UvA-DARE (Digital Academic Repository)

Dependency and valency

From structural syntax to constructive adpositional grammars

Gobbo, Federico; Benini, Marco

DOI
[10.3233/978-1-61499-352-0-113](10.3233/978-1-61499-352-0-113)
[10.3233/978-1-61499-352-0-113](10.3233/978-1-61499-352-0-113)

Publication date
2013

Document Version
Author accepted manuscript

Published in
Computational Dependency Theory

Link to publication

Dependency and Valency: From Structural Syntax to Constructive Adpositional Grammars

Federico GOBBO [a] and Marco BENINI [b]

[a] *DISIM – Dept. of Inf. Eng., Comp Sc. and Maths, University of L'Aquila, Italy*
[b] *DiSTA – Dept. of Theoretical and Applied Sciences, University of Insubria, Italy*

Abstract. In linguistics, the influence of the results by Tesnière cannot be underestimated: above all, he introduced the concepts of dependency and valency, which had a considerable influence in the development of linguistics by the second half 20th century. However, his Structural Syntax remains still uninvestigated in most parts: in particular, there is still no grammar formalism directly inspired by it, that is suitable for theoretical and practical applications in Artificial Intelligence and computational linguistics. The aim of this paper is to fill this gap, in proposing a formal grammar that adopts Tesnière's intuitions and concepts of Structural Syntax – as much as possible – in adapting them to the needs of contemporary linguistics, notably natural language processing. The result of this modelling process is a new formalism derived from Tesnière's, where natural language grammars are expressed in *constructive* mathematical terms (and therefore suitable for computational treatment) where the abstract notion of *adposition* is of the greatest importance. For these reasons, they are called Constructive Adpositional Grammars (CxAdGrams). This paper explains the linguistic and formal reasons of CxAdGrams, with a special regard to the heritage of Tesnière and its relations with existing dependency grammar formalisms in terms of similarities and differences.

Keywords. dependency grammar, structural syntax, adposition, constructive mathematics, constructive adpositional grammar

1. Introduction

Although "somewhat removed from the European structuralist mainstream" [2, p. 195], the work by Lucien Tesnière [1] in linguistics is acknowledged to be central by researchers in dependency linguistics – being member of the Prague circle, he influenced in particular the so-called 'second' Prague school [3]. In fact, it was the French linguist who introduced, in modern times, the key concepts of dependency and valency. Nonetheless, unlike valency, there is no agreement among scholars and specialists on how to treat precisely the concept of dependency. In fact, even if the theoretical assumption behind all dependency-based models is fairly the same, namely "the syntactic structure of sentences resides in binary asymmetrical relations holding between lexical elements" [4, p. 6], in practice the use of this assumption is different among the authors and the formalisms proposed thereby. This paper shows how the Tesnièrian notion of dependency – together with the notion of valency – can be taken up again in a grammar formalism which re-

spects the word classes used in Tesnière's Structural Syntax and is mathematically and computationally sound.

The paper is structured as follows: in Section 2 the most significant dependency-based grammar formalisms are presented, so to investigate how the Tesnièrian notion of dependency is interpreted by different authors. Section 3 collocates the concept of dependency in Structural Syntax, while Section 4 explains why word classes are so important and how to use them (again) in grammar formalisms. After the introduction of valency in Section 5, adpositional trees can be introduced: they are the basic structure of a new formalism, i.e., constructive adpositional grammars (CxAdGrams), that takes account of most concepts of Structural Syntax, namely, dependency, valency, syntactic functions, word classes and transference. Two different representations of adpositional trees are given, so to simplify the comparison with the dependency-based formalisms already presented in Section 2. Section 6 is entirely devoted to transference, one of the most original contribution introduced by Tesnière: in CxAdGrams, transference is used to put different constructions into relation. Finally, Section 7 explains the fundamentals of the formal model behind CxAdGrams, expressed in terms of category theory. In the Conclusion the influence of Structural Syntax in CxAdGrams is evaluated.

2. Related works in dependency and valency

The fortunes of dependency-based grammar formalisms changed over time. In fact, during the years of the 'linguistic wars' [5], grammar analysis was mainly based on the Bloomfieldian tradition of immediate-constituent analysis, strengthened by the formalism introduced by Chomsky's generativism: dependency-based grammars were considered outsiders, differences were underlined over similarities. Luckily, this period is over: a sophisticated formalism spread out from generativism, i.e., Head-driven Phrase Structure Grammar (HPSG), was recently presented as a possible formalisation of an extended dependency-based approach [6]. Furthermore, there is a growing interest by computational linguists, because of the availability of corpora such as the Prague Dependency Treebank 2.0 [7], so that parsing of dependency-based treebanks has been explored seriously in the last years [8].

There are many formalisms based on the notion of dependency, all having three features in common: first, relations are binary and asymmetrical (i.e., with a governor and a dependent); second, relations are stated normally between words as the final unit of analysis – never among groups of words (phrases), but sometimes among parts of words (morphemes); third, all formalisms foresee different levels of analysis – in other words, different types of dependencies. Here, we have taken into consideration the more used dependency-based formalisms in order to show similarities and differences with CxAdGrams, which will be presented in the following Sections. It is important to note that the grammar formalisms are analysed *only* under the perspective of dependency and valency, because they are the central notions introduced by Tesnière that are commonly used: a complete survey is out of the scope of this paper. These formalisms are: Functional Generative Description (FGD); Meaning-Text Theory (MTT); eXtensible Dependency Grammars (XDG); Word Grammar (WG).

2.1. Functional Generative Description

The Prague Dependency Treebank 2.0 [7] is based on the FGD, perhaps the oldest-in-use dependency-based grammar ever, being in use at least since 1969 [9]. FGD is based on Czech, a Slavic language which shows a high degree of freedom in word order (variations often indicating topic-focus articulation) and a considerable amount of morphology as well: traditionally, constituent-based grammars based on Chomsky failed to give reasonable account of these two phenomena, which are widely attested in many languages of the world. FGD assigns a valency value to verbs and have a complex use of dependency. In fact, FGD distinguishes six levels of abstractions – called layers of analysis: phonological, morphematical, morphonological, analytical, tectogrammatical. In particular, the analytical layer represents the surface syntax (i.e., explaining the word order), while the tectogrammatical layer represents the deep syntax (i.e., explaining the 'propositional meaning'). As already underlined by Starosta, this choice is motivated by the desire to have a level of analysis free from functional words (such as adpositions) so to motivate topic-focus articulation: eventually, this choice causes a loss of economy and elegance of the whole model, [10, p. 532]. At the moment, CxAdGrams do not have a level of abstraction for phonetics, phonology and phonotactics: under this perspective, FGD are more expressive and powerful. However, we feel that these levels of analysis are governed by different rules, compared with morphology and syntax, which are treated as two different linguistic phenomena explained within a *single* mechanism called construction – represented through adpositional trees. As we will show below, CxAdGrams take into account topic-focus articulation in adpositional trees too.

2.2. Meaning-Text Theory

MTT also originates in the 1960, by Žolkovskij and Mel'čuk, during a research in machine translation in Moscow. After ten years, a nucleus for Russian was formed, while in the late 1980s Mel'čuk formed a research group at the University of Montreal, where the 'modern' form was defined around French [11]. Afterwards, MTT was improved but not drastically changed. MTT postulates two levels: a *textual* level, defined by phonetics (ability of speaking), and a *meaning* level (ability of speech understanding). MTT is aimed to describe the interface from meaning to text, as the name suggests, as a many-to-many relation (representing synonymy, homonymy, polysemy) caught by a formal language – technically, a cascade of transducers. Semantics, syntax, morphology and phonology are the main levels of analysis; with the important exception of phonology, MTT postulates three types of of syntagmatic dependency relations: semantic dependency, syntactic dependency and morphological dependency. Furthermore, each level of analysis is split in two, the first one being text-oriented while the second is meaning-oriented, with semantics being the notable exception. Six modules are obtained in total, their interfaces being guaranteed by the formal model. Semantics is represented as a whole in form of graphs, where semantic dependencies are the arcs between the words, and their labels are numbered according to the actants – in that way, MTT represents valency. Nodes are semantemes, which should be decomposed until reaching the semantic atoms or primitives. On the contrary, CxAdGrams are agnostic in respect to the world model, as there is no agreement about the number and the properties of the semantic primitives, and there cannot be either. In fact, it is the history of a priori 'perfect' lan-

guages that shows that taxonomies (or ontologies, in the jargon of computer scientists) are doomed to be transient and local. Moreover, it is a *long* story: it starts at least since 1668, when John Wilkins published its Real Character [12]. This is the reason why there is *no* apart level of abstraction for semantics, interpreted in terms of semantemes and their relations. Another point of difference is that CxAdGrams consider morphosyntax as a whole, while MTT devotes four modules, which is undoubtedly more complex. However, MTT has two distinctive advantages. On one hand, the formal model is very accurate both at a mathematical and at a computational level and henceforth MTT formalisms can not only be presented as transducers and logic but also compared with other linguistic formalisms on a strong basis. On the other hand, the equivalence between a surface-syntactic tree and the linearly ordered tree was a source of inspiration in depicting the dependency adtrees in our model – see below for details.

