The Representation of Discontinuity and the Correspondence Principle

Ratna Nirupama
Indian Institute of Technology Hyderabad
Sangareddy
Telangana, India
la19resch11006@iith.ac.in

Prakash Mondal
Indian Institute of Technology Hyderabad
Sangareddy
Telangana, India
prakashmondal@la.iith.ac.in

Abstract

Discontinuity is a nearly universal phenomenon observed in natural languages. Several approaches have been proposed so far by different grammar formalisms but they are widely regarded as distinct approaches owing to their theoretical motivations. This paper proposes the *correspondence principle* which will enable the representation of discontinuity by way of the unification of the representations of linguistic structures in three grammar formalisms: Phrase Structure Grammar (PSG), Dependency Grammar (DG), Categorial Grammar (CG). The goal is not to unify PSG, DG and CG, but rather to sketch out a way of representing discontinuity by uniting constituency relations (as in PSG), head-dependent relations (as in DG) and functor-argument relations (as in CG) for the encoding of discontinuous expressions in natural languages. The implications for natural language syntax and computational linguistics will be offered towards the end of the paper.

1 Introduction

Syntactic discontinuity is a grammatical phenomenon in which a constituent of a sentence is split into two (or more) parts because of the insertion of an element which is not a part of the constituent. The evidence for discontinuity is frequently found in languages with relatively free word-order such as Turkish, Russian, Japanese, Croatian, German, Tamil, Warlpiri etc. In the sentence from Malayalam below, the predicate and its argument are not contiguous as per the linear order because the subject is located between them, indicating a case of discontinuity.

i.	*Kaṇṭu*	*kuṭṭi*	*aanaye*.	(Falk, 2001:19)
	saw	child.NOM	elephant.ACC	

‘The child saw the elephant.’
[NOM=nominative case marking;
ACC=accusative case marking]

It is also observed in rigid word order languages such as English, but is limited to long-distance dependencies such as topicalisation, long-distance *Wh*-questions etc. Various theories of grammar have accounted for discontinuity in natural languages (both rigid and free word order) in different ways, as per their theoretical motivations. For example, PSG has rules and analyses syntactic structures only in terms of constituents/phrases, making it well-nigh impossible to account for discontinuous constituents. DG and related formalisms accommodate discontinuous constituents by analysing expressions on the basis of head-dependent relations. The paper introduces *the correspondence principle* which will help in the unification of the representations of linguistic structures in these grammar formalisms for some cases of discontinuity. Firstly, a brief introduction to each of the grammar formalisms is given followed by a mention of the previous approaches towards discontinuity. This is followed by an illustration of *the correspondence principle* which will help in the derivations, considering one discontinuous sentence from Croatian language. Finally, a brief conclusion is provided towards the end of the paper.

2 The Three Grammar Formalisms: PSG, DG and CG

2.1 PSG

Analysing sentences as constituents is central to the PSG formalism which was first put forth by Noam Chomsky in his book 'Syntactic Structures' (1957) and later developed in the Extended Standard Theory, The Revised Extended Standard Theory, Government/Binding theory and the Minimalist Program. They all exhibit certain common characteristics: all syntactic representations are analysed as phrases and depicted using trees; the grammatical functions are derived from the constituent structures; the configuration of the subject is higher and external; certain operations called transformations (hence transformational grammar) on an existing constituent structure change it into a similar, but not identical, constituent structure called the 'surface structure'. Thus, the traditional PSG greatly emphasizes the interdependence of grammatical relations, thematic roles and constituency. Rewriting rules based on PSG trees specify the manner in which each of the word/phrase is combined to form constituents. For instance, in the discontinuous sentence (i), PSG depicts the predicate *Kaṇṭu* and its argument *aanaye* as V and NP respectively (S → V NP NP). However, *Kaṇṭu* and *aanaye* are considered to form a single VP constituent (VP → V NP) if it were a continuous sentence. Thus, any word order variations are encoded in the rewriting rules based on PSG trees depicting the dominance and linear relations among various constituents (Gazdar, 1983; Chomsky, 1995; Newmeyer, 2001).

