A method for reconstructing first-order arguments in natural language

Gobbo, F.; Wagemans, J.H.M.

Published in:
CEUR Workshop Proceedings

A method for reconstructing first-order arguments in natural language*

Federico Gobbo[1,2][0000−0003−1748−4921] and Jean H. M. Wagemans[3][0000−0001−9304−5766]

[1] University of Amsterdam, ACLC – Amsterdam Center for Language and Communication, Spuistraat 134, Amsterdam 1012VT, The Netherlands
f.gobbo@uva.nl http://uva.nl/profile/f.gobbo
[2] University of Turin, StudiUm – Department of Humanities, via Giulia di Barolo 3/A, 10124 Turin, Italy federico.gobbo@unito.it
https://linguistica.campusnet.unito.it/persone/federico.gobbo
[3] University of Amsterdam, ACLC – Amsterdam Center for Language and Communication, Spuistraat 134, Amsterdam 1012VT, The Netherlands
j.h.m.wagemans@uva.nl http://uva.nl/profile/j.h.m.wagemans

Abstract. This paper develops a new method for reconstructing arguments in natural language by combining the linguistic representation framework of *Constructive Adpositional Grammars* (CxAdGrams) with the argument classification framework of the *Periodic Table of Arguments* (PTA). The method centers around the notion of 'argumentative adpositional adtree' ('arg-adtree'). After an explanation of the two frameworks involved, the method is illustrated by providing the arg-adtrees of two concrete examples of so-called 'first-order arguments'. It is argued that the resulting ad-trees provide a theoretically informed and empirically reliable reconstruction of an argumentative text or discussion. As such, the method developed in this paper is especially suitable as a point of departure for developing instruments for computer-assisted argumentation analysis.

Keywords: argumentation · Constructive Adpositional Grammars · formal linguistics · natural language processing · Periodic Table of Arguments.

1 Introduction

Over the last decade, computational argumentation has emerged as an independent field of research. One of the core challenges within this field is to develop

* This paper is the result of joint work by the two authors. Federico Gobbo is mostly responsible for Section 2, Jean H. M. Wagemans for Section 3, whereas all other sections are rather the result of collaborative efforts. The authors thank Marco Benini for his thorough reading of the manuscript, and in particular for checking the parts concerning constructive mathematics. The present paper is an amended version of a paper submitted to the proceedings of the ISSA 2018 Conference on Argumentation.

methods for representing argumentative texts and discussions so as to enable their analysis and evaluation. So far, scholars have developed various computational models of argument that are used, for example, in developing tools for argument mapping, argument mining and computer-aided human decision making. A common characteristic of these models and tools is that they operate on the level of complete propositions and the interactions between them. This goes, for example, for approaches inspired on Dung's abstract argumentation frameworks, which study sets of atomic arguments and their interrelations. But it also applies to approaches that take Walton's argument schemes as a point of departure, in which an argument scheme is taken to consist of a conclusion and a set of premises [4].

In order to enable a more detailed analysis of argumentative discourse, as well as a fruitful application of the rich but rather informal insights regarding their evaluation developed within the adjacent field of argumentation theory, a method for representing arguments is needed that does not only operate on the level of complete propositions, but also on the level of their individual linguistic elements. Such a method should represent the linguistic and pragmatic information that is relevant for the analysis and evaluation of arguments in an adequate way. At the same time, it should be formal(izable) to the extent that it can be used for the purpose of building tools that automatize these tasks.

In this paper, we provide the theoretical foundation of a high precision method for representing arguments in natural language and illustrate its working by means of examples. For the representation of the linguistic elements of the original text, we use the theoretical framework of *Constructive Adpositional Grammars* (CxAdGrams) developed by Gobbo and Benini [7]. Regarding the identification of different types of arguments, we follow the formal linguistic classification of the *Periodic Table of Arguments* (PTA) developed by Wagemans [12]. The combination of these frameworks is eventually aimed at developing a fully-fledged method for reconstructing argumentative discourse. In this paper, we focus on developing the first step of this method, which is aimed at representing the linguistic elements of the conclusion and the premise of a single argument in such a way that it becomes possible to identify its type. In subsequent work, we extend this first step so as to enable the representation of concatenations of arguments within a complete argumentative text.

The method operates on the level of the individual words and therefore enables the analyst to justify decisions regarding the representation of the original text in a very detailed way, while at the same time leaving it up to the analyst to choose which details to show and hide, according to her needs. It is a valuable tool because it helps the analyst to identify those linguistic elements that have a pragmatic function in the argumentation and are therefore relevant to include in the reconstruction. The parsing largely depends on the natural language in use, while the pragmatic information is language-independent to a large extent.

