'il utilise militent donc également en faveur de sa pertinence.

Evidemment, le travail présenté ici est loin d'être achevé. L'algorithme est en cours d'implémentation et ses performances doivent encore être testées rigoureusement. Son intégration dans les critères d'apprenabilité de Gold ou de Valiant reste pour l'instant problématique. Son intérêt linguistique, théorique et cognitif semble néanmoins prometteur.

6. Références

[Adriaans 92] Adriaans P. W., *Language Learning from a Categorial Perspective*, Ph.D. dissertation, University of Amsterdam, 1992.

[Anderson 77] Anderson J. R., "Induction of Augmented Transition Networks", *Cognitive Science* 1, p125-157, 1977.

[Bar Hillel 53] Bar Hillel Y., "A quasi-arithmetical notation for syntactic description", *Language* 29, p47-58, 1953.

[Bar Hillel & *alii* 60] Bar Hillel Y., Gaifman C., Shamir E., "On categorial and phrase structure grammars", *Bulletin of the Research Council of Israel* 9F, p1-16, 1960.

[Buszkowski & Penn 90] Buszkowski W., Penn G., "Categorial grammars determined from linguistic data by unification", *Studia Logica* 49, p431-454, 1990.

[Chambreuil 89] Chambreuil M., *Grammaire de Montague; langage, traduction, interprétation*, Adossa, Clermont-Ferrand, 1989.

[Chomsky 65] Chomsky N., *Aspects of the Theory of Syntax*, Cambridge, MIT Press.

[Chomsky 68] Chomsky N., *Language and Mind*, Brace & World, 1968.

[Denis & Gilleron 97] Denis F., Gilleron R., "PAC learning under helpful distributions", actes du 8ième ACM Workshop on Computational Learning Theory, 1997.

[Dowty 81] Dowty D.R., Wall R.E., Peters S., *Introduction to Montague Semantics*, Reidel, Dordrecht, 1989.

[Finkel & Tellier 96] Finkel A., Tellier I. : "A polynomial algorithm for the membership problem with categorial grammars", *Theoretical Computer Science* 164, p207-221, 1996.

[Gold 67] Gold E.M., "Language Identification in the Limit", *Information and Control* 10, P447-474, 1967.

[Hamburger & Wexler 75] Hamburger H., Wexler K., "A mathematical Theory of Learning Transformational Grammar", *Journal of Mathematical Psychology* 12, p137-177, 1975.

[Hill 83] Hill J.C., "A computational model of language acquisition in the two-year-old", *Cognition and Brain Theory* 6(3), p287-317, 1983.

[Kanazawa 96] Kanazawa M., "Identification in the Limit of Categorial Grammars", *Journal of Logic, Language & Information*, vol 5, n°2, p115-155, 1996.

[Lambek 58] Lambek J., "The mathematics of Sentence Structure", *American Mathematical Monthly*, n°65, p154-170, 1958.

[Langley 82] Langley P., "Language acquisition through error discovery", *Cognition and Brain Theory* 5, p211-255, 1982.

[Li & Vitanyi 93] Li M., Vitanyi P., *An introduction to Kolmogorov complexity and its applications*, Springer Verlag, 1993. [Montague 74] : R. Montague, *Formal Philosophy; Selected papers of Richard Montague*, Yale University Press, New Haven, 1974.

[Moortgat 88] Moortgat M., *Categorial investigations, logical and linguistic aspects of the Lambek Calculus*, Foris, Dordrecht, 1988.

[Oehrle & *alii* 88] Oehrle R.T., Bach E., Wheeler D. (Eds.), *Categorial Grammars and Natural Language Structure*, Reidel, Dordrecht, 1988.

[Pentus 92] Pentus M., "Lambek grammars are context-free", in : 8th Annual IEEE Symposium on Logic in Computer Science, Montreal, Canada, p429-433, 1992.

[Pinker 79] Pinker S., "Formal models of language learning", *Cognition* 7, p217-283, 1979.

[Rozier & Tellier 92] Rozier O., Tellier I., "Système de Lambek étendu pour la traduction logique de phrases en langage naturel", actes des premières rencontre nationale des jeunes chercheurs en Intelligence Artificielle, p308-324, Rennes.

[Sakakibara 92] Sakakibara Y., "Efficient learning of context-free grammars from positive structural examples", *Information & Computation* 97, p23-60, 1992.

[Schieber 85] Schieber S., "Evidence against the context-freeness of natural languages", *Linguistics and Philosophy* 8, p333-343, 1995.

