Proc. 5th M. of Mathematics of language (MOL5), Becker & Krieger (eds), Saarbrücken: DFKI, 70-76 (1997).

BUBBLE TREES AND SYNTACTIC REPRESENTATIONS[1]

Sylvain KAHANE[2]

Abstract. A new mathematical object, the bubble tree, is introduced and applied to the syntactic representation of sentences. Two themes are explored: first, the comparison of dependency and constituency models; second, the application of bubble trees to a particular syntactic model.

1. Introduction

One of the major issues in modern linguistics is the lack of a common language among linguists (as opposed to, say, mathematics), which leads to problems of finding correspondences between different idiolects. In particular, two differents models can hide more similarities than it appears at first sight. In Section 4, the two main models of syntactic representation, dependency and constituency, will be compared with the support of a common representational device, the **bubble tree**. Intuitively, bubble trees are trees whose nodes are bubbles which in turn contain sub-bubbles linked to other bubbles and so on. The only formal study of such mathematical structures, as far as I know, is by Gladkij 1968.

Section 5 describes some complex syntactic phenomena, such as coordination, extraction and word order, using representations based on bubble trees. Detailed linguistic descriptions and computational applications cannot be presented in this short communication. They will be the subject of a further communication.

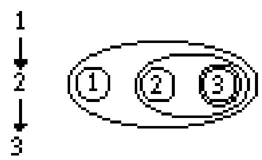
2. Prerequisites

A **tree** can be viewed as an oriented graph or as a binary relation $\lhd$ (in this case we will call it a **tree relation**) ($x \lhd y$ if and only if (y, x) is a link of the corresponding graph). A tree relation induces a **domination relation** $\preceq$ defined by $x \preceq y$ if and only if $x = x_1 \lhd \cdots \lhd x_n = y$ $(n \geq 0)$. The **root** of a tree is the only node which dominates all other nodes. A **terminal node** is a node without dependents.

A **constituency tree** on X (Bloomfield 1933, Chomsky 1957) is a four-tuple $(X, \mathcal{B}, \varphi, \lhd)$ where $\mathcal{B}$ is the set of **constituents**, $\lhd$ is a relation on $\mathcal{B}$ and φ is a map from $\mathcal{B}$ to the non-empty[3] subsets of X (which describes the **content** of constituents) such that:

P1. $\lhd$ is a tree relation.
P2. Any one-element subset of X is the content of one and only one terminal node.
P5. If $\alpha \lhd \beta$, then $\varphi(\alpha) \subseteq \varphi(\beta)$.

A **dependency tree** on X (Tesnière 1934, 1959, Hays 1960, Lecerf 1961) is in fact a plain tree on X.[4] Any dependency tree $(X, \lhd_1)$ induces a constituency tree $(X, \mathcal{B}, \varphi, \lhd_2)$: each node x produces two constituents, noted x and $\bar{x}$ such that $\varphi(x) = \{x\}$ and $\varphi(\bar{x})$ is the **projection** of x, i.e. the set of nodes of X dominated by x (variant: when x is a terminal node x for $\lhd_1$, x and $\bar{x}$ can be identified). The relation $\lhd_2$ is $\lhd_1$ on the bar-constituents and $x \lhd_2 \bar{x}$ for every $x \in X$.



A dependency tree and
the corresponding constituency tree

To give a constituency tree heads (resp. co-heads) means to choose a (resp. a set of) head sub-constituent(s) in each constituent (Pittman 1948). So a **co-headed constituency tree** on X is a quintuple $(X, \mathcal{B}, \varphi, \lhd, \tau)$, where $(X, \mathcal{B}, \varphi, \lhd)$ is a constituency tree and τ a map from $\bar{\mathcal{B}}$ (= the subset of $\mathcal{B}$ of non terminal constituents) to the non-empty subsets of $\mathcal{B}$ such that $\beta \lhd \alpha$ for each $\beta \in \tau(\alpha)$. If $\tau(\alpha)$ has a single element, α is said to be **headed**, otherwise α is said to be **co-headed** or **sub-headed** (in the latter case, α is considered to be a potentially headed constituent for which the head is subspecified). A **head node** of a bubble α is a node obtained in descending from α following only head constituents. The **kernel** of α is the subset of **head nodes** of α.