We begin our paper with providing a general explanation of the two frameworks involved. In Section 2, we will pay attention to CxAdGrams. After having explained the theoretical starting points of this framework, we discuss in more detail how the linguistic features of statements expressed in natural language can be depicted in a so-called 'adpositional tree' ('adtree' in short). In Section 3, we will expound the method for identifying types of arguments employed in the PTA. We describe three basic characteristics of the various types of arguments and subsequently describe two concrete examples of so-called 'first-order arguments'. Then, in Section 4, we combine the two frameworks so as to develop a method for reconstructing the statements that constitute an argument. For this purpose, we develop the notion of 'argumentative adpositional tree' ('arg-adtree' in short) and illustrate, by means of the two examples, how the linguistic and pragmatic features of first-order arguments can be depicted in terms of such adtrees. Finally, in Section 5, we briefly summarize and discuss our findings, situate the results within the field of formal linguistics and argumentation theory, and indicate the main directions for further research.

2 Constructive Adpositional Grammars

Constructive Adpositional Grammars (CxAdGrams) is the theoretical framework resulting from the application of constructive mathematics to the adpositional paradigm in linguistics. Before explaining the constituents of the framework, we will briefly indicate the meaning of 'constructive' and 'adpositional' in this particular context. Constructive mathematics is an approach to mathematics that is premised on the idea that regarding the formulas of a theorem, the information content of any statement should be strictly preserved – see Bridges and Richman [2]. This is established by avoiding the use of the *Law of Excluded Middle*, unlike classic logic. Mathematical representations of natural language grammars following the constructive approach are well known in mathematical and computational linguistics, the first ones being proposed by Adjukiewicz [1] and Church [3].

Of the various constructive models for natural languages so far, CxAdGrams specifically are based on topos-theory. It thus permits to use Grothendieck's topoi as the mathematical instrument to formalize natural languages, and their regularities, both *intra* a single language and between two or more natural languages in comparison. The linguistic and formal rules of CxAdGrams are not discussed here for reasons of conciseness. The interested reader can see Gobbo and Benini [7] for a comprehensive presentation of this approach to linguistic analysis; in particular, the formal model is described in the Appendix B, which guarantees the possibility to add pragmatic information maintaining the formality of the model. An example of pragmatic adpositional trees applied to Searle [8] is illustrated in Gobbo and Benini [7, ch. 6].

The adpositional paradigm in linguistics follows the idea that relations between linguistic elements can be described as hierarchical in that the one element 'governs' the other (which then 'depends' on the former). A very basic example is *children play*, in which *play* is the governing element (in short: *gov*) and *children* is the dependent element (in short: *dep*).

Regarding the way of representing the linguistic elements and their relations, within the adpositional paradigm the linguistic constructions are decomposed in terms of so-called 'adpositional trees' ('adtrees' in short). From a linguistic point of view, adtrees represent natural language expressions in terms of recursive trees, whose syntax will be presented – in its main traits – immediately below. From a formal point of view, they can be seen as formulas, which means that they are suitable for natural language processing (for an example, see Figure 7 below).

Each adtree represents a minimal pair of linguistic elements and their relation, expressed in terms of adpositions. The governing element (*gov*) is conventionally put on the right leaf at the bottom of the rightmost branch, while conversely the dependent element (*dep*) is put on the left leaf at the bottom of the leftmost branch. Finally, the adposition (*adp*), which represents the relation between the governor and the dependent, is depicted as a hook under the bifurcation of the two branches. In Figure 1, we pictured the abstract adtree structure, adapted from Gobbo and Benini [7, p. 15].

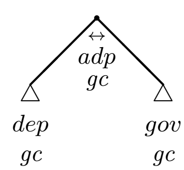

Fig. 1. The abstract adtree structure

The triangles on the leaves indicate the possibility of recursion, i.e. the fact that another adtree can be appended to each adtree leaf recursively, if needed. In this case, it indicates the possibility of performing a morphological analysis of the noun *children*, which is irrelevant in this context; this possibility illustrates the fact that the analyst can hide or show details through the use of triangles, according to her needs. The variable *gc* means 'grammar character'. Gobbo and Benini [7, ch. 2] clarify that the left and right branches of a linguistic adtrees follow different rules of construction: while governors (right branches) can have more dependents, dependents (left branches) can have one and one only governor.[4] In Figure 2, the adtree represents the example *children play* mentioned above – the arrows above *adp* and ϵ respectively in Figure 1 and 2 will be explained below.

[4] The number of dependents of each governor give the structure of the adtree, which is defined by the Tesnerian concept of valency. Readers unfamiliar with the original concept of valency are referred to Tesnière's fundamental book, either in French [10] or in its English translation [9]. The relation between Tesnerian structural syntax and CxAdGrams is clarified in Gobbo and Benini [6].