2.2 DG

DG is a descriptive tradition in linguistics that can be traced back to Panini and was later developed by the French linguist Lucien Tesnière (1959). It analyses sentences in terms of head and dependent relations, motivated by the grammatical functions. A DG can be specified by a 4-tuple: DG = $<V_N, V_T, D, R>$ where V_N is the set of auxiliary/non-terminal items (syntactic categories), V_T is the set of terminal items (actual words realized from syntactic categories), D is the set of dependency rules and R is the initial symbol at the root of the tree (that is, $R \in V_N$). A dependency rule in D is a statement consisting of one auxiliary element functioning as the governing element or head (I) and any finite number of auxiliary elements as the dependents. There are two important rules in D: Rule 1: $I(D_1,\ldots,D_m * D_i,\ldots,D_k)$ (i, m, k ≥ 0 ; not always i=m=k) ; Rule 2: I (*); 'I' is the governing element and indicates the presence of only one independent category; $D_1,\ldots,D_m$ represents the dependent categories towards the left of the root word/head; $D_i,\ldots,D_k$ represents the dependent categories towards the right of the head word; *m* and *k* are the number of dependents on the left and right of the head word respectively. The asterisk '*' indicates the location of 'I' in the linear order of words. As per this rule, the valence of 'I' will be the total number of dependent elements i.e. '(m+k)'. Therefore, the terminal elements, the non-terminal elements, the correspondences (dependency functions) which exist between them, and the rules constitute the core of the dependency theory (Hays, 1964; Gaifman, 1965; de Marneffe and Nivre, 2019). Accordingly, the rule in D for the sentence *i* is V(*NP, NP) which will be realized as *Kaṇṭu* (**kuṭṭi*, *aanaye*). This indicates that V *Kaṇṭu* is the head and NPs *kuṭṭi* and *aanaye* are its dependents. The central idea is that in a sentence, all except one word ('the root') depend on another word. A dependent Y depends on a head X when Y is usually optional with respect to X, and/or X selects Y, and/or Y agrees or is governed by X, and/or the linear position of Y is with reference to that of X (de Marneffe and Nivre, 2019: 203).

2.3 CG

Thirdly, CG is a context-free grammar formalism first defined by the logician Kazimierz Ajdukiewicz (1935). The notion of 'category' and analysis of sentences in terms of functor-argument relations constitute the core idea of this formalism. Words are assigned a category in terms of N and S, based on their combining properties (Steedman, 1992, 2014). The widely used 'slash' notations for directional categories were pioneered by Bar-Hillel (1953) and Lambek (1958). Lambek's notation uses a forward slash '/' to indicate an argument on the right and a backward slash '\' to indicate an argument on the left. It needs to be emphasized that for the CG analyses in this paper, the standard Lambek notation of functor-argument relations (by using only the categories N and S) has been adopted. Given this basic understanding, the next section makes a brief note on the solutions proposed so far to solve the problem of discontinuity. A CG can be defined by a 4-tuple: CG = <V, C, R, F>. Here, V is the set of all lexical items in a language; C is the set of primitive categories ({N, S}); R is the set of functional composition rules for the generation of categories for lexical items. It specifies the process of generation of category of any given lexical item. F is a function that maps each lexical item (LI) to its set of categories (each element of V is mapped to its corresponding element(s) which can be a set of primitive/atomic categories from the set C and/or categories derived by means of R), whose form is: F(LI) = $\{C_1,\ldots,C_n\}$. For instance, for the Malayalam sentence (i), V = {*Kaṇṭu*, *aanaye*, *kuṭṭi*}; C = {N,S}; as per the definition of R and F, the category of *Kaṇṭu* is '(S/N)/N' (with '/' indicating the argument is to the right) and 'N' is for *aanaye* and *kuṭṭi*. In Step 1 of the CG derivation, *Kaṇṭu* is the functor and its argument *aanaye* is to its right. Here, the cancellation of 'N' results in the output '(S/N)' (*Kaṇṭu aanaye*). In Step 2, (S/N) is the functor and N (*kuṭṭi*) is the argument. Here, 'N' is cancelled out, which results in the final output S. We shall now look into the previous approaches to discontinuity proposed so far in the literature.

3 Previous Approaches to Discontinuity

Questioning the validity of a universal constituent, several linguists have proposed alternative approaches towards discontinuity.

3.1 Characterisation of Non-Configurational Languages

Hale's (1982, 1983) work on Australian languages such as Warlpiri, Navajo and Dixon's (1972, 1977) work on Dyirbal and Yidiny provided rich evidence for the existence of discontinuity in natural languages. Hale associated three key properties with 'non-configurational languages':

(i) free word order

(ii) the use of syntactically discontinuous expressions, and

(iii) the extensive use of the null anaphora (an argument such as subject and object that is not represented by an overt nominal expression in the phrase structure).

This is because the syntactic nature of these languages is not the same as that of more familiar languages which admit of analyses in terms of phrase structure constituency (the structure of a clause, configurations of NPs and VPs), subordination, *wh*-movement and extraposition (Nordlinger, 2014). Austin and Bresnan's (1996) claim of Warlpiri phrase structure as flat and characterised by free base-generation of elements is another approach towards discontinuity.