1. The present paper was read and commented on by Anne Abeillé, David Beck, Dick Hudson and Igor Mel'čuk. I thank them.

2. TALANA (Univ. Paris 7) and Univ. Paris 10 - Nanterre. E-mail: sk@ccr.jussieu.fr

3. That does not mean that we ignore empty constituents. Elements of X are abstract nodes which are associated with signs, which do not need a phonological realisation.

4. In a more general way, dependency trees are trees whose nodes are labelled with grammatical categories and links with functional attributes, but we are not directly concerned with the labelling here.

A headed constituency tree induces a dependency tree (Lecerf 1961, Gaifman 1965, Robinson 1970), but the dependency relation is not explicit. We will introduce equivalent structures to headed constituency trees, which makes dependency relations more explicit.

3. Bubble trees

A **bubble tree** is a four-tuple $(X, \mathcal{B}, \varphi, \triangleleft)$, where X is the set of **basic nodes**, $\mathcal{B}$ is the set of **bubbles**, φ is a map from $\mathcal{B}$ to the non-empty subsets of X (which describes the **content** of bubbles) and $\triangleleft$ is a relation on $\mathcal{B}$ verifying P1, P2[5] and

P3. If $\alpha, \beta \in \mathcal{B}$, then $\varphi(\alpha) \cap \varphi(\beta) = \emptyset$ or $\varphi(\alpha) \subseteq \varphi(\beta)$ or $\varphi(\beta) \subseteq \varphi(\alpha)$.

P4. If $\varphi(\alpha) \subset \varphi(\beta)$, then $\alpha \prec \beta$. If $\varphi(\alpha) = \varphi(\beta)$, then $\alpha \preceq \beta$ or $\beta \preceq \alpha$.

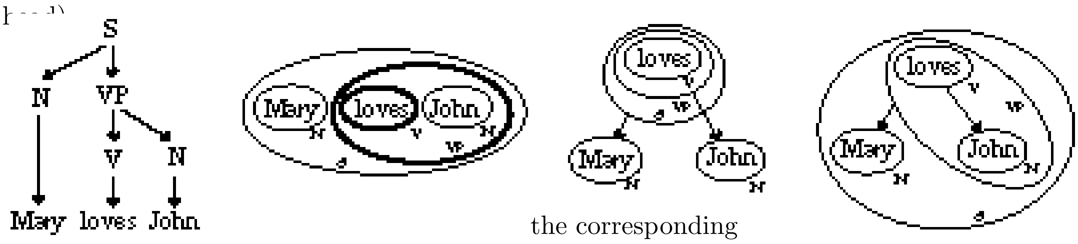
Bubble trees are thus defined.[6] The relation $\triangleleft$ is called **dependency-embedding relation**. Two sub-relations of $\triangleleft$ are considered, the **dependency relation** $\lll$ defined by $\alpha \lll \beta$ if $\alpha \triangleleft \beta$ and $\varphi(\alpha) \cap \varphi(\beta) = \emptyset$ and the **embedding relation** $\circledless$ defined by $\alpha \circledless \beta$ if $\alpha \triangleleft \beta$ and $\alpha \subseteq \beta$. We will say that α **depends** on β if $\alpha \lll \beta$ and α is **directly embedded** in β if $\alpha \circledless \beta$. In the following figures, $\lll$ will be represented by links and $\circledless$ by inclusion of bubbles. The **projection** of a bubble α is the union of the contents of all the bubbles dominated by α, including α.

4. Comparison between dependency and constituency

4.1. Various representations of the same structure

Consider a co-headed constituency tree $B_1 = (X, \mathcal{B}_1, \varphi_1, \triangleleft_1, \tau)$. We associate it with a bubble tree $B_2 = (X, \mathcal{B}_2, \varphi_2, \triangleleft_2)$ where $\mathcal{B}_2 = \mathcal{B}_1$, $\triangleleft_2 = \triangleleft_1$ and $\alpha \circledless_2 \beta$ if and only if $\alpha \in \tau(\beta)$. Such a bubble tree is called a **bi-tree**. Note that $\varphi_2(\alpha)$ is the kernel of α in B_1, and $\varphi_1(\alpha)$ is the projection of α in B_2. Therefore, co-headed constituency trees and bi-trees are in one-to-one correspondence. A bi-tree corresponding to a headed constituency tree is called a **stratified tree**; in this case, every bubble contains a unique element (the head of the constituent). Any stratified tree T trivially induces a dependency tree T'; to obtain T' from T, it is sufficient to collapse all bubbles with the same content.[7]

Example. We will give four representations of an X-bar tree of *John loves Mary* (X-bar trees are headed constituency trees; τ is generally encoded in the node labelling; here τ is represented by lining up any node with its head).



