Une approche mathématique de la notion de structure syntaxique : raisonner en termes de connexions plutôt que d'unités

Anonyme

xxx

xxx

RÉSUMÉ. *Cet article propose d'évaluer l'équivalence des structures syntaxiques (arbres de constituants, arbres de dépendance, etc.) en fonction des combinaisons d'unités qu'elles définissent et des connexions que cela induit. La notion de connexion est définie comme une classe d'équivalence de combinaisons, ce qui nous permet d'introduire une notion de structure syntaxique qui s'abstrait en partie de la notion d'unité syntaxique. Les conséquences que cela a sur la conception de la structure syntaxique sont présentées.*

ABSTRACT. ***Syntactic combinations and equivalency between syntactic structures.*** *The aim of this paper is to evaluate syntactic structures (constituency trees, dependency trees. etc.) according to the combinations of units that they define and the connections that are induced. The notion of connection is defined as an equivalence class of combinations, which allows us to introduce a notion of syntactic structure that cut itself off from the notion of syntactic unit. Consequences of that on the conception of the syntactic structure are investigated*

MOTS-CLÉS : *connexion syntaxique, dépendance, constituance, relation d'équivalence.*

KEYWORDS: *syntactic connection, dependency, constituency, equivalence relation*.

Préambule. Cette proposition d'article n'est pas à proprement parler du TAL. Il s'agit de linguistique mathématique. Néanmoins nous pensons que le public auquel cet article s'adresse est en partie celui de la revue TAL et que les résultats théoriques que nous présentons ici peuvent avoir des incidences sur la modélisation des langues et leur implémentation.

1. Introduction

L'objectif de cet article est de s'interroger sur la nature formelle de la structure syntaxique. Il ne s'agit pas de discuter les critères linguistiques qui permettent de définir la structure syntaxique ou de montrer quelle structure permet de modéliser au mieux tel ou tel phénomène syntaxique en fonction des critères retenus. Nous nous intéressons ici à la nature mathématique de la structure en fonction des objets

linguistiques considérés par la théorie. Nous pensons que la conception que les linguistes ont de la structure syntaxique et des objets qu'encode cette structure (constituants ou dépendances par exemple) est directement contrainte par les objets formels que nous utilisons pour les représenter. Les modes de représentation que nous utilisons, comme les arbres de constituants ou les arbres de dépendance, limitent notre conception de la structure syntaxique et peuvent induire des conceptions erronées de la nature de cette structure. Une meilleure compréhension de la structure syntaxique passe, à notre avis, par l'introduction de structures mathématiques plus riches que les traditionnels arbres, les arbres en question n'étant que des vues particulières et réduites de la véritable structure.

Nous souhaitons en particulier mettre au centre de la discussion la notion de *connexion*, qui est, à notre avis, une notion naturelle, mais formellement complexe, puisqu'une connexion est un ensemble de combinaisons équivalentes, ensemble que l'on peut difficilement représenter par un diagramme de type graphe. Les diagrammes habituellement utilisés en linguistique formelle, et notamment les arbres de constituants et de dépendance, ne sont de ce point de vue que des représentants particuliers des structures abstraites que nous allons introduire. De telles vues, si elles restent utiles pour le linguiste, doivent être appréhendées comme vues simplifiées de la nature réelle de la structure. Nous pensons que prendre en compte la structure dans toute sa complexité formelle permet de mieux comprendre pourquoi les linguistes proposent des vues si variées de la structure sans arriver à un accord. Nous pensons également que prendre en compte toute la richesse de la structure permettrait d'envisager des traitements automatiques plus élégants, mais cette dernière question ne sera qu'effleurée dans cet article qui se limite à présenter des objets mathématiques pour définir la structure.

Dans la section 2, nous présenterons quelques exemples de la diversité des diagrammes syntaxiques considérés dans la littérature. Ce n'est pas la diversité des analyses linguistiques qui nous intéresse ici, mais bien la diversité des objets mathématiques considérés. Dans la section 3, nous montrons comment passer d'une évaluation de la structure en termes d'unités à une évaluation en termes de combinaison, puis de connexion. La section 4 introduit la notion formelle de *combinaison* et montre l'interprétation de différentes structures syntaxiques en termes d'*ensembles de combinaisons*. La section 5 rappelle les notions de relations d'équivalence et de relation d'ordre dont nous ferons grand usage dans la suite de l'article. Dans la section 6, nous introduisons une relation de *compatibilité* entre combinaisons, qui nous permet de définir les notions formelles de *connexion* et d'*ensemble de connexion*. La section 7 montre comment associer à un arbre de dépendance un ensemble de combinaisons plus riche, puis étend les notions d'ensemble de combinaisons minimal et maximal à tout type de structure. Nous étudions dans la section 8 différentes relations de compatibilité, d'ordre ou d'équivalence entre les ensembles de combinaisons, ce qui nous permet de donner une cartographie des différents ensembles de connexions envisageables. La section 9

revient sur la question des unités à travers la notion de *granularité*. Nous concluons à la section 10.

2. Diversité mathématique des structures syntaxiques

Rappelons que nous appelons *arbre* un graphe orienté dont les arêtes, appelées également branches, vont d'un nœud appelé le *gouverneur* à un nœud appelé le *dépendant*. Pour être un arbre, ce graphe doit être *connexe* et *acyclique* et chaque nœud doit posséder un gouverneur à l'exception d'un nœud qu'on appelle la *racine* de l'arbre. Dans cet article, nous ne nous intéresserons pas à l'*ordre linéaire* et nos arbres n'incluent pas d'ordre additionnel sur les nœuds.

Commençons par les *arbres de constituants*, comme l'arbre T0 de la figure 1a qui représente une analyse en constituants immédiats de la phrase (1). Cet arbre indique que *le* et *chat* forme ensemble une unité *le chat* ou de manière équivalent que *le chat* a pour *constituants immédiats le* et *chat*. De même, *vu* et *Zoé* forment ensemble une unité *vu* Zoé qui se combine avec a pour forme une unité *a vu Zoé* qui à son tour se combine avec *le chat* pour donner la phrase entière. Il est intéressant de noter que dans ses premiers travaux, Noam Chomsky ne représente pas l'analyse en constituants immédiats par un arbre, mais par une structure un peu différente, comme on peut le voir sur le diagramme de droite la figure 1 extrait de (Chomsky 1957). Plus précisément, la règle S → NP + VP est représentée par un lien entre NP et PP indiquant que ces deux constituants se combinent entre eux, puis par un petit lien entre ce premier lien et S indiquant que le résultat de la combinaison donne S. Une structure où une arête a pour sommet une autre arête est généralement appelée un polygraphe (Burroni 1993, Bonfante & Guiraud 2008). La représentation en arbre de constituants n'apparaît que dans les travaux suivants de Chomsky.[1]

(1) *Le chat a vu Zoé.*

[1] Dans cet article, nous ne nous intéresserons qu'aux arbres de constituants binaires. D'une certaine façon, le formalisme des arbres de constituants est utilisé de deux façons assez différentes dans la littérature. Les arbres de constituants binaires cherchent à représenter des

T0 Chomsky (1957)

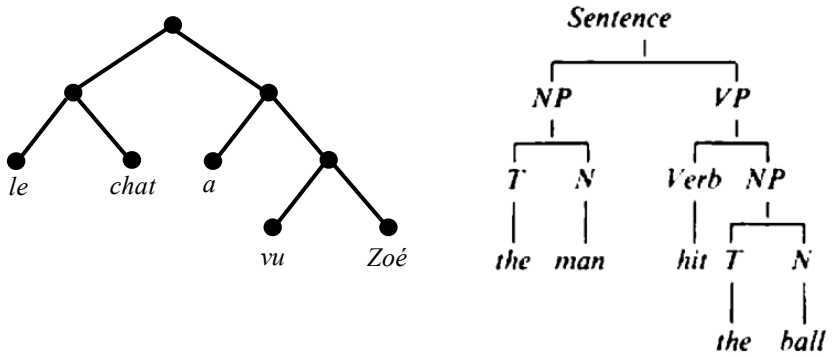

Figure 1. Arbre de constituants binaire et diagramme de Chomsky (1957)

Passons aux *arbres de dépendance*. La figure 2 propose deux arbres de dépendance pour la phrase (1) : D1 est un arbre selon le schéma Universal Dependencies (Nivre et al. 2016), où les liens entre les mots pleins sont privilégiés, D2 un arbre à la Hudson (1984, 2007), où les mots fonctionnels sont traités comme des têtes. Si les analyses linguistiques sont différentes et obéissent à des critères différents, le formalisme est exactement le même, chaque branche représentant la combinaison entre deux mots.