3.2 Phenogrammatical Structure

Dowty (1996) makes two important assumptions. First, he proposes a 'minimalist theory of syntax' to describe various discontinuous syntactic phenomena by taking linear structure as the norm rather than hierarchical structure, that is, 'a clause or a group of words is *only* a string'. Second, some words and constituents are more tightly bound (attached) to adjacent words than others. The linear structures/representations of expressions are treated as unordered lists.

3.3 Sequence Union Operation/Shuffle

Donohue and Sag's (1999) adopted Reape's (1996) 'sequence union operation' or 'shuffle'. The sequence union of two lists l_1 = <a,b> and l_2 = <c,d> is the list l_3 iff each of the elements in l_1 and l_2 is present in l_3 and the original order of the elements in l_1 and l_2 is preserved. For example, the sequence union of l_1 and l_2 is any of the following lists/sequences: <a,b,c,d>, <a,c,d,b>, <a,c,b,d>, <c,d,a,b>, <c,a,d,b>, <c,a,b,d> but not <b,a,c,d>, <a,b,d,c> etc. This allows discontinuous elements to intervene in the linear order of a constituent, thus accounting for discontinuity.

3.4 Tangled Trees

McCawley's tangled trees (McCawley, 1982; Iwakura, 1988; Blevins, 1990) relax the *no-crossing constraint* and the *single mother condition* of the standard PSG trees to account for discontinuity.

3.5 Parallel Merge

Citko's (2011) 'parallel merge' relaxes the *single root/mother condition* to linearize multidominant structures, thus accounting for discontinuous structures.

3.6 Encoding Constituents in terms of Dependency Relations

These include Barry and Pickering's (1990, 1993) 'dependency constituent' linking dependency relations with constituent relations, the formulation of 'subtrees' from DG trees by Hays (1964), Gaifman's (1965) formulation of weak equivalence between 'parenthetical expressions' of PSG trees and dependency graphs.

However, these are limited to showing the correspondences between PSG and DG. This paper goes beyond this and attempts to show a way of unifying CG functor-argument relations with DG head-dependent relations and constituency rules in PSG. Now we shall introduce *the correspondence principle* and the motivations behind this proposal.

4 The Correspondence Principle

Before proceeding to show the desired way of uniting the representations, an elaboration of the principle proposed here, called as the *Correspondence Principle* is noteworthy. In order to achieve a unified system of representation, one needs to establish an equivalence relation between (a) PSG and CG and also between (b) DG and CG. The CG derivations would piggyback on PSG constituents in the analyses as the CG derivations proceed as per the constituency relations in PSG with the *wrapping* [1] operation allowed for the functor-argument distance over more than one (constituent) expression, but the CG relations defined on the relevant constituents have to be mapped onto the dependency relations. This warrants a principle that can help unify the DG and CG representations. Therefore, *the correspondence principle* has been proposed.

5 Motivations for The Correspondence Principle

PSG trees fail to capture discontinuous constituents unless the trees are tangled, that is, the *no-crossing constraint* and the *single mother condition* are relaxed as proposed by McCawley (1982). The 'parallel merge approach' too relaxes the *single mother condition*. This relaxation may seem theoretically gratuitous, because this indicates that the *no-crossing constraint* and the *single mother condition* need to be adhered to in cases of non-discontinuity and these very conditions need to be relaxed in cases of discontinuity. This in turn gives rise to two different and separate structural representations for continuous and discontinuous constituents, the former without the above mentioned two conditions relaxed and the latter, with the conditions relaxed, be it PSG trees or tangled trees. On the other hand, DG captures cases of discontinuity, with CG remaining in-between. Given this situation, there arises incompatibility between PSG, CG and DG in analyses of linguistic structures. The *Correspondence Principle* is the 'glue' that can bind the principles of PSG, CG and DG together in a non-superfluous manner for both continuous and discontinuous structures. Once the Correspondence Principle is applied, the need for the *no-crossing constraint* and the *single mother condition* disappears, precisely because all cases demanding these conditions are re-interpreted and re-analysed in terms of functioning of the basic principles of DG and CG united together. Accordingly, this principle would be used for the DG → CG and CG → DG derivations illustrated in the fifth section of this paper.