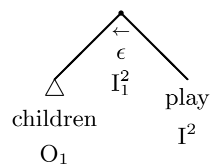

Fig. 2. The adtree of the phrase *children play*

Each proposition is analysed in terms of phrases, which are depicted as subtrees; each phrase presenting a ruling verb is built around that verb, which is posed as the rightmost leaf of the respective subtree. The variable *gc* will take the value of I_{act}^{val} in the case of verbs. A valency value (*val*) is assigned to each verb on the basis of its use in terms of constructions and it is expressed by an apex. Each valency value is fulfilled by its dependent subtree, expressed in terms of a definite actant value (*act*), i.e. a nominal expression (e.g. noun, pronoun) that fulfils the semantic role described by the valency value itself. Actant values are expressed by pedices both in verbs (I_{act}^{val}) and nominal expressions (O_{act}).

Let us provide an analysis of a prototypical example of a trivalent verb. In the case of the English verb *to open*, we will have a first actant that fulfils the role of the opener (e.g. a concierge), a second actant indicating the opened object (e.g. a door), and a possible third actant for the instrument (e.g. a key). Note that the semantic role of the beneficiary (e.g. the client, in the phrase *the concierge opens the door with the key for the client*) is an extra-valent actant, as it cannot be advanced (e.g. the phrase *the key opens the door* is incomplete but still depicts the same scene, while *the lady opens the door* changes the picture substantially).

The formal representation of the grammar characters of the morphosyntactic material by Tesnière in its original French version [10] was indicated using four letters (A, E, I, O). This notation method is preserved in CxAdGrams so as to remain consistent with the original model. However, for technical reasons, an additional grammar character was inserted (U) that did not exist in Tesnière's structural syntax [10]. In Table 1 below, we provide information regarding the meaning of the letters – adapted from Gobbo and Benini [7, p. 41].

Table 1. Grammar characters in adtrees

Value	*Name*	*Function*	*Examples*
A	adjunctive	modifier of O	adjectives, articles
E	circumstantial	modifier of I	adverbs, adverbial expressions
I	verbant	valency ruler	verbs, interjections
O	stative	actants	nouns, pronouns, name-entities
U	underspecified	transferer	prepositions, derivational morphemes

Direct objects of transitive verbs are often the second actant in English, and so they will be indicated as: O_2. In the previous example, *door* is O_2, *concierge* is O_1, and *key* is O_3. True adverbs, such as *here*, *now*, or sentence adverbs – which modifies the whole phrase structure – such as *obviously* or *technically* will be indicated as: E. As we mentioned above, although the main application of the constructive adpositional approach is morphosyntactic analysis, it can be applied to pragmatics as well.

Our current purpose is to show how CxAdGrams can be applied to the relation between premise and conclusion in an argument. In order to prepare the ground for a detailed explanation of this new application, we turn now to expounding a formal linguistic approach to argument classification.

3 The *Periodic Table of Arguments*

The *Periodic Table of Arguments* (PTA) is a classification of argument that integrates the traditional multitude of incomplete, informal and sometimes even inconsistent descriptions of the types of argument into a systematic and comprehensive whole.[5] The theoretical framework of the table is based on three partial characterisations of an argument, namely as (1) a first-order or second-order argument; (2) a predicate or subject argument; and (3) a specific combination of types of statements. The superposition of these three partial characterisations yields a factorial typology of argument that can be used in order to develop tools for analysing, evaluating, and generating arguments in natural language.

The types of arguments described in the PTA consist of exactly one premise and one conclusion, both of which are expressed by means of a statement that consists of a subject and a predicate. Closely following logical conventions, subjects are indicated with letters a, b, etc., predicates with letters X, Y, etc. (predicate $\top$ having the fixed meaning 'true'), and complete propositions with letters p, q, etc.

The theoretical framework of the PTA distinguishes between four different argument forms, a notion that comprises the first two partial characteristics mentioned above. In the visual representation of the table, these forms – first-order predicate arguments, first-order subject arguments, second-order subject arguments, and second-order predicate arguments – correspond to four different quadrants, which are indicated with letters α, β, γ, and δ respectively. In Table 2, for each quadrant we list the corresponding argument form and provide a concrete example.

The argument types situated within each of the quadrants are further differentiated on the basis of the third partial characteristic, which indicates the combination of the types of statements instantiated by the premise and the conclusion of the argument. For this purpose, the PTA makes use of a tripartite typology of statements that distinguishes between statements of fact (F), statements of value (V), and statements of policy (P), which means that an

[5] The present explanation of the theoretical framework of the *Periodic Table of Arguments* is based on Wagemans [12, 11].

