

Modeling and Analysis of Elliptic Coordination by Dynamic Exploitation of Derivation Forests in LTAG parsing

Djamé Seddah (1) & Benoît Sagot (2)

(1) NCLT - Dublin City University - Ireland
djame.seddah@computing.dcu.ie

(2) Projet ATOLL - INRIA - France
benoit.sagot@inria.fr

Abstract

In this paper, we introduce a generic approach to elliptic coordination modeling through the parsing of Ltag grammars. We show that erased lexical items can be replaced during parsing by informations gathered in the other member of the coordinate structure and used as a guide at the derivation level. Moreover, we show how this approach can be indeed implemented as a light extension of the LTAG formalism through a so-called “fusion” operation and by the use of tree schemata during parsing in order to obtain a dependency graph.

1 Introduction

The main goal of this research is to provide a way of solving elliptic coordination through the use of Derivation Forests. The use of this device implies that the resolution mechanism depends on syntactic information, therefore we will not deal with anaphoric resolutions and scope modifier problems. We show how to generate a derivation forest described by a set of context free rules (similar to (Vijay-Shanker and Weir, 1993)) augmented by a stack of current adjunctions when a rule describes a spine traversal. We first briefly discuss the linguistic motivations behind the resolution mechanism we propose, then introduce the **fusion** operation and show how it can be compared to the analysis of (Dalrymple et al., 1991) and (Steedman, 1990) and we show how it differs from (Sarkar and Joshi, 1996). We assume that the reader is familiar with the Lexicalized Tree Adjoining Grammars formalism ((Joshi and Schabes, 1992)).

2 Linguistic Motivations : a parallelism of Derivation

The LTAG formalism provides a derivation tree which is strictly the history of the operations nee-

ded to build a constituent structure, the derived tree. In order to be fully appropriate for semantic inference ¹, the derivation tree should display every syntactico-semantic argument and therefore should be a graph. However to obtain this kind of dependency structure when it is not possible to rely on lexical information, as opposed to (Seddah and Gaiffe, 2005a), is significantly more complicated. An example of this is provided by elliptic coordination.

Consider the sentences Figure 3. They all can be analyzed as coordinations of S categories² with one side lacking one mandatory argument. In (4), one could argue for VP coordination, because the two predicates share the same continuum (same subcategorization frame and semantic space). However the S hypothesis is more generalizable and supports more easily the analysis of coordination of unlike categories (“John is a republican and proud of it” becomes “John_i is_j a republican and $\varepsilon_i \varepsilon_j$ proud of it”).

The main difficulty is to separate the cases when a true co-indexation occurs ((2) and (4)) from the cases of a partial duplication (in (1), the predicate is not shared and its feature structures could differ on aspects, tense or number³). In an elliptic construction, some words are unrealized. Therefore, their associated syntactic structures are also non-realized, at least to some extent. However, our aim is to get, as a result of the parsing process, the full constituency and dependency structures of the sentence, including erased semantic items (or units) and their (empty) syntactic positions. Since their syntactic realizations have been erased, the construction of the dependency structure can not

¹As elementary trees are lexicalized and must have a minimal semantic meaning (Abeillé, 1991), the derivation tree can be seen as a dependency tree with respect to the restrictions defined by (Rambow and Joshi, 1994) and (Candito and Kahane, 1998) to cite a few.

²P for Phrase in french, in Figures given in annex

³see “John loves_i Mary and children_i their gameboy”

be anchored to lexical items. Instead, it has to be anchored on non-realized lexical items and guided by the dependency structure of the reference phrase. Indeed, it is because of the parallelism between the reference phrase and the elliptical phrase that an ellipsis can be interpreted.