2.3. Extensible Dependency Grammars

XDG is the more recent dependency-based formalism that we will analyse here. In fact, its roots can be found in 2001, with Topological Dependency Grammar, proposed by Ralph Debusmann as a declarative formalism for dependency grammars [13]. Here, 'topology' refers to three fields, i.e. forward, in the middle, and backward (in German: *Vorfeld*, *Mittelfeld*, *Nachfeld*); this notion is often used to describe Dutch and German grammars, so to explain distant dependencies with a *divide et impera* strategy. As a result, the notion of valency is split in two: on one side, there is the syntactic valency, which more or less adheres to the original Tesnièrian vision; on the other hand there is a topological valency, which describes valency in terms of fields, that can be obligatory or optional. In particular, the middle field can be expanded from zero to *n*, so to represent the phenomenon of centre embedding. In parallel, the notion of dependency is split in two: the surface dependency is topological (in the sense explained above) while the deep dependency shows the syntactic relations. Both dependencies are represented in terms of trees. Research went further: in 2006 Debusmann proposed XDG, a framework derived from Topological Dependency Grammar, with a strong orientation to practical use especially in computational settings. In fact, an ad-hoc software for linguistic annotation is also provided [14]. XDG was built with the other grammar models, so to integrate dependency-based frameworks and lexicalized approaches such as HPSG in general [15]. As the name suggests, XDG let notions such as valency and dependency to be split not in two but in *i* dimensions: the result is not a tree-oriented representation, but dependency graphs, where non-projectivity (i.e., the fact that a dependency subtree can entail non-contiguous words, in terms of word order) is banned, essentially because it is intractable if no restrictions are required. Another notable point is that the three fields of the ancestor (*Vorfeld*, *Mittelfeld*, *Nachfeld*) are not mandatory. The main advantages of XDG are two: first, its strong modularity; second, the fact that it is substantially agnostic, i.e., different grammar theories can be put in XDG, with the proviso of dependency and valency. In contrast, CxAdGrams are not modular, being focused on morphosyntax and pragmatic reflexes in terms of marked word order, and they are supported by a strong grammar theory, that will be presented in the sequel of this paper.

2.4. Word Grammar

WG was developed since the 1980s [16] and it was subjected to many revisions, until the last paradigmatic presentation, interestingly called 'language networks' by Hudson – therefore, dependency is represented in graphs, not trees [17]. A textbook for undergrad is also available [18]. WG is different from the formalisms presented above, and it is arguable if WG can be counted as a dependency-based formalism, although it is normally presented as such [19]. In fact, unlike the other formalisms just presented, in WG a dependent can have *more* than one governor. For instance, in constructions as English verbal gerunds, such as *John is walking*, where *is* and *walking* are considered both depending on *John*, for semantic reasons. Furthermore, WG allows non-projectivity, in order to give account of structure sharing such as coordination, which is a known problem in dependency-based formalisms, especially for parsing; the number of intervening words is called dependency distance [18], p. 151. The price in computational terms is high: it is not clear how language networks generated by WG can be parsed. WG retains the usual notion of valency, but introduces the term 'valent' in order to indicate the elements that saturates valency – in CxAdGrams, we retain the Tesnièrian term 'actant', which is widely used across different traditions with this meaning. Analogously, extravalent elements are called 'adjuncts'. In CxAdGrams, 'adjuncts' indicate the modifiers of actants and extravalent statives such as nouns or pronouns, while modifiers of the verbants are called 'circumstantials': in other terms, WG-adjuncts are both adjuncts and circumstantials in CxAdGrams (see below for details). Although WG spread out from syntax as the main level of analysis, it was applied by to morphology and semantics too, and it takes into account pragmatic data such as 'event', 'speaker', 'time' and so forth, which resembles functions – in the sense of functionalism in linguistics. This choice is motivated by the fact that WG is a branch of cognitive grammar whose main tenet is that language is "only" a part of general cognition, which essentially functions as a network of associations. CxAdGrams also adopt the view of cognitive grammarians that language is part of cognition, not a distinguished "faculty", since its transitional version, called adpositional grammars (AdGrams) [20]. In CxAdGrams, dependents can have only one governor, as in most dependency-based framework. The need of multiple governors is solved by means of the mechanism of adtree transformations – for example, gerunds are packed constructions that hide more explicit constructions, where dependencies are clearly stated. Another important point of difference is the admissibility of zero morphs, i.e. no explicit morphemic signifier: unlike many other dependency-based models, CxAdGrams do not suffer of *horror vacui*.

3. The structure of adpositional trees

A direct formalisation of Structural Syntax, which was the first hypothesis considered by the authors, is simply not possible, essentially for two reasons, the first one being theoretical, the second technical. The theoretical reason is rooted in the nature of the most relevant publication of the French linguist. The *summa* of Tesnière's work is a posthumous book where his language analysis system is explained in detail, being approximately 600-pages long [1]. His system, called Structural Syntax[1] is very reach in details, but

---

unfortunately not always coherent in all parts, as we will show below. In the book, every paragraph is numbered referring to a Chapter that belongs to an internal Book (from A to F) belonging to a Part (from 1 to 3). In the sequel, references to that work will take the original form. For instance, paragraphs 1–8 of Chapter 21 belonging to Book A of Part 1 will be referred like this: (1, A, ch. 21, par. 1–8). Analogously, it was decided to retain the original numbers of Tesnière's examples (*stemma*) in order to help the reader in the comparison between Structural Syntax and the model presented in this paper, while new examples are numbered through capital letters.

The second reason of the difficulties of a straightforward formalisation is technical. Tesnière, working in the 1930-50 years, was a son of his time: he could take advantage of the fruits of the great tradition of linguistic structuralism that spread out in francophone Europe in the first half of the past century, but on the other hand he could not have the proper formal and mathematical instruments to be applied to his linguistic results, not to mention computational instruments – computational linguistics in the 1950s was in its infancy stage. In particular, CxAdGrams is formally based on category theory (see Section 7 for details), introduced by Samuel Eilenberg and Saunders Mac Lane in the 1940s: in those times it was not mature enough for linguistic applications. For these two reasons, linguistics in the second half of the 20th century adopted the concept of dependency and valency, but generally did not use Structural Syntax as such – an exception being a representation of Japanese for computational purposes in the late 1990s, whose influence has been scarce [21].

In spite of a direct formalisation of Structural Syntax, the authors decided to retain as much as possible the linguistic concepts by Tesnière, and at the same time to let themselves free to use modern mathematical and computational tools, for the necessary adaptation: the results is CxAdGrams. For the linguistic part, the starting point was the notion of dependency. How did it all begin? In other words, how Tesnière *really* defined dependency? What can be saved and adapted?