The Correspondence Principle:

$$A(B^*) \vee A(^*B) \equiv A|B$$

For any two words A and B, A(B*) indicates B is dependent on A and B is to the left of A and A(*B) indicates B is dependent on A and B is to the right of A. Here, '∨' is the logical disjunction, '≡' is a special equivalence sign and A|B indicates that either A or B can be the functor in categorial relations, with '|' indicating the neutral direction of the functor. This implies that the other element will be the argument. The logical relation is that of an *implication*, but not of *an entailment*, because when one element (A or B) is the functor, nothing is said about the other element. In cases where there is a direct dependency relation between the functor and the argument, A and B on the Left-Hand Side (LHS) and Right-Hand Side (RHS) turn out to be the same. However, this is not the case always. In exceptional cases, only either A or B

[1] Wrapping rules usually infix, by way of a sort of swapping, a discontinuous string element in a place where another element was initially located (see for details, Steedman, 1985: 527).

tends to be the same on LHS and RHS, and the other category can vary across sides. If, for example, we suppose that A is the same on both sides, the exact value of B may differ on the LHS and the RHS (that is, B can take a word X, for example, on the LHS, while it takes a word Y, for example, on the RHS). Given this understanding of the *Correspondence Principle*, we shall now turn to the derivations to apply this principle and arrive at a unified system of representation for a Croatian sentence.

6 Towards a Unified Representation: An Illustrative Case of a Croatian Sentence

This section provides an illustration of the unified system of representation for a discontinuous Croatian sentence. An outline of the strategy followed is given below. The following four steps (not necessarily in the same order) are essential: (a) PSG to CG derivation (b) DG to CG derivation (c) CG to DG derivation (d) CG to PSG derivation. For the PSG to CG derivation, the starting point would be the PSG tree, hence the CG derivation is depicted in the PSG tree. For the CG to PSG derivation, each step of the CG derivation is mapped onto an appropriate PSG constituent. The final PSG tree can be derived after the last step of the CG derivation. For the CG to DG derivation, the CG derivation would be the starting point. For a functor-argument relation in each step of the CG derivation the corresponding head-dependent relation is established. The DG tree of the expression can then be drawn based on the outputs of the individual steps. This is where the proposed *correspondence principle* will come into picture. Similarly, for the DG to CG derivation, for each head-dependent relation in the DG graph a functor-argument relation is derived by using *the correspondence principle*. These account for the forward and converse derivations for establishing the equivalence relations CG ≡ PSG and DG ≡ CG. This has been illustrated for the Croatian sentence below:

ii. *Naša je učionica udobna*.
Our is classroom comfortable
'Our classroom is comfortable.' (Van Valin, 2001:88)

In this sentence, discontinuity arises since *je* and *udobna* are not contiguous in the linear order of the sentence.

(iia) A CG derivation in the phrase structure tree (PSG → CG)

Figure 1 depicts the CG derivation of (ii) in its PSG tree and Figure 2 depicts the CG derivation of (ii).

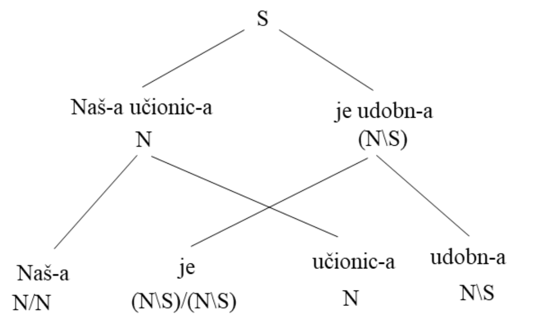

Figure 1. The CG derivation in PSG tree

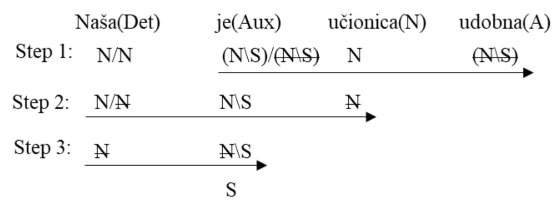

Figure 2. The CG derivation of (ii)

The illustration of Fig 2:

- In step1, the category of *je* is cancelled out with respect to the category of *udobna*.
- In step 2, the category of *Naša* is cancelled out with respect to the category of *učionica*.
- In step 3, the output of step 2 (category of *Naša učionica*) now becomes the input and it is cancelled out with respect to the category of *je udobna*, resulting in the final output S.

It may be noted that though standard PSG trees do not allow for criss-crossing lines in the tree diagrams, the crossing lines are drawn in order to explain how the cancellation of categorial functions can be implemented with the help of the PSG tree as seen in Figure 1. The cancellation of arguments of a function proceeds in accordance with the constituency relations in PSG, as seen in Figure 2. The exact manner in which tree branches are or can be tangled reflects the way categorial derivations can work, thus uniting CG derivations with PSG. That the crossing lines are made insignificant in CG is substantiated by the specification of the series of steps for the categorial derivation of the sentence which is shown right in Figure 2.