3 The Fusion Operation

In this research, we assume that every coordinator, which occurs in elided sentences, anchors an initial tree α_{conj} rooted by P and with two substitution nodes of category P (Figure 1). The fu-

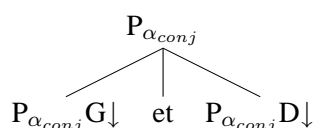


FIG. 1 – Initial Tree α_{conj}

sion operation replaces the missing derivation of any side of the coordinator by the corresponding ones from the other side. It shall be noted that the fusion provide proper node sharing when it is syntactically decidable (cf. 6.4). The implementation relies on the use of non lexicalized trees (*ie tree schemes*) called *ghost trees*. Their purpose is to be the support for partial derivations which will be used to rebuild the derivation walk in the elided part. We call the partial derivations *ghost derivations*. The incomplete derivations from the tree γ are shown as a broken tree in Figure 2. The ghost derivations are induced by the inclusion of the *ghost tree* α' which must be the scheme of the tree α . When the two derivation structures from γ and α' are processed by the fusion operation, a complete derivation structure is obtained.

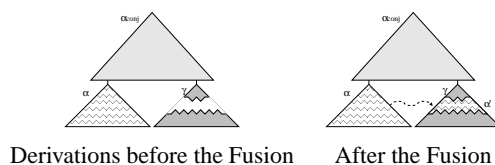


FIG. 2 – Derivation sketch of the Fusion Operation

4 examples analysis

Let us go back to the following sentences :

(1) Jean aime _i Marie et Paul ε _i Virginie John loves Mary and Paul Virginia
(2) Paul _i aime Virginie et ε _i déteste Marie Paul loves Virginia and hates Mary

Obviously (1) can have as a logical formula :

$aimel'(jean', Marie') \wedge aimel'(paul', virginie')$ whereas (2) is rewritten by $eat(paul', apple') \wedge buy'(Paul', cherries')$. The question is to differentiate the two occurrence of $aimel'$ in (1) from the $paul'$ ones. Of course, the second should be noted as a sharing of the same argument when the first is a copy of the predicate $aimel'$. Therefore in order to represent the sharing, we will use the same node in the dependency graph while a ghosted node (noted by $ghost(\gamma)$ in our figures) will be used in the other case. This leads to the analysis figure 4. The level of what exactly should be copied, speaking of level of information, is outside the scope of this paper, but our intuition is that a state between a pure anchored tree and an tree schemata is probably the correct answer. As we said, aspect, tense and in most case diathesis for ⁴ are shared, as it is showed by the following sentences :

(3)*Paul killed John and Bill by Rodger
(4)*Paul ate apple and Mary will pears

As opposed to (4), we believe “Paul ate apples and Mary will do pears” to be correct but in this case, we do not strictly have an ellipsis but a semi-modal verb which is susbsumed by its co-referent. Although our proposition focuses on syntax-semantic interface, mainly missing syntactic arguments.

5 Ghost Trees and Logical Abstractions

Looking either at the approach proposed by (Dalrymple et al., 1991) or (Steedman, 1990) for the treatment of sentences with gaps, we note that in both cases⁵ one wants to abstract the realized element in one side of the coordination in order to instantiate it in the other conjunct using the coordinator as the pivot of this process. In our analysis, this is exactly the role of *ghost trees* to support such abstraction (talking either about High Order Variable or λ -abstraction). In this regard, the fusion operation has only to check that the derivations induced by the *ghost tree* superimpose well with the derivations of the realized side.

This is where our approach differs strongly from (Sarkar and Joshi, 1996). Using the fusion operation involves inserting partial derivations, which are linked to already existing ones (the realized derivation), into the shared forest whereas using

⁴w.r.t to the examples of (Dalrymple et al., 1991), i.e “It is possible that this result can be derived (...) but I know of no theory that does so.”

⁵Footnote n°3, page 5 for (Dalrymple et al., 1991), and pages 41-42 for (Steedman, 1990).

the *conjoin* operation defined in (Sarkar and Joshi, 1996) involves merging nodes from different trees while the tree anchored by a coordinator acts similarly to an auxiliary tree with two foot nodes. This may cause difficulties to derive the now dag into a linear string. In our approach, we use empty lexical items in order to leave traces in the derivation forest and to have syntactically motivated derived tree (cf fig. 5) if we extract only the regular LTAG “derivation item” from the forest.