Table 2. Overview of argument forms

quadrant	*conclusion*	*premise*	*argument (variant)*	*example in normal form*
α	a is X	a is Y	a is X, because a is Y (a is Y, so a is X)	The suspect (a) was driving fast (X), because he (a) left a long trace of rubber on the road (Y)
β	a is X	b is X	a is X, because b is X (b is X, so a is X)	Cycling on the grass (a) is forbidden (X), because walking on the grass (b) is forbidden (X)
γ	q is $\top$	r is $\top$	q is $\top$, because r is $\top$ r is $\top$, so q is $\top$	He must have gone to the pub (q), because the interview was cancelled (r)
δ	q is $\top$	q is Z	q is $\top$, because q is Z (q is Z, so q is $\top$)	We only use 10% of our brain (q), because that (q) was said by Einstein (Z)

argument can be said to instantiate one of nine different combinations of types of statements (PP, PV, PF, VP, VV, VF, FP, FV, FF). The argument *The government should invest in jobs, because this will lead to economic growth*, for instance, can be characterized as a PF argument since its combines a statement of policy (P) in its conclusion with a statement of fact (F) in its premise.

When taken together, the three partial characterizations of argument constitute a theoretical framework that allows for 2 x 2 x 9 = 36 systematic types of arguments. Depending on the linguistic formulation of the relation between the premise and the conclusion, each of these systematic types hosts a number of isotopes, which are named in accordance with the existing dialectical and rhetorical traditions of argument classification. The argument *The suspect was driving fast, because he left a long trace of rubber on the road*, for instance, is to be identified as a first-order predicate argument (1 pre) that combines a statement of fact (F) with another statement of fact (F). The systematic name of this argument is therefore '1 pre FF'. Given that the relation between the premise and the conclusion can be captured by saying that the predicate of the statement expressed in the premise, *leaving a long trace of rubber on the road*, is an 'effect' for the predicate of the conclusion, *driving fast*, the traditional name of this specific isotope of '1 pre FF' is 'argument from effect'. Within every quadrant, the systematic place of the type of argument is determined by the specific combination of types of statements that it instantiates (FF, VF, PF, etc.), while the isotopes representing the linguistic variations in the traditional names are placed in a vertical line. In Figure 3, we picture the relevant part of the current version of the PTA – for updates and more detailed analyses of examples, see its official web site [11].

In the next section we shall demonstrate how the theoretical framework of CxAdGrams can be applied to the types of argument situated in the Alpha Quadrant and the Beta Quadrant of the PTA. More in particular, we will provide an analysis of the concrete examples of arguments situated within these quadrants that were mentioned in Table 2.

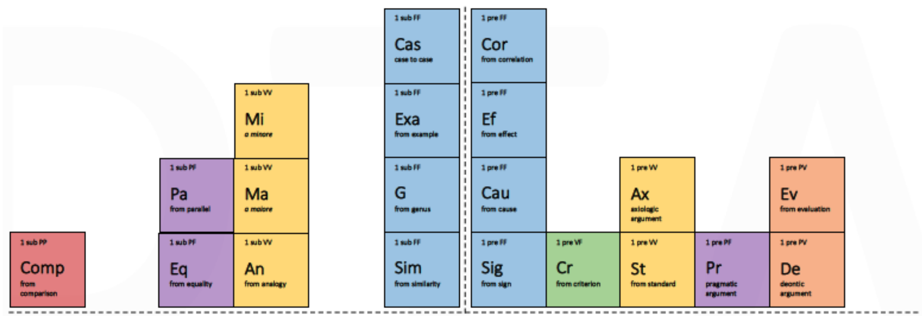

Fig. 3. Quadrants Beta (left) and Alpha (right) of the PTA – Wagemans [11]

4 Combining CxAdGrams and PTA

In this section we combine the two formalisms together. In order to do so, we will develop the notion 'argumentative adtree' (arg-adtree). We will explain how such an adtree makes use of all the expressive of linguistic adtrees taken from the framework of CxAdGrams, while at the same time incorporates the pragmatic information resulting from the argument analysis taken from the framework of in the PTA.[6]

In arg-adtrees, a particular emphasis is put on the first actant, which is the subject - a, as defined in the two concrete examples above - because its identification permits to classify the argument itself as a subject or predicate argument. In the arg-adtree that we are constructing for the purposes of the argument analysis, this emphasis is represented putting the first actant O_1 in evidence, as the leftmost subtree of the given phrase. After individuating the subject, the analyst considers at first in-valent actants, that is the actants that are either explicitly expressed by the verb ruling the proposition or implied in its semantic role structure. The procedure of explicitation of the in-valent actant structure permits to flesh out the inner functioning of the conclusion of the argument, and eventually it deepens the analysis of the argument itself in terms of robustness.