An X-bar tree — the same X-bar tree — the corresponding stratified tree — the corresponding Gladkij tree

Bi-trees can be easily characterised. The bubble β is a **sub-bubble** of α if $\beta \subseteq \alpha$ and $\beta \prec \alpha$. The bubble β is an **immediate sub-bubble** of α if β is a sub-bubble of α but not a sub-bubble of a sub-bubble of α. Note that an immediate sub-bubble of α is either directly embedded in α or depends on another immediate sub-bubble of α. A bubble tree is a **bi-tree** if and only if any immediate sub-bubbles of any bubble α is directly embedded in α. In other words, a link cannot be included in a bubble.

There is another way to encode a headed constituency tree with a bubble tree, but this does not work with a co-headed constituency tree. A headed constituency tree $B_1 = (X, \mathcal{B}_1, \varphi_1, \triangleleft_1, \tau)$ can be associated with a bubble tree $B_3 = (X, \mathcal{B}_3, \varphi_3, \triangleleft_3)$ where $\mathcal{B}_3 = \mathcal{B}_1$, $\varphi_3 = \varphi_1$ and $\alpha \triangleleft_3 \beta$ if and only if either $\alpha = \tau(\beta)$ or there exists γ such that $\alpha \triangleleft_1 \gamma$ and $\beta = \tau(\gamma)$. The resulting bubble tree is called a **Gladkij tree**.[8] A combination of both types of representation will give us hybrid representations of bi-trees and Gladkij trees (for example Vergne's model (Vergne 1994) uses such structures).[9]

---

5. P2 can be weakened to authorize a "lexical" bubble to have descendents.

6. Gladkij 1968 defines a particular case of bubble trees (he supposes in particular that each link must be contained in a bubble and that there must exist a bubble which contains all the nodes). Moreover, $\triangleleft$ and $\lll$ are not clearly distinguished and P1 is only partially stated.

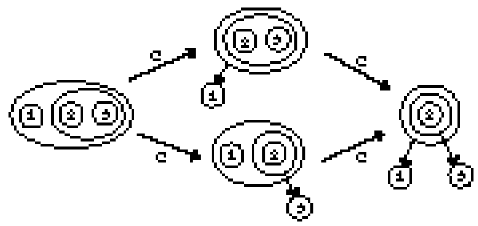
7. If the stratified tree is labelled by categories, the induced dependency tree must be labelled by sequences of categories.

8. This kind of bubble trees are bubble trees in the sense of Gladkij 1968. But Gladkij's formal and linguistic interpretations of these trees are very different from what is offered here.

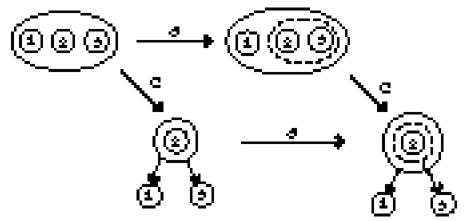
9. A dependency tree $T=(X,\triangleleft_1)$ is said to be **compatible** with a bubble tree $B=(X,\mathcal{B},\varphi,\triangleleft_2)$ if for each link $x \triangleleft_1 y$ of T, there exists a link $\alpha \lll_2$ of B with $x \in \varphi(\alpha)$ and $y \in \varphi(\beta)$. Note that the dependency tree induced by a stratified tree is compatible with it and is also compatible with the corresponding Gladkij tree. Therefore, the two representations can be mixed without problems: the induced dependency tree can be recovered by compatibility.

4.2. Concentration vs stratification, dependency vs constituency

The bi-tree $B_1 = (X, \mathcal{B}, \varphi_1, \triangleleft)$ is said to be more **concentrated** than the bi-tree $B_2 = (X, \mathcal{B}, \varphi_2, \triangleleft)$ if for every α, $\varphi_1(\alpha) \subseteq \varphi_2(\alpha)$. The **concentration** is a partial order on bi-trees. Maximal elements are stratified trees and minimal elements constituency trees. **Concentrating** a given bi-tree consists in changing an immediate sub-bubble β of a bubble α (which has at least two immediate sub-bubbles) into a dependent bubble of α.