[Shinohara 90] Shinohara T., "Inductive inference of monotonic formal systems from positive data", p339-351 in : *Algorithmic Learning Theory*, S. Arikara, S. Goto, S. Ohsuga & T. Yokomori (eds), Tokyo : Ohmsha and New York and Berlin : Springer.

[Stabler 96] Stabler E., "Acquiring and parsing languages with movement", Linguistics 185/209 Lecture notes, http://128/97.8.34/utrecht, 1996.

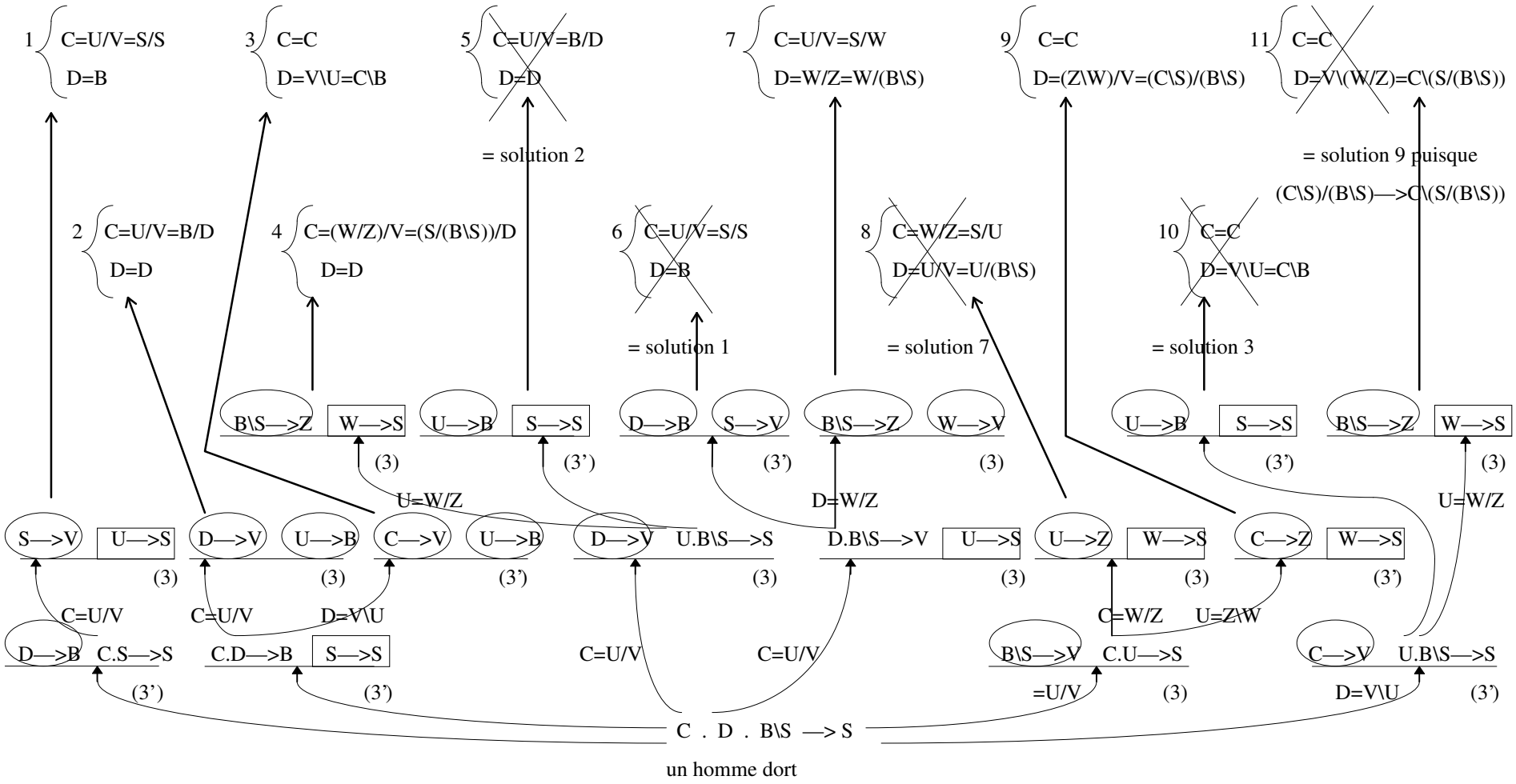
[Valiant 84] Valiant L.G., "A theory of the learnable", *Communication of the ACM*, p1134-1142, 1984.

[Wexler & Culicover 80] Wexler K., Culicover P., *Formal Principles of Language Acquisition*, Cambridge, MIT Press.

[Zielonka 81] Zielonka W., "Axiomatizability of Ajdukiewicz-Lambek calculus by means of cancellations schemes", *Zeischrift für Mathematsche Logik und grunlagen der Mathematik* 27, p215-224, 1981.