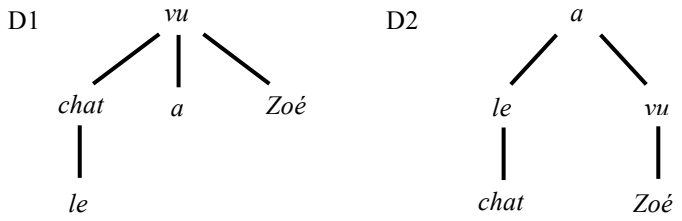

Figure 2. Arbres de dépendance à la UD et à la Hudson

On peut en fait donner d'autres interprétations des branches d'un arbre de dépendance et voir la combinaison non pas entre deux mots, mais entre un mot et une unité qui serait la projection du nœud dépendant. La projection d'un nœud x est l'unité formée des nœuds dominés par x, x inclus. Ainsi la dépendance entre *a* et *vu* de D2 peut être interprétée comme une combinaison entre *a* et la projection de *vu*, c'est-à-dire *vu Zoé*. Cette double interprétation de l'analyse en dépendance a déjà été mise en évidence par Nicolas Beauzée (1765) dans l'article *Régime* de l'*Encyclopédie* de Diderot et d'Alembert. Elle a été formalisée par Gladkij (1968) qui propose une structure de type arbre à bulles (Kahane 1997), où les branches peuvent liées des bulles qui possèdent elle même une analyse interne. La structure

S2 de la figure 3 est un « arbre » à la Beauzée-Gladkij pour la phrase (1) suivant la même analyse qu'en D2.

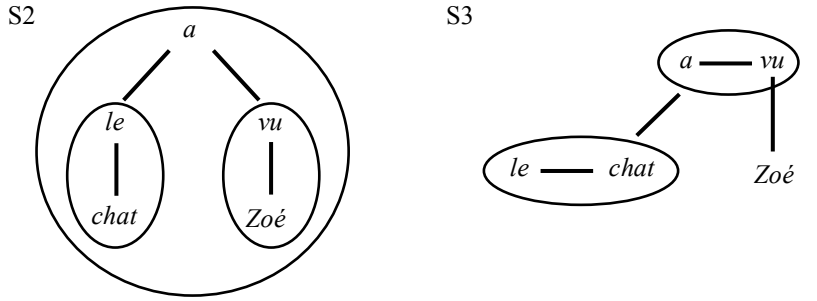

Figure 3. Un « arbre » à la Beauzée-Gladkij et un stemma à la Tesnière

Considérons pour terminer les structures introduites par Lucien Tesnière (1959). Si Tesnière est généralement crédité d'avoir marqué un pas décisif dans le développement de la syntaxe de dépendance, force est de constater que ses stemmas ne sont généralement pas des arbres. L'introduction des nucléus est particulièrement intéressante. Tesnière considère que les mots grammaticaux comme les auxiliaires ou les articles forment avec les mots pleins des nucléus et que c'est entre les nucléus que les connexions ont lieu. La structure S3 de la figure 3 donne un stemma à la Tesnière pour la phrase (1). Dans ce stemma, *a* et *vu* se combinent entre eux et forment le nucléus *a vu* dont dépend le nucléus *le chat*, qui résulte de la combinaison de *le* et *chat*. Par contre, *Zoé* dépend de *vu* seul et non du nucléus *a vu*.[2]

3. Des unités aux connexions

La question des unités syntaxiques est devenue centrale en linguistique à partir de l'émergence de l'analyse en constituants immédiats (Bloomfield 1933). Définir la structure consiste alors à déterminer les constituants majeurs de la phrase, puis les constituants immédiats ce chaque constituant et ainsi de suite. L'amorce même du processus nécessite de déterminer l'unité maximale de la syntaxe, la phrase. On retrouve cette contrainte jusqu'à aujourd'hui en TAL, puisque la quasi-totalité des analyseurs syntaxiques automatiques suppose une segmentation préalable du texte à analyser en phrases.

[2] Selon les stemmas, Tesnière utilise, pour représenter les nucléus (translatifs), une notation en T ou une bulle comme ici. Dans les notations en T, l'élément lexical peut avoir ses propres dépendants. C'est l'analyse que nous avons retenu ici en faisant dépendre *Zoé* du verbe lexical *vu* et non de la totalité du nucléus. Nous avons par ailleurs ajouté un lien entre les deux éléments du nucléus pour bien indiquer qu'ils sont combinés. Voir également Kahane & Osborne (2015) pour l'interprétation des stemmas et une représentation en terme de polygraphe (Kahane & Maziotta 2015).

Pour comparer deux analyses en constituants, on compare généralement les unités considérées. Il en va de même pour comparer une analyse en dépendance et une analyse en constituants : la méthode la plus courante consiste à regarder la projection[3] de chacun des éléments de l'arbre de dépendance et de les comparer avec les constituants de l'analyse en constituants (Lecerf 1961, Kahane 2001).

Nous pensons que ce n'est pas en terme d'unités, mais plutôt en terme de combinaisons que les analyses syntaxiques doivent être comparées. Expliquons-nous sur un exemple :

(2) *Le petit frère de Zoé m'a parlé de ça*.

Une analyse en constituants immédiats (Bloomfield 1933, Chomsky 1957) va postuler que la phrase (S) est la combinaison d'un NP (*le petit frère de Zoé*) et d'un VP (*m'a parlé de ça*), une analyse traditionnelle que la forme verbale *a parlé* a pour sujet *le petit frère de Zoé*, une analyse en dépendance qu'il y a une dépendance entre *parlé* et *frère* (UD), d'autres analyses encore pourront considérer que la dépendance est entre l'auxiliaire *a* et le déterminant *le* (Hudson 1984, 2007) ou entre les nucléus *a parlé* et *le frère* (Tesnière 1959) ou encore entre le chunk verbal *m'a parlé* et le chunk nominal *le petit frère* (Vergne 2000). D'autres encore peuvent considérer que c'est en fait la flexion verbale ou le mode indicatif qui gouverne le sujet (IP → DP + I' ; Abney 1987). Mais finalement toutes ces analyses s'accordent sur le fait qu'il y a, à un certain endroit, une *connexion*, que nous appellerons la *connexion subjectale*, et que cette connexion est instanciée de différentes façons selon les analyses et les formalismes sous-jacents (figure 4). Certaines de ces analyses sont compatibles, d'autres sont concurrentes, certaines analyses sont plus fines que d'autres. Nous allons donner un sens mathématique à ce que nous venons de dire et voir ce qu'on peut en faire.

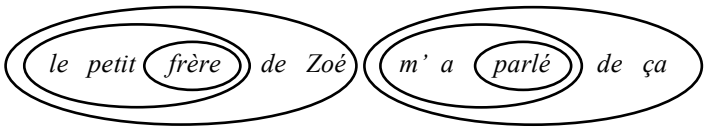

Figure 4. Différentes instanciations de la connexion subjectale

4. Combinaisons et ensembles de combinaisons

Même si, comme nous le verrons, on peut s'abstraire en partie de la question des unités, notre définition des combinaisons reposent principalement sur la notion d'unité. Nous appelons *unité* tout signe linguistique, vu ici comme une portion d'un texte. Notre définition est très lâche : nous ne faisons pas d'hypothèse de continuité

[3] On appelle projection (maximale) d'un nœud x de l'arbre de dépendance l'unité formée par l'ensemble des (mots qui étiquettent les) nœuds dominés par x, x compris.

en particulier. Les unités sont des suites (éventuellement discontinues) d'unités élémentaires. Dans cet article, nous ne considérons que les unités élémentaires d'une phrase sont les mots, mais la méthode peut bien sûr s'appliquer à des décompositions morphosyntaxiques plus fines en lexèmes et morphèmes flexionnels. Nous reviendrons sur la question de la *granularité* de la structure dans la section 9.

L'analyse d'une phrase, quelque soit le formalisme, repose au départ sur l'identification d'unités et de combinaisons entre ces unités. Nous nous limiterons dans cet exposé aux combinaisons binaires même si on peut tout à fait envisager d'étendre ce travail à des combinaisons ternaires (par exemple avec le traitement de certains morphèmes comme des marqueurs de connexions)[4]. Il y a deux principales façons d'indiquer que deux unités se combinent : soit en indiquant que ces deux unités forment ensemble une autre unité (c'est la méthode de l'analyse en constituants immédiats), soit en indiquant explicitement qu'il y a une connexion entre les deux unités (c'est la méthode des grammaires de dépendance). Il a déjà été montré que les deux méthodes sont en grande partie équivalentes (voir la notion de *grouping* chez Kahane & Mazziotta 2015), mais toutes les conséquences n'en ont pas été tiré. Nous représentons une combinaison par une paire (non ordonnée)[5] de deux unités, c'est-à-dire un ensemble à deux éléments. Pour que {A,B} soit une combinaison, il faut que A et B soient des unités et que A et B soient disjoints ($A \cap B = \emptyset$).