To **stratify** a constituency tree consists simply in adding any intermediate constituents. For a bi-tree, this is generally more complicated, because we want to ensure the commutativity between the operations of concentration and stratification. **Stratification** of a given bi-tree consists in adding to some non-terminal bubble α an immediate sub-bubble β and in distributing α's dependencies among α and β. A bi-tree B_1 is said to be more **stratified** than the bi-tree B_2 if B_1 is obtained by stratifying B_2. The stratification is a partial order on bi-trees. Stratification has no maximal elements. Minimal stratified trees are dependency trees or, more exactly, bi-trees corresponding to the headed constituency trees associated with these dependency trees.



Note that concentration does not change stratification and vice-versa, and the two operations commute (see figure above). Clearly, concentration measures the degree of dependency or headedness and stratification, the degree of constituency. Contemporary syntactic models are generally split into dependency models (MTT (Mel'čuk 1988), WG (Hudson 1990), ...) and constituency models (GB (Chomsky 1981), G/HPSG (Gazdar & al. 1985, Pollard & Sag 1994), LFG (Kaplan & Bresnan 1982), TAG (Joshi 1987), CG (Bar-Hillel 1953, Moortgaat 1988), ...). In fact, all these models use dependency (and of course constituency, in the sense that any dependency tree canonically induces a constituency tree) and their respective classification depends on how these models are presented (i.e. whether the dependency relation is explicit or not), not on concentration and stratification. For example, GB is a real dependency model (c-command and government cannot be defined without the notion of head) and it is even more concentrated than Tesnière's model.[10]

There is a third important operation on bubble trees: **granularisation**. The **aggregation** of a bi-tree consists in collapsing together two "adjacent" basic nodes. A bi-tree B_1 is said to be more **granular** than B_2 if B_2 is obtained by aggregating B_1. For example, GB is more granular than other models because it considers two nodes V and $INFL$ where other models consider only the node V. More generally, models that work with morphemes are more granular than models that work with words.

5. A particular model based on bubble trees

We will now defend a particular syntactic representation using bubble trees. Our basic representation is a standard dependency tree which is roughly equivalent to Tesnière's stemma or the Surface Syntactic Structure of MTT, the deep structure of WG, the f-structure of LFG, the derivation tree of TAG, the d-structure of GB or the subcategorization structure of G/HPSG. These theories differ most in the representation of some complex phenomena such as coordination or extraction, to which we now turn.

On the one hand, the **well-formedness** of syntactic structures is controlled, by some general principles such as projectivity, coordination principles, ... (see §§5.1-2) and, on the other, by lexical frames[11] (a **lexical frame**, which is a part of a given lexical entry a, describes an acceptable structural environment for a, that is, the nature and the government (= *régime*) of its arguments or the nature of its head if it is a modifier)[12].

5.1. Coordination

Dependency links are a good way to formalize subordination. But coordination is an orthogonal operation and must be formalize in an orthogonal way. Bubbles offer us a good solution.

10. Tesnière's stemma is not exactly a tree as most linguists think. Some phrases, such as determinant-noun, auxiliary-participle, complementizer-verb ..., are grouped in a bubble, called a **nucleus**. Thus, the stemma is a bubble tree with co-head bubbles.

11. The border line between lexical frames and general principles is not clear: for example, agreement rules can either be general principles directly applied to the structure or can be encoded in lexical frames. Nevertheless, in the latter case, an agreement rule must not be introduced independently in each particular lexical frame, but must be stated by a rule at a meta-level: agreement rules belong to a meta-lexicon, which controls the correctness of elements of the lexicon, i.e. the lexical frames).

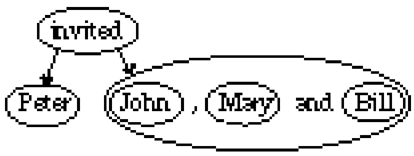
12. It can be supposed that a lexical frame can deal only with adjacent nodes of the node occupied by the lexical unit in question (this is generally called the **locality principle**). In dependency structures, the locality principle assumes that a lexical unit controls only its governor and dependents. In a certain sense, that defines the head; the head of a constituent is the lexical unit which determines the possibilities of insertion of the constituent (Pittman 1948, Garde 1967, Mel'čuk 1988). Nevertheless, these possibilities do not necessarily depend on only one lexical unit and co-heads can be relevant.