Figure 1. How connection works for Tesnière (original image from Tesnière 1959)

At the beginning of the book, in (1, A, ch. 1, 4-12), Tesnière introduces the notion of connection (*connexion*). Figure 1 shows the examples used by the French linguist to introduce this key concept: in *Alfred parle* ('Alfred speaks'), the verb *parle* is the governor (*régissant*), the noun *Alfred* being the dependent (*élément subordonné*). Their relation, "indicated by nothing" (1, A, ch. 1, 4) is their connection. Connections can be recursive, as in example 2 *mon ami parle* ('my friend speaks'): governors are put above, dependents are put thereunder. It is interesting to note that Tesnière did not use the word 'dependency' (*dependance*) but 'connection', even if the asymmetry in the relation between

the governor and the dependent is obvious[2]. In other words, the Tesnièrian notion of 'connection' corresponds to what now linguists call 'dependency'. This choice becomes clear when the dichotomy 'structural *vs.* semantic influence' is introduced by the author (1, A, ch. 2, 3).
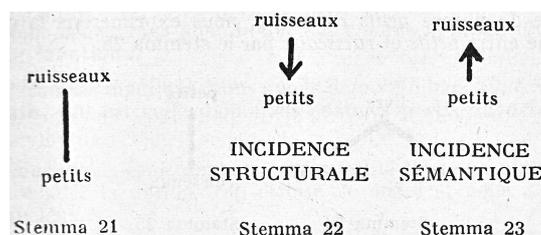

Figure 2. Structural and semantic influence (original image from Tesnière 1959)

Figure 2 shows that two types of connections are possible: in the example, *petits ruisseaux* ('little streams') can be structurally driven, if the governor *ruisseaux* influences the dependent *petits*; on the contrary, if the dependent is stronger, there is a semantic influence over the governor – usually by grammaticalization, for instance in the proverb *Les petits ruisseaux font les grandes rivières*, 'tall oaks from little acorn grow', literally, "little streams form big rivers". Here, it seems that the French linguist wants to give the apart status of dependency only to semantically influenced connections. Unfortunately, this crucial point is never mentioned anymore throughout the whole book; in fact, only generic, underspecified connections are actually used. Nonetheless, we argue that this point is crucial for the Tesnièrian concept of connection and for dependency as well. In fact, Tesnièrian Structural Syntax shows a triple in order to describe the connection between two linguistic elements: governor, dependent, relation. In sum, connections can be of three types: generic or underspecified (no direction of the arrow), structural (informed by the governor) and semantic (informed by the dependent).

CxAdGrams strictly adheres to the idea of Tesnièrian connection, unlike the other dependency-like formalisms. However, the representations of this triple by unary trees – called by the author *représentation stemmatique*, let them be 'structural syntactic trees' from here – seems not to be the best way to describe such a structure, at least under a formal point of view. In particular, the third element of the triple gets opaque: that "indication by nothing" *is* a crucial element that we prefer to put explicitly by an epsilon (ε), and justifies the admissibility of zero morphs. For this reason, in CxAdGrams a special form of binary trees is preferred: this form is called 'adpositional tree' (by now, 'adtree'), where the triple is explicitly rendered. Adtrees can be represented in three forms: the standard form (Figure 3), the linearised form (Figure 4) and the dependency-oriented form (Figure 5). In order to facilitate the reader used to other linguistic modelling of natural language grammars, and in particular the dependency-based models, we will present every example in at least two of the three forms.

Figure 3 shows the reinterpretation of the structural syntactic trees of Tesnière's example 2 in terms of standard adtree representation: this representation is standard, be-
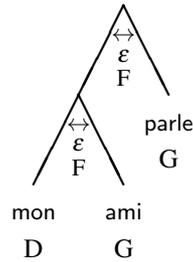

Figure 3. Standard adtree of Tesnière's example 2 [1]

$$\varepsilon_F^{\leftrightarrow}(\varepsilon_F^{\leftrightarrow}(\text{mon}_D, \text{ami}_G), \text{parle}_G)$$

Figure 4. Linear adtree of Tesnière's example 2 [1]
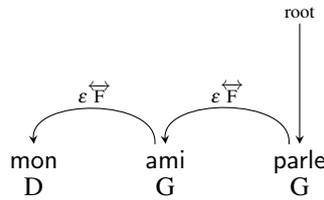

Figure 5. Dependency adtree of Tesnière's example 2 [1]

cause all structural elements are put in evidence. The governor (indicated by G) of each subtree is always put on the right leaf by convention, while the correspondent dependent (indicated by D) is put on the left. The third element of the triple, which determines the connection type (underspecified, structural or semantic) is put as a "hook" under the root of the tree, and indicated by F – meaning 'final'. In the first formalisation of adpositional grammars (AdGrams) [20], connection type was understood in Cognitive Linguistics terms as information prominence, and this interpretation is kept in CxAdGrams – see Section 5.1 below for details. However, the hook under the root is a marker not only of information prominence (the Tesnièrian connection type) but also of the final grammar character resulting from the interaction between the governor and the dependent – see the next Section 4 below, for an in-depth analysis. In fact, what Tesnière conceived as the connector – i.e., the element indicating the connection – is individuated as an *adposition*. In fact, within the adpositional paradigm, proposed at first under a more Cognitive Linguistic approach, in particular close to Construction Grammar formalisms [20], and now in CxAdGrams [21], linguistic analysis has shown that it is in a preposition, a postposition or another form of adposition that determines the final character (F) to the connection, in many languages of the world. This fact gives the same dignity to morphology and syntax. For instance, in example 2, the connections between *mon ami* and *parle* as well as between *mon* and *ami* are morphologically unmarked. In the terminology adopted in CxAdGrams, they are syntactic adpositions. Underspecified connections are signed by left-right arrows ($\leftrightarrow$).

Figure 4 shows the linear adtree, which is equivalent to the standard adtree. This representation has two advantages and an obvious disadvantage. The first advantage is compactness in terms of space: governors are at the right side, while dependents are put

at the left. The second advantage is that adpositions are put in evidence as the operators, preceding their respective governors and dependents, being the operands. Linear adtrees are similar to the Polish notation of mathematics introduced by Jan Łukasiewicz, which is adopted in some programming languages such as LISP. The obvious disadvantage is the loss of human readability, if the adtree is complex, because of the proliferation of parentheses. Other linear representations for trees have been tested, e.g., reverse Polish notation: although they reduce parentheses, they are even more difficult to read.

Dependency adtrees are the most recent form of representation – see example 2 in Figure 5. In fact, it is presented in this paper for the first time. We realised that both standard and linear adtrees are not intuitive for linguists and language specialists used to other representations of linguistic structures, while dependency trees in general are very clear, also for non-experts. Therefore, we have found a dependency-oriented representation which is equivalent to the others. It is important to stress the fact that arcs in dependency adtrees indicate Tesnièrian connections or dependency, as commonly intended; in order to avoid ambiguity, 'structural dependency' will be explicitly used to indicate the arcs, while 'informational dependency' will be the non-ambiguous term that indicates dependency in the sense of information prominence (see Section 5.1). In comparison with the other representations presented above, there is only one notable difference: the vertical arrow that indicates the adtree root is explicit, while the adtree root is implicitly individuated in standard adtrees as the rightmost leaf (*parle* in example 2). Governors G are put under the sources of the arcs, while dependents D are put under the targets. In CxAdGrams, each dependent has one and one only governor, as in the dependency-based formalisms presented above – with the exception of WG. The arcs are labelled with all adpositional information, i.e., the final grammar character (F, see Section 4), the information prominence ($\leftrightarrow$, if underspecified, see Section 5.1) and the morph (ε if zero). The great advantage of dependency adtrees is human readability, as they tend to respect the word order. The disadvantage is that they do not provide a representation suitable to direct mathematical and computational formalisation.

3.1. The importance of information hiding

In every dependency-based formalism presented in Section 2 dependency subtrees cannot be grouped as units of analysis – phrases are not allowed. Tesnière allowed grouping only to explain grammaticalization – the 'paradox of change' (Coseriu) for which yesterday's morphology becomes today's syntax, and vice versa, paraphrasing Givón. For instance, examples 36-37 in Figure 6 shows how the Latin syntactic expression *bona mente* ('with a good mind') became in French *bonne-ment* ('quite simply') because of grammaticalization.
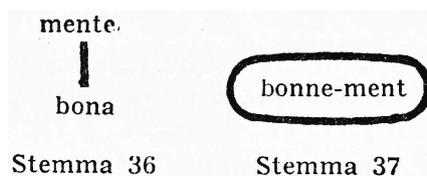

Figure 6. Examples 36-37 of grammaticalization (original image from Tesnière 1959)

CxAdGrams generalise the possibility of grouping, introducing an important difference with Structural Syntax. While structural syntactic trees are unary, adtrees are binary: from the point of view of the verbal governor *parle*, the dependent is the whole tree depicting *mon ami* (Figure 7). This does not imply that *mon ami* is a phrase: it is only a view, not a constituent. In other words, while in constituent-based formalisms phrases are ontological, in the adpositional paradigm dependencies can be grouped epistemologically, so to simulate phrases, without posing them as linguistic primitives. The mechanism to manage grouping is called information hiding.
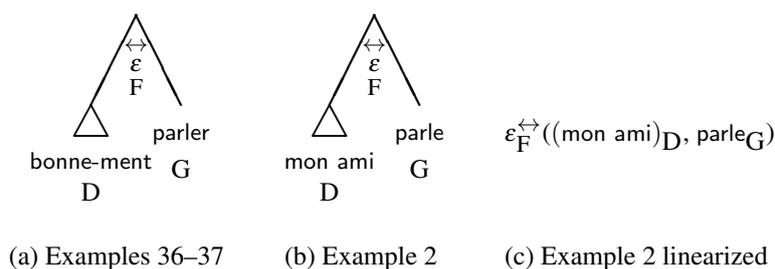

(a) Examples 36–37   (b) Example 2   (c) Example 2 linearized

Figure 7. How information hiding works in grammaticalization (a) and grouping (b)

The small triangles ($\triangle$) in Figure 7 indicate that a binary tree was 'packed' in order to increase human readability: however, no information was lost, but only *hidden* – i.e., it is always possible to get that information explicitly, exploding the subtree accordingly. The correspondent linear adtree is shown in Figure 7c. However, there is at least one example of pseudo-information hiding in Tesnière's, used to explain diachronic change from Latin to French: example 34 was the source in order to represent the small triangles in dependency adtrees as rectangles around words (see Figure 8). The advantage of dependency adtrees is the possibility of representing at the same time dependencies inside groups (in the example, these are the arcs over the words, i.e., between *il* and *aime* as well as *les* and *roses*) and between groups (the arc below the words). The standard adtrees provide an evident formal advantage as the group dependencies are rendered uniformly, with no need of special notation, see Figure 8.

