



HAL
open science

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

Sylvain Kahane

► **To cite this version:**

Sylvain Kahane. Une approche mathématique de la notion de structure syntaxique : raisonner en termes de connexions plutôt que d'unités. *Revue TAL : traitement automatique des langues*, 2018, 59 (1), pp.13-37. hal-04067475

HAL Id: hal-04067475

<https://hal.parisnanterre.fr/hal-04067475v1>

Submitted on 11 May 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

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

Sylvain Kahane

*Modyco, Université Paris Nanterre et CNRS
sylvain@kahane.fr*

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. A mathematical approach of the notion of syntactic structure: reasoning in terms of connections rather than units.

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, combinaison syntaxique, dépendance, constituance, relation d'équivalence, graphe à bulles.

KEYWORDS: syntactic connection, syntactic combination, dependency, constituency, equivalence relation, bubble graph.

Préambule. Cet 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 des 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, cependant 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

combinaisons, puis de connexions. 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 connexions. 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, en tirant les conséquences de notre étude en ce qui concerne la notion de structure syntaxique et le statut des diagrammes syntaxiques les plus utilisés, les arbres de constituants et les arbres de dépendance.

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 1 qui représente une analyse en constituants immédiats de la phrase (1). Cet arbre indique que *le* et *chat* forment ensemble une unité *le chat* ou, de manière équivalente, 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 former 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 $\text{Sentence} \rightarrow \text{NP} + \text{VP}$ est représentée par un lien entre NP et VP indiquant que ces deux constituants se combinent entre eux, puis par un petit lien entre ce premier lien et Sentence indiquant que le résultat de la combinaison donne Sentence. La structure n'est donc pas un graphe¹, mais un

¹ Comme l'ont montré Mazziotta et Kahane (2017), il ne semble pas y avoir eu de représentation des analyses en constituants immédiats sous forme d'arbre (c'est-à-dire de graphe pointé connexe et acyclique) avant Chomsky (1956) et toutes les représentations proposées avant cela mettent en avant la connexion. Le diagramme proposé par Chomsky (1957), réifiant les connexions, s'inscrit donc parfaitement dans la tradition de son époque.

polygraphe (Kahane et Mazziotta, 2015), où une arête a pour sommet une autre arête. On notera qu’une telle représentation, qui encode explicitement les connexions (comme la connexion entre NP et VP), n’est possible que pour des grammaires qui n’ont que des décompositions binaires².

(1) *Le chat a vu Zoé.*

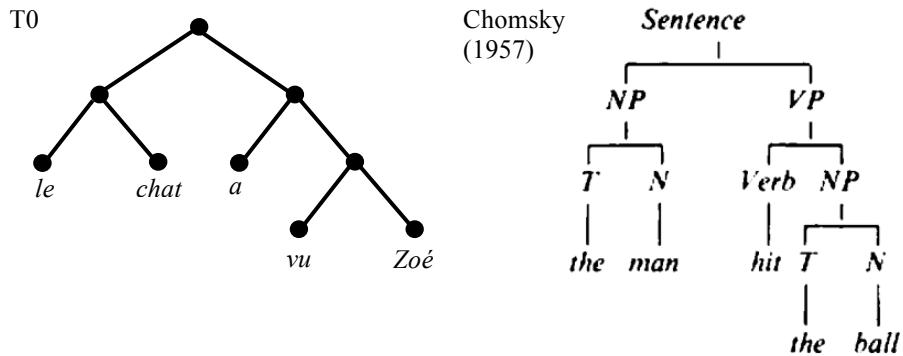


Figure 1. Arbre de constituants binaire (T0) et diagramme de Chomsky (1957)

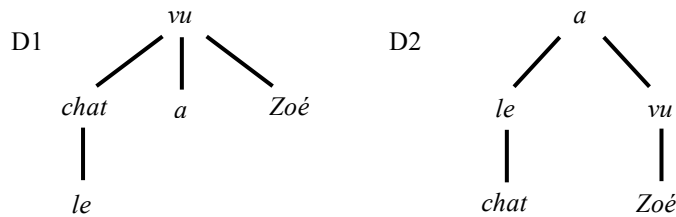


Figure 2. Arbres de dépendance à la UD (D1) et à la Hudson (D2)

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 [UD] (Nivre *et al.*, 2016), où les liens entre les mots pleins sont

² 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 combinaisons binaires d’unités : autrement dit, les deux constituants immédiats d’une unité donnée se combinent entre eux pour former cette unité. L’interprétation des arbres plats à branchements multiples est différente. Dans une phrase comme *Pierre donne un livre à Marie*, si l’on considère $VP \rightarrow V NP PP$, on ne veut pas dire que NP et PP se combinent entre eux, mais bien que c’est V qui se combine avec NP et avec PP. La règle $VP \rightarrow V NP PP$ n’est donc réellement interprétable que si l’on ajoute que V est la tête de VP et l’on voit alors que ce branchement encode en fait deux combinaisons, V avec NP et V (ou V NP) avec PP.

privilegiés, 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.