A minimal argument is made of two statements, i.e. a conclusion σ, standing for the Greek equivalent $\sigma\upsilon\mu\pi\epsilon\rho\alpha\sigma\mu\alpha$ (sumperasma), and a premise π, standing for the Greek equivalent $\pi\rho o\theta\alpha\sigma\iota\sigma$ (protasis). Both are expressed by means of propositions consisting of a subject (e.g. a, b) and a predicate (e.g. X, Y). In order for the premise to effectively support the acceptability of the conclusion, there should always be a common element. This yields two basic possibilities: (1) in so-called predicate arguments, the common element is the subject, which means that the conclusion 'a is X' is supported by the premise 'a is Y' and (2) in so-called subject arguments, the common element is the predicate, which means

[6] Such a transformation is formally justified by the so-called conjugate construction in the formal model of CxAdGrams – see Gobbo and Benini [7, Definition B.1.4, p. 211].

that the conclusion 'a is X' is supported by the premise 'b is X'.[7] Linguistically, each argument can be presented in two different forms: progressive and retrogressive – see van Eemeren and Snoeck Henkemans [5, p. 33]. The progressive form presents at first the premise π and then the conclusion σ. In Table 2, this is indicated by the conjunction *so*, which in arg-adtrees is represented formally by a left arrow: $\leftarrow$. The second form is called retrogressive, as the conclusion σ is presented before the premise π. The conjunction *because* is used to indicate them in Table 2, while in arg-adtrees its formal representation is a right arrow: $\rightarrow$. For the sake of simplicity, since now we consider only retrogressive arguments, i.e. our concrete examples make use of *because* in natural language terms and of right arrow ($\rightarrow$) in formal terms. Figure 4 shows the abstract adtree of first-order predicate arguments (on the left) and the abstract adtree of the first-order subject arguments (on the right).

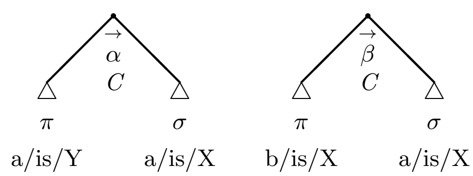

Fig. 4. The abstract argumentative adtrees of first-order arguments

The Greek letters alpha and beta (α, β) in the hook indicate the respective quadrants in the PTA. In the concrete examples below, the subtrees indicating the linguistic material will be always identified with an epsilon (ϵ), put in the hook, or by linguistic material directly, for reasons of compactness. Finally, the letter C indicates the combination of types of statements, which in concrete examples can take the values of FF, VF, PF, etc.

In order to illustrate how all this works, we turn now to reconstructing two examples in terms of the framework explained above. The examples are both first-order arguments, one being a predicate argument and the other a subject argument, and they instantiate different combinations of types of statements.

Example 1: An argument from effect The first example we analyse is *The suspect was driving fast, because he left a long trace of rubber on the road.* It has been identified as a first-order predicate argument that supports a statement of fact with another statement of fact, and its trivial name is 'argument from effect'. In the PTA, this type of arguments is indicated by means of the symbol 'Ef' (see Figure 3). In order to analyse this example in terms of CxAdGrams,

[7] What follows is a different presentation of the illustration of Figure 1 in Wagemans [12, p. 4]. Here, the roles of the subjects and predicates are put in more evidence.

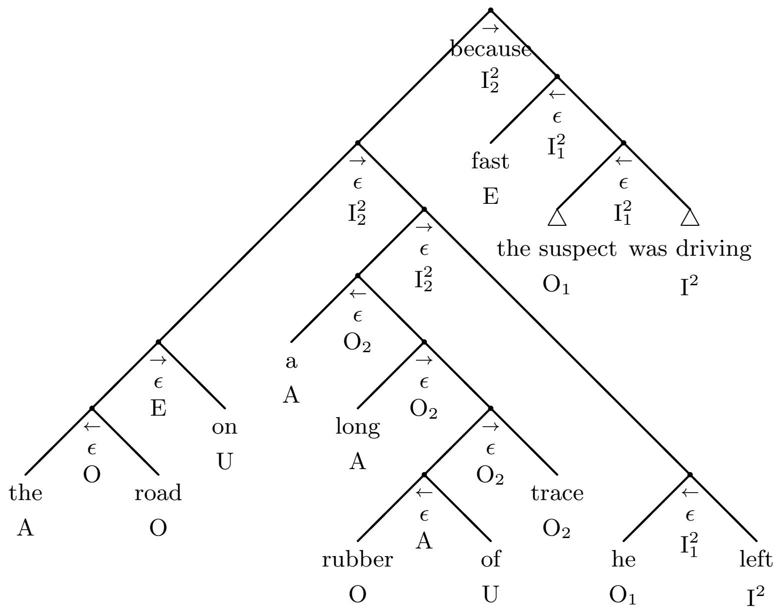

Fig. 5. The fully expanded linguistic adtree of Example 1

we first present the linguistic analysis of the statements in the premise and the conclusion without including any information about the type of argument. We will then explain how the transformation from the linguistic adtree to the arg-adtree takes place.

