To what extent is Immediate Constituency Analysis dependency-based? A survey of foundational texts

Nicolas Mazziotta
Université de Liège
Universität Stuttgart
Belgium, Germany
nicolas.mazziotta@ulg.ac.be

Sylvain Kahane
Modyco
Université Paris Nanterre
CNRS
France
sylvain@kahane.fr

Abstract

This paper investigates the seminal texts on Immediate Constituent Analysis and the associated diagrams. We show that the relations between the whole and its parts, that are typical of current phrase structure trees, were less prominent in the early diagramming efforts than the relationships between units of the same level. This can be observed until the beginning of the 1960's, including in Chomsky's *Syntactic Structures* (1957). We discuss whether such analyses could be said "dependency-based", according to an attempt to define this term.

1 Introduction

Chomsky's *Syntacic structures* (1957) is famous for the formalization of immediate constituent analysis (henceforth *ICA*) it introduces, using string-rewriting systems. After the first example of such a system, Chomsky introduces a corresponding diagram (reproduced here in fig. 1(a)) representing a set of equivalent derivations. Such a structure is now called a *derivation tree* and represented by tree, but it will appear later in this paper that Chomsky's first diagram was not exactly a tree. Let us compare fig. 1(a) with fig. 1(b), which should be an equivalent diagram, since it appears in the French translation of the same text (Chomsky, 1969(1957)). Fig. 1(b) is similar to phrase structure trees in (Chomsky, 1965): each internal node except the root, is linked to an upper node by a stroke encoding a part-whole relation. The original diagram (fig. 1(a)) does not display the same configuration of strokes.

Synctacticians of all kinds are familiar with diagrams, but most of the time, they use them without questioning their origins or the implications of the structural choices they represent. Studies on this subject, such as (Coseriu, 1980) on Tesnière's stemmas, (Stewart, 1976) on linguistic diagrams in general and (Mazziotta, 2016b) on the representation of syntactic knowledge, are not frequent, but we think they contribute to the definition of our epistemological field. Thus, the aim of this paper is to understand Chomsky's first diagram, as well as the other diagrams proposed for the formalization of ICA until tree-based diagrams become the norm in the mid 1960's. These diagrams will be compared with dependency trees and we will discuss whether such analyses can be deemed as "dependency-based".

Section 2 introduces the mathematical and graphical notion of *tree* as well as the notion of *reification*, that helps understanding how diagrams are conceptualized. Section 3 attempts to define the meaning of the term *dependency*, in connection with the usage of trees in dependency and phrase structure syntax. Chomsky 1957's diagram is analyzed in section 4 in order to evaluate to what extent it is "dependency-based". The same section surveys the foundational works in ICA in the light shed by those preliminary notions (Barnard, 1836; Bloomfield, 1933; Wells, 1947; Nida, 1943; Gleason, 1955; Hockett, 1958). In the conclusion, we point out what distinguishes dependency syntax from ICA.

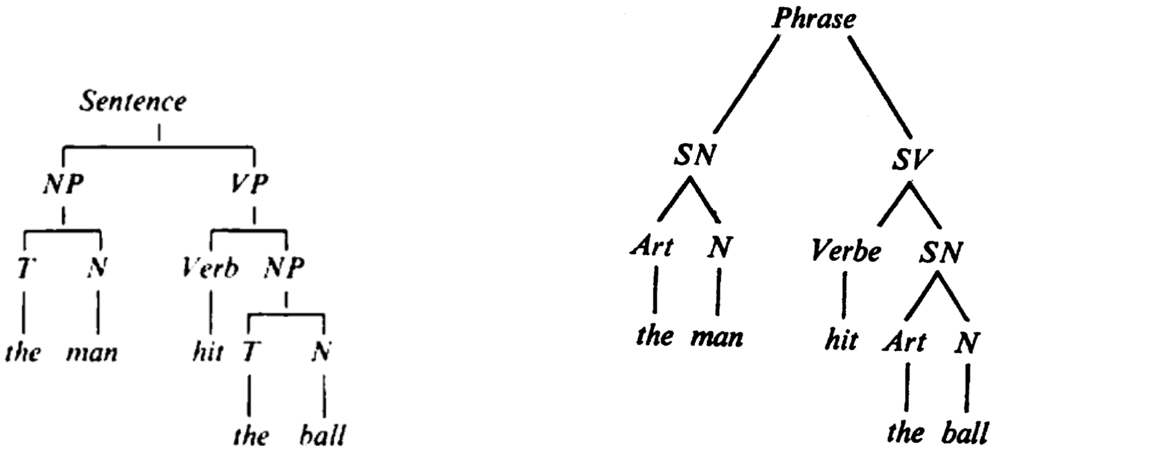
2 Trees and reification

This section introduces the notion of *tree*, from an algebraic as well as a graphical perspectives (section 2.1). The notion of *reification*, i.e. the fact that conceptual elements are represented by discrete graphical entities in diagrams, is discussed under 2.2.

2.1 Algebraic and graphical notion of *tree*

To understand Chomsky's first diagram and other ICA diagrams, we need to bear in mind what a tree is. In graph theory, a tree T is algebraically de-

Proceedings of the Fourth International Conference on Dependency Linguistics (Depling 2017), pages 116-126,
Pisa, Italy, September 18-20 2017



(a) Original diagram (Chomsky, 1957) (b) Diagram in the French translation (Chomsky, 1969(1957))

Figure 1: Diagrams corresponding to the first derivation structure in (Chomsky, 1957)

fined as a kind of directed graph (with nodes and edges pairing them) that satisfies two additional constraints: it is connected and it does not contain any cycle. (1) is a simple example of the algebraic expression of a tree T, with N a set of nodes, E a set of edges, and π a map associating edges with their vertices, that is, ordered pairs in $N \times N$.