6 LTAG implementation

6.1 Working on shared forest

A *shared forest* is a structure which combines all the information coming from derivation trees and from derived trees. Following (Vijay-Shanker and Weir, 1993; Lang, 1991), each tree anchored by the elements of the input sentence is described by a set of rewriting rules. We use the fact that each rule which validates a derivation can infer a derivation item and has access to the whole chart in order to prepare the inference process. The goal is to use the shared forest as a guide for synchronizing the derivation structures from both parts of the coordinator.

This forest is represented by a context free grammar augmented by a stack containing the current adjunctions (Seddah and Gaiffe, 2005a), which looks like a Linear Indexed Grammar (Aho, 1968).

Each part of a rule corresponds to an item *à la* Cock Kasami Younger described by (Shieber et al., 1995), whose form is $\langle N, POS, I, J, STACK \rangle$ with N a node of an elementary tree, POS the situation relative to an adjunction (marked \top if an adjunction is still possible, \perp otherwise). This is marked on figure 5 with a bold dot in high position, \top , or a bold dot in low position, \perp). I and J are the start and end indices of the string dominated by the N node. $STACK$ is the stack containing all the call of the subtrees which has started an adjunction et which must be recognized by the foot recognition rules. We used S as the starting symbol of the grammar and n is the length of the initial string. Only the rules which prove a derivation are shown in figure 6.

The form of a derivation item is

$$Name : \langle Node_{\gamma_{to}}, \gamma_{from}, \gamma_{to}, Type, \gamma_{ghost} \rangle$$

where $Name$ is the derivation, typed $Type^6$, of the tree γ_{from} to the node $Node$ of γ_{to} .⁷

6.2 Overview of the process

We refer to a *ghost derivation* as any derivation which occurs in a tree anchored by an empty element, and *ghost tree* as a tree anchored by this empty element. As we can see in figure 5, we assume that the proper ghost tree has been selected. So the problem remains to know which structure we have to use in order to synchronize our derivation process.

Elliptic substitution of an initial ghost tree on a tree α_{conj} :

Given a tree α_{conj} (see Fig. 1) anchored by a coordinator and an initial tree α_1 of root P to be substituted in the leftmost P node of α_{conj} . Then the rule corresponding to the traversal of the Leftmost P node would be

$$P_{\alpha_{conj}G}(\top, i, j, -, -) \longrightarrow P_{\alpha_1}(\top, i, j, -, -).$$

So if this rule is validated, then we infer a derivation item called

$$D1 : \langle P_{\alpha_{conj}G}, \alpha_1, \alpha_{conj}, subst, - \rangle.$$

Now, let us assume that the node situated to the right of the coordinating conjunction dominates a phrase whose verb has been erased (as in *et Paul _ Virginia*) and that there exists a tree of Root P with two argument positions (a quasi tree like $NOVN1$ in LTAG literature for example). This ghost tree is anchored by an empty element and is called α_{ghost} . We have a rule, called *Call-subst-ghost*, describing the traversal of this node :

$$P_{\alpha_{conj}D}(\top, j+1, n, -, -) \longrightarrow P_{\alpha_{ghost}}(\top, j+1, n, -, -).$$

For the sake of readability, let us call $D1'$ the pseudo-derivation of call-subst-ghost :

$$D1' : \langle P_{\alpha_{conj}D}, \boxed{?}, \alpha_{conj}, subst, \alpha_{ghost} \rangle,$$

where the non-instantiated variable, $\boxed{?}$, indicates the missing information in the synchronized tree. If our hypothesis is correct, this tree will be anchored by the anchor of α_1 . So we have to prepare this anchoring by performing a synchronization with existing derivations. This leads us to infer a ghost substitution derivation of the tree α_1 on the node $P_{\alpha_{conj}D}$. The inference rule which produces the

⁶which can be an adjunction ($type = adj$), a substitution ($subst$), an axiom (ax), an anchor which is usually an implicit derivation in an LTAG derivation tree ($anch$) or a “ghosted” one ($adj_g, subst_g, anch_g$)