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 lier des bulles qui possèdent elles-mêmes 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.

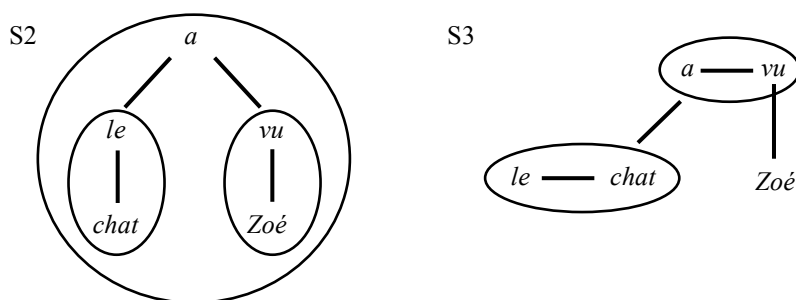


Figure 3. Un « arbre » à la Beauzée-Gladkij et un stemma à la Tesnière

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 $chat$, tandis que $Zoé$ dépend de vu seul et non du nucléus $a vu$.³

³ 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 retenue 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

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 de 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.

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 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 termes d'unités, mais plutôt en termes 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 à la UD 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 (ce que sous-entend la décomposition $IP \rightarrow DP + I'$ (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*, entre une unité verbale et son sujet, et que cette connexion est instanciée de différentes façons selon les théories 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 que l'on peut en faire.

éléments du nucléus pour bien indiquer qu'ils sont combinés. Voir également Kahane et Osborne (2015) pour l'interprétation des stemmas et une représentation en termes de polygraphe.

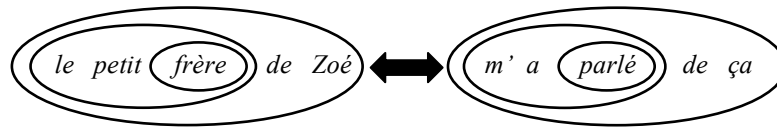


Figure 4. Différentes instanciations de la connexion subjectale

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 repose 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, en particulier, d’hypothèse de continuité. Les unités sont des suites (éventuellement discontinues) d’unités élémentaires, que nous traiterons comme des ensembles d’unités élémentaires (en particulier, nous considérerons l’intersection ou la réunion de deux unités). Dans cet article, nous considérons que les unités élémentaires d’une phrase sont les mots⁴, 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, quel que 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 l’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)⁵. 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 et Mazziotto, 2015), mais toutes les conséquences n’en ont pas été tirées. Nous représentons une combinaison par une paire (non ordonnée)⁶ de

4 Il s’agit plus précisément des occurrences de mots (les différentes occurrences d’un même mot doivent être distinguées).

5 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 et Manning, 2008).

6 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

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 non vides 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 et Kahane, 2017, sur l'interprétation de la constituance en termes de connexions). Si nous reprenons l'arbre de constituants T0 de la section 2, nous avons :

$$C(T_0) = \{ \{le, chat, a, vu, Zoé\}, \{le, chat\}, \{a, vu, Zoé\}, \{vu, Zoé\} \}$$

On peut associer à tout ensemble de combinaisons une représentation, que nous appellerons un *graphe à bulles*, où les unités non élémentaires sont représentées par des bulles et les combinaisons par des arêtes entre les unités. On associe ainsi à $C(T_0)$ le graphe à bulles B0 de la figure 5.

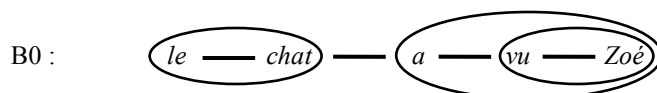


Figure 5. Graphe à bulles associé à $C(T_0)$

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 :

$$C_{\min}(D_1) = \{ \{le, chat\}, \{chat, vu\}, \{a, vu\}, \{vu, Zoé\} \}$$

Avec les mêmes conventions que précédemment, on peut associer un graphe à bulles B1 à $C_{\min}(D_1)$ (figure 6). B1, qui ne contient que des unités élémentaires et donc pas de bulles à proprement parler, est le graphe (non orienté) sous-jacent à l'arbre D1 :