(iib) Dependency functions in terms of CG formulae (DG → CG)

The dependency graph for the sentence (ii) is depicted in Figure 3. It may be observed that δ is a dependency valuation function that takes a node as an input and returns a real value as an output (see Levelt, 2008: III:51). If A~B (meaning that A is dependent on B), then δ(A)>δ(B). The real value is set to 0 at the top of the tree, but we can start from 1 at the top of the tree. This function will be useful for recoding CG functor-argument relations in terms of dependency relations as discussed below:

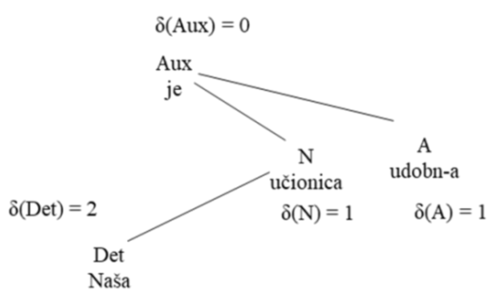

Figure 3. A dependency graph of (ii)

The above figure illustrates the following dependencies:

i. *udobna* is dependent on *je*.
ii. *Naša* is dependent on *učionica*.
iii. *učionica* is dependent on *je*.

Based on Figures 2 and 3, the dependency functions capturing all the functor-argument relations/cancellations can be formulated as follows:

Step 1: δ(Aux)/~~δ(A)~~ ~~δ(A)~~

This step captures the functor-argument relation between *udobna* (A) and *je* (Aux) corresponding to step 1 of the CG derivation in Figure 2. This step builds the meaning conveyed through *is comfortable*.

Step 2: δ(Det)/~~δ(N)~~ ~~δ(N)~~

This step captures the functor-argument relation between *Naša* and *učionica* corresponding to step 2 of the CG derivation in Figure 2. This step builds the meaning conveyed through *our classroom*.

Step 3: δ(Aux)/~~δ(N)~~ ~~δ(N)~~

This step captures the functor-argument relation between the outputs of Step1 and Step2 of the CG derivation. By using the correspondence principle, we have that je(*učionica) ≡ Naša\je. In other words, Aux(*N) ≡ Det\Aux. We can also express it as: A(*B) ≡ B\A (A = *je*). Since 'Naša' and 'je' do not participate in any (direct) dependency relation as seen in Figure 3, the functor-argument relation is constructed through Aux and N in step 3. This step builds the meaning conveyed through *our classroom is comfortable*.

(iic) CG → DG derivation

Here *the correspondence principle* is used to show how the dependency relations can be derived from the categories assigned to the words and the subsequent CG derivation. When each step of the CG derivation is taken into account and expressed in terms of head-dependent relations, the corresponding dependency relation between the functor and the argument can be established.

In this derivation, the equivalence relation is established from RHS to LHS. Hence the CG derivation would be the starting point. The aim is to establish a DG relation for each step of the CG derivation. In other words, for every functor-argument relation, the equivalent head-dependent relation is to be established. Finally, by considering all the head-dependent relations that are established from the CG derivation and other possible head-dependent relations (if any), the DG graph of the sentence can be drawn.

Step 1: The CG relation between 'je' and 'udobna'

In this CG relation, 'je' (Aux) is the functor and 'udobna' (Adj) is the argument. The direction of the argument is to the right. If we consider 'je' (Aux) to be A and 'udobna' (Adj) to be B, the RHS would be Aux/Adj or je/udobna or A/B. There is a direct dependency relation between the functor and the argument - 'je' and 'udobna', with 'je' as the head and 'udobna' as its dependent. Accordingly, the LHS would be je(*udobna) or Aux(*Adj) or A(*B). Thus, the equivalence relation for this CG relation would be:

je(*udobna) ≡ je/udobna. It can also be expressed as: Aux(*Adj) ≡ Aux/Adj or A(*B) ≡ A/B.

Step 2: The CG relation between 'Naša' and 'učionica'

In this CG relation, 'Naša' (Det) is the functor and 'učionica' (N) is the argument. The direction of the argument is to the right. If we consider 'učionica' (N) to be A and 'Naša' (Det) to be B, the RHS would be Det/N or Naša/učionica or B/A. There is a direct dependency relation between the functor and the argument - 'Naša' and 'učionica', with 'učionica' as the head and 'Naša' as its dependent. Accordingly, the LHS would be the following: učionica(Naša*) or N(Det*) or A(B*). Thus the equivalence relation for this CG relation would be: učionica (Naša*) ≡ Naša/učionica. It can also be expressed as: N(Det*) ≡ Det/N or A(B*) ≡ B/A.