Un arbre de constituants binaire T induit naturellement un ensemble de combinaisons $C(T)$: à chaque fois qu'un constituant C se décompose en deux constituants A et B (ce que l'on note habituellement par $C \rightarrow A\ B$), on postule une combinaison {A,B}, puisque A et B se combinent pour donner C (cf. Chomsky 1957, Nida 1966, Mazziotta & Kahane 2017 sur l'interprétation de la constituance en terme de connexion). Si nous reprenons l'arbre de constituants T0 de la section 2, nous avons :

$$C(T0) = \{\ \{\textit{le chat, a vu Zoé}\}, \{\textit{le, chat}\}, \{\textit{a, vu Zoé}\}, \{\textit{vu, Zoé}\}\ \}$$

[4] La considération de dépendances ternaires apparaît à plusieurs reprises dans l'histoire de la syntaxe de dépendance. On en trouve chez Tesnière (1934) pour la coordination (la conjonction de coordination *et* étiquette la relation entre les conjoints), chez Débili (1982) pour les prépositions régimes et plus récemment dans les Collapsed Stanford Dependencies (CSD) (de Marneffe & Manning 2008).

[5] On pourrait prendre en compte la notion de *tête* en travaillant avec des couples (A,B) et la convention que le premier élément est la tête de la combinaison. Tout ce que nous faisons dans cet article peut être étendu à cette situation. Considérer à la fois des combinaisons avec et sans tête demanderait davantage de précautions. Ce problème est comparable à la prise en compte à la fois d'arêtes orientées et non orientées dans un graphe.

On peut associer à tout ensemble de combinaisons une représentation où les unités non élémentaires sont représentées par des bulles et les combinaisons par des arêtes entre les unités. On associe ainsi à C(T0) la représentation B0 de la figure 5.

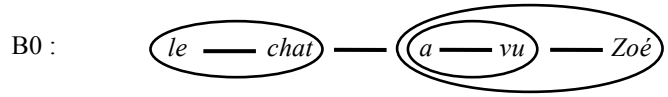

Figure 5. Représentation associée à C(T0)

Un arbre de dépendance D induit également naturellement un ensemble de combinaisons C_{min}(D) : si deux unités A et B sont connectées dans l'arbre de dépendance (A et B sont généralement des mots), alors on postule une combinaison {A,B}. Si nous reprenons l'arbre de dépendance D1 de la section 2, nous avons :

C_{min}(D1) = { {*le, chat*}, {*chat, vu*}, {*a, vu*}, {*vu, Zoé*} }

Comme précédemment, on peut associer une représentation B1 à C_{min}(D1) (figure 6). B1 est le graphe (non-orienté) sous-jacent à l'arbre D1:

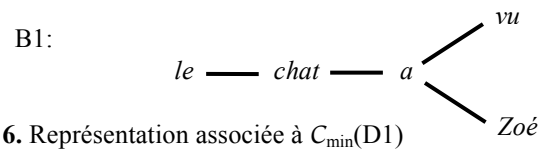

Figure 6. Représentation associée à C_{min}(D1)

On peut également associer naturellement des ensembles de combinaisons aux structures S2 et S3 de la figure 3 :

C_{min}(S2) = { {*le, chat*}, {*vu, Zoé*}, {*le chat, a*}, {*a, vu Zoé*} }

C_{min}(S3) = { {*le, chat*}, {*vu, Zoé*}, {*a, vu*}, {*le chat, a vu*} }

Les représentations associés à ces ensembles de combinaisons, donnés figure 7, sont les mêmes que ceux de la figure 3, à la différence que la hiérarchie induite par le positionnement vertical n'est plus pris en compte.

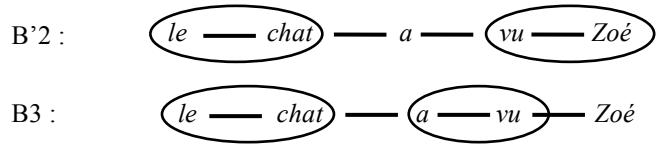

Figure 7. Représentations associées à C_{min}(S2) et C_{min}(S3)

5. Notions mathématiques

La suite de cet article fait appel à des notions de théorie des ensemble qu'il nous semble préférable de rappeler. Si E est un ensemble, on note E×E l'ensemble des couples d'éléments de E. Tout sous-ensemble de E×E peut-être vue comme une *relation binaire* sur les éléments de E. On dit d'une relation binaire R qu'elle est *réflexive* si tout élément de E est en relation avec lui-même ($\forall$x, xRx), *symétrique* si toute relation peut être inversée ($\forall$x,y, xRy $\rightarrow$ yRx), *antisymétrique* si au contraire elle ne peut jamais être inversée sauf dans les cas triviaux ($\forall$x,y [xRy et yRx] $\rightarrow$ x = y) et *transitive* si elle se propage systématiquement ($\forall$x,y,z [xRy et yRz] $\rightarrow$ xRz).

Une relation qui est réflexive, symétrique et transitive est appelée une *relation d'équivalence*. Une relation d'équivalence définit une partition de E, c'est-à-dire qu'elle découpe E en sous-ensembles deux à deux disjoints au sein desquels tous les éléments sont en relation les uns avec les autres. Ces ensembles sont appelées les *classes d'équivalence* de la relation R. L'ensemble des classes d'équivalence de la relation R sur E est appelé *ensemble quotient* de E par R et est noté $E_{/R}$. Il paraît un peu difficile de se représenter un tel ensemble au premier abord, car il s'agit d'un ensemble d'ensembles. Si on note $\mathcal{P}$(E) l'*ensemble des parties* de E, les éléments de $E_{/R}$ appartiennent à $\mathcal{P}$(E) et donc $E_{/R}$ est une partie de $\mathcal{P}$(E), c'est-à-dire un élément de $\mathcal{P}(\mathcal{P}(E))$.

Nous allons beaucoup utiliser la notion d'ensemble quotient et travailler non seulement avec des ensembles d'ensembles, mais avec des ensembles d'ensembles d'ensembles et ainsi de suite. Nous noterons $\mathcal{P}^2(E) = \mathcal{P}(\mathcal{P}(E))$, $\mathcal{P}^3(E) = \mathcal{P}(\mathcal{P}^2(E))$ et ainsi de suite.

Nous allons manipuler deux notions qu'il nous faudra bien distinguer : les *combinaisons* qui sont des paires d'unités, comme on vient de le voir, et les *connexions* qui sont des ensembles de combinaisons équivalentes, pour une relation d'équivalence que nous allons définir dans la prochaine section. On part d'un ensemble X d'*unités élémentaires*, les mots d'une phrase dans nos exemples. On considère l'ensemble U des unités sur X qui des suites sur X. Les combinaisons sont des paires d'unités et donc des éléments de $\mathcal{P}$(U). Les connexions sont des éléments de $\mathcal{P}^2$(U).

Pour ceux qui ne sont pas familiers avec les ensembles quotients, il est certainement nécessaire de faire le parallèle avec l'ensemble des nombres rationnels. On sait que $1_{/2}$ ou $2_{/4}$ ou encore $50_{/100}$ représentent le même nombre rationnel. Autrement dit, l'ensemble des rationnels est un ensemble quotient obtenu à partir de l'ensemble des couples de nombres entiers et de la relation d'équivalence suivante : deux couples (a,b) et (a',b') représentent le même nombre rationnel si et seulement si ab' = a'b. Chaque couple (a,b), ou fraction a/b pour prendre la notation usuelle, est un représentant du nombre relationnel. De même, les combinaisons sont des *représentants* des connexions.

Revenons sur les relations générales. A partir de n'importe quelle relation R, on peut définir une relation R* qu'on appelle la *clôture réflexive et transitive* de R. L'idée est que deux éléments x et y sont en relation par R* (xR*y) s'il existe une chaîne d'éléments en relation par R qui va de x à y, ce qui se formalise par :

xR^*y ssi $\exists n \geq 0$, $\exists x_0, x_1, \ldots, x_n$ tels que $x = x_0$, $y = x_n$ et $(\forall i = 0,\ldots,n\text{-}1\ x_i R x_{i+1})$

Par exemple, si R est la *relation de dépendance* dans un arbre (xRy ssi y dépend de x), la relation R* est la relation est la *relation de dominance* (xR*y ssi x domine y). La projection de x (voir note 3) est l'unité formée par l'ensemble des nœuds dominés par x.