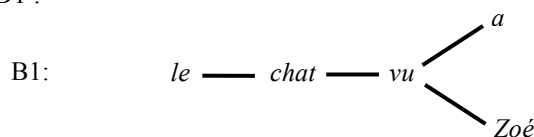


Figure 6. Graphe à bulles associé à $C_{\min}(D_1)$

Plus généralement, à tout diagramme comportant l'identification d'unités (par exemple, par des bulles) et de liens binaires entre les unités, on peut associer l'ensemble de combinaisons formé de toutes les paires d'unités liées par un lien

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.

binaires. Les liens binaires entre les unités sont généralement représentés soit par des arêtes entre les unités (comme dans le cas des arbres de dépendance ou de structures plus complexes telles que S2 ou S3), soit par l'enchâssement d'unités (comme dans les arbres de constituants). On peut ainsi associer 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 associées 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 prise en compte.

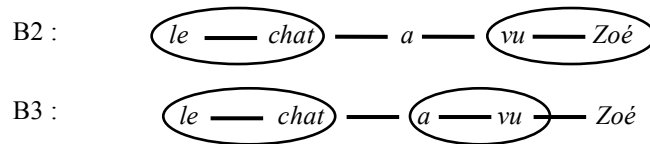


Figure 7. Graphes à bulles associés à $C_{\min}(S2)$ et $C_{\min}(S3)$

5. Notions mathématiques

La suite de cet article fait appel à des notions de théorie des ensembles qu'il nous semble préférable de rappeler. Si E est un ensemble, on note $E \times E$ l'ensemble des couples d'éléments de E . Tout sous-ensemble de $E \times E$ peut être vu 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 ($\forall x,y [xRy \text{ et } yRx] \rightarrow x = y$) et *transitive* si elle se propage systématiquement ($\forall x,y,z [xRy \text{ et } yRz] \rightarrow xRz$).

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 appelés 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 (c'est-à-dire les ensembles inclus dans 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 sont des parties de X , que nous représentons par des suites sur X par souci de lisibilité, bien que l'ordre linéaire ne soit pas pris en compte dans cette étude. Ainsi la suite a vu Zoé représente en fait l'ensemble $\{a, vu, Zoé\}$. Les combinaisons sont des paires d'unités et donc des éléments de $\mathcal{P}(U)$. Les connexions sont des éléments de $\mathcal{P}^2(U)$, c'est-à-dire de $\mathcal{P}^3(X)$.

Pour ceux qui ne sont pas familiers avec les ensembles quotients, il est certainement utile 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 rationnel. De même, les combinaisons sont des *représentants* des connexions.

Revenons sur les relations générales. À 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 :

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

Par exemple, si R est la *relation de dépendance* dans un arbre (xRy ssi y dépend de x), la relation R^* est la *relation de dominance* (xR^*y ssi x domine y). La projection de x (voir section 2) 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* (ang. *partial order*). 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 ($\subseteq = R^*$).

Pour un ordre (partiel ou non), on peut définir plusieurs notions. On appelle *maximum* un élément qui est plus grand que tous les autres et *minimum* un élément

qui est plus petit que tous les autres. E est le maximum de l'inclusion sur $\mathcal{P}(E)$, tandis que l'ensemble vide, noté \emptyset , est le 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.

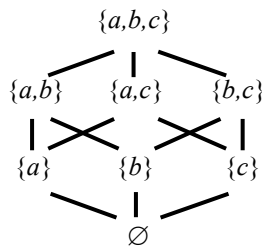


Figure 8. Treillis pour l'inclusion sur $\mathcal{P}(\{a,b,c\})$

Une relation qui est seulement réflexive et transitive est un *préordre* (Bourbaki, 1939). Si la relation \leq est un préordre, elle définit naturellement une relation d'équivalence \equiv par la relation suivante : $x \equiv y$ si et seulement si $x \leq y$ et $y \leq x$. On peut alors projeter la relation \leq sur l'ensemble quotient E_{\equiv} en considérant que deux classes d'équivalence X et Y vérifient $X \leq Y$ si pour tous éléments x de X et y de Y , on a $x \leq y$. La relation \leq est 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* :

$$\{A,B\} \approx \{A',B'\} \text{ ssi } \begin{array}{l} A \cap A' \text{ et } B \cap B' \text{ sont non vides} \\ \text{et } A \cup A' \text{ et } B \cup B' \text{ sont disjoints.} \end{array} \text{ } ^7$$

La première condition ($A \cap A'$ et $B \cap B'$ non vides) assure que $\{A \cap A', B \cap B'\}$ est encore une combinaison. La deuxième condition ($A \cup A'$ et $B \cup B'$ disjoints) assure que $\{A \cup A', B \cup B'\}$ est également une combinaison. Autrement dit :

$$\{A,B\} \approx \{A',B'\} \text{ ssi } \{A \cap A', B \cap B'\} \text{ et } \{A \cup A', B \cup B'\} \text{ sont également des combinaisons.}$$

Dire que deux combinaisons sont compatibles revient à dire qu'elles peuvent représenter la même connexion, en l'instanciant chacune à leur façon. $\{A \cap A',$

⁷ Rappelons que, par définition de la notion de combinaison, on a aussi A et B disjoints, ainsi que A' et B' .