Figure 5 illustrates the *linguistic* adtree of the example.[8] In particular, the linguistic adtree of the premise (π) of the argument, *he left a long trace of rubber on the road* is depicted as the mostleft subtree, while the linguistic adtree of the conclusion (σ), *the suspect was driving fast* is the rightmost subtree, as the conjunction is retrogressive, in this case *because*. The reader is invited to note the presence of every grammar character mentioned in Table 1.

The arg-adtree of the argument can be derived from its linguistic counterpart by adding information that is relevant for identifying the type of argument and by condensing information that is too detailed for the purposes of the analysis. For the sake of simplicity, let us concentrate at the premise (π) first. Figure 6 shows the transformation from the linguistic to the arg-adtree. Linguistic details – already illustrated in Figure 5 – have been conveniently compacted through triangles: $\triangle$.

First, the upmost hook does not indicate linguistic information only (ϵ) but it marks the whole adtree as the premise (π) and as a statement of fact (F). Then,

[8] In adtrees, some branches are longer than others just for human readability. While linearised for machine coding, all branches become equally long.

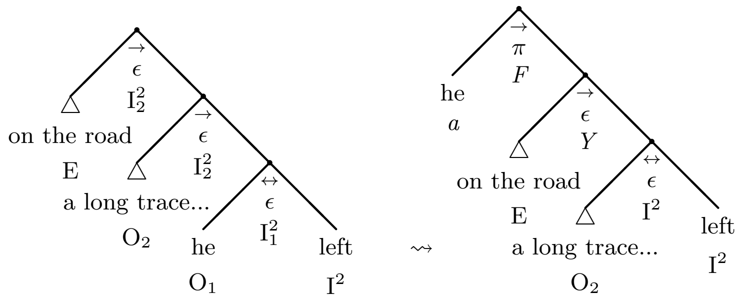

Fig. 6. The transformation from linguistic to argumentative adtree

the subject put in evidence is not only an actant of the verb (O_1) but a crucial part of the argument itself (a, which subsumes the linguistic information). For this reason, it is found as the leftmost sub-branch. All the rest is part of the predicate, identified by the hook ypsilon (Y). The reader is invited to note the corresponding linearised form of the argumentative adtree (Figure 7), useful for constructing a treebank of arg-adtrees suitable for natural language processing.

$$\pi_{\overrightarrow{F}}(\text{he}_a,\ \epsilon_{\overrightarrow{Y}}((\text{on the road})_{\text{E}},\ \epsilon_{\overleftrightarrow{\text{I}^2}}((\text{a long trace...})_{\text{O}_2},\ \text{left}_{\text{I}^2}))\ )$$

Fig. 7. Linearisation of the argumentative adtree (Figure 6, right)

Figure 8 shows the complete argumentative adtree of the example. On the top hook, there is information regarding the quadrant (in this case, α) and the combination of types of statement (indicated with a generic C in Figure 4) under the hook that connects premise and conclusion. In this case, it is a combination of two factual statements (FF), which correspond to 'Ef' in the PTA (see Figure 3).

CxAdGrams help the analyst through the analysis of the in-valent structure. In fact, the analyst has seen that the verb ruling the conclusion (σ), 'to drive', has two actants: the driver (O_1) and the vehicle (O_2). In the example, the information carried by the second actant is unexpressed; however, that does not imply that it does not exist, rather that is hidden. In other words, adtrees permit to show this information under the form of a barred subtree. Let us suppose that we have to analyse not a single argument but a whole argumentative text. In such a case, unexpressed actants can be helpful to show what is present in the argument structure and what is – on purpose or not – omitted.