Step 3: The CG relation between 'Naša' and 'je'

In this CG relation, 'je' (Aux) is the functor and 'Naša' (Det) is the argument. The direction of the argument is to the left. If we consider 'je' (Aux) to be A, the RHS would be Det\Aux or Naša\je or B\A (B = Det). However, there is no direct dependency relation between the functor and the argument - 'je' and 'Naša'. Rather, 'učionica' (N) is dependent on 'je' (Aux). In other words, Aux(*N) or je(*učionica) or A(*B) [B = 'učionica'] would be the LHS. Since the functor and the argument do not participate in a direct head and dependent relationship, considering 'je' to be A on the RHS would implicitly indicate that B on the RHS is its argument 'Naša' and B on the LHS is its dependent 'učionica'. Thus, the equivalence relation for this CG relation would be: je(*učionica) ≡ Naša\je. It can also be expressed as, Aux(*N) ≡ Det\Aux or A(*B) ≡ B\A. This clearly shows that 'je' (Aux) is the head of 'učionica' (N) but is the functor of the argument 'Naša' (Det).

Thus, combining the DG relations of all the steps, we get the following.

(i) Aux(*Adj) or 'udobna' is dependent on 'je'

(ii) N(Det*) or 'Naša' is dependent on 'učionica'

(iii) Aux(*N) or 'učionica' is dependent on 'je'

Based on these dependency relations, we arrive at the DG graph with the same dependency functions, that is, Figure 3.

(iid) CG → PSG derivation

In this Croatian sentence, discontinuity arises because (i) 'Naša' and 'učionica' and (ii) 'je' and 'udobna' are not contiguous in the linear order of the sentence. However, in a PSG tree, the cancellation of arguments proceeds as per the constituent structure.

Step1 of the CG derivation would mean that Aux and AP form a single constituent AuxP with Aux as the head. The analysis of 'udobna' as a separate constituent 'AP' is drawn from the fundamental principles of PSG as seen in Figure 4.

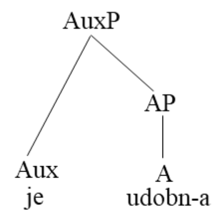

Figure 4. The PSG tree corresponding to step 1 of the CG derivation

Step2 of the CG derivation would mean that Det and N form a single constituent NP with N as the head as seen in Figure 5.

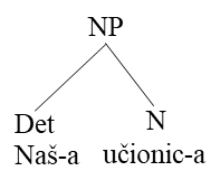

Figure 5. The PSG tree corresponding to step 2 of the CG derivation

Step3 of the CG derivation indicates that NP and AuxP form a single constituent S. Hence, in this step the remaining arguments are cancelled out resulting in the final output S. The corresponding tangled diagram of the sentence is shown below in Figure 6. Accordingly, the corresponding PSG rules are: (i) S → NP AuxP (ii) NP → Det N (iii) AuxP → Aux AP (iv) AP → A.

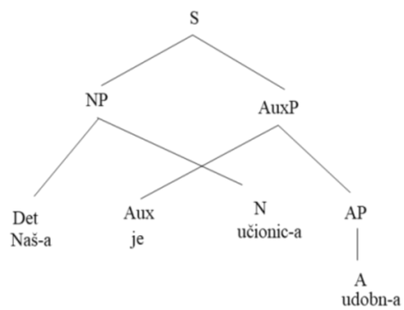

Figure 6. The final PSG tree

(iie) A unified representation (from DG to CG to PSG)

Finally, Figure 7 depicts a unified representation taking into account the constituency relations, dependency relations and the functor-argument relations in the discontinuous Croatian sentence.

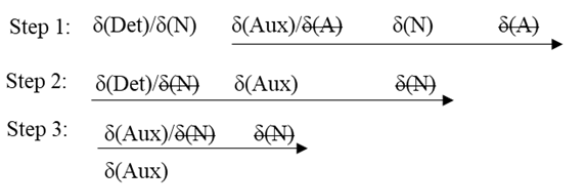

Fig 7. A unified representation of the discontinuous sentence

In all, the derivations formulated for the above illustration show how the conversions, namely PSG→CG, DG→CG, CG→DG and CG→PSG (not necessarily always in that order), can establish the desired equivalence of representations in those formalisms. Therefore, establishing PSG→CG, DG→CG, CG→DG and CG→PSG is tantamount to establishing PSG≡CG≡DG in their representational principles for natural language constructions.