$B \cap B'$ et $\{A \cup A', B \cup B'\}$ seront également des instanciations possibles de cette connexion. On a en fait :

$$\{A, B\} \approx \{A', B'\} \approx \{A \cap A', B \cap B'\} \approx \{A \cup A', B \cup B'\}.$$

On peut représenter la relation de compatibilité par la configuration de la figure 9.

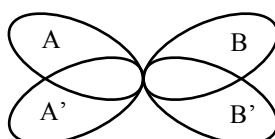


Figure 9. Deux combinaisons $\{A, B\}$ et $\{A', B'\}$ compatibles

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

$$c0 = \{le\ chat, a\ vu\ Zoé\}, c1 = \{chat, vu\} \text{ et } c2 = \{le, a\}.$$

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

On peut éviter la non-transitivité en ne prenant en compte qu'une partie des combinaisons. La relation \approx est une relation d'équivalence sur de nombreux sous-ensembles \mathcal{E} de combinaisons. 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 syntaxiques. 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'ont les arbres de constituants et les arbres de dépendance dans ce paysage.

Reprenons les exemples introduits dans la section 4 et considérons maintenant l'ensemble de combinaisons $\mathcal{E}4 = C_{\min}(D1) \cup C(T0) = \{ \{le, chat\}, \{chat, vu\}, \{le\ chat, a\ vu\ Zoé\}, \{a, vu\}, \{a, vu\ Zoé\}, \{vu, Zoé\} \}$. La relation \approx est bien une relation d'équivalence sur $\mathcal{E}4$. Les classes d'équivalence sont :

$$\begin{aligned} C0 &= \{ \{le, chat\} \} \\ C1 &= \{ \{chat, vu\}, \{le\ chat, a\ vu\ Zoé\} \} \\ C2 &= \{ \{a, vu\}, \{a, vu\ Zoé\} \} \\ C3 &= \{ \{vu, Zoé\} \} \end{aligned}$$

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. Ensembles 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* par (Osborne *et al.*, 2012). Les *catenae* de $D1$ sont :

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

À la suite de Gerdes et Kahane (2013), 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 $A \cup B$ 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 $\{A,B\}$ de $C_{\max}(D)$ 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, \approx est une relation d'équivalence sur cet ensemble. En fait, les $C_{\max}(D)$ sont les plus gros ensembles sur lesquels \approx est une relation d'équivalence.

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

$C'0 = \{ \{le, chat\}, \{le, chat vu\}, \{le, chat a vu\}, \{le, chat vu Zoé\}, \{le, chat a vu Zoé\} \}$

$$\begin{aligned}
C'1 &= \{ \{chat, vu\}, \{le chat, vu\}, \{chat, a vu\}, \{chat, vu Zoé\}, \{le chat, a vu\}, \\
&\quad \{le chat, vu Zoé\}, \{chat, a vu Zoé\}, \{le chat, a vu Zoé\} \} \\
C'2 &= \{ \{a, vu\}, \{a, vu Zoé\}, \{a, chat vu\}, \{a, le chat vu\}, \{a, chat vu Zoé\}, \{a, \\
&\quad le chat vu Zoé\} \} \\
C'3 &= \{ \{vu, Zoé\}, \{a vu, Zoé\}, \{chat vu, Zoé\}, \{le chat vu, Zoé\}, \{le chat a vu, \\
&\quad Zoé\} \}
\end{aligned}$$

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 équivalentes par un diagramme, comme nous le proposons dans la figure 10 pour la connexion subjectale $C'1$ de $C_{\max}(D1)/\approx$.

