3.3.1 Premier fragment

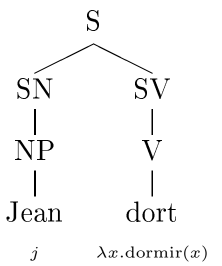
Nous construisons de façon progressive, avec les outils dont nous disposons, un algorithme de calcul de représentations sémantiques (en logique du premier ordre). Cela nous conduira, pour chaque phénomène considéré, à :

1. définir une analyse syntaxique (une portion de grammaire)
2. définir un mode de composition des représentations (application fonctionnelle)
3. définir une représentation pour chaque lexème participant à la construction

3.3.1.1 Phrases simples

(5) a. Jean dort
    b. dormir(j)

Si l'on admet la représentation sémantique donnée en (5b), et l'arbre syntaxique de la figure 3.2, on est conduit à proposer la grammaire et les règles de calcul données dans la même figure[6].



S	$\rightarrow$	SN	SV
0	$\leftarrow$	(2)	1
SN	$\rightarrow$	NP	
0	$\leftarrow$	1	
SV	$\rightarrow$	V	
0	$\leftarrow$	1	

Fig. 3.2 – Arbre syntaxique et grammaire pour (5a)

Il est inutile de détailler plus le calcul : les valeurs sémantiques sont « propagées » le long de l'arbre jusqu'à la combinaison qui aboutit à $(\lambda x.\text{dormir}(x))j$, ce qui donne bien, par β-réduction, dormir(j).

3.3.1.2 Quantificateurs (généralisés)

Considérons maintenant une phrase à peine plus compliquée, mais qui fait intervenir, en position sujet, un syntagme nominal quantifié.

(6) a. Un homme marche
    b. $\exists x\ (\text{homme}(x) \wedge \text{marche}(x))$

[6] Une notation classique pour représenter la composition sémantique associée à une règle syntaxique utiliserait la représentation usuelle de la dénotation : ainsi, pour la première règle de la figure 3.2, on aurait : $[\![S]\!] \leftarrow ([\![SV]\!])\ [\![SN]\!]$, pour expliciter le fait que la valeur sémantique associée au nœud S est le produit de l'application fonctionnelle de la valeur sémantique asociée au nœud SV sur la valeur sémantique associée au nœud SN.

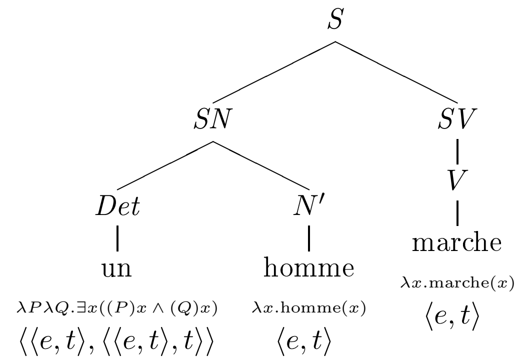
A cette notation assez lourde, nous préférons dans ce texte une notation chiffrée qui permet de mieux voir le sens de composition des éléments, ainsi que les régularités dans la composition.

Les contributions respectives des différents éléments sont les suivantes : *homme* : le prédicat homme ; *marche* : le prédicat marche ; enfin le déterminant (ou quantificateur) *un* : la quantification existentielle **et** la structure générale de la formule.

D'où la proposition pour le lexique :

	homme	$\lambda x.\text{homme}(x)$
	marche	$\lambda x.\text{marche}(x)$
	un	$\lambda P \lambda Q.\exists x((P)x \wedge (Q)x)$

Si on adopte ce lexique, inspiré de l'idée des quantificateurs généralisés, alors il est nécessaire de réviser les règles de construction : en effet, c'est la valeur sémantique du déterminant qui devient le foncteur (à deux places), s'appliquant d'abord au N', puis au VP. On peut indiquer tout cela dans la figure suivante.



S	$\rightarrow$	SN	SV
0	$\leftarrow$	(1)	2
SN	$\rightarrow$	Det	N'
0	$\leftarrow$	(1)	2
SV	$\rightarrow$	V	
0	$\leftarrow$	1	

FIG. 3.3 – Arbre syntaxique et grammaire pour (6a)

Ce fragment de grammaire est incompatible avec le précédent : la règle de composition sémantique associée à $S \rightarrow SN\ SV$ n'est pas la même dans les deux cas.