Nous allons maintenant considérer un autre type de relation, tout aussi utile que les relations d'équivalence. Une relation qui est réflexive, antisymétrique et transitive est appelée un *ordre*. Par exemple, la relation $\leq$ est un ordre sur les nombres entiers. Un ordre qui s'apparente davantage aux ordres que nous considérons dans cet article est la relation d'inclusion $\subseteq$ entre ensembles. Cet ordre est *partiel* sur $\mathcal{P}$(E) dans le sens où de nombreux couples de parties de E ne sont pas ordonnés l'un par rapport à l'autre, comme par exemple les paires {a,b} et {b,c}. Un ordre partiel de ce type peut être représenté par un treillis. Nous donnons figure 8 le treillis de l'inclusion sur $\mathcal{P}$({a,b,c}). En fait, ce treillis ne représente pas directement la relation d'inclusion, mais une relation de « précédence » immédiate R dont la relation d'inclusion est la clôture réflexive et transitive ($\subseteq = R^*$).

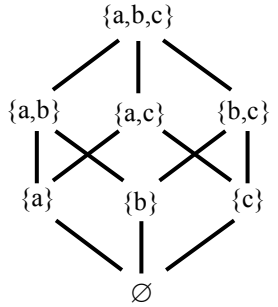

Figure 8. Treillis pour l'inclusion sur $\mathcal{P}$({a,b,c})

Pour un ordre (partiel ou non), on peut définir plusieurs notions. On appelle *élément maximum* un élément qui est plus grand que tous les autres et *élément minimum* un élément qui est plus petit que tous les autres. E est l'élément maximum de l'inclusion sur $\mathcal{P}$(E), tandis que l'ensemble vide, noté $\varnothing$, est l'élément minimum. On appelle *supremum* de x et y, noté $x \vee y$, le plus petit des éléments plus grands que x et y et *infimum* de x et y, noté $x \wedge y$, le plus grand des éléments plus petits que x et y. Les supremums et infimums pour l'inclusion sont respectivement l'*union* et l'*intersection* : $A \vee B = A \cup B$; $A \wedge B = A \cap B$. On repère facilement les supremums et infimums de x et y dans le treillis correspondant à un ordre, puisqu'il

s'agit des éléments directement liés à x et y qui se trouvent respectivement au dessus ou en dessous.

Une relation qui est seulement réflexive et transitive est un *quasi-ordre* (Louveau & Rosendal 2001). Si la relation $\preccurlyeq$ est un quasi-ordre, elle définit naturellement une relation d'équivalence $\equiv$ par la relation suivante :

$x \equiv y$ si et seulement si $x \preccurlyeq y$ et $y \preccurlyeq x$.

On peut projeter la relation $\preccurlyeq$ sur l'ensemble quotient $E_{|\equiv}$ en considérant que deux classes d'équivalence X et Y vérifient $X \preccurlyeq Y$ si pour tous éléments x de X et y de Y, on a $x \preccurlyeq y$. La relation $\preccurlyeq$ est alors un ordre sur $E_{|\equiv}$.

6. Compatibilité de combinaisons et connexions

On peut maintenant introduire une relation binaire sur les combinaisons que nous notons $\approx$ et que nous appelons la relation de *compatibilité* :

$\{A,B\} \approx \{A',B'\}$ ssi $A \cap A'$ et $B \cap B'$ sont non vides
et $A \cup A'$ et $B \cup B'$ sont disjoints

Dire que deux combinaisons sont compatibles revient à dire qu'elles représentent la même connexion, mais éventuellement réalisée par des combinaisons d'unités différentes. On peut représenter la relation de compatibilité par la configuration de la figure 9.

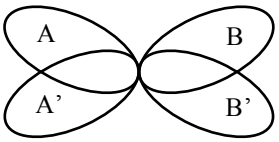

Figure 9. Deux combinaisons {A,B} et {A',B'} compatibles

La relation $\approx$ est presque une relation d'équivalence. Elle est réflexive et symétrique, mais elle n'est pas en générale transitive. Si nous considérons les connexions subjectales de T0, D1 et D2, on récupère les combinaisons respectives suivantes :

c0 = {*le chat, a vu Zoé*}, c1 = {*chat, vu*} et c2 = {*le, a*}

On a $c1 \approx c0$ et $c0 \approx c2$, mais $c1 \not\approx c2$, ce qui montre la non-transitivité.

Malgré cela, la relation $\approx$ est une relation d'équivalence sur de nombreux ensembles $\mathcal{E}$ de combinaisons et notamment sur ceux que nous souhaitons considérer. L'avantage d'une relation d'équivalence est qu'elle induit une partition de l'ensemble (section 5). Dans le cas d'un ensemble de combinaisons $\mathcal{E}$, les éléments de l'espace quotient $\mathcal{E}_{/\approx}$ sont appelés des *connexions*. Autrement dit, à

partir de nos structures de départ, nous induisons des ensembles de combinaisons, puis des ensembles de connexions.

Les *ensembles de connexions*, du type $\mathcal{E}_{/\approx}$ où $\mathcal{E}$ est un ensemble de combinaisons, sont selon notre point de vue les véritables structures avec lesquelles les linguistes devraient travailler. Avant d'expliquer pourquoi, nous allons présenter différents exemples d'ensembles de connexions, ainsi que différentes relations entre ces ensembles, ce qui nous permettra d'avoir une bonne idée des différentes structures possibles et de la place qu'on les arbres de constituants et les arbres de dépendance dans ce paysage.

Prenons un premier exemple d'ensemble de combinaison avec $\mathcal{E}4 = C_{min}(D1) \cup C(T0)$ = { { {*le, chat*}, {*chat, vu*}, {*le chat, a vu Zoé*}, {*a, vu*}, {*a, vu Zoé*}, {*vu, Zoé*} }. La relation ≈ est bien une relation d'équivalence sur $\mathcal{E}4$. Les classes d'équivalence sont :

C0 = { {*le, chat*} }
C1 = { {*chat, vu*}, {*le chat, a vu Zoé*} }
C2 = { {*a, vu*}, {*a, vu Zoé*} }
C3 = { {*vu, Zoé*} }

Chacune de ces classes est une connexion et $\mathcal{E}4_{/\approx}$ = { C0, C1, C2, C3 } est un ensemble de connexions. Rappelons que si U est l'ensemble des unités, les combinaisons sont des éléments de $\mathcal{P}(U)$, les ensembles de combinaisons, comme $C_{min}(D1)$, $C(T0)$ et $\mathcal{E}4$, et les connexions sont des éléments de $\mathcal{P}^2(U)$ et les ensembles de connexions, comme $\mathcal{E}4_{/\approx}$, sont eux des éléments de $\mathcal{P}^3(U)$.

Les connexions C1 et C2 ont deux représentants. L'un des représentants appartient à $C_{min}(D1)$ et l'autre à $C(T0)$. On peut voir les ensembles de combinaisons $C_{min}(D1)$ et $C(T0)$ comme des instanciations particulières de l'ensemble de connexions $\mathcal{E}4_{/\approx}$, où une combinaison a été sélectionnée dans chaque connexion. Nous allons maintenant préciser le lien entre les différents ensembles de connexions et leurs différentes instanciations, les ensembles de combinaisons.

7. Ensemble de combinaisons minimal et maximal

On peut associer à un arbre de dépendance D un ensemble de combinaisons *étendu* que nous appelons $E_{max}(D)$ et qui est, d'un certain point de vue que nous allons définir, équivalent à $E_{min}(D$. Nous considérons maintenant comme unités toutes les portions connexes de notre arbre de dépendance, encore appelées les catenae de (Osborne et al. 2012). Les catenae de D1 sont :

Catenae(D1) = { *le, chat, a, vu, Zoé, le chat, chat vu, a vu, vu Zoé, le chat vu, chat a vu, chat vu Zoé, le chat a vu, le chat vu Zoé, chat a vu Zoé, le chat a vu Zoé* }.

On peut, à partir de tout ensemble X d'unités, définir un ensemble de combinaisons, que nous appelons Combi(X) :

{A,B} appartient à Combi(X) ssi A, B et A ∪ B sont des éléments de X.

Nous pouvons ainsi définir, à partir d'un arbre de dépendance D, un nouvel ensemble de combinaisons, que nous nommons C_{max}(D) :

C_{max}(D) = Combi(Catenae(D)).

On peut construire C_{max}(D) directement à partir de D : deux portions connexes A et B de D forment une combinaison {A,B} dans C_{max}(D) ssi A et B sont disjointes et connectées par une dépendance. L'ensemble de combinaisons C_{max}(D) obtenu est assez gros. Ainsi C_{max}(D1) contient à la fois C_{min}(D1) et C(T0) et encore plein d'autres combinaisons. Malgré cela, ≈ est une relation d'équivalence sur cet ensemble. En fait, les C_{max}(D) sont les plus gros ensembles sur lesquels ≈ est une relation d'équivalence, à condition que l'on ajoute des conditions de bonnes formation qui restreignent les ensembles de combinaisons possibles, comme la connexité que nous introduirons à la section 8.