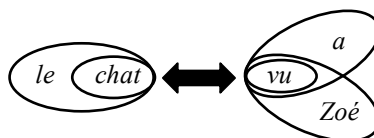


Figure 10. Les combinaisons de la connexion subjectale $C'1$ de $C_{\max}(D1)/\approx$

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 \approx . Tout ensemble de combinaisons 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 d'acyclicité de l'ensemble de combinaisons 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 combinaisons $[\mathcal{E}]_{\min}$ 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 \approx n'exclut pas des connexions telles que $C = \{ \{ab,d\}, \{bc,d\}, \{ac,d\} \}$. 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 $\{A,B\}$ dans C telle que, pour toute combinaison $\{A',B'\}$ dans C , on ait $A \subseteq A'$ et $B \subseteq B'$.⁸ 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 $\text{Unités}(\mathcal{E})$ l'ensemble des unités qui apparaissent dans les combinaisons de \mathcal{E} . \mathcal{E} définit une relation binaire sur $\text{Unités}(\mathcal{E})$ que nous notons \square : $A \square B$ ssi $\{A,B\} \in \mathcal{E}$. Dans le cas où $\mathcal{E} = C_{\min}(D)$ avec D un arbre de dépendance, $\text{Unités}(\mathcal{E})$ est un ensemble de mots et deux mots sont en relation par \square s'ils sont liés par une dépendance. On peut maintenant définir $\text{Catenae}(\mathcal{E})$ comme l'ensemble des unités qui sont des catenae pour la relation \square sur $\text{Unités}(\mathcal{E})$, c'est-à-dire des unités obtenues en réunissant des unités liées les unes aux autres par \square . Dans le cas où $\mathcal{E} = C(T)$ avec T un arbre de constituants binaire, $\text{Unités}(\mathcal{E})$ est l'ensemble des constituants de T et $\text{Catenae}(\mathcal{E}) = \text{Unités}(\mathcal{E})$, 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 A de \mathcal{E} non élémentaire est un catena d'unités strictement incluses dans A . 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 \square associée à \mathcal{E} doit être acyclique, car sinon la relation \approx ne serait 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 $\{ab,c\} \in [\mathcal{E}]_{\max}$. Mais $\{a,c\} \approx \{ab,c\}$, $\{ab,c\} \approx \{b,c\}$ et $\{a,c\} \not\approx \{b,c\}$, ce qui montre la non-transitivité de \approx .

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

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

⁸ 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 $\{b,c\}$.

La connexité et l'acyclicité assure que \approx 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 (figures 3 et 7). La structure S3 est une structure qui sous-spécifie la connexion subjectale en liant *le chat* à *a vu* et qui est ainsi la structure la plus fine qui soit compatible aussi bien avec D1 que D2. S3 est donc une sorte de supremum de D1 et D2. Nous allons formaliser cela.

8. Ordre et équivalence entre ensembles de combinaisons connexes

Nous avons vu 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 . Ceci revient à dire que $C \cup C'$ est un ensemble de combinaisons équivalentes et définit également une connexion. On est tenté de dire que C, C' et $C \cup C'$ définissent plus ou moins la même connexion, mais il ne faut pas oublier que ces connexions restent quand même différentes car C et C' peuvent être de granularité différente (par ex., $C = \{ \{chat, vu\} \}$ et $C' = \{ \{le\ chat, a\ vu\ Zoé\} \}$).

On peut maintenant définir une relation de compatibilité sur les ensembles de combinaisons que nous notons \approx . Deux ensembles de combinaisons \mathcal{E} et \mathcal{F} sont dits *compatibles*, ce que nous notons $\mathcal{E} \approx \mathcal{F}$, si leurs connexions sont deux à deux compatibles par \approx . Cela revient à dire que \mathcal{E} et \mathcal{F} définissent plus ou moins la même structure de connexion. La relation \approx a la propriété suivante :

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

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, $C_{\min}(D) \approx C_{\max}(D)$, 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, $C(T0) \approx C_{\max}(D1)$, car $C(T0) \subseteq C_{\max}(D1)$ et $C(T0)$ possède quatre connexions comme $C_{\max}(D1)$. On a également $C(T0) \approx C_{\min}(D1)$, 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, \approx 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 nous donne, entre autres, les ensembles de combinaisons (incompatibles) $C_{\min/\max}(D1)$ et $C_{\min/\max}(D2)$. 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. Si maintenant, nous prenons un ensemble de combinaisons \mathcal{E} ayant $\{ab,c\}$ comme combinaison minimale et que nous considérons $\mathcal{F} = \mathcal{E} \cup \{a,c\}$ et $\mathcal{F}' = \mathcal{E} \cup \{b,c\}$, nous aurons $\mathcal{E} \approx \mathcal{F}$ et $\mathcal{E} \approx \mathcal{F}'$, mais \mathcal{F} et \mathcal{F}' ne seront pas compatibles entre eux pour \approx . 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. À 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 $\mathcal{E} \approx \mathcal{F}$.