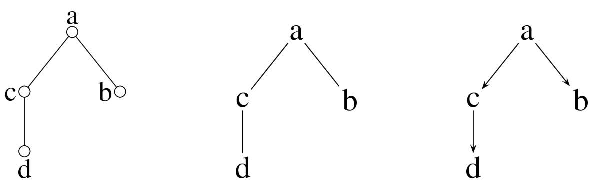
(1) $T = (N, E, \pi)$
$N = \{n1, n2, n3, n4\}$
$E = \{u, v, w\}$
$\pi : E \to N \times N$
with
$\pi(u) = (n1, n2)$,
$\pi(v) = (n1, n3)$,
$\pi(w) = (n3, n4)$

(1) is an *algebraic inscription* of a tree. Other inscriptions are possible; e.g. it is possible not to introduce the map π and to directly define E as a set of ordered pairs, i.e. as a *binary relation* on N.

Trees are often labeled, i.e. their nodes or their edges can be associated with labels; e.g. the nodes of (1) could be labeled using a labelling map λ as follow: $\lambda(n1) = a$, $\lambda(n2) = b$, $\lambda(n3) = c$, $\lambda(n4) = d$.

Fig. 2 depicts three alternate *graphical inscriptions* of the labeled structure of (1), where strokes or arrows correspond to edges. Nodes are either represented by discs or by their labels. Other variants are of course possible.

In an algebraic inscription, it is possible to part the expression of the binary relation that symmetrically links nodes and the direction of this relation, e.g. by using unordered pairs to encode



(a) (b) (c)

Figure 2: Graphical inscriptions of a tree

edges and a typing of the vertices to encode direction. The use of arrows in a graphical inscription (fig. 2(c)) is similar to this typing operation, but direction can be expressed by other means. When directed edges correspond to bare strokes without arrows, direction can be expressed by the verticality of the diagram: the source of the edge is placed at a higher level than the target (fig. 2(a,b).

2.2 Reification

In graphical trees, nodes and edges are turned into discrete graphical objects. This encoding operation is called *reification* (from Lat. *rēs* 'thing'; hence *to reifiy* 'to turn into a thing'). Theoretical objects can be expressed by graphical objects, in which case, they are indeed reified (Kahane and Mazziotta, 2015; Mazziotta, 2016b). However, as illustrated by the alternative between the use of arrows or the use of vertically ordered strokes, the fact that diagrams are drawn on a bidimensional plane allows for the configurational expression of theoretical objects. Configurational expression competes with reification – e.g. in phrase structure trees (henceforth *PST*), words are often linearly ordered, which is a configurational means of expression of their precedence relations; this precedence could be reified by arrows instead.

As an example of linguistic entities that are conceived as distinct notions in the argumentation but not reified in the diagrams, one can introduce S.W. Clark's diagrams. The diagrams in his *Practical grammar* (1847), a pedagogical handbook on the grammar of English, do not reify the relations between the words – see Mazziotta's comprehensive study (2016a), although the text acknowledges that some words *modify* or *complete* others. In the diagrams, words are depicted as labeled bubbles that are but aggregated to one another (fig. 3).

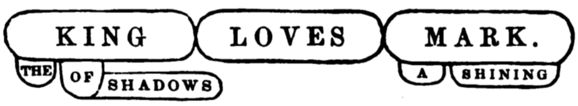


Figure 3: Bubble diagram (Clark, 1847, 23)

It is clear in Clark's diagrams that bubbles in contact correspond to word in syntagmatic relation (cf. section 3.2). Their configuration conveys information about the syntactic analysis they encode. It is possible to reify these contacts and we obtain a diagram that, intuitively, is very similar to a classical dependency tree (fig. 4) – the only difference is that the connection between the verb and the subject and between the verb and the object are not directed.

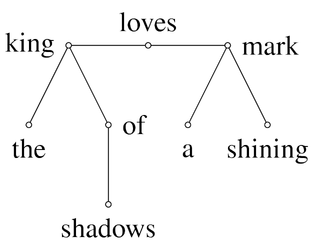


Figure 4: Clark's diagram, reified

In the diagrams, the choice of what is reified and what is not is closely bound to the theoretical stance chosen, but, as it will appear, some options are not always taken in full awareness.

3 What does dependency-based mean?

The difference between constituency and dependency is presented through their use of tree structures under 3.1 and the definitional attributes of dependency trees are reviewed under 3.2.

3.1 Phrase structure trees vs. dependency trees

Since trees are pure formal objects, they imply no *a priori* interpretation as such. The formal objects in a tree (or a graph) can represent different kinds of relations, with respect to the theoretical framework they are conventionally correlated to. The edges of PST do not represent the same information as the edges of dependency trees.

Bloomfield does not provide any ICA diagram, but he quite clearly defines constituents in terms of part-whole relations (1933, § 10.2):

> A linguistic form which bears a partial phonetic-semantic resemblance to some other linguistic form, is a *complex form*. The common part of any (two or more) complex forms is a linguistic form; it is a *constituent* (or *component*) of these complex forms. The constituent is said to be *contained in* (or to be *included in* or to *enter into*) the complex form.