Roughly speaking, coordination boils down to the fact that two or more elements together occupy one syntactic position. These elements can be grouped in a bubble, called a **coordination bubble**, which occupies this position. Paradoxically, our description of coordination rests on the notion of head, but cannot be properly encoded in a plain dependency tree.

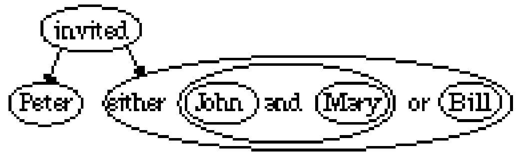
Note that coordination can be developped in two ways:

- the **iterativity** of coordination is the fact that an illimited number of elements can be coordinated and that a coordination bubble can have an illimited number of elements;

- the **recursivity** of the coordination is the fact that coordination bubbles can be coordinated, as in *Peter invited John, Mary and Bill*; the recursivity is linguistically limited to one step and must be well marked (by special words, such as *either*, or prosody).

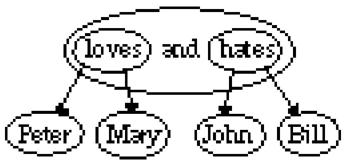


Peter invited John, Mary and Bill
(iterativity of coordination)

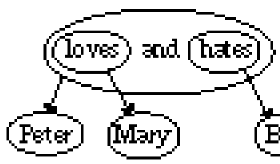


Peter invited either John and Mary or Bill
(recursivity of coordination)

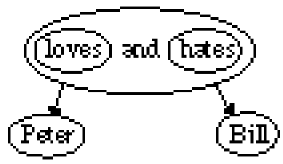
Sharing. Coordinated elements necessarily share their governor (if there is one) and they can share all or part of their dependents. Sharing is constrained: for example, in English, it is easier for two coordinated verbs to share their subject than their object ($^?$ *Mary loves and Peter hates Bill*). The whole sharing (*Peter loves and hates Bill*) is a special case of coordination with particular constraints (it is generally called lexical coordination).



Peter loves Mary and John hates Bill



Peter loves Mary and hates Bill

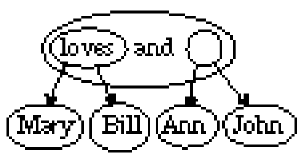


Peter loves and hates Bill

Coordination and lexical frames. A coordination bubble is a special kind of co-headed bubbles. Nevertheless, a coordination bubble requires a particular interpretation. First, the coordination bubble contains two different sorts of objects: coordinated elements on the one hand, and coordinating conjunctions on the other hand. Second, lexical frames apply to coordination bubbles (for example, the verb agrees with its subject which can be a coordination bubble), but also to the coordinated elements by taking into account that the valency (= sub-categorization) of any coordinated element is the union of the valency of every coordination bubble containing it (it is this resulting graph that Tesnière 1959 and Hudson 1990, ∞ adopt as a representation of the coordination). In particular, lexical order rules, such as "the subject is before its governor", apply to a dependent of a bubble just as they do to a dependent of lexical node: thus, *Peter loves and hates Bill*, for example, is the only possible projection of its syntactic structure.

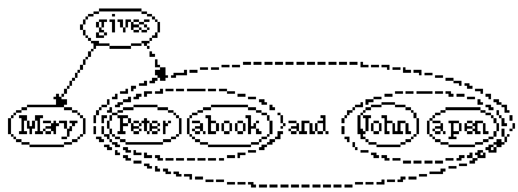
We will now describe two particular kinds of coordination, gapping coordination and valency slot coordination.

Gapping coordination. If two clauses with the same main verb are coordinated, the second occurrence of the verb can be omitted. So we have a verbal bubble with an empty phonological realisation.