Nous allons donner un sens précis à cette idée de finesse des connexions et de sur et sous-spécification. On peut, à partir de \approx , définir une relation d'ordre \leq telle que \mathcal{E} soit *plus fin* que \mathcal{F} ($\mathcal{E} \leq \mathcal{F}$) 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) :

$$\mathcal{E} \leq \mathcal{F} \quad \text{ssi} \quad \mathcal{E} \approx \mathcal{F} \text{ et } [\mathcal{F}]_{\max} \subseteq [\mathcal{E}]_{\max}.$$

La relation \leq est un préordre (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 \leq la relation \equiv :

$$\mathcal{E} \equiv \mathcal{F} \quad \text{ssi} \quad (\mathcal{E} \leq \mathcal{F} \text{ et } \mathcal{F} \leq \mathcal{E}) \quad \text{ssi} \quad [\mathcal{E}]_{\max} = [\mathcal{F}]_{\max} \quad \text{ssi} \quad [\mathcal{E}]_{\min} = [\mathcal{F}]_{\min}.$$

La relation \equiv 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 arbre de dépendance D .

Dans la figure 11, nous donnons un fragment de la relation \leq . Il s'agit du treillis de relations pour toutes les structures plus fines qu'un arbre de constituants binaire T1 donné. (Par souci de lisibilité, nous n'avons pas mis les trois arbres de dépendance qui sont en dessous de la 4^e structure sous T1 ; on peut facilement les déduire en regardant ceux qui sont liés à la 2^e structure.) Chaque structure S' qui est sous une 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 (1^{re} structure), soit à cd (3^e structure), tandis que la connexion mettant en jeu b peut être attribuée soit à c (2^e structure), soit à d (4^e structure). De ces quatre structures, celle qui donne le plus de possibilités est la 3^e, 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 \leq 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 structures comme le montre la figure 11. On peut aussi se reporter aux exemples présentés au début de cet article : la combinaison $c_0 = \{\textit{le chat, a vu Zoé}\}$ 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 $\{\textit{le chat, a vu}\}$ et qu'un arbre de dépendance comme D1 ou D2 attribue cette connexion à une combinaison encore plus fine (entre 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, les éléments minimaux sont bien des arbres 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 que \mathcal{E} et \mathcal{F} n'ont pas de 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.

L'ordre \leq est déjà introduit dans Gerdes et Kahane (2013) sous une forme plus géométrique. L'objectif des auteurs est de montrer comment, à partir d'un ensemble d'unités syntaxiques assez riche, il est possible de construire une structure de connexion. Pour cela, les auteurs proposent à partir d'un ensemble d'unités U donné de construire n'importe quel arbre de constituants binaire dont les constituants appartiennent à U , 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 $\text{Combi}(U)$. Ce que montre notre article, c'est que $\text{Combi}(U)$ est l'infimum de tous les arbres de constituants sur U . Et si nous voulons faire un diagramme pour $\text{Combi}(U)$, il suffit de prendre la combinaison minimale de chaque connexion, les connexions étant définies grâce à la relation d'équivalence \approx . Nous avons ainsi fourni 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 nombres rationnels : de même que $1/2$ et $50/100$ représentent le même nombre, deux combinaisons équivalentes représentent la même connexion, indépendamment des unités 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 et qui disparaîtront).

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 de 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'unités 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 $\text{Combi}(Y)$ des combinaisons d'éléments de Y . Nous appelons cet infimum 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 flexionnels. 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, lorsque l'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. Commençons par les résultats bruts de cet article. Nous avons introduit trois relations entre les structures de connexion : la relation de compatibilité \approx , la relation d'équivalence \equiv et la relation d'ordre \leq . Un arbre de dépendance D correspond à plusieurs structures de connexion équivalentes : si le diagramme lui-même représente $C_{\min}(D)$, c'est-à-

dire les combinaisons entre mots, c'est en tant que $C_{\max}(D)$ que la structure de dépendance doit être interprétée. On comprend alors que l'arbre de dépendance et les arbres de constituants sont différentes façons d'instancier les connexions de la structure de dépendance (voir figure 12).