La solution que nous allons adopter s'appuie sur la théorie des quantificateurs généralisés [Barwise and Cooper, 1981, Keenan and Stavi, 1986, Westerståhl, 1989] : elle consiste à « monter » le type du SN lorsqu'il se dérive en un nom propre, pour qu'il soit du même type que *un homme* dans la figure précédente. On propose que $[\![[_{\text{SN}}\ \text{Jean}]]\!] = \lambda P.(P)j$: la dénotation d'un nom propre est l'ensemble des propriétés possédées par le référent.

Cette proposition permet d'unifier les règles de composition, on peut récapituler le fragment obtenu jusqu'à présent :

S	$\rightarrow$	SN	SV
0	$\leftarrow$	(1)	2
SN	$\rightarrow$	NP	
0	$\leftarrow$	1	
SN	$\rightarrow$	Det	N'
0	$\leftarrow$	(1)	2
N'	$\rightarrow$	N	
0	$\leftarrow$	(1)	2
SV	$\rightarrow$	V_i	
0	$\leftarrow$	1	

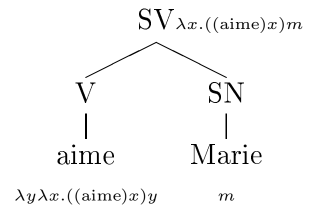
Det	$\rightarrow$	un
0	$\leftarrow$	$\lambda P \lambda Q.\exists x((P)x \wedge (Q)x)$
N	$\rightarrow$	homme
0	$\leftarrow$	$\lambda x.\text{homme}(x)$
V_i	$\rightarrow$	marche
0	$\leftarrow$	$\lambda x.\text{marche}(x)$
NP	$\rightarrow$	Jean
0	$\leftarrow$	$\lambda P.(P)j$ [7]

[7] On pourrait aussi imaginer que la contribution de *Jean* se réduit à une constante (ici j), et que c'est lorsque le NP devient un SN que s'opère la « montée de type » : la règle sémantique associée à la règle '$SN \rightarrow NP$' serait alors : $0 \leftarrow (\lambda x \lambda P.(P)x)1$.

3.3.1.3 Verbe transitif

(7) a. Jean aime Marie
b. $((\text{aime})j)m$

Considérons maintenant le cas des verbes transitifs. Une généralisation directe à partir des cas de verbes intransitifs pourrait conduire à proposer $\lambda x \lambda y.((\text{aime})x)y$. Cette proposition serait compatible avec le premier cas simple considéré, par exemple en présence d'un nom propre (et à condition d'inverser les arguments, l'objet étant combiné avant le sujet), comme le montre la figure suivante :



Malheureusement, cette option n'est plus possible avec les SN quantifiés, qui sont de type $\langle e, \langle e, t\rangle\rangle$ et non de type e. Si l'on veut que le verbe reste le foncteur, opérant sur le SN objet, il faut faire la proposition suivante : aime : $\lambda P \lambda x.(P)\lambda y.((\text{aime})x)y$. Cette expression est de type $\langle\langle e, \langle e, t\rangle\rangle, \langle e, t\rangle\rangle$.

Naturellement, la nouvelle règle sera

$$\begin{array}{cccc} SV & \rightarrow & V & SN \\ 0 & \leftarrow & (1) & 2 \end{array}$$

On peut vérifier sur l'exemple courant qu'au bout de 3 β-réductions, on obtient bien le résultat escompté :

$$\begin{array}{l} \Big(\lambda P \lambda x.(P)\lambda y.((\text{aime})x)y\Big)\, \lambda P.(P)m \\ \longrightarrow_\beta \quad \lambda x.\, \Big(\lambda P.(P)m\Big)\, \lambda y.((\text{aime})x)y \\ \longrightarrow_\beta \quad \lambda x.(\lambda y((aime)x)y)m \\ \longrightarrow_\beta \quad \lambda x.((aime)x)m \end{array}$$