⁷ γ_{ghost} is here to store the name of the ‘ghost tree’ if the Node belongs to one or – otherwise.

item called $\text{ghost}(\alpha_1)$ on Figure 5, is therefore :

$$\frac{D1' : \langle P_{\alpha_{conj}D}, \boxed{?}, \alpha_{conj}, subst, \alpha_{ghost} \rangle}{D1 : \langle P_{\alpha_{conj}R}, \alpha_1, \alpha_{conj}, subst, - \rangle} \\ \text{Ghost} - D1 : \langle P_{\alpha_{conj}R}, \alpha_1, \alpha_{conj}, subst_g, \alpha_{ghost} \rangle$$

The process which is almost the same for the remaining derivations, is described section 6.4.

6.3 Ghost derivation and Item retrieving

In the last section we have described a ghost derivation as a derivation which deals with a tree anchored by an empty element, either it is the source tree or the destination tree. In fact we need to keep marks on the shared forest between what we are really traversing during the parsing process and what we are synchronizing, that is why we need to have access to all the needed informations. But the only rule which really knows which tree will be either co-indexed or duplicated is the rule describing the substitution of the realized tree. So, we have to get this information by accessing the corresponding derivation item. If we are in a two phase generation process of a shared forest⁸ we can generate simultaneously the substitution rules for the leftmost and rightmost nodes of the tree anchored by a coordination and then we can easily get the right synchronized derivation from the start. Here we have to fetch from the chart this item using unification variables through the path of the derivations leading to it.

Let us call “climbing” the process of going from a leaf node N of a tree γ to the node belonging to the tree anchored by a coordinator (α_{conj}) and which dominates this node. This “climbing” gives us a list of linked derivations (ie. $\langle \gamma_x(N), \gamma_y, \gamma_x, Type, IsGhost \rangle, \langle \gamma_z(N), \gamma_x, \gamma_z, Type_1, IsGhost_1 \rangle, \dots$) where $\gamma(N)$ is the node of the tree γ where the derivation takes place⁹). The last returned item is the one who has an exact counterpart in the other conjunct, and which is easy to recover as shown by the inference rule in the previous section. Given this item, we start the opposite process, called “descent”, which use the available data gathered by the climbing (the derivation starting nodes, the argumental position marked by an index on nodes in TAG gram-

⁸The first phase is the generation of the set of rules, (Vijay-Shanker and Weir, 1993), and the second one is the forest traversal (Lang, 1992). See (Seddah and Gaiffe, 2005b) for a way to generate a shared derivation forest where each derivation rule infers its own derivation item, directly prepared during the generation phase.

⁹The form of a derivation item is defined section 6.1

mars..) to follow a parallel path. Our algorithm can be considered as taking the two resulting lists as a parameter to produce the correct derivation item. If we apply a two step generation process (shared forest generation then extraction), the “descent” and the “climbing” phase can be done in parallel in the same time efficient way than(2005a).

6.4 Description of inference rules

In this section we will describe all of the inferences relative to the derivation in the right part, resp. left, of the coordination, seen in figure 5.

In the remainder of this paper, we describe the inference rules involved in so called predicative derivations (substitutions and ghost substitutions). Indeed, the status of adjunction is ambiguous. In the general case, when an adjunct is present on one side only of the conjunct, there are two possible readings : one reading with an erased (co-indexed) modifier on the other side, and one reading with no such modifier at all on this other side. In the reading with erasing, there is an additionnal question, which occurs in the substitution case as well : in the derivation structure, shall we co-index the erased node with its reference node, or shall we perform a (partial) copy, hence creating two (partially co-indexed) nodes ? The answer to this question is non-trivial, and an appropriate heuristics is needed. A first guess could be the following : any fully erased node (which spans an empty range) is fully co-indexed, any partially erased node is copied (with partial co-indexation). In particular, erased verbs are always copied, since they can not occur without non-erased arguments (or modifiers).