Accordingly, in a PST, edges represent part-whole relations between a phrase and one of its immediate constituent.[1] This kind of relation can be called a *constituency relation*. Consequently, diagrams containing constituency relations will be said *constituency-based* (Kahane and Osborne, 2015, lv).

In a classical dependency tree, such as fig. 5, edges represents *dependencies* between pairs of words. The rationales at work are not the same at all: dependency trees match the five definitional attributes described in section 3.2.

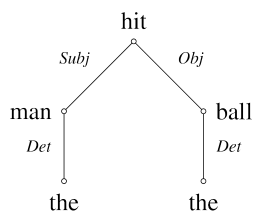


Figure 5: Dependency tree

[1]The widespread use that consists in calling constituents the nodes of a phrase structure tree (cf. "constituency tests") rather than to use the term *constituent* as a relational term denoting an (immediate) constituent of a phrase is confusing at best. The term *constituent* will be used in this latter sense, as it is in the first works on constituency, since we think it fits ICA better.

3.2 Dependency trees: definitional attributes

Dependency trees have five theoretical attributes that distinguish them from phrase structure trees, namely: *connection-basedness*, *binarity*, *headedness*, *flatness*, and *node-to-word mapping*.

As a preliminary remark, word order is abstracted away from the following discussion. It is generally assumed that PSTs encode word order: many of them actually represent the order of the words by sequentially organizing their terminal nodes from left to right (or the opposite, depending on the language). By contrast, dependency trees often encode other pieces of information by the same means – e.g., in Tesnière's stemmas, the dependents of the verb are linearly organized with respect to their status (the subject comes first, then the object, etc.). However, the correspondance between the order of the words and the sequence of terminals in a PST necessitates the tree to be projective.[2] Additionally, a genuine dependency tree can encode word order with the same restrictions as a PST (Groß and Osborne, 2009).

Connection-basedness. Words combine pairwise, they are in a *syntagmatic* relationship in the sense of de Saussure (2013(1916), 170):

> Words as used in discourse, strung together one after another, enter into relations based on the linear character of languages. Linearity precludes the possibility of uttering two words simultaneously. They must be arranged consecutively in spoken sequence. Combinations based on sequentiality may be called *syntagmas*. The *syntagma* invariably comprises two or more consecutive units: for example, *re-lire* ('re-read'), *contre tous* ('against all'), *la vie humaine* ('the life of man'), *Dieu est bon* ('God is good'), *s'il fait beau temps, nous sortirons* ('if it's fine, we'll go out').

Since the term *syntagma* has been led astray – this is especially the case in French linguistic: Fr. *syntagme* has been used to translate *phrase* (Chomsky, 1969(1957)) –, we suggest to use the term *connection* introduced by Tesnière (2015(1959), ch. 1, § 3-5):

> Each word in a sentence is not isolated as it is in the dictionary. The mind perceives **connections** between a word and its neighbors. The totality of these connections forms the scaffold of the sentence. [...] [A] sentence of the type *Alfred speaks* is not composed of just the **two** elements, *Alfred* and *speaks*, but rather of **three** elements, the first being *Alfred*, the second *speaks*, and the third the connection that unites them – without which there would be no sentence.

Elaborating from this quotation, we call *connection* the undirected relation underlying any dependency.[3] Hence, in a dependency tree, syntagmatic relations are encoded by edges. By contrast, in a PST, edges represent constituency relations – see also (Mel'čuk, 1988, 13-14). Analyses and diagrams that make use of connections to describe the syntactic structure of constructions are *connection-based*.

Binarity. In a dependency tree, a connection always involves exactly two words. In a PST, a phrase can have more than two immediate constituents. Binarity is a central property of ICA until the 60's and still remains preeminent.[4] It seems that binarity is the consequence of the connection-basedness of these ICAs. Non-binary structures appear later, cf. fig. 6 (Chomsky, 1965, 65).[5]

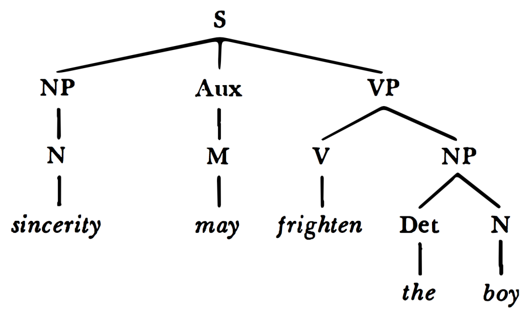


Figure 6: First PST in (Chomsky, 1965)

Headedness. Connections are directed, as explained by Tesnière (2015(1959), ch. 2, § 1-3):

[2]See (Gerdes, 2006) for an in-depth discussion on the relation between X-bar syntax and word order and its consequences.

[3]Tesnière's theory actually lacks a term to designate such a general undirected relation: his *connexion structurale* is equivalent to a dependency.

[4]Some ternary constructions are considered, such as the coordination (Wells, 1947, § 53 sqq.) and (Hockett, 1958).

[5]This first diagram in (Chomsky, 1965) is a tree containing unary, binary, and ternary branchings.

> Structural connections establish **dependency** relations between words. In principle, each connection unites a **superior** term and an **inferior** term. The superior term is called the **governor**, and the inferior term the **subordinate**. Thus in the sentence *Alfred speaks* (Stemma 1), *speaks* is the governor and *Alfred* is the subordinate. We say that the subordinate depends on the governor and that the governor governs the subordinate. Thus in the sentence *Alfred speaks* (Stemma 1), *Alfred* depends on *speaks*, and *speaks* governs *Alfred*.