Nous allons maintenant nous intéresser aux ensembles de connexions C_{max}(D)/≈. Tout ensemble C_{max}(D)/≈ contient le même nombre de connexions que C_{min}(D)/≈. Par exemple, les connexions dans C_{max}(D1)/≈ sont :

C'0 = { {*le, chat*}, {*le, chat vu*}, {*le, chat a vu*}, {*le, chat vu Zoé*}, {*le, chat a vu Zoé*} }
C'1 = { {*chat, vu*}, {*le chat, vu*}, {*chat, a vu*}, {*chat, vu Zoé*}, {*le chat, a vu*}, {*le chat, vu Zoé*}, {*chat, a vu Zoé*}, {*le chat, a vu Zoé*} }
C'2 = { {*a, vu*}, {*a, vu Zoé*}, {*a, chat vu*}, {*a, le chat vu*}, {*a, chat vu Zoé*}, {*a, le chat vu Zoé*} }
C'3 = { {*vu, Zoé*}, {*a vu, Zoé*}, {*chat vu, Zoé*}, {*le chat vu, Zoé*}, {*le chat a vu, Zoé*} }

On voit que chaque connexion dans l'arbre de dépendance correspond à tout un ensemble de combinaisons et pas seulement à une combinaison entre mots. On peut représenter cet ensemble de combinaisons équivalents par un diagramme, comme nous le proposons dans la figure 10 pour la connexion subjectale C'1 de C_{max}(D1)/≈.

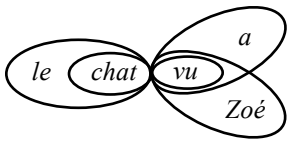

Figure 10. Les combinaisons de la connexion subjectale C'1 de C_{max}(D1)/≈

Nous avons associé deux ensembles de combinaisons à tout arbre de dépendance, C_{min}(D) et C_{max}(D). Si nous avons appelé ces deux ensembles C_{min} et C_{max}, c'est qu'ils sont d'un certain point de vue (qu'il nous reste à définir) le plus petit et le plus grand ensemble de combinaisons qu'on peut associer à un arbre de dépendance.

C_{min}(D) est clairement le plus petit ensemble de combinaisons qu'on peut associer à D, car il ne contient qu'une combinaison par connexion, c'est-à-dire par classe d'équivalence de la relation ≈. Tout ensemble de combinaison qui ne contiendrait qu'une combinaison par connexion et qui serait différent d'un C_{min}(D) représente une autre structure, « autre » en un sens qu'il nous reste à définir. C_{max}(D) est le plus grand ensemble de combinaisons qu'on peut associer à D, sous certaines conditions de connexité et acyclicité de l'ensemble de combinaions que nous allons introduire plus bas.

Ces deux ensembles nous intéressent pour des raisons opposées : l'avantage d'un ensemble comme C_{min}(D), avec un seul représentant par connexion, est qu'il peut facilement donner lieu à une représentation graphique, un *diagramme syntaxique* comme ceux des figures 5, 6 et 7. Mais fondamentalement, un tel diagramme doit être interprété comme C_{max}(D), c'est-à-dire comme contenant un foisonnement de combinaisons possibles.

Nous voudrions généraliser ce que nous avons fait pour les ensembles de combinaisons associés aux arbres de dépendance à tous les ensembles de combinaisons. Autrement dit, nous voulons associer à tout ensemble de combinaisons $\mathcal{E}$ des ensembles de combinaison $[\mathcal{E}]_{min}$ et $[\mathcal{E}]_{max}$ tels que : $[\mathcal{E}]_{min}$ possède un représentant par connexion et permet de tracer un diagramme lisible, $[\mathcal{E}]_{max}$ représente la multitude de combinaisons associée à chaque connexion de la structure.

Commençons par $[\mathcal{E}]_{min}$. Notre définition de la relation ≈ n'exclut pas des connexions telles C = { {ab,d}, {bc,d}, {ac,d} }. Pour une telle connexion, il est impossible de définir un représentant minimal, car *d* n'est finalement connecté ni à *a*, ni à *b*, ni à *c*. Nous ajoutons donc une condition : un ensemble de combinaisons $\mathcal{E}$ est *bien fondé* si toutes ses connexions sont bien fondées ; une connexion C est *bien fondée* si C contient une *combinaison minimale*, c'est-à-dire s'il existe une combinaison {u,v} dans C telle pour toute combinaison {u',v'} dans C, on ait u ⊆ u' et v ⊆ v'.[6] Si $\mathcal{E}$ est bien fondé, on définit alors $[\mathcal{E}]_{min}$ comme l'ensemble des combinaisons minimales de $\mathcal{E}$. On a bien sur $[\mathcal{E}]_{min} \subseteq \mathcal{E}$.

Pour définir $[\mathcal{E}]_{max}$, nous devons étendre à toute structure la notion de catena introduite section 7 pour les arbres de dépendance. Pour commencer, appelons Unités($\mathcal{E}$) l'ensemble des unités qui apparaissent dans les combinaisons de $\mathcal{E}$. $\mathcal{E}$ définit une relation binaire sur Unités($\mathcal{E}$) que nous notons □ : u□v ssi {u,v} ∈ $\mathcal{E}$. Dans le cas où $\mathcal{E} = C_{min}$(D) avec D un arbre de dépendance, Unités($\mathcal{E}$) est un ensemble de mots et deux mots sont en relation par □ s'ils sont liés par une dépendance.

---

[6] Il n'est pas nécessaire d'imposer que la combinaison minimale de C appartienne à C. On pourrait accepter des connexions telles que C = { {ab,c}, {b,cd} }, qui induit la combinaison minimale {b,c}.

On considère maintenant l'ensemble Catenae($\mathcal{E}$) des unités qui sont des catenae pour la relation □ sur Unités($\mathcal{E}$), c'es-à-dire des unités obtenues en réunissant des unités liées les unes aux autres par □. Dans le cas où $\mathcal{E} = C(T)$ avec T un arbre de constituants binaires, Unités($\mathcal{E}$) est l'ensemble des constituants de T et Catenae($\mathcal{E}$) = Unités($\mathcal{E}$), car chaque constituant ne peut se combiner qu'avec un seul autre constituant (en raison de la binarité) et le résultat donne un constituant.

Nous considérons une deuxième propriété de bonne formation de nos ensembles de combinaisons, qui est une forme de connexité : nous dirons que $\mathcal{E}$ est *connexe* si toute unité u de $\mathcal{E}$ non élémentaire est un catena d'unités strictement incluse dans u. Cette condition assure que la structure soit suffisamment fine pour que toute unité soit décomposable jusqu'aux unités élémentaires. Nous imposons une troisième propriété de bonne formation qui est l'*acyclicité* de $\mathcal{E}$: la relation □ associée à $\mathcal{E}$ doit être acyclique. En effet, si □ n'est pas acyclique, la relation ≈ n'est pas une relation d'équivalence sur $[\mathcal{E}]_{max}$. En effet, supposons que $\mathcal{E}$ contienne {a,b}, {b,c}, {a,c}, alors ab est un catena, donc {ab,c} ∈ $[\mathcal{E}]_{max}$, {a,c} ≈ {ab,c}, {b,c} ≈ {ab,c}, mais {a,c} ≉ {b,c}, ce qui montre la non-transitivité de ≈.

Comme dans le cas des arbres de dépendance, on associe à tout ensemble de combinaison $\mathcal{E}$ connexe et acyclique l'ensemble de combinaisons $[\mathcal{E}]_{max}$ défini par :

$[\mathcal{E}]_{max}$ = Combi(Catenae($\mathcal{E}$)).

La connexité et l'acyclicité assure que ≈ est bien une relation d'équivalence et que $\mathcal{E} \subseteq [\mathcal{E}]_{max}$.

On peut vérifier que $[C_{min}(D)]_{max} = C_{max}(D)$ et $[C_{max}(D)]_{min} = C_{min}(D)$ pour tout arbre de dépendance et que $[C(T)]_{min} = [C(T)]_{max} = C(T)$ pour tout arbre de constituants binaire. On peut généraliser ce type de propriété : pour tout ensemble de combinaisons bien fondé, connexe et acyclique, on a :

$[[\mathcal{E}]_{min}]_{max} = [\mathcal{E}]_{max}$

$[[\mathcal{E}]_{max}]_{min} = [\mathcal{E}]_{min}$.

Les deux derniers résultats découlent directement de nos propriétés de bonne formation : $[[\mathcal{E}]_{min}]_{max} = [\mathcal{E}]_{max}$, car, en raison de la connexité, toute unité de $\mathcal{E}$ peut être reconstruite à partir de sous-unités qui se combinent entre elles, et en particulier d'unités appartenant à des combinaisons minimales en raison de la bonne fondation des connexions. Quant à $[[\mathcal{E}]_{max}]_{min} = [\mathcal{E}]_{min}$, cela découle trivialement de la bonne fondation des connexions.