7 Implications and Conclusion

This paper is an attempt to show how a flexible account of functor-argument relations can be decoded from the rigid constituents of phrase structures and how in turn these functor-argument (categorial) relations can also be formulated in terms of dependency relations. Hence the PSG rules in trees could be redrawn in terms of the CG formula which in turn could be rewritten in terms of the DG functions. This can have far-reaching implications for theories of natural language since most current linguistic theories do adopt and subscribe to constituency-based analyses, although specific treatments of particular phenomena such as labelling phrases may differ. But one emerging conclusion is that not all aspects of natural language (especially syntax) can be accounted for by binary branching and headed rules (see Müller, 2013). The unified system of representation for continuous and discontinuous structures cuts across and in fact (somewhat) neutralizes the traditional distinction between derivational theories (as in Chomsky, 1995) and constraint-based formalisms because both types of formalisms have to define their derivations or constraints on the structural organization of linguistic structures.

Though there have been many solutions proposed so far, to account for discontinuity, we argue that considering an alternative approach will merely add to the solutions existing in the literature. The present unified system of representation differs from the solutions proposed so far in that it attempts to face up to the problem of discontinuity by enabling a direct analysis of discontinuous structures from the basic underlying assumptions in each of the grammar formalisms without introducing any extra assumptions/rules or even constraints. There is no expansion/manipulation of the features of the system as in the case of most of the proposed solutions. This reduces the number of types of structures and strikes a balance between rigidity and flexibility to account for continuous constituents as well as discontinuous constituents which are grammatical. This can comprehensively capture and help analyse both continuous and discontinuous expressions for a range of natural language phenomena including movement/displacement, long-distance dependencies etc. Regarding long distance dependencies, gaps occur when a phrase is fronted. This missing phrase is anaphorically interpreted as in the case of *Wh*-questions, topicalisation etc. The unified system of representation works in a non-local way and, therefore can account for the gaps created as a result of fronting of constituents in long-distance dependencies. This thereby eliminates the need to separately represent the movement of constituents/gaps using arrows or bars. However, an illustration of this is beyond the scope and space requirements of the current paper.

Most importantly, what makes this approach different from the earlier ones is that the equivalence relation between PSG, DG and CG is in terms of the representational descriptions of natural language constructions. Only the representational principles are unified (constituency relations in PSG, head-dependent relations in DG and functor-argument relations in CG), *not* the grammar formalisms as such, in their descriptions of natural language constructions. The very flexibility that PSG can have in allowing for both normal trees and tangled trees (by relaxing the 'no-crossing' constraint) is nothing other than the flexibility DG or CG inherently permits. DG or CG is inherently neutral with respect to line crossing or no-line crossing. Hence the desired flexibility in PSG for continuous and discontinuous structures is an expression or instantiation of the principles of DG/CG itself. Thus, the apparent tension between the three grammar formalisms can perhaps be neutralized by this way of working towards mutually unifying *the most basic principles* of the three formalisms. The novelty of this unified representation is it can help draw correspondences between representations in parsing systems based on head-dependent relations and the parsing systems based on PSG and/or CG relations. Thus, systems of dependency parsing and PSG-constituency-based parsing can have representational interrelation that can help achieve representational economy. A parser can then (de)code dependency parses into constituency parses and vice versa, without any extra burden on computational resources, since one single representational system may suffice. Hence exploring the full range of practical applications of the unified representation is beyond the scope of the current study and would be left as a follow-up.

Another argument substantiating the unified system of representation pertains to the representation of language in our cognitive system. In the real world a speaker of a language with a predominantly continuous system can also learn, comprehend, speak a language with a non-continuous system. Though linguists advance each of these grammatical formalisms on distinct grounds having varying theoretical motivations, it is more likely that for a speaker of any language there exists just one representation in their cognitive system which is equipped to deal with the features of both kinds of systems. However, this is too left open for further study.

References

Barbara Citko. 2011. Symmetry in syntax: Merge, move and labels. Cambridge: Cambridge University Press.

Cathryn Donohue and Ivan A. Sag. 1999. Domains in Warlpiri. In Sixth International Conference on HPSG–Abstracts. 04–06 August 1999, 101–106.University of Edinburgh.

David G. Hays. 1964. Dependency theory: A formalism and some observations. Language, 40: 511–25.