We call this property *headedness*.

It is noteworthy to mention that although the notion of *head* is absent from (Chomsky, 1957), headedness is considered as a central notion in many early ICA-based presentations, and especially in (Bloomfield, 1933). Bloomfield's work emphasizes constituency relations, but connections are also considered: "Every syntactic constructions shows us two (or sometimes more free forms combined in a phrase, which may call the *resultant* phrase." (§ 12.10) This last definition allows Bloomfield to oppose *endocentric* vs. *exocentric* constructions, according to the fact that the resultant phrase may belong or not to the "form-class" (i.e. distributional class) of one of the constituents (called the *head*). In a dependency tree, every construction is *endocentric*, i.e. connections are directed from a governor to a dependent. In a PST, endocentric constructions can be encoded by marking one of their constituents as the head.

Flatness (i.e. absence of stratification). In a dependency tree, dependents that have the same governor are not hierarchized. In a PST, phrases are embedded: if a head word has several complements (or specifiers, or adjuncts), each of them can belong to a different stratum (Kahane, 1997; Kahane and Mazziotta, 2015). E.g., the dependency tree of a sentence such as *Mary gives Peter a book* represents *Mary*, *Peter* and *a book* as co-dependents of *gives* that belong to the same level, whereas a PST of the same sentence can attach *Mary*, *Peter* and *a book* at different levels. Stratification remains the main difference between dependency syntax and ICA-based syntax. This point will be developed in Section 4.

Node-to-word mapping. Dependency trees do not encode connections by the means of nodes: these are used exclusively to encode words.[6] As a result, one can state:

> A dependency structure for a sentence is a one-to-one mapping between the nodes of a tree (the dependency tree) and the words of the sentence. (Kahane, 1996, 45)

By contrast, classical PST use nodes to encode words as well as constituents. Thus the mapping between nodes and words is not one-to-one. As it will appear in the next section, node-to-word mapping does not imply flatness.

As soon as additional nodes are introduced, labels on these nodes can be used to reify other information. E.g., X-bar syntax (Chomsky, 1970) uses *XP* vs. *X* labels to express headedness.

Summary. The definitional attributes can be summarized in a table (tab. 1). In the next section, ICA diagrams wil be evaluated in comparison with this table.

	Connection	Binarity	Headedness	Flatness	Node-to-word
Dependency tree (fig. 5)	×	×	×	×	×

Table 1: Definitional attributes of dependency trees.

4 Interpreting ICA diagrams

Chomsky's commentary on the diagram of fig. 6 deserves to be mentioned: "The interpretation of such a diagram is transparent, and has been frequently discussed elsewhere." (Chomsky, 1965, 64). The assumed "transparency" of syntactic diagrams in general could lead to overlook important characteristics that only emerge when the graphical elements are scrutinized.

A stroke, an arc, or an arrow in a diagram generally correspond to an edge of a binary rela-

[6] It should be noted that the very definition of the term *word* has to be stated precisely. We assume that, in a dependency tree, words are abstract units. Depending on the descriptive stance chosen, they can be "zero" forms as well as elements of amalgamated complexes, such as Fr. *au* = *à* 'to' + *le* 'the' (Mel'čuk, 1988, 15).

tion.[7] From the perspective of a linguistic analysis, such an edge in a syntactic diagram reifies a constituency relation or a connection.

4.1 Chomsky, 1957

Chomsky's first diagram (fig. 1(a)) displays a continuous arc between NP and VP nodes and a small stroke between the S node and this arc. The diagram is introduced in the text. Chomsky first introduces the rewriting rules in the first page of ch. 4, entitled "Phrase structure":

> As a simple example of the new form for grammars associated with constituent analysis, consider the following: (13) (i) *Sentence* $\rightarrow$ *NP* + *VP* [...] Suppose that we interpret each rule $X \rightarrow Y$ of (13) as the instruction "rewrite X as Y". [...] [T]he second line of (14) is formed from the first line by rewriting *Sentence* as *NP* + *VP* in accordance with rule (i) of (13) [...] We can represent the derivation (14) in an obvious way by means of the following diagram.

It seems reasonable to interpret the arc between the NP node and the VP node in fig. 1(a) as a notation of the relation between the nodes: they combine to form *NP* + *VP*. Moreover, the operation corresponding to this connection is noted down in the rewriting rule (i.e. the algebraic inscription) by the symbol "+". Accordingly, the arc between NP and VP would reify the syntagmatic combination of *NP* and *VP*, i.e. a connection edge. The small stroke that stands between the S node and this arc reifies the rewriting operation: *Sentence* is rewritten as *NP* + *VP*. This corresponds to the symbol "$\rightarrow$" in the algebraic inscription. According to this interpretation, the small stroke and the arc are to be considered as the reifications of two distinct elements that encode two binary relations: the connection between the ICs and the rewriting operation.

Headedness is partially encoded in an indirect way: by using similar labels for *NP* and *N*, the diagram shows that *N* is the most important element in the NP.

The diagram is not a dependency tree, but it shares some of the definitional attributes of such structures (as shown in tab. 2).

	Connection	Binarity	Headedness	Flatness	Node-to-word
Chomsky, 1957 (fig. 1(a))	×	×	?		