In particular, what is interesting here is that the categorial grammar subject 'the suspect', is metonymically identical with the driver O_1, even if, strictly speaking, it is the motor vehicle O_2 that left the long trace on the road. Interest-

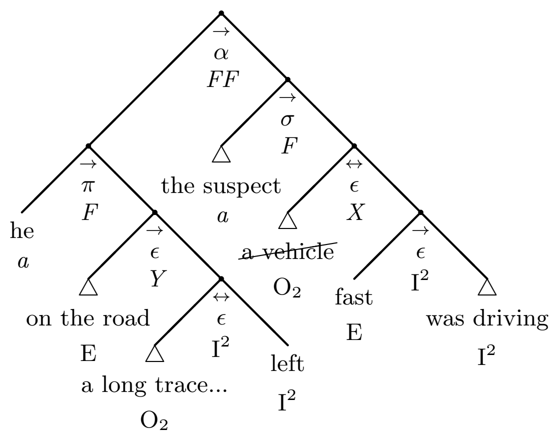

Fig. 8. The argumentative adtree of Example 1

ingly, the effect in the argument is expressed by apparently peripherical elements in the linguistic structure of the propositions, in particular: *fast*, which is the circumstantial (E) of the premise (π); the second actant O_2, *a long trace*; finally, its circumstantial (E), *on the road*. In other words, even if circumstantials (E) represent inessential information from the point of view of linguistic soundness, on the contrary they are central for the sake of the argument: if the suspect weren't driving fast long traces on the road possibly couldn't be left; in other words, the argumentative correlation is sustained by both circumstantials (E) and the explicit second actant O_2, *a long trace*.

If we forget to represent the second actant O_2 of the conclusion σ, we risk to lose an important piece of information, and that's why it is important to represent it in the argumentative adtree.

Example 2: An argument from analogy The second example we analyse is *Cycling on the grass is forbidden, because walking on the grass is forbidden.* In Section 3, this example has been identified as a first-order subject argument linking a statement of value to another statement of value. The systematic name of the argument type is '1 sub VV' and its trivial name is 'argument from analogy'. In the *Periodic Table of Arguments*, it is indicated by means of the symbol 'An' (see Figure 3). Like with the previous example, we first present the linguistic analysis of the statements in the premise and the conclusion without including any information about the type of argument and then explain how the transformation from the linguistic adtree (Figure 9) to the arg-adtree (Figure 10) takes place.

The linguistic adtree is rather symmetric, as the premise (π) and the conclusion (σ) share the same structure. The prepositional groups *on the grass* modify

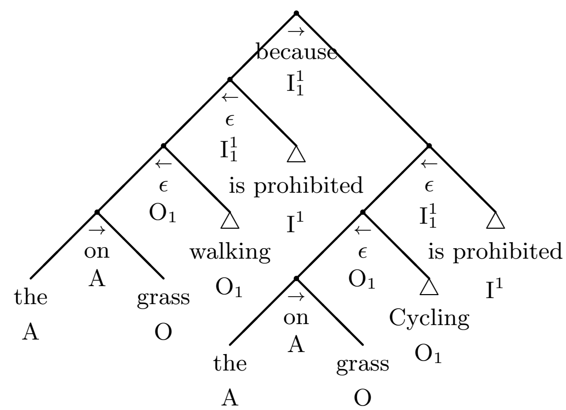

Fig. 9. The linguistic adtree of Example 2

respectively the subjects *Cycling* and *walking* (O_1), and therefore it is an adjunct (grammar character: A; see Figure 9).

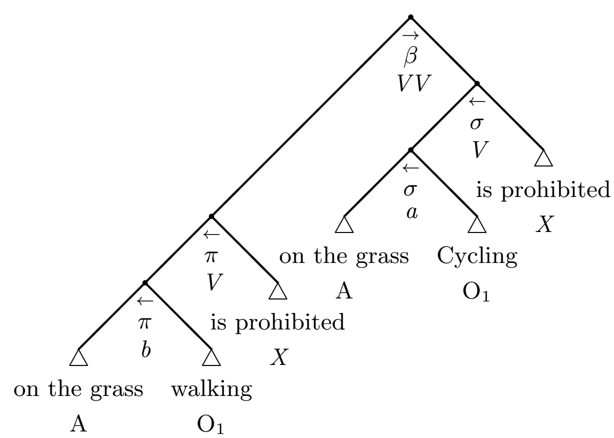

Fig. 10. The argumentative adtree of Example 2

In this case, the arg-adtree (Figure 10) appears very similar to the linguistic adtree. The subtrees of the subjects a and b, respectively of the conclusion (σ) and the premise (π), put in evidence the similar parts (*on the grass*), which are essential parts of the argument. In fact, if we cut them, the resulting phrase becomes: *Cycling is prohibited because walking is prohibited*, which loses all its pragmatic force.

We argue that these two examples show that arg-adtrees are powerful tools in order to show where the pragmatic force is placed within the linguistic material.