David R. Dowty. 1996. Towards a minimalist theory of syntactic structure. In Bunt, H. & van Horck, A. (Eds.). *Discontinuous constituency*, 11–62. Berlin: Mouton de Gruyter.

Frederick J. Newmeyer. 2001. Grammatical functions, thematic roles, and phrase structure: their underlying disunity. In Davies, W. D. & Dubinsky, S. (eds.). Objects and other subjects, 53-75. Springer, Dordrecht.

Gerald Gazdar. 1983. Phrase structure grammar. Kulas, J., Fetzer, J. H. & Rankin, T. L. (eds.). Philosophy, language and artificial intelligence, 163-218. Dordrecht: Springer.

Guy Barry and Martin J Pickering. 1990. Dependency and constituency in categorial grammar. In Barry, G. & Morrill, G. (eds.), Edinburgh working papers in cognitive science, vol. 5. Studies in categorial grammar, 23-45. Edinburgh: Centre for Cognitive Science, University of Edinburgh.

Guy Barry and Martin J. Pickering. 1993. Dependency and categorial grammar and coordination. Linguistics, 31(5): 855 - 902.

Haim Gaifman. 1965. Dependency systems and phrase-structure systems. Information and Control, 8(3): 304-337.

James D. McCawley. 1982. Parentheticals and discontinuous constituent structure. Linguistic Inquiry, 13(1): 91-106.

James P. Blevins. 1990. Syntactic complexity: Evidence for discontinuity and multidomination. Ph.D.

dissertation, University of Massachusetts at Amherst.

Joachim Lambek. 1958. The mathematics of sentence structure. American Mathematical Monthly, 65(3): 154–170.

Kazimierz Ajdukiewicz. 1935. Die syntaktische konnexität. *Studia Philosophica.* 1, 1-27, transl. *Syntactic connexion* in S. McCall, Polish Logic. Oxford 1967, 207–231.

Ken Hale. 1982. Preliminary remarks on configurationality. In Pustejovsky, J. & Sells, P. (eds.). North East Linguistic Society, 12: 86-96.

Ken Hale.1983. Warlpiri and the grammar of non-configurational languages. Natural Language and Linguistic Theory, 1(1): 5-47.

Kunihiro Iwakura. 1988. Review article on Huck, G. and Ojeda, A. (eds.) 1987. English Linguistics, 5: 329-46.

Lucien Tesniére. 1959. Eléments de syntaxe structurale. Paris: Ed. Klincksieck.

Marie-Catherine de Marneffe and Joakim Nivre. 2019. Dependency grammar. Annual Review of Linguistics. 5: 197-218.

Mark Steedman. 1985. Dependency and coordination in the grammar of Dutch and English. Language, 61(3): 523-568.

Mark Steedman. 1992. Categorial Grammar. Lingua, 90: 221-258.

Mark Steedman. 2014. Categorial Grammar. In Carnie, A., Sato, Y. & Siddiqi, D. (eds.), Routledge handbook of syntax, 670–701. New York: Routledge.

Noam Chomsky. 1957. Syntactic structures. Hague: Mouton de Gruyter.

Noam Chomsky. 1995. The minimalist program. Cambridge: MIT Press.

Peter Austin and Joan Bresnan. 1996. Non-configurationality in Australian aboriginal languages. Natural Language & Linguistic Theory, 14(2): 215-268.

Rachel Nordlinger. 2014. Constituency and grammatical relations in Australian languages. Koch, H. & Nordlinger, R (eds.). The languages and linguistics of Australia: A comprehensive guide, 215-261. Berlin: Mouton de Gruyter.

R. M. W Dixon. 1972. The Dyirbal language of north Queensland. Cambridge: Cambridge University Press. Dixon, R. M. W. 1977. A grammar of Yidiny. Cambridge: Cambridge University Press.

Robert Van Valin Jr. 2001. An introduction to syntax. Cambridge: Cambridge University Press.

Stefan Müller. 2002. Complex predicates: Verbal complexes, resultative constructions, and particle verbs in German. Stanford, CA: CSLI Publications.

Stefan Müller. 2013. Unifying everything: Some remarks on Simpler Syntax, Construction Grammar, Minimalism and HPSG. Language, 89(4): 920–950.

Willem J. M. Levelt. 2008. Formal grammars in linguistics and psycholinguistics. Vol.3. Amsterdam: John Benjamins Publishing Company.

Yehoshua Bar-Hillel. 1953. A quasi-arithmetical notation for syntactic description. Language, 29: 47–58.

Yehuda N. Falk. 2001. Lexical functional grammar: An introduction to parallel constraint-based syntax. Stanford: CSLI Publications.