Unfortunately, one of the greatest problems in Tesnière's work [1] is that examples are illustrated in different ways throughout the book incoherently. In fact, some information got lost during the way, some other gets introduced: in particular, structural *vs.* semantic influence, and grouping, got completely lost. In spite of this fact, we argue that the Tesnièrian triple 'governor, dependent, relation' is powerful and can be rightly captured by adtrees and recursion. Information hiding is needed essentially for humans – retrieving linguistic information whenever need. This is the reason why we adopted the triple as valid in general.

4. Word classes and syntactic functions

Tesnière [1] quite early introduces a set of symbols which "express the deep nature [of structure] without keeping the accidental contingencies" (A, ch. 33, par. 1). For Tesnière, morphology is the "shallow facet" while syntax is the "essential facet" of structure, i.e., Humboldtian *Innere Sprachform* – in modern terms, deep structure (1, A, ch. 12, note
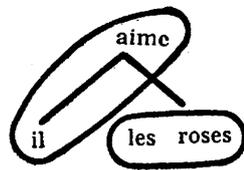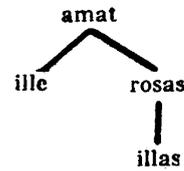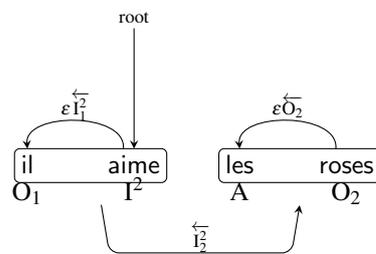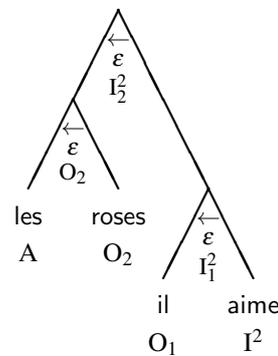
Figure 8. Grouping in Structural Syntax (a) and in dependency adtrees (b) of example 34 – (a) being the original image from Tesnière 1959

1). This emphasis on syntax is a severe limit, perhaps a cultural heritage of the times when the French linguist lived, where so much importance was given to morphology, almost neglecting syntax, and structuralism reacted in the opposite way. However, times changed, and it is now possible to express both syntactic and morphological information in adtrees through a unique mechanism – although, from a formal and computational point of view, syntax and morphology still may be kept separate for practical reasons. In other words, CxAdGrams extend Structural Syntax in order to comprehend morphologic phenomena, adopting the Tesnièrian symbols both for morphs and syntactic relations.

Figure 9. Tesnièrian classes and functions at work (examples 43–44, original image from Tesnière 1959)

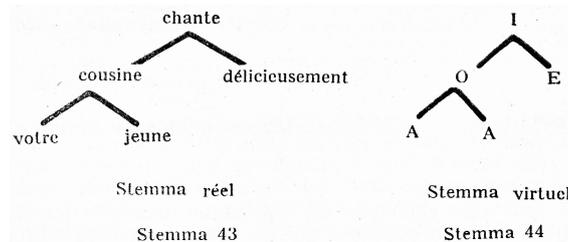Figure 9 shows the instantiated example 43 (the accidental contingencies of the *stemma réel*, the shallow facet) and its correspondent abstract syntactic tree (example 44, the deep nature of the *stemma virtuel*, or essential facet) – example 43 means 'your young cousin sings lovely'. However, in Structural Syntax there is no example of representation of both shallow and essential facets at the same time: in CxAdGrams, these levels are put together in adtrees, regardless of the representation.

Tesnière individuates four 'word classes' (*classes de mots*), valid in general and across languages. Their marker and respective *fonctions syntactiques*, 'syntactic functions', are the following ones:

- I = verbants (presence of predication),
- O = statives (expressions of reference),
- E = circumstantials (modifiers of verbants),
- A = adjunctives (modifiers of statives).

There is general agreement among linguists that the presence of expression of reference (i.e., "things") and the presence of predication (i.e., "events") are conceptual archetypes, i.e., always-valid universals of language – see for example [23,24]. Within the Standard Average European sprachbund, verbants (I) include verbs and interjections, while statives (O) include common and proper nouns, personal pronouns. Normally verbants and statives are the governors of their respective subtrees, while their modifiers play the role of dependents. Adjunctives (A) are the modifiers of statives, including adjectives, determiners, possessive pronouns – beware that in WG 'adjunct' is used as hyperonym of both modifiers, while adjunctives is the special term for stative modifiers. Circumstantials (E) are the modifiers of verbants, e.g., in English, adverbs and adverbials. In fact, example 34 in Figure 9 (right) shows that both modifiers (A and E) are dependents, respectively of the stative (O) *cousine* and the verbant (E) *chante*; the counterproof is that they are put below in the Tesnièrian abstract syntactic tree, as by convention governors are higher than their respective dependents. Tesnière explains that the choice of the vowels is a borrowing from the planned language Esperanto, used as a "mnemonic tool" (1, A, ch. 33, par. 1).

While the original Tesnièrian vowels are retained here for adherence with the original proposal, in order to help the reader in the comparison of the two models, their original names – e.g., "substantives" or "verbs" – were not adopted in Constructive Adpositional Grammars, because these names are too closely related to the grammar tradition belonging to the Standard Average European sprachbund [25]. However, it is important to notice that Tesnière [1] gives examples from a lot of different languages through the book, in order to show how Structural Syntax is valid across sprachbunds: mainly French, German, Latin, Russian, Greek, but also Coptic, Chinese, Samoan, Turk, Tatar, Votiak. The cross-linguistic validity of the four word classes is strengthened by the same result obtained in a completely independent way, by Benjamin Whorf [26]. The American linguist addressed the problem of grammar categorisation out of Standard Average European, with results similar to Tesnière's. Since Whorf's names are valid across typologically distant sprachbunds, they were adopted here, with some adaptation.

However, Whorf introduces the concept of selective and collocational lexemes that Tesnière did not notice. In particular, in every language there are some *selective* lexemes: these lexemes have their proper grammar category carved inside, as in the English adjunctive (A) *honest*. No collocation can turn the grammar category of the selective ad-
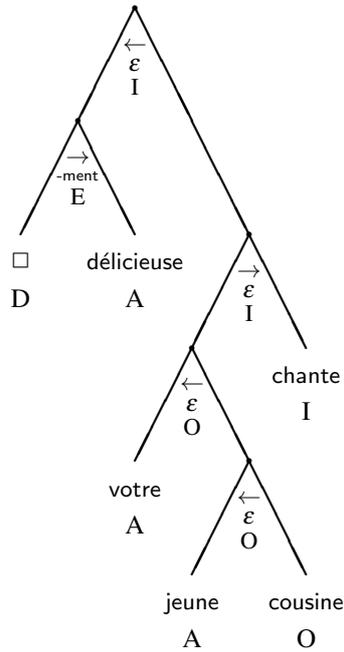
←
ε
I

→
-ment
E

□   délicieuse
D   A

→
ε
I

←
ε
O

chante
I

votre
A

←
ε
O

jeune   cousine
  A   O

Figure 10. Reinterpretation of Tesnièrian examples 43-44 in a standard adtree

junctive, but only morphology, e.g., *honest-y*, in order to obtain a stative (O), or *honest-ly*, in order to obtain a circumstantial (E). In other words, selective lexemes are never affected by syntax, but only by morphology.