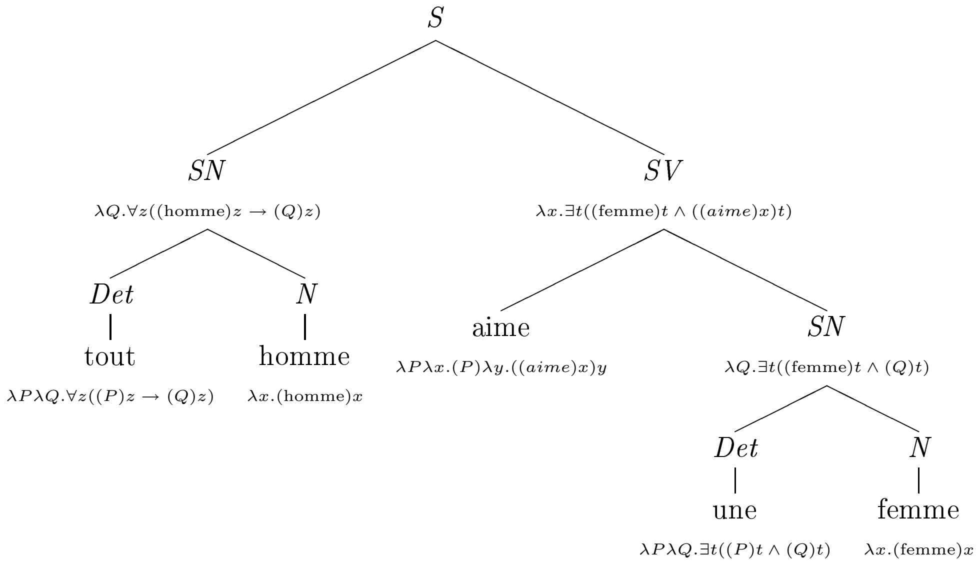
On peut aussi vérifier facilement que cette proposition est compatible avec la présence de plusieurs SN quantifiés. Considérons (8a). Une représentation classique (de l'une des interprétations) de cette phrase est donnée sous (8b). Nous montrons à la figure 3.4 que les principes précédents nous permettent d'obtenir la forme équivalente donnée sous (8c).

(8) a. Tout homme aime une femme
b. $\forall x \exists y((\text{homme}(x) \wedge \text{femme}(y)) \rightarrow \text{aime}(x, y))$
c. $\forall z((\text{homme})z \rightarrow \exists t((\text{femme})t \wedge ((aime)z)t))$

3.3.1.4 Négation

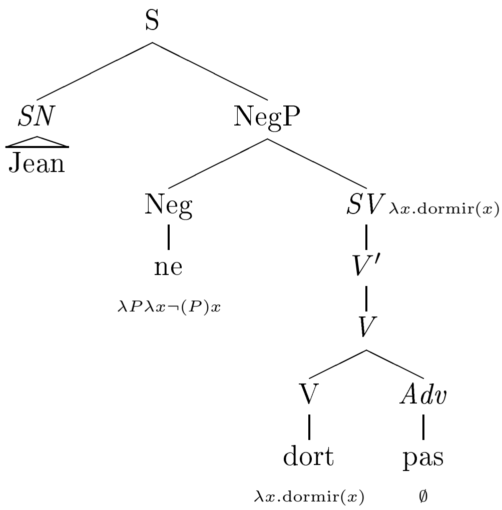
Pour le traitement de la négation en français, il est nécessaire de faire des choix syntaxiques, à cause de la forme discontinue de la négation.

FIG. 3.4 – Composition pour (8a)



(9) a. Jean ne dort pas
b. $\neg(\text{dort}')j$
c. *ne* : $\lambda P\lambda x.\neg(P)x$

Une des hypothèses possibles consiste à considérer que *pas* est un adverbe adjoint à V, sans contribution sémantique. Alors on peut charger le mot *ne* d'introduire la négation dans la formule finale.



Conséquence Le fragment ainsi défini permet de vérifier que le problème connu de « montée de la négation » (*neg-raising*) est un problème de compositionalité : la contruction d'une représentation pour la phrase (10a) donne (10b) au lieu de (10c).

(10) a. Tous les enfants ne dorment pas
b. $\forall x(\text{enfant}(x) \rightarrow \neg\text{dormir}(x))$
c. $\neg\forall x(\text{enfant}(x) \rightarrow \text{dormir}(x))$