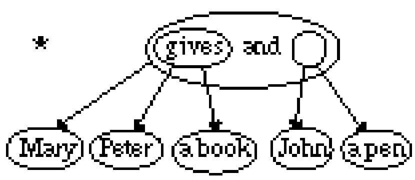
Mary loves Bill and Ann, John

Valency slot coordination. A **valency slot bubble** is a subset of the valency of a governing element grouped in a bubble (with a single link to governor); two valency slot bubbles of the same kind can be coordinated.



Mary gives Peter a book and John a pen

Gapping coordinations and valency slot coordinations are close, and formally they could be represented in the same way: opposite, we propose representing a valency slot coordination in the same way as a gapping coordination. In fact, valency slot coordination are closer to ordinary coordination than gapping coordination; gapping coordination is more constrained. For example, in French, with the coordinating cunjunction *ainsi que*, only valency slot coordination is possible: *Pierre donne un livre à Pierre, ainsi qu'un crayon à Jean* vs ?? *Marie parle à Paul, ainsi qu'Anne à Jean.* The same is true for *as well as* in English: *Peter drinks coffee at 11 as well as tea at 4* vs ?? *Peter drank coffee as well as John tea.*



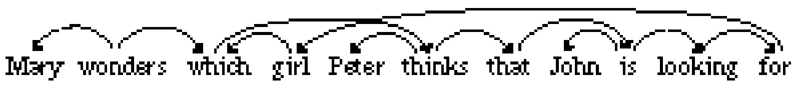
Mary gives Peter a book and John a pen
(as gapping coordination)

On the other hand, gapping coordinations are not valency slot coordinations, because the governing verb agrees with only the first subject. Moreover, our valency slot representation would pose problems for the control of the linear order of gapping coordinations, because we have to assume that the projection of a valency slot bubble, as well as a coordination bubble, must be continuous (see §5.2)

5.2. Word order, projectivity and nuclei

A **dependency** (resp. **constituency**) **structure** is a dependency (resp. constituency) tree on a *linearly ordered* set X. A dependency structure on X is said to be **projective** if links do not cross each other and no link covers an ancestor. A constituency structure is said to be **continuous** if the content of any constituent is continuous. Generally, constituency structures are always supposed to be continuous. Note that a dependency structure is projective if and only if the induced constituency structure is continuous (Lecerf 1961, Gladkij 1966).

Very often, the syntactic dependency tree of a sentence is not projective. The most common type of non-projectivity in English is due to extraction (topicalisation, interrogation, relativisation ...), which we will now study. Following Tesnière 1959 or Hudson ∞, we think that it is better to associate two nodes to a *wh*-word because it assumes two functions: a pronominal function and the same function as other conjunctions, such as *that*, which have the role of subordinating a verb. Both nodes can be combined: we obtain a structure which is not exactly a tree because the *wh*-word has two governors, but which does not make formal problems.



Note that the above "tree" is not projective. Nevertheless, the linear order of such constructions is very constrained and it is not possible to totally renounce projectivity. We need a weaker property which allows all the possible linearisations, but which, combined with the order constraints of the lexical frames, allows only the possible linearisations. Our solution is to use bubble trees.

Roughly, a node of a clause can be **extracted** (= topicalized, relativized, interrogated ...) only if it depends on some main verbal nucleus, where **verbal nuclei** of English are verbs and complex units such as auxiliary−participle (*be eating*, *have eaten*), verb−infinitive (*want to eat*), verb−conjunction−verb (*think that eat*), verb−preposition (*look for*) and all units built by transitivity from these (*thinks that is looking for*).[13] The extractee is a nominal nucleus containing a *wh*-pronoun. **Nominal nuclei** are nouns (*who*) and complex units such as determinant−noun (*which girl*, *whose girl*) and noun−noun complement (*the daughter of which man*). Verbal and nominal nuclei can be represented by bubbles and certain links can or must be allocated to the nucleus such as the governing link of the extractee.

The previous ordered bubble tree represented is projective in a sense that I will now make clear.

A linearly ordered bubble-tree is said to be **projective** if bubblinks do not cross each other and no bubblink covers an ancestor or a co-head (where a **bubblink** is either a bubble or a link).[14]

13. Every language can have nuclei, but each language develops its own proper types of nuclei. For example, in French, the nucleus verb−preposition does not exist (**Marie se demande quelle fille Pierre* ***parle à***), but there exists a nucleus verb−subject and verb−direct object (*l'homme dont la* ***fille dort***, *l'homme dont Pierre* ***aime*** *la* ***fille***) and no other nucleus verb-complement (**l'homme dont Pierre* ***parle à*** *la* ***fille***).