Elliptic substitution of an initial tree α on a ghost tree γ_{ghost} : If a tree α substituted in a node N_i of a ghost tree γ_{ghost} (ie. Derivation g-Der2' on figure 5), where i is the traditional index of an argumental position ($N_0, N_1 \dots$) of this tree ; and if there exists a ghost derivation of a substitution of the tree γ_{ghost} into a coordination tree α_{conj} (Der. g-Der1) and therefore if this ghost derivation pertains to a tree α_X where a substitution derivation exists node N_i , (Der. Der2) then we infer a ghost derivation indicating the substitution of α on the forwarded tree α_X through the node N_i of the ghost tree γ_{ghost} (Der. Ghost-Der2).

$$\begin{array}{l}
\text{g-Der2}' : \langle N_{i_\alpha}, \alpha, \boxed{?}, \text{subst}_g, \gamma_{ghost} \rangle \\
\text{g-Der1} : \langle P_{\alpha_{conj}D}, \alpha_X, \alpha_{conj}, \text{subst}_g, \gamma_{ghost} \rangle \\
\text{Der2} : \langle N_{i_{\alpha_X}}, -, \alpha_X, \text{subst}, - \rangle \\
\hline
\text{ghost-Der2} : \langle N_{i_\alpha}, \alpha, \text{ghost}(\alpha_X), \text{subst}_g, \gamma_{ghost} \rangle
\end{array}$$

This is the mechanism seen in the analysis of “Jean aime Marie et Pierre Virginie” to provide the derivation tree.

Elliptic substitution of a initial ghost tree α_{ghost} on a tree γ substituted on an tree α_{conj} : We are here on a kind of opposite situation, we have a realized subtree which lacks one of its argument such as *Jean_i dormit puis ϵ_i mourut* (John_i slept then ϵ_i died). So we have to first let a mark in the shared forest, then fetch the tree substituted on the left part of the coordination, and get the tree which has substituted on its i^{th} node, then we will be able to infer the proper substitution. We want to create a real link, because as opposed to the last case, it’s really a link, so the resulting structure would be a graph with two links out of the tree anchored by *Jean*, one to [*dormir*] (to sleep) and one to [*mourir*] (to die).

If a ghost tree α_{ghost} substituted on a node N_i of a tree α (Der. g-Der1’), if this tree α has been substituted on a substitution node, P_{conjD} , in the rightmost part of a tree α_{conj} , (Der. Der1) anchored by a coordinating conjunction, if the leftmost part node, P_{conjL} , of α_{conj} received a substitution of a tree α_s , (Der. Der2) and if this tree has a substitution of a tree α_{final} on its i^{th} node, (Der. Der3) then we infer an item indicating a derivation between the tree α_{final} and the tree α on its node N_i , (Der. g-Der1)¹⁰.

$$\begin{array}{l}
\text{g-Der1}' : \langle N_{i_{\alpha_{ghost}}}, \boxed{?}, \alpha, \text{subst}_g, \alpha_{ghost} \rangle \\
\text{Der1} : \langle P_{\alpha_{conj}D}, \alpha, \alpha_{conj}, \text{subst}, - \rangle \\
\text{Der2} : \langle P_{\alpha_{conj}L}, \alpha_s, \alpha_{conj}, \text{subst}, - \rangle \\
\text{Der3} : \langle N_{i_{\alpha_s}}, \alpha_{final}, \alpha_s, \text{subst}, - \rangle \\
\hline
\text{g-Der1} : \langle N_{i_\alpha}, \alpha_{final}, \alpha, \text{subst}, \alpha_{ghost} \rangle
\end{array}$$

7 Conclusion

We presented a general framework to model and to analyze elliptic constructions using simple mechanisms namely partial sharing and partial duplication through the use of a shared derivation forest in the LTAG framework. The main drawback of this approach is the use of tree schemata as part of parsing process because the anchoring process

¹⁰This mechanism without any restriction in the general case, can lead to an exponential complexity w.r.t to the length of the sentence.