In contrast, collocational lexemes are also affected by syntax – as well as morphology. In other words, their word classes are defined only if put in collocation – on the syntagmatic axis. In isolation, we can have cues about their most probable class, but we cannot be sure. For instance, the English lexeme *walk* is probably a verbant (I), as in the group *I walk in the park*. Nonetheless, it can also be a stative (O), as in *Let's have a walk* or even an adjunctive (A), as in *walking distance*.

It is the adtree that shows the final function of the lexeme in context. As collocational lexemes do exist, we prefer to consider the function as a label put on the lexeme (or the word) instead of something carved inside, and as a result in CxAdGrams instead of 'word classes' we prefer to say 'grammar characters'. Consequently, selective lexemes are considered as a special case.

5. Adpositional trees and valency

Adtrees retain all the features of Tesnière's model in a single representation, in particular both the concrete and abstract syntactic trees – surface and essential facets – as shown in the adtree of examples 43-44 (Figure 10 for the standard adtree, Figure 11 for the dependency-oriented adtree, both without any grouping imposed).

It is important to note that the verbant *chante* and the stative *cousine* are the governors of their respective adtrees, as expected from example 44. The reader is invited to note that the final grammar character of the stative group *votre jeune cousine* is indicated
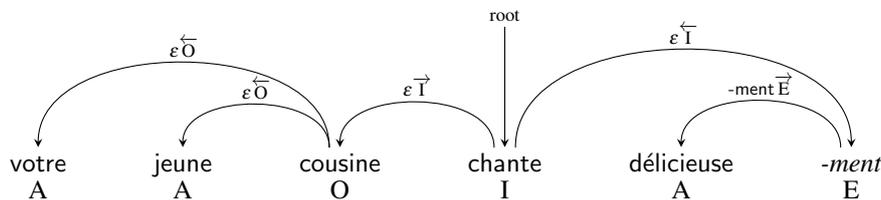
Figure 11. Alternative representation of examples 43-44 as a dependency adtree

by the syntactic adpositions (ε); analogously, the stative-verbant connection is syntactic as well. On the contrary, the adverb *délicieusement* is obtained by the application of the suffix *-ment* to the lexeme, whose function is to impose the circumstantial grammar character E to the adjective *délicieuse*. In the standard adtree, the dependent branch is left underspecified (D states for 'dependent' in general), while the zero morph is indicated by the box (□) and it cannot be further expanded by the application of other morphs. In contrast, the dependency adtree do not need boxes, but on the other hand it shows redundantly the adposition *ment* as an adposition (on the arc) and as a morph (in collocation) to explain more explicitly the grammar character rotation (A > E) or 'transference', in Tesnièrian terms: with the use of grouping, this kind of dependency arcs can be hidden, if needed –see Section 6 below for details. It is important to notice that both representations are equally valid: they have advantages and disadvantages, already explained above. Moreover, this particular example 43-44 shows how in CxAdGrams morphosyntax is a single mechanism.

5.1. Trajectories of information prominence

The Tesnièrian dichotomy 'structural *vs.* semantic influence' became one of the core features of CxAdGrams [22], as the cognitive-based adpositional functions of AdGrams [20] could not represent ergativeness easily, while information prominence solves this problem elegantly. In fact, in the last years' research in cross-linguistic typology on ergativeness has shown that a good grammar theory "would have to recognise that there are three basic syntactic-semantic primitives (A, S and O) rather than just two 'subject' and 'object' – however these are defined" [27], p. 236. The arrows proposed by Tesnière the solution of this problem (see again Figure 2, if needed). Within a stative-verbant connection, if the stative actively "does" the action, then the stative will be the most prominent element of the pair: in the terms proposed by Langacker [23], the stative (O) is the trajectory (TM) while the verbant (I) is the landmark (LM). Following the conventions of the standard adtrees, the trajectory of information prominence will be from left to right ($\rightarrow$). In other words, the stative, being the dependent, is prominent (TR), and hence the connection will be a *informational dependency* ('semantic influence', according to Tesnière). In the opposite case, if the action marked by the verbant (I) "happens" to the stative (O), then the verbant will be trajectory (TR) and the stative the landmark (LM): accordingly, the trajectory will be from right to left ($\leftarrow$). As the verbant is the governor, the connection will be called *government* ('structural influence', according to Tesnière). Within the adpositional paradigm, the word 'dependency' can assume a very technical and precise meaning. It is important to note that what stated for the stative-verbant connection is valid for every grammar character connection, as exemplified in Figure 10.
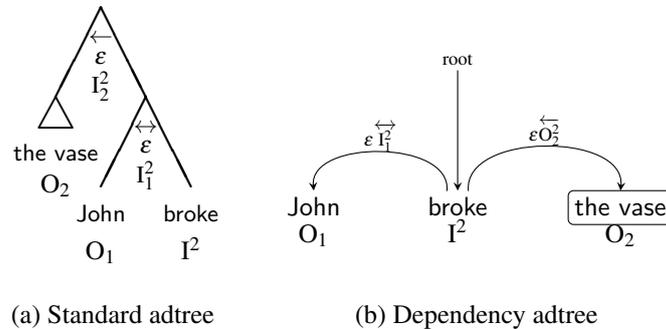
| (a) Standard adtree | (b) Dependency adtree |

Figure 12. Two different but equivalent representations of *John broke the vase*

The trajectory of information prominence permits to have three basic syntactic-semantic primitives, so to deal with ergativeness as well as accusativeness elegantly. The adtree of *John broke the vase* is an example of the expressive power of information prominence within the adpositional paradigm (Figure 12). Let assume that our world model is standard, i.e., vases are non-animated objects, without will or beliefs, and John is a man – as explained above with reference to MTT, CxAdGrams are agnostic in respect of the world model, i.e., they do not model semantics in terms of ontologies or taxonomies. While John can have broken the vase by accident (government, ←) or willingly (dependency, →), the vase for sure happened to be broken, from its point of view, and hence its connection is a government (←).

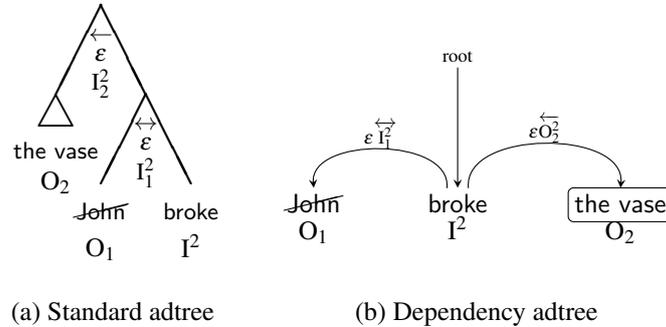
| (a) Standard adtree | (b) Dependency adtree |

Figure 13. How the representation of *The vase broke* is derived from *John broke the vase*

Trajectory of information prominence explains why some "subjects" are statives in dependency – $\overrightarrow{O_1}$, 'A' in Dixon's terms – while others are in government, i.e. $\overleftarrow{O_1}$, 'S' for Dixon [28]. This treatment has important consequences: in particular, it permits different constructions to be put into relation, opening the door to adtree transformations (see Section 6.2). In particular, the adtree of *the vase broke* (Figure 13) can be considered a reduced or *transferred* adtree of *John broke the vase* (Figure 12), where the subject (either in government or dependency, i.e., generically O_1) got lost.

Finally, information prominence gives account to topic-focus articulation phenomena. For instance, if we look at the adjective-noun dependency, Italian shows a clear difference in the word order of the stative group in many semantic domains: in the adjective is put before the noun, as in *un alto funzionario* ('a high official'), the information promi-

nence relies in the governor *funzionario* ($\varepsilon\overleftarrow{O}$); in the opposite case, where the adjective is postponed, such as in *un funzionario alto* ('a tall official'), it is the dependent *alto* to be more prominent ($\varepsilon\overrightarrow{O}$).

5.2. The treatment of valency

The notion of valency and the actants – i.e., how many statives are needed to saturate the valency value – is one of the most successful part of his Structural Syntax, being adopted in most dependency-based frameworks as well as in CxAdGrams, presented in this way by Tesnière:

> one could indeed compare the verb to a kind of **crossed atom**, which can attract a number more or less high of actants, in proportion to the number more or less high of hooks needed to maintain the respective dependencies (2, D, ch. 97, par. 3).