Table 2: Description of fig. 1(a) with respect to definitional attributes of dependency trees.

Constituency relations are not reified in the diagram, whereas connections are. Could it be that previous ICA diagrams share this characteristic? To answer this question, the rest of this section scrutinizes previous and contemporary ICA diagrams in a chronological order.

4.2 Barnard, 1836

To our knowledge, the first diagram representing an ICA (fig. 7) appears in Frederick A. P. Barnard's *Analytic Grammar with Symbolic Illustrations* (1836). Syntactic categories of units are represented by special symbols and braces that indicate in a configurational way that a list of units combine together to form another unit. In his text, Barnard compares *man* and *a rational animal* or *quadruped* and *a four-footed animal* and says (Barnard, 1836, 243-244):

> We thus construct *phrases* standing in the *places of nouns*, and answering all their purpose. [...] Contemplating, then, a noun and its adjective, we say that they constitute, together, a compound noun. Contemplating an adjective and its accompanying adverb, we say, in like manner, that they constitute a compound adjective.

E.g., in fig. 7, *in* and *disposition* form together a unit with the same category as *very* and *who is mild* and *in disposition* form together a unit with the same category as *many*.[8]

Barnard's diagrams have no discrete means to express individual part-whole relations: the brace

[7] It is not always the case. For instance, (Reed and Kellogg, 1876) makes use of syntactic diagrams where words are represented as labeled strokes, which connect to each other to represent the way they combine. See also the discussion on Nida's diagrams below (section 4.3).

[8] Categories are represented by symbols in Barnard's diagrams. These symbols are probably inspired by symbols used for sign language writing systems, since Barnard was a 27-year-old professor of English in a deaf institute when his book was published. The fact that he taught deaf people is likely to be the reason for the use of diagrams in his book.

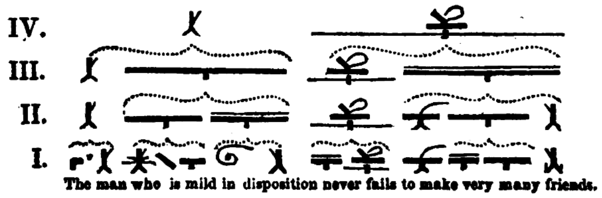


Figure 7: Barnard's diagram (1836)

is equivalent to Chomsky's rewriting operator as well as the "+" symbol, linking a phrase with the entire set of its immediate constituents. There is no independent reification for the two operations. Syntagmatic relations are not represented in a discrete way either. The brace inscribes the whole construction. According to our terms (section 3), such a diagram is thus neither exactly connection-based nor exactly constituency-based.

As shown in tab. 3, the diagram is very different from a canonical dependency tree: not a single definitional attribute firmly holds.

	Connection	Binarity	Headedness	Flatness	Node-to-word
Barnard, 1836 (fig. 7)	?				

Table 3: Description of fig. 7 with respect to definitional attributes of dependency trees.

4.3 Nida, 1943; 1966

It seems that Barnard's diagram was overlooked by his contemporaries. More than one century passed between this attempt and the next ICA diagram.[9] It appears in Nida's *Morphology* (1949(1943), 87).[10] Fig. 8 shows the first ICA diagram published by Nida and fig. 9 is a diagram from (Nida, 1966).

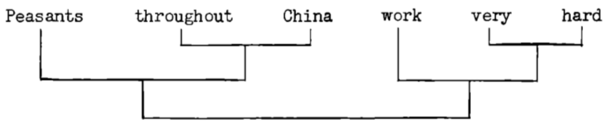


Figure 8: Nida's first diagram (1949(1943))

[9]In the mid time, other diagrams, which are much more dependency-based and that will not be discussed here, have been proposed by several authors (Clark, 1847; Reed and Kellogg, 1876; Kern, 1883; Tesnière, 1934).

[10]We could not access the fist edition of Nida's *Morphology* (1943).

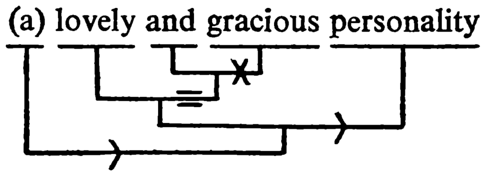


Figure 9: Nida's diagram (1966)

At first glance, it would seem that Nida's first diagram could be interpreted as a PST. It is tempting to consider that fig. 8 is completely equivalent to fig. 10, where constituency relations are reified as distinct graphical entities.

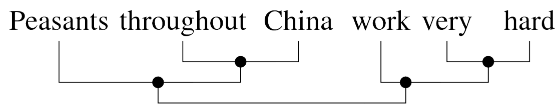


Figure 10: Nida, 1943's diagram, reified

However, fig. 9, which elaborates on the same rationales as fig. 8, demonstrates that it is not the case. Both diagrams consist of arcs between words and arcs between words and other arcs. Every single node in these diagrams corresponds to a word. Thus, the contact point between strokes are not equivalent to reifications, since they are not discrete graphical entities and they possibly allow for several interpretations.

To fully understand fig. 9, let us recall that Nida's work was preceded by Bloomfield's seminal text on constructions (section 3.1). Hence, in his fig. 9, arcs bear additional symbols (">", "×", "=") and the accompanying text clearly explains how to interpret them (Nida, 1966, 17):

> In addition to the usual set of lines used to show relationships between immediate constituents, an additional set of symbols has been employed to mark exocentric, endocentric, and paratactic relationships.