ANNEXE

- arbre de recherche des hypothèses syntaxiques pour la phrase « un homme dort ».



Jusqu'à présent, nous avions illustré le mécanisme d'apprentissage en utilisant les règles R1 et R'1, qui permettent de mieux visualiser les arbres produits. Pour cette nouvelle donnée, nous avons plutôt appliqué les règles du calcul de Lambek en chaînage arrière, en arrêtant l'inférence dès que les séquents obtenus sont soit des axiomes (ils sont alors entourés d'un rectangle) soit des séquents unifiables avec des axiomes (ils sont alors présentés entourés d'une ellipse). La différence entre les deux cas réside dans la nature du conséquent du séquent (c'est-à-dire la catégorie à droite de la flèche) : si c'est une catégorie fixée connue (typiquement la catégorie S qui est la seule connue au départ et est indécomposable), la seule solution est que l'antécédent du séquent soit identique ou unifiable à cette catégorie (premier cas) ; alors que si le conséquent est une variable inconnue, rien n'empêcherait de la décomposer afin de lui appliquer la règle (2) ou (2') en chaînage arrière mais on n'essaie cette solution que lorsqu'aucune autre n'est possible.

• traduction sémantique : notons α=q(un, C) et β=q(homme,D)

* la traduction de la solution 1 produit l'équation : α(dort'(β))=$\exists$x[homme'(x) $\wedge$dort'(x)].

  L'identification des deux formules impose de poser β=x. Mais il est interdit d'introduire des variables libres dans les traductions logiques des mots et cette solution est donc abandonnée.

* la traduction de la solution 2 produit l'équation : dort'(α(β))=$\exists$x[homme'(x) $\wedge$dort'(x)].

  Cette équation n'admet aucune solution et la solution est abandonnée.

* la traduction de la solution 3 produit l'équation : dort'(β(α))=$\exists$x[homme'(x) $\wedge$dort'(x)].

  Comme dans le cas précédent, la solution est abandonnée.

* la traduction de la solution 4 produit l'équation : (α(β))(dort')= $\exists$x[homme'(x) $\wedge$dort'(x)].

  L'identification de ces deux formules n'est possible qu'à condition de procéder à des lambdas-abstractions sur le second membre (on ne s'autorise à en faire que sur les *constantes* logiques) :

  $\exists$x[homme'(x) $\wedge$dort'(x)]= λP$\exists$x[homme'(x) $\wedge$P(x)](dort') donc α(β)=λP$\exists$x[homme'(x) $\wedge$P(x)]

  et λP$\exists$x[homme'(x) $\wedge$P(x)]=λQλP$\exists$x[Q(x)$\wedge$P(x)](homme')

  donc la solution la plus simple est : α=λQλP$\exists$x[Q(x)$\wedge$P(x)] et β=homme'.

* la traduction de la solution 7 produit l'équation : α(β(dort'))= $\exists$x[homme'(x) $\wedge$dort'(x)].

  avec $\exists$x[homme'(x) $\wedge$dort'(x)]= λP$\exists$x[homme'(x) $\wedge$P(x)](dort') on obtient deux solutions :

  - α=λx.x et β=λP$\exists$x[homme'(x) $\wedge$P(x)]

  - α=λP$\exists$x[homme'(x) $\wedge$P(x)] et β= λx.x

  (les solutions faisant appel à la fonction constante λx.x ne sont prises en compte que dans les cas où il n'est pas possible de faire autrement ; c'est pourquoi elles n'ont pas été proposées plus tôt)

* la traduction de la solution 9 produit l'équation : (β(dort'))(α)=$\exists$x[homme'(x) $\wedge$dort'(x)].

  après abstractions, la solution la plus simple est : α=homme' et β=λPλQ$\exists$x[Q(x)$\wedge$P(x)].

Finalement, les nouvelles hypothèses apportées par cette donnée sont les suivantes :

{ (un, (S/(B\S))/D, λQλP$\exists$x[Q(x)$\wedge$P(x)]), (homme, D, homme') }

OR { (un, S/W, λx.x), (homme, W/(B\S), λP$\exists$x[homme'(x) $\wedge$P(x)]) }

OR { (un, S/W, λP$\exists$x[homme'(x) $\wedge$P(x)])), (homme, W/(B\S), λx.x) }

OR { (un, C, homme'), (homme, (C\S)/(B\S), λPλQ$\exists$x[Q(x)$\wedge$P(x)]) }.

Il est évident que seul le premier de ces sous-ensembles d'hypothèses sera compatible avec de nouvelles données d'entrée contenant un de ces mots.