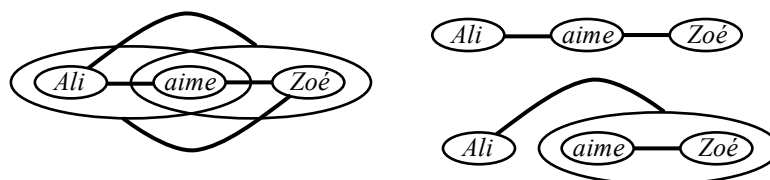


Figure 12. Une structure de dépendance $C_{\max}(D)$ et deux instantiations, par un arbre de dépendance et par un arbre de constituants

Arbres de dépendance et arbres de constituants sont des structures de connexion compatibles, mais pas équivalentes. L'arbre de dépendance instancie les connexions par des combinaisons minimales, des combinaisons entre mots, et l'arbre de constituants par des combinaisons maximales, des combinaisons entre constituants immédiats. On peut les voir comme deux modes de représentation de la structure syntaxique. Nous considérons qu'aucune des deux représentations ne correspond à la structure syntaxique, mais que ces représentations constituent deux vues différentes de la structure, avec des instantiations différentes des connexions. C'est ce qui nous permet de supposer que les connexions sont des classes d'équivalence de combinaisons, qui contiennent aussi bien les combinaisons maximales considérées par les arbres de constituants que les combinaisons minimales considérées par les arbres de dépendance.

Nous n'avons pas encore parlé du rôle joué par la notion de *tête* d'une unité (Bloomfield, 1933 ; Hudson 1984 ; Mel'čuk, 1988 ; Kahane 2001). C'est une notion complémentaire à la notion de connexion, qui permet de raffiner les combinaisons. Dès que l'on a une combinaison $\{a, bc\}$ et que l'on sait que a est la tête de abc et b est la tête de bc , on peut postuler une combinaison $\{a, b\}$ (Gerdes et Kahane 2013). C'est par cette méthode qu'à partir d'un arbre de constituants avec tête, on peut induire un arbre de dépendance (Lecerf, 1961). Cette méthode peut s'appliquer à n'importe quelle structure de connexion intermédiaire entre un arbre de constituants et un arbre de dépendance. Ceci a des conséquences immédiates sur la stratégie pour construire la structure syntaxique, aussi bien du point de vue du TAL que du point de vue de l'enseignement de la syntaxe. Cela signifie que, pour construire la structure, il suffit de construire n'importe quelle structure qui contienne au moins une instantiation de chaque connexion, puis de raffiner, si cela est possible, les différentes connexions en utilisant la notion de tête, afin d'obtenir la structure la plus fine possible.

Du point de vue cognitif, 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 instantiations 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. 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). À notre avis, du point de vue du TAL, cela a des conséquences importantes et signifie que les algorithmes de parsing pourraient prendre en compte le fait que l'on ne cherche pas à construire une instanciation particulière de la structure syntaxique (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 de connexion, quitte à la raffiner ensuite.

Si l'on considère maintenant la génération de texte, ce que nous combinons, ce sont des constructions (locutions, constructions de phrase, etc.) correspondant aux différents éléments de sens que l'on veut exprimer. L'ensemble de ces combinaisons constitue ce que Mel'čuk (1988) nomme la *structure syntaxique profonde*. Les connexions entre ces unités lexicales et grammaticales porteuses de sens vont ensuite être raffinées, quand ces mêmes unités seront réalisées par des combinaisons de lexèmes et de morphèmes flexionnels. On peut donc voir la structure syntaxique profonde comme une instanciation particulière de la structure syntaxique de surface, instanciation où sont regroupées ensemble les éléments formant ensemble une même unité sémantique élémentaire. Il existe encore un domaine du TAL qui considère des instanciations particulières de la structure syntaxique : la traduction automatique. En effet, les stratégies actuelles basées sur des mémoires de traduction consistent à rechercher les plus grandes sous-unités d'une phrase dont on a une traduction et à combiner leurs traductions. Ceci revient à instancier les connexions d'une manière particulière, les unités considérées n'étant généralement pas des constituants.

Revenons sur la nature même de la structure syntaxique. Il est possible que certaines connexions ne puissent pas être instanciées par une combinaison entre mots. Par exemple, considérer que, dans la phrase *le chien dort*, l'unité *le chien dort* possède une tête bien identifiée et que l'on puisse raffiner la combinaison $\{\textit{le chien, dort}\}$ de manière unique est probablement un biais de l'analyse par des arbres de dépendance, qui oblige à instancier chaque connexion par une combinaison entre mots ou, ce qui revient finalement au même, à choisir une tête pour chaque unité. En fait, par leur côté extrême, les deux modes de représentation, dépendance et constituance, présentent des biais. Le biais pour les arbres de constituants est la *stratification* : il faut à chaque étape de l'analyse en constituants immédiats décider quelles sont les deux unités qui se connectent, même lorsqu'il y a plusieurs connexions disponibles. Il n'est pas possible avec un arbre de constituants de dire simplement que, dans *Ali aime Zoé*, *aime* se combine avec *Ali* et *Zoé*. On doit stratifier, c'est-à-dire traiter les connexions dans un certain ordre (en considérant par exemple que *Ali* se combine avec *aime Zoé* qui est lui-même la combinaison de *aime* et *Zoé*). Il en découle qu'il y a autant d'arbres de constituants associés à une structure de connexion donnée que de façons de stratifier l'ensemble des connexions.