Figure 14 shows how Tesnière indicates the numbers of the actants saturating the valency value, in case of a trivalent verb – not coherently, only somewhere across the book. The examples are in Latin and in German, where an English equivalent can be *Alfred gave the book to Charles*.
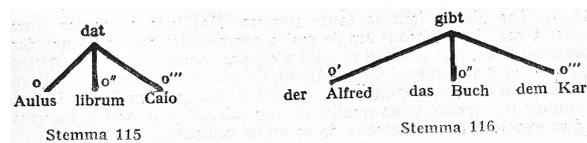
Figure 14. Examples of trivalent verbs (original image from Tesnière 1959)

(a) Standard adtree   (b) Dependency adtree
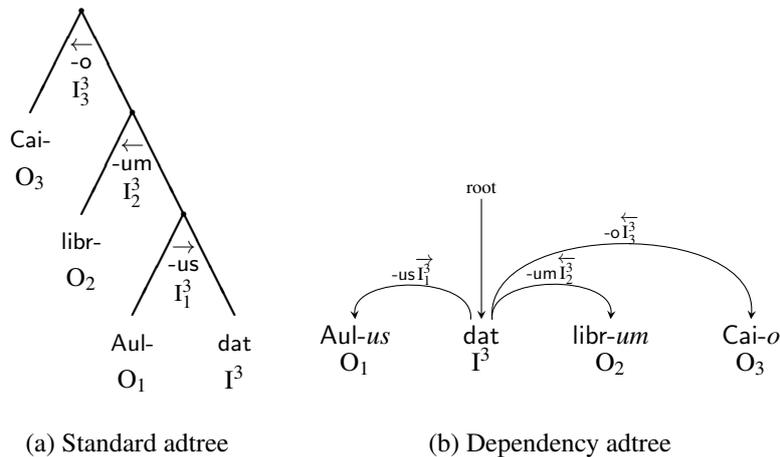
Figure 15. Example 115 in two adpositional tree representations

Figure 15 shows the adtree of example 115 in Latin. In CxAdGrams, the verbant is the governor of the phrasal adtree – 'phrase' to be intended as an adtree governed by a uniquely identified verbant, as an epistemological construct, not ontological, as explained above. If the verbant is a trivalent verb, as in example 115, three actants (i.e.,

O_1, O_2, O_3) are provided to saturate the valency value, along with their respective adtrees. The superscript number of the verbant indicates the absolute valency value – e.g. I^2 for a bivalent verb. The subscript number of the verbant indicates the degree of saturation in that point of the adtree, while the subscript of the actant indicates the actant number, following Tesnière's usage (Figure 14). Example 115 shows that in Latin substantive finals act as adpositions of the stative-verbant connection, with an indication of information prominence: *Aulus* ('Alfred') performs the giving (*dat*) and hence it is in a relation of informational dependency ($\rightarrow$), while the giving happens both to *Caio* ('Carl', the beneficiary), and *librum* ('the book', the actual object which was given), and therefore they are are both in a relation of government ($\leftarrow$).

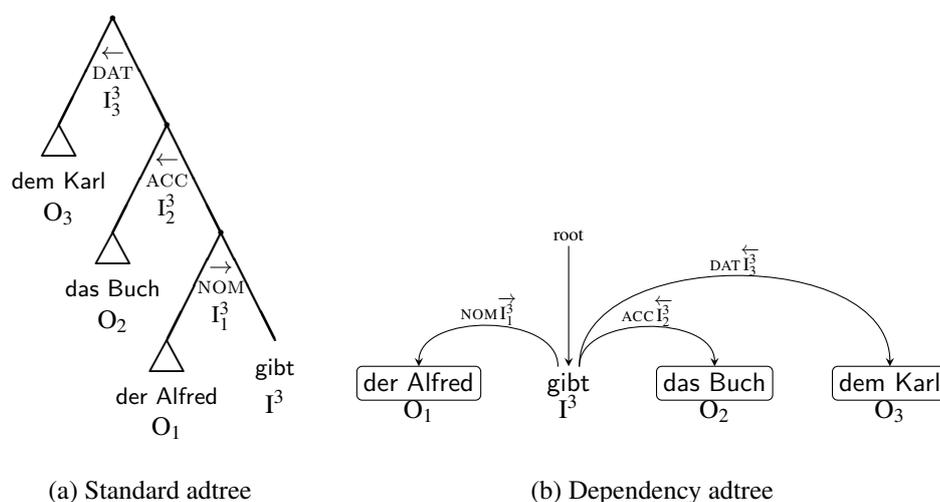(a) Standard adtree   (b) Dependency adtree

Figure 16. Example 116 in two adpositional tree representations

Sometimes adpositions are marked through sememes, i.e., structural well-defined traits within a given language, even if the morph is absent. For instance, example 116 shows that in German the case markers, like DATIVE, are not morphologically marked in the stative-verbant connection, but still well present in every German speaker's competence. In these cases, sememes can be written explicitly instead of epsilons, for clarity, if there is no possible ambiguity.

The detail of the adtree hidden under the third actant *dem Karl* (Figure 17) shows that the sememe DATIVE is an additional trait under the adjunctive grammar character. Moreover, the adtree clarifies that there is a number agreement requirement, indicated by the sememe SINGULAR, between the stative *Karl* and the adjunctive *dem*, in order to get everything work under a grammatical point of view. No such level of detail is present in Structural Syntax, most probably because Tesnière was not interested in. However, it is important that such level of detail is possible within the model here proposed if needed, e.g. for language learning purposes.

6. Transference

Every language – regardless of its sprachbund – has a class apart within its morphemes devoted to convey the most part of semantics, called lexemes. In fact, while the concept
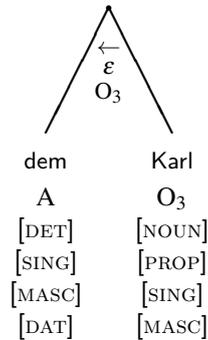
$$\begin{array}{c} \wedge \\ \nwarrow \quad \nearrow \\ \leftarrow \\ \varepsilon \\ O_3 \end{array}$$

dem	Karl
A	O_3
[DET]	[NOUN]
[SING]	[PROP]
[MASC]	[SING]
[DAT]	[MASC]

Figure 17. Detail of the standard adtree of example 116

of 'word' is central only in the grammar traditions of the Standard Average European, the distinction of lexemes within a language's morpheme set is valid in general. For example, in Japanese non-lexemes are written in *kana* (syllabic script), while lexemes are written in *kanji* (Chinese logographic characters). Lexemes are morphemes devoted to represent the relation between the language and the non-linguistic world, with a particular attention to reference. In structural syntactic trees, they are put above, being the governors, and similarly in the adpositional paradigm they are put in the rightmost leaves of their respective standard adtrees. Tesnière noted that the most part of the non-lexical morphemes have the function of "turning" the grammar character of the lexemes they are applied to (3, A, ch. 161, par. 6).

(290) un exemple frapp-**ant** (I > A)
    'a strik-**ing** example'
(292) liber Petr-**i** (O > A)
    'Peter**'s** book'

The French suffix *-ant* (in 290) is applied to verbant lexemes in order to transfer their syntactic function to adjunctive, while the Latin suffix *-i* (in 292) is applied to stative lexemes in order to transfer their syntactic function to adjunctive as well. Of course, there are a lot of differences between the two adjunctives: in CxAdGrams, they would be expressed by different sememes and trajectories of information prominence. This kind of differences are not well formalised in Structural Syntax; however, the fundamental intuition that the morpheme set of a language can be divided in lexemes and non-lexemes on a functional syntactic basis is a remarkable part of Tesnière's heritage in the adpositional paradigm, since its definition in Gobbo [20]. This kind of morphemes (prepositions, postpositions, derivational suffixes and so on) were called by Tesnière *translatifs* – in the model proposed here, morphological, explicit adpositions, i.e., morphs – and the phenomenon as a whole was called *translation*, while in English "an equivalent may be *transference*, as the word *translation* has already the meaning of the French 'traduction'" (3, A, ch. 154, par. 2).