Consequently, the labels over the strokes reify the headedness of the connections. Nida's diagrams are connection-based and not constituency-based. Such a diagram is close to a dependency tree. The only difference between classical dependency trees and Nida's diagrams is that the later are not flat, but stratified: connections are ordered and hierarchized. The consequence of such an analysis is that connections can be connected to one another.

From a mathematical perspective, this means that edges can have other edges as vertices – see (Kahane and Mazziotta, 2015) for a formalization of such a structure, that can be called a *polygraph*.

Tab. 4 shows that the evolution between fig. 8 and fig. 9 consists in encoding headedness in the diagram. Fig. 9 is almost a dependency tree: the only attribute that does not hold is flatness.

	Connection	Binarity	Headedness	Flatness	Node-to-word
Nida, 1943 (fig. 8)	×	×			×
Nida, 1966 (fig. 9)	×	×	×		×

Table 4: Description of fig. 8 and 9 with respect to definitional attributes of dependency trees.

4.4 Wells, 1947

Rulon S. Wells (1947) is more interested in constituency relations than in constructions seen as wholes. The term *construction* itself is used in another meaning – "The reader must constantly bear in mind that our definition of this term is not the same as Bloomfield's" (Wells, 1947, note 19). He proposes a linear diagram (fig. 11).

the || *king* ||| *of* |||| *England* | *open* ||| *ed* || *Parliament*

Figure 11: Well's diagram (1947)

This diagram (Wells uses this very term to designate this inscription) corresponds to the following analysis (Wells, 1947, 84):

> Let us call the ICs of a sentence, and the ICs of those ICs, and so on down to the morphemes, the CONSTITUENTS of the sentence; and conversely whatever sequence is constituted by two or more ICs let us call a CONSTITUTE. Assuming that the ICs of *The king of England opened Parliament* are *the king of England* and *opened Parliament*, that those of the former are *the* and *king of England* and those of the latter are *opened* and *Parliament*, and that *king of England* is divided into *king* and *of England*, *of England* is divided into the morphemes *of* and *England*, and *opened* is divided into *open* and *-ed*-all of which facts may be thus diagrammed [by fig. 11] "

Although this analysis is purely based on the decomposition of wholes ("constitutes") into parts ("constituents"), the symbols made of "|" in Wells's diagrams reify the combination/separation operations (according to the perspective, that can be deductive or inductive) of the elements around them. In a sense, they correspond more to connections than to constituency relations.

Tab. 5 shows that Wells's diagram is equivalent to Nida's first diagram (fig. 8).

	Connection	Binarity	Headedness	Flatness	Node-to-word
Wells, 1947 (fig. 11)	×	×			×

Table 5: Description of fig. 11 with respect to definitional attributes of dependency trees.

4.5 Gleason, 1955

H. A. Gleason's handbook (1961(1955)) also contains interesting diagrams.[11] Gleason has a clear bottom-up vision of the ICA. Considering the sentence *The old man who lives there has gone to his son's house*, he says (Gleason, 1961(1955), § 10.3):

> We may, as a first hypothesis, consider that each of [the words] has some stat-able relationship to each other word. If we can describe these interrelationships completely, we will have described the syntax of the utterance in its entirety. [. . .] At a second step in our procedure, let us assume that these pairs of words function in the utterance as single units. [. . .] If this procedure is valid, there is no reason why it cannot be repeated as many times as may be useful. Something like the following [diagram] might result.

In the mentionned diagram (fig. 12), braces indicates the units that combine together as in Barnard's diagrams (cp. fig. 7).

A characteristic of Gleason's handbook is that it introduces alternate diagrams to inscribe the same

[11] We could only manage to access the 1961 edition and we don't know if diagrams have been changed.

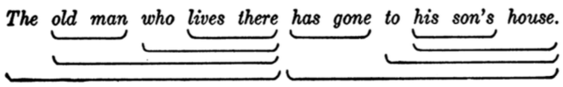


Figure 12: Gleason's first ICA diagram

analysis. Fig. 13 is similar to Wells's diagrams, but where the hierarchy of frontiers is inverted. Gleason, who starts from the bottom, use thin stroke for the most embedded connection, while Wells, who starts from the top, use them for main segmentation of the sentence.

Figure 13: Gleason's second ICA diagram

Gleason introduces a third concurrent diagram (fig. 14) as follows (Gleason, 1961(1955), *ibid.*):

> The procedure which we have just sketched will be useful to us, if it serves as a framework within which all the relationships of the utterance can be effectively and economically described.

This is done in the following diagram, where the heavier line is "intended to indicated the most direct relationship between *old* and *house* [. . .] describable in terms of a chain of relationships each of which individually seems significant."

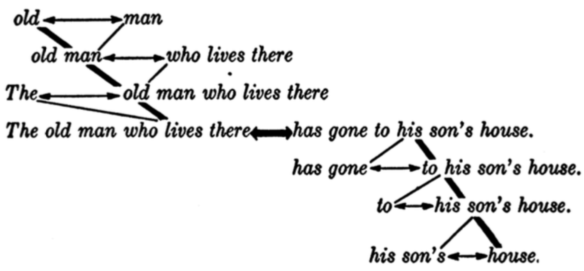


Figure 14: Gleason's third ICA diagram

This last diagram clearly provides both constituency relations (reified by mere strokes) and connections (reified by double arrows). The book does not contain any diagram that is exactly a tree.

The attributes of Gleason's diagrams are summarized in tab. 6.