Par ailleurs, 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 syntaxique peut être envisagée à différents niveaux de granularité. Nous pensons de surcroît 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 leur 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.

Remerciements. Je remercie François Lareau, Nicolas Mazziotta et les 3 relecteurs pour leurs nombreuses remarques qui m'ont permis, je l'espère, de rendre le texte plus facile à lire.

Bibliographie

- Abney S. P., *The English noun phrase in its sentential aspect*. Thèse de doctorat. Massachusetts Institute of Technology, 1987.
- Abney S. P., « Parsing by chunks », in Berwick R. C., Abney S. P., Tenny, C. (éds), *Principle-based parsing [Computation and psycholinguistics, 44]*, Springer, Dordrecht, 1991, 257-278.
- Beauzée N., « Régime », in Denis Diderot & Jean Le Rond D'Alembert J. (eds.), *Encyclopédie ou Dictionnaire raisonné des sciences, des arts et des métiers*, vol. 14, 1765, 5-11.
- Bourbaki N., *Théorie des ensembles*, Hermann, 1939.
- Bloomfield L., *Language*, The University of Chicago Press, 1933.
- Chomsky N., « Three models for the description of language », *IRE Transactions on information theory*, 2(3), 1956, 113-124.
- Chomsky N., *Syntactic Structures*, MIT Press, Cambridge, 1957.
- Debili F., *Analyse syntaxico-sémantique fondée sur une acquisition automatique de relations lexicales-sémantiques*, Thèse de doctorat d'état, Université Paris Sud, Orsay, 1982.
- de Marneffe M.-C., Manning C. D., « The Stanford typed dependencies representation », *Proceedings of the Workshop on Cross-framework and Cross-domain Parser Evaluation*, COLING, 2008.

- Gerdes K., Kahane S., « Defining dependency (and constituency) », in K. Gerdes, E. Hajičová, L. Wanner (éds.), *Computational Dependency Linguistics*, IOS Press, 2013.
- Gladkij A. V., « On describing the syntactic structure of a sentence » (en russe avec résumé en anglais), *Computational Linguistics*, 7, Budapest, 1968, 21-44.
- Hudson R. A., *Word grammar*, Oxford: Blackwell, 1984.
- Hudson R. A., *Language networks: The new word grammar*, Oxford University Press, 2007.
- Kahane S., « Bubble trees and syntactic representations », in *Proceedings of the 5th conference on Mathematics of Language (MoL)*, 1997, 70-76.
- Kahane S., « Grammaires de dépendance formelles et théorie Sens-Texte », tutoriel, *Actes TALN 2001*, vol. 2, 2001, 17-76.
- Kahane S., Mazziotta N., « Syntactic Polygraphs. A Formalism Extending Both Constituency and Dependency », *Proceedings of the 14th Meeting on the Mathematics of Language (MoL)*, Association for Computational Linguistics, 2015, 152-164.
- Kahane S., Osborne T., « Translators' introduction », in L. Tesnière, *Elements of structural syntax*, John Benjamins, 2015, ixxx-lxiii (49 p.).
- Lecerf Y., « Une représentation algébrique de la structure des phrases dans diverses langues naturelles », *Comptes Rendus de l'Académie des Sciences*, 252(2), 1961, 232-235.
- Mel'čuk I., *Dependency syntax : Theory and practice*, SUNY, New York, 1988.
- Mazziotta N., Kahane S., « To what extent is Immediate Constituency Analysis dependency-based? A survey of foundational texts », *Proceedings of the Fourth International Conference on Dependency Linguistics (Depling)*, 2017, 116-126.
- Nida E., *A synopsis of English Syntax*. Mouton and Co., London, The Hague, Paris, 2^e édition, 1966.
- Nivre J., De Marneffe M.-C., Ginter F., Goldberg Y., Hajic J., Manning C. D., ..., Tsarfaty R., « Universal Dependencies v1: A Multilingual Treebank Collection », *Proceedings of LREC*, 2016.
- Osborne T., Putnam M., Gross T., « Catenae: Introducing a novel unit of syntactic analysis », *Syntax*, 15(4), 2012, 354-396.
- Steedman M., « The Surface Compositional Semantics of English Intonation », *Language*, 90, 2014, 2-57.
- Tesnière L., *Eléments de syntaxe structurale*, Klincksieck, Paris, 1959.
- Vergne J., *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, 2000.