14. The first half of this property is stated in Gladkij 1968.

The significance of projectivity depends on the way nuclei are encoded. Although the two properties are not equivalent, projectivity can be described in this case by saying that the projection of every bubble is continuous.[15]

Attention: nuclear structure does not take the place of dependency structure. The former is simply **superimposed** on the latter and it is to the former that projectivity applies. There is fundamental difference between nuclei and coordination bubbles: the nucleus is a marking of a particular string of dependencies which must be consider from a certain point of view as a whole, whereas the coordination bubble is an orthogonal operation which must not be describe in term of dependencies.

5.3. Nuclei and coordination

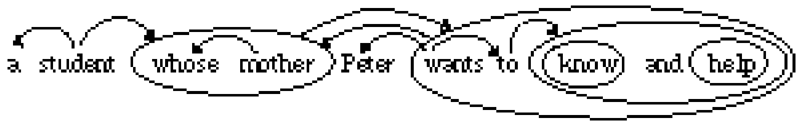
It is well known that there is some constraints between extraction and coordination (Ross 1967):

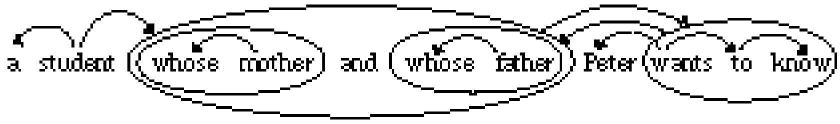
* *a student whose mother Peter knows and helps Mary*

* *a student whose mother and his father Peter knows*

In all the models which I know —constituency or dependency based—, these constructions must be blocked by complicate special constraints.

In our model, possible cases of coordination are of course allowed because a node of a nucleus can naturally be occupied by a coordination bubble:





On the other hand, non acceptable cases of coordination are blocked without additional constraints by our constraint on the extraction (§5.2) and our particular representation of the *wh*-words. Thus, * *a student whose mother Peter knows and helps Mary* is blocked because the nominal nucleus *whose mother* is not a complement of the main verbal nucleus *knows and helps* (but only to *knows*) and * *a student whose mother and his father Peter knows* is blocked because *whose* and *his* do not have the same lexical frame and the two nominal nuclei cannot be coordinated (*wh*-words own very special lexical frame: they have two governors !).

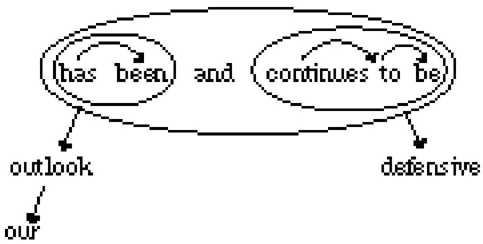
5.4. Coordination and nuclei

The notion of nucleus also allows us to explain some facts of coordination which cannot be described without extending our definition of coordination. We claim that a verbal nucleus can be coordinated with a verb or with another verbal nucleus:

Our outlook ***has been*** *and* ***continues to be*** *defensive.*

a picture that Peter ***likes*** *and* ***is trying to buy****.*

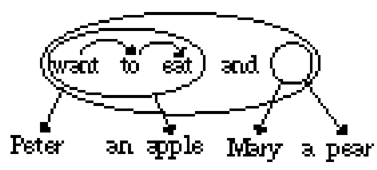
Verbal nuclei in coordination are more constrained than verbal nuclei in extraction: not all kinds of verbal nuclei can be coordinated.



Gapping coordination is possible with verbal nuclei as it is with a verb:

Peter ***wants to eat*** *an apple and Mary, a pear.*

Mary ***is looking for*** *a beautiful landscape and Peter, a pretty girl.*[16]



Nuclei also intervene in a valency slot coordination, in the sense that two nodes can be grouped in valency slot bubble if their governors belong to the same nucleus and are thus coordinated: *John went to London in April and Boston in June.*

Other kinds of nuclei can also be coordinated. For example, one can find in a dictionary this definition: *hedonism:* ***pursuit of*** *or* ***devotion to*** *pleasure.*

15. Projectivity can be ensured in other equivalent ways. For example, all the links of the nucleus can be allocated to the head node (that is what is done by Hudson ∞ with raising or by G/HPSG with the slash feature (see Kahane 1996)), or the links can be labelled as extra- and intranuclear links and projectivity stated in terms of which kinds of links can cut or cover each other (see again Kahane 1996).