During the development of the formal model on which CxAdGrams, the role of transference grew of importance. Tesnière devoted a lot of pages to transfer chains, from 'simple transference' (*translation simple*, e.g. I > O) to 'double transference' (*translation double*, e.g. O > A > O) until, at a limit, sevenfold transference (*translation septuple*). Complex transfer chains, i.e., double or more, can be explained in terms of recursive, nested adtrees, but this solution has two limits. First, from a linguistic point of view,

there is no relation between an abstract adtree and the others belonging to the same language – 'abstract adtree' – meaning what Tesnière called a *stemma virtuel*, i.e., an adtree without morphemic instantiation, something we wanted to avoid. Second, from a computational point of view, the CxAdGram of a given language should contain at least two kind of descriptions: the first one explains how morphs act, along with their morphemes, grammar characters, basic transfers and the appropriate sememes; on the other hand, the second one lists the finite set of admissible abstract adtrees. This list could grow inconveniently, if there were no method to put in relation one abstract adtree with the other within the same CxAdGram – in principle, a machine translation model can be envisaged if the relations are between pairs of CxAdGrams. In fact, one of the goals of the adpositional paradigm is to give a convenient description of natural language grammars, both linguistically and computationally.

6.1. Abstract adtrees as constructions

The Tesnièrian concept of transference shows that most part of the dictionary is in reality the result of transfer chains: for this reason, a constructive dictionary can be built upon the lexemes and a set of transfer chain *patterns* in order to perform grammar character changes. In a cognitive linguistic perspective, these patterns of usage of form-meaning correspondences, that carry meaning beyond that of the single composing morphemes, are called *constructions* [28,29]. As a side note, the community of cognitive linguistics recognised Structural Syntax as a complementary, although dated, approach [30].

After the study of Tesnièrian transference, it seemed more reasonable to see abstract adtrees as constructions instead of describing grammar only in terms of adtrees, so that the relations between constructions are formally represented in terms of adtree transformations, i.e., Tesnièrian transference rendered in formal terms. For example, the active-passive diathesis transference (2, D, ch. 101–102) can be expressed in terms of adtree transformations. Basically, the most primitive construction is the active diathesis, with all valency saturated by the possible actants, then a chain of adtree transformations permits to obtain the desired construction.

(A) (Carl)$_O$ (slept in)$_I$ (the beds)$_O$.
(B) (the beds)$_O$ (were slept in)$_{I>I}$ (by Carl)$_{O>O}$.
(C) (Carl's)$_{O>A}$ (sleeping)$_{I>O}$.

Examples A-B-C were annotated with the main grammar characters of the respective adtrees in order to help the reader in the knowledge of the use of transference within the model proposed here. In particular, example A shows an instantiation of the active diathesis construction of the English verbant *to sleep in*, while example B shows the correspondent passive construction. It is worth noticing that two transfers were performed in order to obtain the appropriate form of the verb (I > I) and of the SLEEPER actant (O > O). Moreover, example C is an example of normalisation: the SLEEPER actant was transferred into a Saxon genitive construction (O > A) while the *ing*-construction transferred the verbant into a stative (I > O).

It is possible to write down classes of lexemes following the admissible patterns of adtree transformations. For example, it can be easily tested that the verbants *to sleep in* and *to melt in* belong to different classes of English verbants:

English	**wh**-ere	**th**-ere	**wh**-en	**th**-en
French	où	là	**qu**-and	alors
Latin	u-bi	i-bi	**qu**-ando	**t**-um
German	**w**-er	**d**-a	**w**-ann	**d**-ann

Table 1. Tesnièrian analysis of correlatives

(D) the ice cube melted in the oven.

(E) *the oven was melted in by the ice cube.

(F) the melting of the ice cube.

Example D is structurally identical to example A; nevertheless, the passive construction obtained by the adtree transformation is ungrammatical (example E), while a different adjunctive construction, head by the adposition *of*, is to be preferred to Saxon's genitive (example F). A full treatment of adtree transformation would need at least another paper devoted to it, so it is left as a further work.

6.2. Second-order transference

Tesnière introduces the second-order transference (*translation du second degré*) in order to explain "what the traditional grammar had already implicitly regarded apartly with the name of 'subordination".' (3, D, ch. 239, par. 2). For example, the sentence *Alfred espère qu'il réussira* ('Alfred hopes that he will achieve') is a second-order transference from the verbant phrase *Alfred réussira* ('Alfred will achieve') to the made stative phrase *qu'il réussira* ('that he will achieve'; 3, D, ch. 241, par. 15). This kind of second-order transference is indicated with the symbol: $\gg$; e.g., a verbant-stative second-order transfer will be indicated as such: $I \gg O$.

Tesnière noticed that the *translatifs* – in the model proposed here, adpositions – devoted to second-order transference show a high degree of regularity in many different languages (3, D, ch. 240, par. 6, adapted in Table 1). Unlike Structural Syntax, in CxAd-Grams there is no need of a second-order level because of the expressive power of the mathematics underlying the formal model (see next Section 7). What is retained from the Tesnièrian analysis is the observation that correlatives are *double* morphemes, made by a fixed part (e.g., *wh-* in English), that is appended to the governor's phrase, and a flexible part (e.g., the English *-ere* for PLACE and *-en* for TIME) that is put in the adtree of the adtree of the subordinate phrase.

(H) I know where she goes.

Figure 18 shows the standard and dependency adtree of example H. In both cases, the adtree of *where she goes* is intact in its inner construction: the relevant fact is that the correlative adposition *wh-ere* transfers the phrase from verbant to the second actant stative ($I > O_2$), from the point of view of the construction of *I know [where she goes]*. This is particular evident in the standard adtree: on the other hand, the dependency adtree put in evidence the role of the secondary phrase *-ere she goes* as the second actant ($\overrightarrow{O_2}$) of the main phrase – see the arc below In sum, adtrees can represent correlatives without any need of a second-order level of analysis. For the interested reader, Gobbo presents an inter-linguistic comparison of correlatives, based on this line of reasoning, in [31].
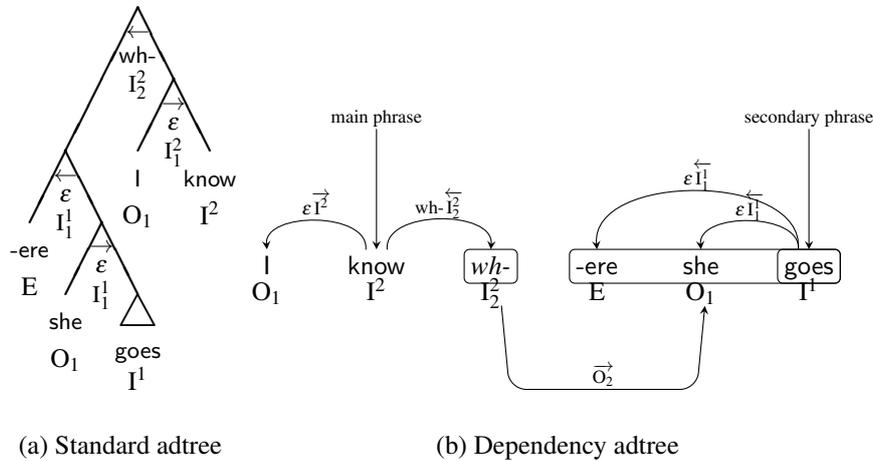
(a) Standard adtree   (b) Dependency adtree

Figure 18. Adtrees of example H, showing correlatives

7. Sketches of the formal model

Tesnière asserts that "the use of symbols [grammar characters O, I, E, A, authors' note] in grammar is equivalent to the use of calculus in algebra" (1, A, ch. 33, par. 10). This statement implies that Structural Syntax can be formalised, at least theoretically. In the fields of mathematical and computational linguistics there are many natural language grammar formalisms currently under investigation. In particular, the constructive adpositional formalism can be put into the realm of the so-called "categorial grammars" – i.e., representations of natural language grammars in terms of categories [32]. At the present stage, the formal model is intended as a guiding reference for the development of linguistic concepts [22]. In fact, 'constructive' is intended linguistically as pertaining to constructions (as already defined) and mathematically as pertaining to constructive mathematics, i.e., any formal, mathematical construct used here have a constructive existence. In other words, it is possible to find an algorithm, non necessarily efficient, to construct any entity of the model.

In particular, adtrees and constructions together form a category, called **AdTree**, in the mathematical sense [33,34]. A mathematical category is an algebraic structure composed by two classes, the *objects* and the *arrows*; arrows lie between two objects, the *source* or *domain*, and the *target* or *codomain*. Also, a category states that there are distinct arrows, the *identities*, one for every object A and such that the source and the target are A. Moreover, a category is equipped with a partial operation allowing to compose two arrows whenever one has the domain which is the target of the other one. Composition is required to be associative and identities act as one expects with respect to composition.