4.6 Hockett, 1958

Hockett (1958) formalizes the concept of *construction* by the means of diagrams consisting of

	Connection	Binarity	Headedness	Flatness	Node-to-word
Gleason, 1955 (fig. 12)	×	×			×
Gleason, 1955 (fig. 13)	×	×			×
Gleason, 1955 (fig. 14)	×	×			

Table 6: Description of fig. 12 to 14 with respect to definitional attributes of dependency trees.

embeddable three-compartment boxes (fig. 15). Two compartments represent immediate constituents and the lower compartment represents the resultant phrase. These boxes can be embedded to give the whole ICA of a sentence (Hockett, 1958, 160-161):[12]

> Sentence A consists of only two ultimate constituents (morphemes), which are therefore also the ICs of the whole sentence: 3 and 2 are the ICs of 1. Sentence B consists of more than two *ultimate* constituents, but, once again, of only two *immediate* constituents: 3 and 2 as in A, are the ICs of 1. Similar remarks apply to sentences C and D. Furthermore, the relationship between the two ICs of each whole sentence is the same. Thus, if we make just one IC-cut in each sentence, ignoring any smaller constituents for the moment, then all four sentences conform to pattern X.

Hockett's boxes can be typed by an additional symbol, "<" or ">", "placed at each junction of ICs, pointing from attribute to head" (fig. 16).

We can observe that, in Hockett's diagrams, constituency relations and connection are indissociable and none of them is favored, although the additional symbols ("<" or ">"), similar to Nida's (1966), are clearly connection-based.

	Connection	Binarity	Headedness	Flatness	Node-to-word
Hockett, 1958 (fig. 16)	?	×	×		

Table 7: Description of fig. 16 with respect to definitional attributes of dependency trees.

[12]Numbers in the text correspond to numbers in the lower right-hand corners of compartments.

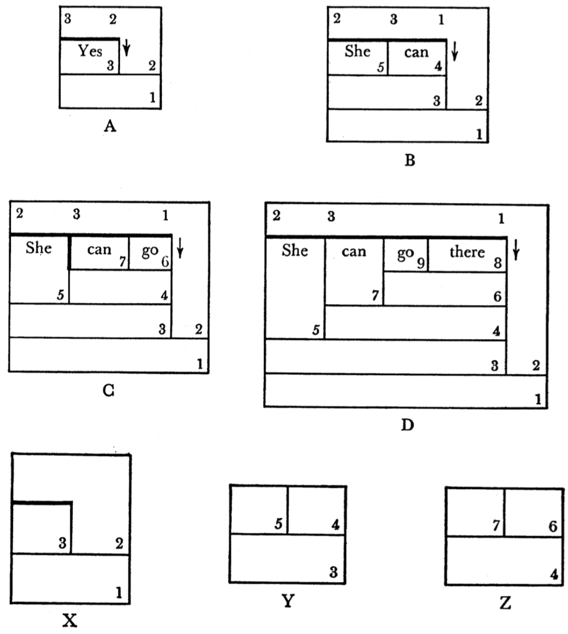


Figure 15: Hockett's boxes (1958)

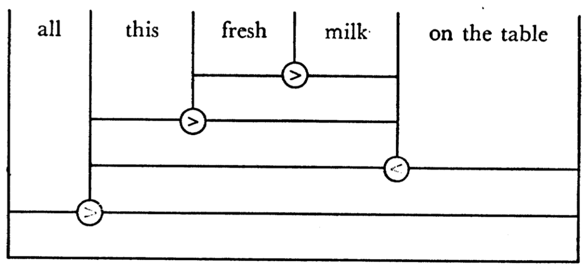


Figure 16: Endocentric construction in Hockett's diagram (1958)

5 Conclusion

Immediate constituent Analysis has been modeled by phrase structure trees only from the middle of the 1960's on. Chomsky's first derivation diagrams is not a genuine modern phrase structure tree; it is partly connection-based and it also contains other edges. Previous ICA diagrams by Nida are totally connection-based. Contemporary diagrams by Hockett or Gleason are more connection-based than constituency-based.

Tab. 8, which merges all previous tables, clearly shows that: (i) until fig. 1(b), all ICA diagrams encoded connections to a certain extent; (ii) the only constant difference between a dependency tree and a PST is the flatness of the former (opposed to the stratification of the later).[13]

	Binarity	Connection	Node-to-word	Headedness	Flatness
Barnard, 1836 (fig. 7)		?			
Chomsky, translated (fig. 1(b))	×				
Gleason, 1955 (fig. 14)	×	×			
Chomsky, 1957 (fig. 1(a))	×	×		?	
Hockett, 1958 (fig. 16)	×	?		×	
Gleason, 1955 (fig. 12)	×	×	×		
Wells, 1947 (fig. 11)	×	×	×		
Nida, 1943 (fig. 8)	×	×	×		
Nida, 1966 (fig. 9)	×	×	×	×	
Dependency tree (fig. 5)	×	×	×	×	×

Table 8: Comparison of the diagrams with respect to definitional attributes of dependency trees (rows and columns are arranged for better visualization).