must have a extremely good precision choose algorithm when selecting the relevant trees. For the best of our knowledge it is one of the first time that merging tree schemata, shared forest walking and graph induction, i.e., working with three different levels of abstraction, is proposed. The mechanism we presented is powerful enough to model much more than the ellipsis of verbal heads and/or some of their arguments. To model elliptic coordinations for a given language, the introduction of a specific *saturation* feature may be needed to prevent over-generation (as we presented in (Seddah and Sagot, 2006)). But the same mechanism can be used to go beyond standard elliptic coordinations. Indeed, the use of strongly structured anchors (e.g., with a distinction between the morphological lemma and the lexeme) could allow a fine-grained specification of partial value sharing phenomena (e.g. zeugmas). Apart from an actual large scale implementation of our approach (both in grammars and parsers), future work includes applying the technique described here to such more complex phenomena.

References

- Anne Abeillé. 1991. *Une grammaire lexicalisée d’arbres adjoints pour le français*. Ph.D. thesis, Paris 7.
- Alfred V. Aho. 1968. Indexed grammars-an extension of context-free grammars. *J. ACM*, 15(4) :647–671.
- Marie-Hélène Candito and Sylvain Kahane. 1998. Can the TAG derivation tree represent a semantic graph ? In *Proceedings TAG+4, Philadelphie*, pages 21–24.
- Mary Dalrymple, Stuart M. Shieber, and Fernando C. N. Pereira. 1991. Ellipsis and higher-order unification. *Linguistics and Philosophy*, 14(4) :399–452.
- Aravind K. Joshi and Yves Schabes. 1992. Tree Adjoining Grammars and lexicalized grammars. In Maurice Nivat and Andreas Podelski, editors, *Tree automata and languages*. Elsevier Science.
- Bernard Lang. 1991. Towards a Uniform Formal Framework for Parsing. In M. Tomita, editor, *Current Issues in Parsing Technology*. Kluwer Academic Publishers.
- Bernard Lang. 1992. Recognition can be harder than parsing. In *Proceeding of the Second TAG Workshop*.
- Owen Rambow and Aravind K. Joshi. 1994. *A Formal Look at Dependency Grammar and Phrase Structure Grammars, with Special consideration of Word Order Phenomena*. Leo Wanner, Pinter London, 94.
- Anoop Sarkar and Aravind Joshi. 1996. Coordination in tree adjoining grammars : Formalization and implementation. In *COLING’96, Copenhagen*, pages 610–615.

Djamé Seddah and Bertrand Gaiffe. 2005a. How to build argumental graphs using TAG shared forest : a view from control verbs problematic. In *Proc. of the 5th International Conference on the Logical Aspect of Computational Linguistic - LACL'05, Bordeaux, France*, Apr.

Djamé Seddah and Bertrand Gaiffe. 2005b. Using both derivation tree and derived tree to get dependency graph in derivation forest. In *Proc. of the 6th International Workshop on Computational Semantics - IWCS-6, Tilburg, The Netherlands*, Jan.

Djamé Seddah and Benoît Sagot. 2006. Modélisation et analyse des coordinations elliptiques via l'exploitation dynamique des forêts de dérivation. In *Proc. of Traitement automatique des Langues Naturelle - TALN 06 - louveau, Belgium*, Apr.

Stuart Shieber, Yves Schabes, and Fernando Pereira. 1995. Principles and implementation of deductive parsing. *Journal of Logic Programming*, 24 :3–36.

Marc Steedman. 1990. Gapping as constituent coordination. *Linguistic and Philosophy*, 13 :207–264.

K. Vijay-Shanker and D. Weir. 1993. The use of shared forests in tree adjoining grammar parsing. In *EACL '93*, pages 384–393.