5 Conclusion

In this paper we demonstrated how the theoretical frameworks of *Constructive Adpositional Grammars* (CxAdGrams) and the *Periodic Table of Arguments* (PTA) can be combined so as to develop a high precision tool for the reconstruction of arguments in natural language. The central notion in this endeavour is that of the so-called 'argumentative adpositional tree'. Apart from representing the linguistic features of the statements that function as the conclusion and the premise of the argument under scrutiny, such an arg-adtree contains pragmatic information regarding the type of argument.

By providing a fully-fledged reconstruction of two examples of so-called 'first-order arguments', we showed how to transform the linguistic adtrees of the statements involved into argumentative adtrees. Such a transformation permits to represent the argumentative text or discussion including all information that may turn out to be relevant for its evaluation. Whereas the state-of-the-art in computational argumentation has automatized the extraction of complete propositions and their relations [4], our method prepares the ground for a more fine-grained computer-assisted analysis of argumentative texts.

By indicating how to apply CxAdGrams to the reconstruction of argument types, we extend its analytical potential to pragmatic aspects of discourse. In doing so, we have shown that CxAdGrams is not only suitable for the purpose of analysing and representing aspects of language itself, but also of the way language is used in communication (i.e., the persuasive efforts that are characteristic of argumentative discourse). Present research in argumentation theory usually separates the analysis of the external organisation of the argumentation – the so-called 'argumentation structure' – from the internal organisation of an argument – the so-called 'argument scheme'. By developing the notion of argumentative adtree, we have provided an instrument that enables an integrated analysis of these two aspects of argumentative discourse. Also, while the analytical tools developed in argumentation theory mostly produce selective representations of premises and conclusions, our reconstruction procedure reveals pragmatic information in detail. It therefore helps in providing a more fine-grained analysis of the linguistic aspects of statements that are used in arguments.

As we indicated in the introduction, the research presented in this paper is the first step in developing a complete procedure for reconstructing argumentative discourse. In this endeavor, the method for creating argumentative adtrees can be extended to second-order arguments. Another extension would be to apply the procedure to concatenations of arguments, thereby providing a complete analysis of an argumentative text. In fact, CxAdGrams provide a way to represent punctuation as conjunctions between sentences, thus they permit to represent a whole text in the terms of a single, comprehensive adtree.

Finally, the formal linguistic model presented here could be implemented in a computational model under a form of a tool. Such a tool would assist the analyst in making decisions regarding what linguistic and pragmatic information to include in specific reconstructions of argumentative discourse. Thanks to the combination of the linguistic and pragmatic information included in our framework with example-based data extracted from past analyses, the aim is to partially automatize the whole procedure using Artificial Intelligence techniques.

References

1. Adjukiewicz, K.: Die syntaxische Konnexität. Studia Philosophica **1**, 1–27 (1935).
2. Bridges, D., Richman, F.: Varieties of Constructive Mathematics. Cambridge University Press: Cambridge (1987).
3. Church, A.: A formulation of the simple theory of types. Journal of Symbolic Logic **5**, 56–58 (1940).
4. Modgil, S., Budzynska, K., Lawrence, J. (eds.): Computational Models of Argument. Proceedings of COMMA 2018. IOS Press: Amsterdam (2018).
5. Eemeren, F.H. van, Snoeck Henkemans, A.F.: Argumentation. Analysis and evaluation. Routledge: New York / London (2016).
6. Gobbo, F., Benini, M.: Dependency and Valency. From Structural Syntax to Constructive Adpositional Grammars. In: Gerdes, K., Hajiov E., Wanner, L. (eds.) Computational Dependency Theory, pp. 113–135. IOS Press: Amsterdam (2013).
7. Gobbo, F., Benini, M.: Constructive Adpositional Grammars. Foundations of Constructive Linguistics. Cambridge Scholars Publishing: Newcastle upon Tyne (2011).
8. Searle, J. R.: Making the Social World. The Structure of Human Civilization. Oxford University Press: Oxford (2010).
9. Tesnière, L.: Elements of Structural Syntax. English translation. John Benjamins: Amsterdam (2015).
10. Tesnière, L.: Éléments of Syntaxe Structurale. Klincksieck: Paris (1959).
11. Wagemans, J.H.M.: Periodic Table of Arguments. The atomic building blocks of persuasive discourse. URL: https://periodic-table-of-arguments.org/ (9 december 2017).
12. Wagemans, J.H.M.: Constructing a Periodic Table of Arguments. In: Bondy, P., Benacquista, L. (eds.) Argumentation, Objectivity, and Bias. Proceedings of the 11th International Conference of the Ontario Society for the Study of Argumentation (OSSA), pp. 1–12. OSSA: Windsor, ON (2016).