Intuitively, there is an arrow f from A to B whenever we can construct the B tree starting from the A tree applying the construction f. We do allow complex constructions obtained by sequentially composing simpler ones; if f and g are constructions such that $f(A) = B$ and $g(B) = C$, that is, if f maps A into B, and g constructs C from B, then $g \circ f$ is the construction which maps A into C by doing g after f.

It is possible to observe that, calling M the free monoid over the alphabet of morphemes of some natural language, i.e., the set of all possible (finite) sequences of mor-

phemes obtained by juxtaposition, the functions mapping the trees in **AdTree** into the sequences of M comprehend the textual renderings of adpositional trees. If the attention is restricted to *contra-variant functors*, i.e., the functions preserving the identical transformation and the reverse composition of adpositional trees, what is obtained is a class of functions which is called *presheaves over M*. Requiring that a presheaf maps the morphemes in the adtree into themselves in the monoid, what is obtained is exactly the lexicalizations of adtrees. In other words, there is a subclass of presheaves which directly corresponds to the texts the adtrees represent and which encodes the transformations that constitute the grammar. It is this space of presheaves which is generally understood as the subject of linguistics. Moreover, considering *endo-functors* on **AdTree**, i.e., functions mapping each adtree into another adtree, and each construction into another one such that they preserve identities and composition, it easily follows that each linguistic transformation, e.g., the mapping of active to passive diathesis, is an endo-functor. In turn, an endo-functor can be represented as an arrow between presheaves, thus showing that the mathematical model of the presheaves space is rich enough to represent and to reason about the foundational elements of CxAdGrams.

As a side effect of this intended model of interpretation, it follows that whatever construction over adtrees which is built by combinatorially composing the fundamental constructions, is an arrow. Lifting the structure of the **AdTree** category into the spaces of presheaves, which is a category, it is possible to reason in a larger and richer environment, where the full power of mathematical methods can be applied: in fact, the presheaves space is a *Grothendieck topos* [35,36], one of the richest mathematical structures. For a complete treatement of the formal model of CxAdGrams, the interested reader is invited to read Appendix B of the book devoted to their foundation [22].

8. Conclusion

The impressive work by Tesnière [1] is a constant source of inspiration for the definition of the new formalism explained in this paper. It is quite astonishing that nobody until now – as far as the authors know – has proposed a dependency-based model that makes use of the grammar characters proposed by the French linguist, i.e O, I, E, A, which are the ground on which Structural Syntax was built.

However, as Hajičová entitled her *lectio magistralis* in occasion of the ACL Lifetime Achievement Award in 2006, old linguists never die, they only get obligatorily deleted on the visible surface. This paper aimed to recover Tesnière's heritage, putting it again on the visible surface. CxAdGrams stands on the shoulders of Tesnière, being an elaboration of Structural Syntax in many aspects.

References

[1] L. Tesnière. *Éléments de syntaxe structurale*, Klincksieck, Paris, 1959.

[2] G. Graffi. *200 Years of Syntax: A Critical Survey*, John Benjamins, Amsterdam, 2001.

[3] B. H. Partee and P. Sgall. *Discourse and Meaning: Papers in Honor of Eva Hajičová*, John Benjamins, Amsterdam, 1996.

[4] J. Nivre. *Dependency grammar and dependency parsing*, Technical report, Växjö University, 2005.

[5] R. A. Harris. *The Linguistic Wars*, Oxford, Oxford University Press, 1995.

[6] S. Kahane: On the status of phrases in head-driven phrase structure grammar: Illustration by a fully lexical treatment of extraction. Chapter in: I. Mel'čuk and A. Polguère. *Dependency in Linguistic Description*, Amsterdam, John Benjamins, 2009.

[7] J. Hajič, J. Panevová, E. Hajičová, P. Sgall, P., P. Pajas, J. Štěánek, J. Havelka, M. Mikulová. Prague Dependency Treebank 2.0. Prague, Linguistic Data Consortium, 2006.

[8] S. Kübler, R. McDonald, J. Nivre. *Dependency parsing*. Morgan & Claypool Publishers, 2009.

[9] P. Sgall, L. Nebeský, A. Goralčiková, E. Hajičová, J. Panevová. *A Functional Approach to Syntax in Generative Description of Language*, New York, PUBLISHER MISSING, 1969.

[10] S. Starosta. Lexicase grammar. Chapter 41 in: H. E. Wiegand (hsg. / ed.) *Dependenz und Valenz / Dependency and Valency*, Berlin & New York, Walter de Gruyter, 2003. 526–545.

[11] I. Mel'čuk. *Dependency Syntax: Theory and Practice*, Albany, N.Y., The SUNY Press, 1988.

[12] U. Eco. *The Search of the Perfect Language*, London, Wiley-Blackwell, 1997.

[13] R. Debusmann. *A declarative grammar formalism for dependency grammar*, Diplomarbeit, Universität des Saarlandes, 2001.

[14] R. Debusmann. *Extensible Dependency Grammar: A modular grammar formalism based on multigraph description*, PhD thesis, Universität des Saarlandes, 2006.

[15] R. Debusmann, M. Kuhlmann. Dependency Grammar: Classification and Exploration. In: M. W. Crocker, J. Siekmann (eds.), *Resource-Adaptive Cognitive Processes*. Berling, Springer Verlag, 2010.

[16] R. Hudson. Word Grammar. Oxford, Blackwell, 1984.

[17] R. Hudson. Language Networks: the New Word Grammar. Oxford, Oxford University Press, 2007.

[18] R. Hudson. Word Grammar. An Introduction to Word Grammar, Cambridge University Press, 2010.

[19] V. Ágel, L. M. Eichinger, H-W. Eroms, P. Hellwig, H. J. Heringer, H. Lobin (eds.). *Dependenz und Valenz / Dependency and Valency Ein internationales Handbuch der zeitgenössischen Forschung / An International Handbook of Contemporary Research*. Berlin and New York, Walter de Gruyter, 2003.

[20] F. Gobbo. *Adpositional Grammars: A Multilingual Grammar Formalism for NLP*, PhD thesis. University of Insubria, 2009.

[21] Y. Lepage, Anto S., Akamine S., Iida H.. *An annotated corpus in Japanese using Tesnière's structural syntax*, DOI 10.1.1.14.8905. Technical report. ATR Research Labs, NEC C&C Media, Human Media Research Labs, Kyoto and Kanagawa, Japan. 1998.

[22] F. Gobbo, M. Benini. *Constructive Adpositional Grammars: Foundations of Constructive Linguistics*. Newcastle upon Tyne, Cambridge Scholars Publishing. 2011.

[23] R. W. Langacker. *Foundations of Cognitive Grammar: Theoretical Prerequisites*. Standord, Stanford University Press. 1987.

[24] M. Tomasello. *Constructing a language: A Usage-Based Theory of Language Acquisition*. Harvard, Harvard University Press. 2003.

[25] M. Haspelmath. The European linguistic area: Standard Average European, in: M. Haspelmath, W. Oesterreicher, W. Raible. *Handbuch der Sprach- und Kommunikationswissenschaft*. Berlin & New York, Mouton de Gruyter, 1492–1510. 2001.

[26] B. L. Whorf. Grammatical Categories. *Language*, 21(1):1–11. 1945.

[27] R. M. W. Dixon. *Ergativity*. Cambridge, Cambridge University Press. 1994.

[28] W. Croft. *Radical Constructions Grammar*. Oxford, Oxford University Press. 2001

[29] A. E. Goldberg. *Constructions at work*. Oxford, Oxford University Press. 2006.

[30] R. W. Langacker. Structural syntax: the view from cognitive grammar, in: G. Dondelinger, F. Madray-Lesigne, J. Richard-Zappella. *Lucien Tesnière aujourd'hui*. Leuven, Peeters Publishers, Leuven. 13–37. 1995.

[31] F. Gobbo. The Case of Correlatives: A Comparison between Natural and Planned Languages. *Journal of Universal Language*. 12(2):45–79. 2011.

[32] G. Morril. A Chronicle of Type Logical Grammar: 1935–1994. *Research on Language & Computation*, 5(3):359–386. 2007.

[33] S. Mac Lane. *Categories for the Working Mathematician*. Springer Verlag, Berlin. 1998.

[34] F. Borceux. *Handbook of Categorical Algebra I*. Cambridge University Press: Cambridge. 1994.

[35] S. Mac Lane and I. Moerdijk. *Sheaves in Geometry and Logic*. Springer Verlag, Berlin. 1992.

[36] P. Johnstone. *Sketches of an Elephant: A Topos Theory Compendium*, Volume 2. Cambridge University Press: Cambridge. 2002.