16. Such a coordination (with the ellipsis of the preposition) is impossible in French where the nucleus verb-preposition does not exist.

5.5. Conclusion

Constituency structures, as well as plain dependency structures, are too poor: they force us to put in a same dimension subordination and coordination. But subordination and coordination are two orthogonal linguistic operations and we need a two dimentional formalism to capture this, such as bubble trees.

References

BAR-HILLEL Yehoshua, 1953, *A quasi-arithmetical notation for syntactic description*, Language **29.1**, 47-58.

BLOOMFIELD Leonard, 1933, *Language*, New York: Holt, 566 p.

CHOMSKY Noam, 1957, *Syntaxic structures*, La Haye: Mouton, 116 p.

CHOMSKY Noam, 1981, *Lectures on Government and Binding*, Dortrecht: Foris.

GAIFMAN Haïm, 1965, "Dependency systems and phrase-structure systems", *Information and Control* ***8***, 304-337; Rand Corporation, 1961, RM-2315.

GARDE Paul, "Ordre linéaire et dépendance syntaxique: contribution à une typologie", *Bull. Soc. Ling. Paris* ***72.1***, 1-26.

GAZDAR Gerald, KLEIN Ewan, PULLUM Geoffrey K. & SAG Ivan A., 1985, *Generalised Phrase Structure Grammar*, Cambridge, Mass.: Harvard Univ. Press.

GLADKIJ Aleksej V., 1966, *Leckii po matematičeskoj linguistike dlja studentov NGU*, Novosibirsk (French transl: *Leçons de linguistique mathématique, fasc. 1*, 1970, Paris: Dunod).

GLADKIJ Aleksej V., 1968, "On describing the syntactic structure of a sentence" (in Russian with English summary), Computational Linguistics **7**, Budapest, 21-44.

HAYS David, 1960, "Grouping and dependency theory", Rand Corporation, RM-2646.

HUDSON Richard A., 1990, *English Word Grammar*, Oxford: Blackwell.

HUDSON Richard A., ∞, "Discontinuity", e-preprint (ftp.phon.ucl.ac.uk).

JOSHI Aravind K., 1987, "Introduction to Tree Adjoining Grammar", in A. Manaster Ramer (ed.), *The Mathematics of Language*, Amsterdam: J. Benjamins, 87-114.

KAHANE Sylvain, 1996, "If HPSG were a dependency grammar ...", *Proc. 3rd TALN Conf.*, Marseille, 45-49.

KAPLAN Ronald M. & BRESNAN Joan, 1982, "LFG: a formal system for grammatical representation", in J. Bresnan (ed.), *The mental representation of grammatical relations*, Cambridge, Mass.: MIT Press.

LECERF Yves, 1961, "Une représentation algébrique de la structure des phrases dans diverses langues natuelles", *C. R. Acad. Sc. Paris* ***252***, 232-234.

MEL'CUK Igor, 1988, *Dependency syntax: theory and practice*, Albany, NY: State Univ. Press NY.

MOORTGAT 1988, *Categorial investigations*, Dordrecht: Foris.

POLLARD Carl & SAG Ivan A., 1994, *Head-Driven Phrase Structure Grammar*, CSLI series, University of Chicago Press.

ROBINSON Jane J., 1970, "Dependency structures and transformational rules", *Language* ***46****.2*, 259-85.

ROSS John, 1967, "Contrainst on variables in syntax", Doctoral dissertation, MIT (Published as *Infinite syntax !*, Norwood, N.J.: Ablex, 1986).

TESNIÈRE Lucien, 1934, "Comment construire une syntaxe", *Bull. Fac. Lettres Strasbourg* ***7***, 12ème année, 219-229.

TESNIÈRE Lucien, 1959, *Éléments de syntaxe structurale*, Paris: Kliencksieck, 670 p.

VERGNE Jacques, 1994, "A non recursive sentence segmentation applied to parsing of linear complexity in time", *Proc. 1st Int. Conf. on New Methods in Lang. Proc.*, Manchester.