Nous allons donner un premier exemple d'utilisation des ensembles minimaux et maximaux. Considérons $\mathcal{E}3 = C_{max}(D1) \cap C_{max}(D2)$. Considérer l'ensemble $\mathcal{E}3$ revient à sous-spécifier l'analyse afin de ne pas prendre les décisions qui rendent $C_{max}(D1)$ et $C_{max}(D2)$ incompatibles. En l'occurrence, ces décisions sont de savoir qui du déterminant ou du nom est la tête du groupe *le chat* et qui de l'auxiliaire ou du verbe lexical gouverne le sujet. Si l'on garde pour chaque connexion de $\mathcal{E}3$ la plus petite combinaison on obtient :

$$[\mathcal{E}3]_{\min} = \{ \{le, chat\}, \{a, vu\}, \{le\ chat, a\ vu\}, \{vu, Zoé\} \} = C_{\min}(S3)$$

où S3 est la structure à la Tesnière introduite dans la section 3 (figure 3). Le diagramme associé à $[\mathcal{E}3]_{\min} = C_{\min}(S3)$ a été donné figure 7.

8. Ordre et équivalence entre ensembles de combinaisons connexes

Nous avons que les $C_{\max}(D)/\approx$ et les $C_{\min}(D)/\approx$ ont le même nombre de connexions. D'une certaine façon, ils définissent la même structure, car leurs connexions sont deux à deux équivalentes. Ceci nous amène à étendre la relation $\approx$ sur les combinaisons aux connexions : les connexions C et C' sont *compatibles* par $\approx$ (ce que nous notons C $\approx$ C') ssi les combinaisons appartenant à C et C' sont compatibles par $\approx$.

On peut maintenant définir une relation de compatibilité sur les ensembles de combinaisons que nous notons $\lessapprox$. Deux ensembles de combinaisons $\mathcal{E}$ et $\mathcal{F}$ sont dits *compatibles*, ce que nous notons $\mathcal{E} \lessapprox \mathcal{F}$, si leurs connexions sont deux à deux compatibles par $\approx$. On a également :

$\mathcal{E} \lessapprox \mathcal{F}$ ssi $\approx$ est une relation d'équivalence sur $\mathcal{E} \cup \mathcal{F}$
et $\mathcal{E}/\approx$, $\mathcal{F}/\approx$ et $(\mathcal{E} \cup \mathcal{F})/\approx$ ont le même cardinal.

En effet, si une connexion C de $\mathcal{E}$ est compatible avec une connexion C' de $\mathcal{F}$, C $\cup$ C' forme une classe d'équivalence de $\approx$ sur E $\cup$ F et donc une connexion. On peut voir $\mathcal{E}$ et $\mathcal{F}$ comme des instanciations différentes des connexions définies par $(\mathcal{E} \cup \mathcal{F})/\approx$.

Comme nous l'avons montré dans la section 7, $C_{\min}(D) \lessapprox C_{\max}(D)$, pour tout arbre de dépendance D. Par ailleurs, les ensembles de combinaisons définis par les arbres de dépendance sont compatibles avec de nombreux ensembles de combinaisons définis par des arbres de constituants. Par exemple, $C(T0) \lessapprox C_{\max}(D1)$, car $C(T0) \subseteq C_{\max}(D1)$ et $C(T0)$ possède quatre connexions comme $C_{\max}(D1)$. On a également $C(T0) \lessapprox C_{\min}(D1)$, comme on l'a vu à la section 6 en étudiant $\mathcal{E}4 = C_{\min}(D1) \cup C(T0)$.

De même que $\approx$ n'est pas transitive sur les combinaisons, $\lessapprox$ n'est pas transitive sur les ensembles de combinaisons. Par exemple, $C(T0)$ est compatible avec $C_{\min}(D1)$ et $C_{\min}(D2)$, alors que ces deux derniers ensembles ne sont pas compatibles entre eux, car $(C_{\min}(D1) \cup C_{\min}(D2))/\approx$ n'a pas le même cardinal que $C_{\min}(D1)/\approx$ et $C_{\min}(D2)/\approx$. En effet, ces deux derniers ensembles ont quatre éléments, tandis que $(C_{\min}(D1) \cup C_{\min}(D2))/\approx$ en a cinq, car les combinaisons c1 = {*chat, vu*} de $C_{\min}(D1)$ et c2 = {*le, a*} de $C_{\min}(D2)$ ne sont pas compatibles et appartiennent à deux connexions distinctes.

Si l'on considère à nouveau $\mathcal{E}3 = C_{\max}(D1) \cap C_{\max}(D2)$, on voit que l'on construit une structure compatible à la fois avec $C_{\min}(D1)$ et $C_{\min}(D2)$ et qui est plus « proche » de $C_{\min}(D1)$ et $C_{\min}(D2)$ que ne l'est $C(T0)$. $\mathcal{E}3$ contient une combinaison

minimale {*le chat, a vu*} qui n'est pas élémentaire, et qu'on peut donc raffiner de différentes façons, ce qui donne des ensembles de combinaisons incompatibles. Dès qu'une combinaison est non élémentaire, on peut construire deux combinaisons plus fines incompatibles entre elles. Par exemple, pour {ab,c}, on a {a,c} et {b,c} qui sont plus fines et incompatibles entre elles. Notons que {a,b} ≈ {ab,c}, {b,c} ≈ {ab,c} et {a,c} ≉ {b,c}. Si maintenant, nous prenons un ensemble de combinaisons $\mathcal{E}$ ayant {ab,c} comme combinaison minimale et que nous considérons $\mathcal{F}$ = $\mathcal{E}$ ∪ {a,c} et $\mathcal{F}$' = $\mathcal{E}$ ∪ {b,c}, nous aurons $\mathcal{E} \approxeq \mathcal{F}$ et $\mathcal{E} \approxeq \mathcal{F}$', mais $\mathcal{F}$ et $\mathcal{F}$' ne seront pas compatibles entre eux pour ≋. Ce que nous venons de montrer, c'est qu'en raffinant une des connexions de $\mathcal{E}$, on obtient un nouvel ensemble de combinaisons qui est compatible avec moins d'ensembles de combinaisons. En effet, $\mathcal{F}$ n'est pas compatible avec $\mathcal{F}$', alors que $\mathcal{E}$ l'est. A l'inverse, on peut montrer que $\mathcal{E}$ est compatible avec tous les ensembles compatibles avec $\mathcal{F}$, car $\mathcal{E} \subseteq \mathcal{F}$ et $\mathcal{E} \approxeq \mathcal{F}$.

Nous allons donner un sens précis à cette idée de finesse des connexions et sur/sous-spécification. On peut, à partir de ≋, définir une relation d'ordre ⪯ telle que $\mathcal{E}$ soit *plus fin* que $\mathcal{F}$ ($\mathcal{E} \preceq \mathcal{F}$) si $\mathcal{E}$ est compatible avec moins d'ensembles que $\mathcal{F}$ et *équivalent* ($\mathcal{E} \equiv \mathcal{F}$) s'il est compatible avec les mêmes ensembles que $\mathcal{F}$. Nous introduisons la relation suivante sur les ensembles de combinaisons bien formés (c'est-à-dire bien fondés, connexes et acycliques) :

$\mathcal{E} \preceq \mathcal{F}$ ssi $\mathcal{E} \approxeq \mathcal{F}$ et $[\mathcal{F}]_{max} \subseteq [\mathcal{E}]_{max}$

La relation ⪯ est un quasi-ordre (voir section 3) sur les ensembles de combinaisons connexes. Il est trivial de vérifier qu'elle est réflexive et transitive. Ce n'est pas un ordre, car la relation n'est pas antisymétrique. Nous définissons donc à partir de ⪯ la relation ≡ :

$\mathcal{E} \equiv \mathcal{F}$ ssi ($\mathcal{E} \preceq \mathcal{F}$ et $\mathcal{F} \preceq \mathcal{E}$)
ssi $[\mathcal{E}]_{max} = [\mathcal{F}]_{max}$
ssi $[\mathcal{E}]_{min} = [\mathcal{F}]_{min}$

La relation ≡ est une relation d'équivalence. D'après ce nous avons vu à la section 7, $[\mathcal{E}]_{min} \equiv [\mathcal{E}]_{max}$ pour tout ensemble de combinaisons bien formé et $\mathcal{F} \equiv \mathcal{E}$ ssi $[\mathcal{E}]_{min} \subseteq \mathcal{F} \subseteq [\mathcal{E}]_{max}$. En particulier, $C_{min}(D) \equiv C_{max}(D)$ pour tout ensemble de dépendance D.