These connection-based diagrams are very close to dependency trees, since they (at least partially) consist of reified connections rather than reified constituency relations. By contrast, modern PSTs do not reify connections directly: one has to infer them from specific configurations. The seemingly trivial differences between the diagrams in fig. 1 are actually very important from the perspective of the history of linguistics. The diagrammatic habits led their users to ignore connections. In consequence, original diagrams were reinterpreted. Fig. 1(b) was already understood as a faithful copy of fig. 1(a) at the time the book was translated into French, and the interpretation of fig. 6 was considered completely transparent by its author. This progression demonstrates that the tools we use to model and to inscribe knowledge about language have a dramatic epistemological impact.

Acknowledgements

The authors would like to thank Kim Gerdes, Timoty Osborne and the anonymous reviewers of the Depling conference for their comments, suggestions and debates.

References

Frederick Augustus Porter Barnard. 1836. *Analytic Grammar, with symbolic illustration.* French, New York.

[13]It is possible to use PSTs for diagramming flat structures, but there is no obvious advantage in using PSTs instead of dependency trees.

Leonard Bloomfield. 1933. *Language*. The University of Chicago Press.

Noam Chomsky. 1957. *Syntactic structures*. Mouton, The Hague.

Noam Chomsky. 1965. *Aspects of the Theory of Syntax*. The MIT Press, Cambridge.

Noam Chomsky. 1969(1957). *Structures syntaxiques,* translation of (Chomsky, 1957) by M. Braudeau. Seuil, Paris.

Noam Chomsky. 1970. Remarks on nominalization. In *On the Nature of Grammatical Relations*, pages 184–221. Ginn and Co., Waltham, Mass.

Stephen W. Clark. 1847. *The science of the English language. A practical grammar; in which words, phrases, and sentences are classified according to their offices, and their relation to each other. Illustrated by a complete system of diagrams*. A. S. Barnes & Co. and Derby, Bradley & Co., New York and Cincinnati.

Eugenio Coseriu. 1980. Un pécurseur méconnu de la syntaxe structurale, H. Tiktin. In *Recherches de linguistique: hommage à Maurice Leroy*, pages 48–62. Bruxelles.

Kim Gerdes. 2006. Sur la non-équivalence des représentations syntaxiques: Comment la représentation en x-barre nous amène au concept du mouvement. *Cahiers de grammaire*, 30:175–192.

Henry A. Gleason. 1961(1955). *An Introduction to Descriptive linguistics*. Holt, Rinehart and Winston.

Henry A. Gleason. 1969. *An Introduction to Descriptive Linguistics*. Holt, Rinehart & Winston, New York, revised edition.

Thomas Groß and Timothy Osborne. 2009. Toward a practical dependency grammar theory of discontinuities. *SKY Journal of Linguistics*, 22:43–90.

Charles F. Hockett. 1958. *A course in modern linguistics*. The MacMillan Company.

Sylvain Kahane and Nicolas Mazziotta, 2015. *Proceedings of the 14th Meeting on the Mathematics of Language (MoL 2015)*, chapter Syntactic Polygraphs. A Formalism Extending Both Constituency and Dependency, pages 152–164. Association for Computational Linguistics.

Sylvain Kahane and Timothy Osborne. 2015. Translators' introduction. In *Elements of structural syntax* (Tesnière, 2015(1959)), pages xxix–lxxiv.

Sylvain Kahane. 1996. If hpsg were a dependency grammar... In *Actes de TALN, Marseille*, pages 45–49.

Sylvain Kahane. 1997. Bubble trees and syntactic representations. In *Proceedings of Mathematics of Language (MOL5) Meeting*, pages 70–76.

Franz Kern. 1883. *Zur Methodik des deutschen Unterrichts*. Nicolai, Berlin.

Nicolas Mazziotta. 2016a. Drawing syntax before syntactic trees. Stephen Watkins Clark's sentence diagrams (1847). *Historiographia Linguistica*, 43(3):301–342.

Nicolas Mazziotta. 2016b. *Représenter la connaissance en linguistique. Observations sur l'édition de matériaux et sur l'analyse syntaxique. Habilitation à diriger des recherches, mémoire de synthèse*. Université Paris-Ouest, Nanterre – La Défense, Paris. `http://hdl.handle.net/2268/204408`.

Igor Mel'čuk. 1988. *Dependency syntax: theory and practice*. State University of New York, Albany.

Eugene Nida. 1949(1943). *Morphology: the descriptive analysis of words*. University of Michigan press, Ann Arbor, 2nd edition.

Eugene Nida. 1966. *A synopsys of English Syntax*. Mouton and Co., London, The Hague, Paris, 2 edition.

Alonzo Reed and Brainerd Kellogg. 1876. *Graded Lessons in English*. Clark and Maynard, New York.

Ferdinand de Saussure. 1916. *Cours de linguistique générale*. Payot, Paris.

Ferdinand de Saussure. 2013(1916). *Course in general linguistics,* translation and annotations of (Saussure, 1916) by Roy Harris, with a new introduction by Roy Harris. Bloomsbury, London and New York.

An Harleman Stewart. 1976. *Graphic representation of models in linguistic theory*. Indiana university press, Bloomington and London.

Lucien Tesnière. 1934. Comment construire une syntaxe. *Bulletin de la Faculté des Lettres de Strasbourg*, 7:219–229.

Lucien Tesnière. 1959. *Éléments de syntaxe structurale*. Klincksieck, Paris.

Lucien Tesnière. 2015(1959). *Elements of structural syntax,* translation by Timothy Osborne and Sylvain Kahane of (Tesnière, 1959). Benjamins, Amsterdam/Philadelphia.

Rulon S. Wells. 1947. Immediate constituents. *Language*, 23(2):81–117.