Dans la figure 11, nous donnons un fragment de la relation ⪯. Il s'agit du treillis de relations pour toutes les structures plus fines qu'un arbre de constituants binaire T1 donné. (Par soucis de lisibilité, nous n'avons pas mis les trois arbres de dépendance qui sont en-dessous de la 4ème structure sous T1 ; on peut facilement les déduire en regardant ceux qui sont liés à la 2ème structure.) Chaque structure S' qui est sous une autre structure S précise une des connexions de S. Lorsqu'on part de l'arbre de constituants T1, on peut préciser soit la connexion mettant en jeu *a*, soit la connexion mettant en jeu *b*. La connexion mettant en jeu *a* peut être attribuée soit à *b* (1ère structure), soit à *cd* (3ème structure, tandis que la connexion mettant en jeu *b* peut être attribuée soit à *c* (2ème structure), soit à *d* (4ème structure). De ces 4

structures, celle qui donne le plus de possibilités est la 3ème, où *cd* est connecté à *a* et *b*. On peut en effet connecter l'un des éléments, *a* ou *b*, à *c* ou *d*, tout en laissant l'autre connecté à *cd*. On obtient au final six arbres de dépendance plus fins que l'arbre de constituants T1. Remarquons encore que comme tout treillis d'ordre, le treillis de la figure 11 permet de visualiser facilement les infimums et supremums de deux structures. On peut par exemple, pour chaque paire d'arbres de dépendance, voir la structure la plus fine les sous-spécifiant en considérant la structure à laquelle ils sont tous les deux reliés.

Comme le laisse supposer la figure 11, les éléments minimaux pour la relation $\preceq$ sont les $C_{\min/\max}(D)$ avec D arbre de dépendance, tandis que les éléments maximaux sont les $C(T)$ avec T arbre de constituants binaire.

Avant de montrer ces résultats, donnons en une interprétation. Ces résultats signifient que les arbres de constituants binaires sont les structures qui spécifient le moins les connexions, alors que les arbres de dépendance sont celles qui les spécifient le plus. Entre les deux, il existe toute une gamme de structure comme le montre la figure 11. On peut aussi se reporter aux exemples présentés au début de cet article : la combinaison c0 = {*le chat, a vu Zoé*} de l'arbre de constituants T0 indique juste qu'il y a une combinaison entre deux grandes unités, alors que la stemma à la Tesnière S3 attribue cette connexion à une combinaison un peu plus fine {*le chat, a vu*} et qu'un arbre de dépendance comme D1 ou D2 attribue cette connexion à une combinaison de mots.

Les $C_{\min/\max}(D)$ sont bien les seuls éléments minimaux puisque tout $\mathcal{E}$ minimal contient nécessairement des combinaisons minimales entre unités élémentaires, sinon les connexions correspondantes pourraient être raffinées. Vu les hypothèses de connexité et d'acyclicité par ailleurs, il s'agit donc bien d'un arbre de dépendance.

Le fait que les $C(T)$ sont maximaux est une conséquence directe du fait que les $[C(T)]_{\max} = C(T)$. Comme $C(T)$ possède une unique combinaison par connexion, il ne peut pas y avoir d'ensemble de combinaisons $\mathcal{E}$ bien formé avec $\mathcal{E} \subset C(T)$.

On peut montrer facilement que :

$\mathcal{E} \vee \mathcal{F} \equiv [\mathcal{E}]_{\max} \cap [\mathcal{F}]_{\max}$

$\mathcal{E} \wedge \mathcal{F} \equiv [\mathcal{E}]_{\max} \cup [\mathcal{F}]_{\max} \equiv [\mathcal{E}]_{\min} \cup [\mathcal{F}]_{\min}$

Il est possible que ces ensembles soient mal formés, ce qui signifie $\mathcal{E}$ et $\mathcal{F}$ n'ont pas supremum ou pas d'infimum. Par exemple, $C_{\min/\max}(D1)$ et $C_{\min/\max}(D2)$ sont des éléments minimaux non équivalents et n'ont donc pas d'infimum ($C_{\min}(D1) \cup C_{\min}(D2)$ contient un cycle). De même, deux $C(T)$ pour des arbres de constituants différents ne pourront pas avoir de supremum.

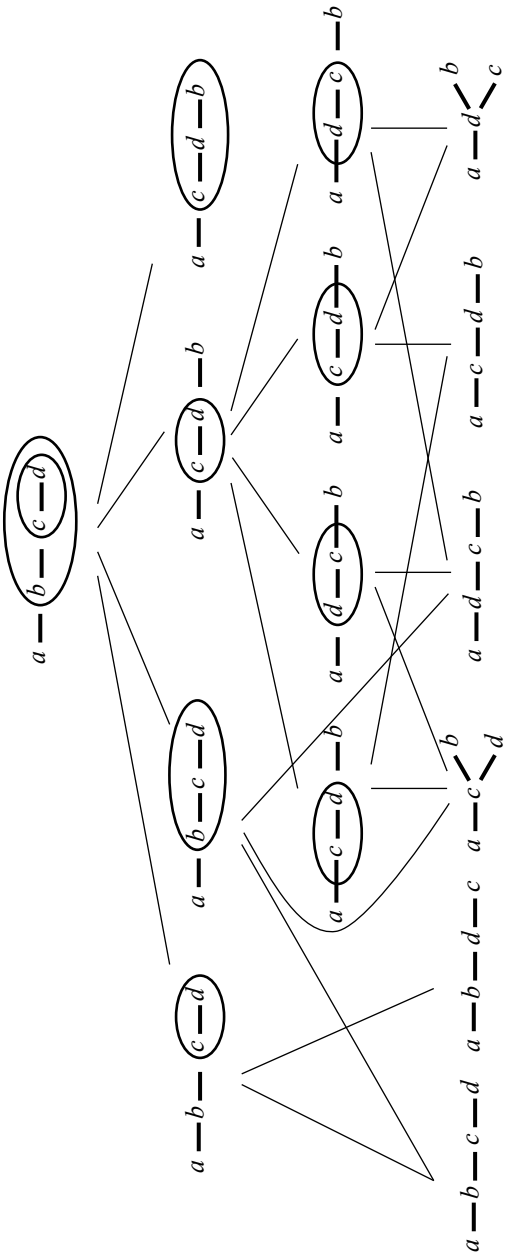

Figure 11. Treillis de la relation d'ordre $\preceq$ pour les structures plus fines qu'un arbre de constituants donné

L'ordre $\preceq$ était déjà introduit dans Gerdes & Kahane (2013) sous une forme plus géométrique. L'objectif des auteurs est de montrer comment à partir d'un ensemble d'unités syntaxiques V assez riche de construire une structure dite de connexion. Pour cela, les auteurs proposent à partir d'un ensemble d'unités V donné de construire n'importe quel arbre de constituants binaires dont les constituants appartiennent à V, puis de raffiner cet arbre jusqu'à obtenir une structure qui ne peut plus être raffinée. Cette structure peut en fait être construite directement : il s'agit de Combi(V). Ce que montre notre article, c'est que Combi(V) est l'infimum de tous les arbres de constituants sur V. Et si nous voulons faire un diagramme pour Combi(V), il suffit de prendre la combinaison minimale de chaque connexion, les connexions étant définies grâce à la relation d'équivalence $\approx$. Nous avons ainsi fournit un outil plus propre et direct pour définir une structure de connexion à partir d'un ensemble d'unités.

9. Granularité de la structure

Nous avons annoncé dans le titre de cet article la nécessité d'appréhender la structure en termes de connexions plutôt que d'unités. Il a malgré tout été beaucoup question d'unités, puisque les connexions se définissent en termes de combinaisons d'unités. Néanmoins, les unités n'ont pas plus de valeurs pour les connexions que les nombres entiers qui apparaissent dans la définition des nombre rationnels : de même que $1/2$ et $50/100$ représentent le même nombre, deux combinaisons équivalentes représente la même connexion, indépendamment des unité qui les composent. Si l'on change la granularité de l'analyse, les unités élémentaires et les combinaisons minimales des connexions changent sans que cela ne change fondamentalement la structure (à l'exception des connexions qui sont plus fines que le grain).

Nous allons formaliser cette question de granularité de la structure et de changement de grain. Le fait de voir la structure comme un ensemble de connexions permet modifier facilement la granularité de la structure. Considérons un ensemble de combinaisons $\mathcal{E}$ défini sur un ensemble X d'unités élémentaires. On s'intéresse à un nouvel ensemble d'unité Y plus grossier que X, autrement dit un ensemble Y d'unités sur X. Nous voulons maintenant prendre Y comme ensemble d'unités élémentaires et regarder notre structure avec ce nouveau grain. Il suffit de considérer l'infimum de $\mathcal{E}$ et de l'ensemble des combinaisons d'éléments de Y, que nous appelons la *restriction* de $\mathcal{E}$ à Y.

Par exemple, si Y est l'ensemble des chunks (qui sont des combinaisons particulières de mots, cf. Abney 1991, Vergne 2000), la restriction d'un arbre de dépendance à Y donnera un arbre dont les nœuds seront des chunks définis par l'arbre de dépendance original. Inversement, on peut considérer que la structure syntaxique se construit non pas sur les mots de la phrase, mais sur des unités plus fines, les lexèmes et les morphèmes. Un arbre de dépendance traditionnel, qui est un

arbre sur les mots, peut alors être vu comme la restriction aux mots d'une structure plus fine.

On voit à nouveau l'intérêt, lorsqu'on appréhende un arbre de dépendance, de voir cette structure syntaxique non pas comme une structure avec une combinaison par connexion (C_{min}(D)), mais comme une structure avec de nombreuses instanciations possibles pour chaque connexion (C_{max}(D)) et dont l'instanciation minimale dépend du grain considéré.

10. Conclusion

Nous allons essayer dans cette conclusion d'interpréter les résultats que nous avons présentés, au delà de leur intérêt purement mathématique.

Premièrement, si l'on évalue les représentations syntaxiques usuelles uniquement en termes de connexions, on voit que les arbres de constituants sont les moins spécifiés et que les arbre de dépendance sont les plus spécifiés. De là à en conclure que les arbres de constituants sont sous-spécifiés et que les arbres de dépendance sont sur-spécifiés, il n'y a qu'un pas à franchir. Il y a entre ces deux extrêmes tout un panel de structures que l'on peut envisager et qu'il y aurait un intérêt à étudier davantage à notre avis.

Deuxièmement, il ne faut pas confondre la structure syntaxique et les représentations qu'on en fait. Nous considérons que la syntaxe est l'étude de la façon dont les signes linguistiques se combinent. Lorsqu'on représente cette structure par un arbre de dépendance, le choix de la représentation, avec des liens entre les mots, laisse supposer que les combinaisons ont uniquement lieu entre les mots. Nous avons vu qu'il était possible d'avoir une interprétation beaucoup plus riche de cette structure et que derrière chaque connexion liant deux mots se cache un grand nombre de combinaisons, qui sont autant d'instanciations de cette connexion. De ce point de vue, un arbre de constituants binaire apparaît comme une des instanciations possible d'un arbre de dépendance. Si on ajoute la notion de tête, on peut à partir de n'importe quelle instanciation récupérer la structure complète (cf. la méthode de Lecerf (1961) ou celle de Kahane & Mazziotta (2015) pour passer d'un arbre de constituants avec têtes à un arbre de dépendance).

Troisièmement, même si nous n'avons pas vraiment d'éléments pour étayer cela, nous postulons que lors de l'analyse d'une phrase, les connexions sont instanciées par des combinaisons particulières portant sur des unités particulières et que ces instanciations peuvent être différentes d'une situation à l'autre et ne correspondent a priori ni à un arbre de constituants, ni à un arbre de dépendance. Du point de vue théorique, nous pensons que la prosodie (y compris la prosodie silencieuse du lecteur) joue un rôle important et que les unités prosodiques sont des candidats incontournables de cette instanciation (voir Steedman 2014 pour une approche similaire dans le cadre des grammaires catégorielles). Du point de vue du TAL, cela a, à notre avis, des conséquences importantes et signifie que les algorithmes de

parsing pourraient prendre en compte le fait qu'on ne cherche pas à construire une instanciation particulière de la structure abstraite (un arbre de constituants ou un arbre de dépendance en l'état actuel des systèmes), mais à construire n'importe quelle instanciation de la structure abstraite.

Quatrièmement, les analyses se placent généralement à un niveau de granularité particulier, celui des mots ou des unités morphosyntaxiques (lexèmes et morphèmes flexionnels). Là encore il s'agit d'instanciations particulières des connexions. Il est important de concevoir que la même structure abstraite peut être envisagée à différents niveaux de granularité. Il est probable de surcroit que les locuteurs manipulent des unités de différents niveaux aussi bien lorsqu'ils produisent que lorsqu'ils analysent des énoncés. Nous avons montré que notre formalisation des connexions ne fait aucune hypothèse sur la nature des unités mises en jeu lors de sa réalisation. Nous défendons l'idée que l'on peut faire de la syntaxe sans poser a priori la question des unités, laquelle est très délicate, tant il est difficile de définir des notions comme celles de mot ou de phrase. De ce point de vue, une définition qui dirait que la syntaxe est l'étude de l'organisation des mots au sein de la phrase nous semble à rejeter totalement. Pour nous, la syntaxe est avant tout l'étude des combinaisons (libres, régulières) entre signes linguistiques, sans préjuger du niveau de granularité de ces signes.

Bibliographie

ABNEY S. P. (1987) *The English noun phrase in its sentential aspect*. Thèse de doctorat. Massachusetts Institute of Technology.

ABNEY S. P. (1991). Parsing by chunks. In Berwick R. C., Abney S. P., Tenny, C. (éds), *Principle-based parsing* [*Computation and psycholinguistics*, 44], Springer, Dordrecht, 257-278.

BEAUZÉE N. (1765) Régime, in Denis Diderot & Jean Le Rond D'Alembert J. (eds.), *Encyclopédie ou Dictionnaire raisonné des sciences, des arts et des métiers*, vol. 14, 5-11.

BONFANTE G., GUIRAUD. Y. (2008) Intensional properties of polygraphs. *Electronic Notes in Theoretical Computer Science*, 203(1):65–77.

BURRONI A. (1993) Higher-dimensional word problems with applications to equational logic. *Theoretical computer science*, 115(1):43–62.

BLOOMFIELD L. (1933) *Language*, The University of Chicago Press.

CHOMSKY N. (1957) *Syntactic Structures*, MIT Press, Cambridge.

DEBILI F. (1982) *Analyse syntaxico-sémantique fondée sur une acquisition automatique de relations lexicales-sémantiques*, Thèse de doctorat d'état, Université Paris Sud, Orsay.

DE MARNEFFE M.-C., MANNING C. D. (2008) The Stanford typed dependencies representation. Proceedings of *Workshop on Cross-framework and Cross-domain Parser Evaluation*, COLING.

GERDES K. & KAHANE S. (2013) Defining dependency (and constituency), in K. Gerdes, E. Hajičová, L. Wanner (éds.), *Computational Dependency Linguistics*, IOS Press.

GLADKIJ A. V. (1968) On describing the syntactic structure of a sentence (en russe avec résumé en anglais), *Computational Linguistics, 7*, Budapest, 21-44.

HUDSON R. A. (1984) *Word grammar*, Oxford: Blackwell.Hudson

HUDSON R. A. (2007) *Language networks: The new word grammar*, Oxford University Press.

KAHANE S. (1997) Bubble trees and syntactic representations. In *Proceedings of the 5th conference on Mathematics of Language (MoL 5)*, 70-76.

KAHANE S. (2001) Grammaires de dépendance formelles et théorie Sens-Texte, Tutoriel, *Actes TALN 2001*, *vol. 2*, 17-76.

KAHANE S. & MAZZIOTTA N. (2015) Syntactic Polygraphs. A Formalism Extending Both Constituency and Dependency. In *Proceedings of the 14th Meeting on the Mathematics of Language (MoL 2015)*. Association for Computational Linguistics, pp. 152-164.

KAHANE S. & OSBORNE T. (2015) Translators' introduction, in L. Tesnière, *Elements of structural syntax*, John Benjamins, ixxx-lxiii (49 p.).

LECERF Y. (1961) Une représentation algébrique de la structure des phrases dans diverses langues naturelles, *Comptes Rendus de l'Académie des Sciences*, *252*(2), 232-235.

LOUVEAU A. & ROSENDAL C. (2001) Relations d'équivalence analytiques complètes. *Comptes Rendus de l'Académie des Sciences-Series I-Mathematics*, *333*(10), 903-906.

MAZZIOTTA N. & KAHANE S. (2017) To what extent is Immediate Constituency Analysis dependency-based? A survey of foundational texts. In *Proceedings of the Fourth International Conference on Dependency Linguistics (Depling 2017)*, 116-126.

NIDA E. (1966) *A synopsys of English Syntax*. Mouton and Co., London, The Hague, Paris, 2e édition.

NIVRE J., DE MARNEFFE M.-C., GINTER F., GOLDBERG Y., HAJIC J., MANNING C. D., ... & TSARFATY R. (2016). Universal Dependencies v1: A Multilingual Treebank Collection. Proceedings of *LREC*.

OSBORNE T., PUTNAM M. & GROSS T. (2012) Catenae: Introducing a novel unit of syntactic analysis. *Syntax*, *15*(4), 354-396.

STEEDMAN M. (2014) The Surface Compositional Semantics of English Intonation, *Language*, 90, 2-57.

TESNIERE L. (1959) *Eléments de syntaxe structurale*, Klincksieck, Paris.

VERGNE J. (2000) *Étude et modélisation de la syntaxe des langues à l'aide de l'ordinateur - Analyse syntaxique automatique non combinatoire*, Thèse d'HDR, Université de Caen.