8 Figures

1) Jean aime _i Marie et Paul ε _i Virginie <i>John loves Mary and Paul Virginia</i> Predicate elision
2) Paul _i mange une pomme et ε _i achète des cerises <i>Paul eats an apple and buys cherries</i> Right subject elision
3) Marie cuit ε _i et Pierre vend des crêpes _i <i>Mary cooks and Peter sells pancakes</i> Left object elision
4) Marie _i cuit ε _j et ε _i vend des crêpes _j <i>Mary cooks and sells pancakes</i> Left object and right subject elision

FIG. 3 – Exemples de elliptic constructions

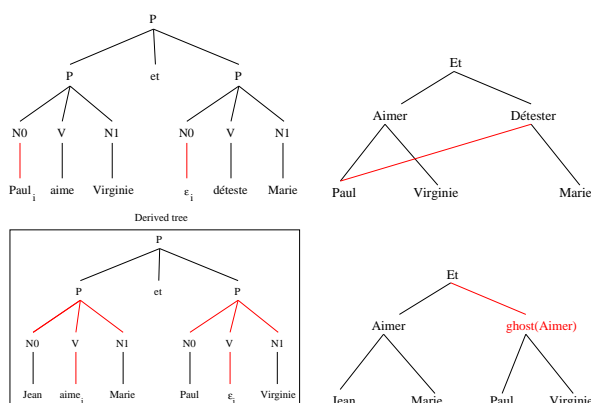


FIG. 4 – Gapping and Forward Conjunction reduction

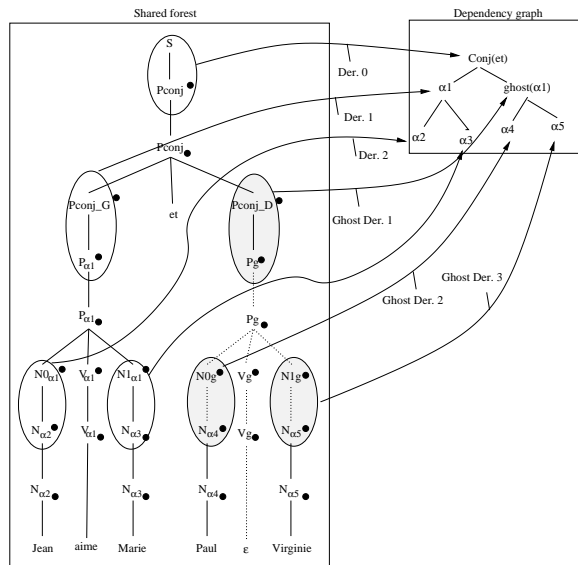


FIG. 5 – Shared forest and relative dependency graph for “Jean aime Marie et Paul Virginie”(John loves Mary and Paul Virginia)

call transition	rules
Call subst	$\langle \perp, N_\gamma, i, j, -, -, R, Stack \rangle \rightarrow \langle \top, N_\alpha, i, j, -, -, R, Stack \rangle$
Call adj	$\frac{\langle \top, N_\gamma, i, j, -, -, R, Stack \rangle \rightarrow \langle \top, N_\beta, i, j, -, -, R, [N_\gamma Stack] \rangle}{S \rightarrow}$
Call axiom	$\langle \top, N_\alpha, 0, n, -, -, \emptyset, \emptyset \rangle$
Call no subs	$\langle \perp, N_\gamma, i, j, -, -, R, Stack \rangle \rightarrow \text{true}$
Call foot	$\frac{\langle \perp, *N_\beta, i, j, -, -, R, [N_\gamma Stack] \rangle \rightarrow \langle \top, N_\gamma, i, j, -, -, R, [Stack] \rangle}{}$

The “Call subst” rule is the rule which starts the recognition of a substitution of the initial tree α on the node N of the tree γ between the indices i and j . “Call adj” starts the recognition of the adjunction of the auxiliary tree β on the node N of an elementary tree γ between i and j . “Call axiom” starts the recognition α of an elementary tree spawning the whole string. “Call no subs” starts the recognition of a node N of a elementary tree γ dominating the empty node between the indices i and j . “Call foot” starts the recognition of a subtree dominated by the node N_γ between the indices i and j , the node N_γ was the start of the adjunction of the auxiliary tree β and $*N_\beta$ its foot node.

In order to avoid the “call adj” rule to be over generating, we control the size of the stack by the number of possible adjunctions at a given state : if the automata has no cycle and if each state of the automata goes forward (j always superior to i), the number of possible adjunctions on a spine (the path between the root of an auxiliary tree and its foot) is bounded by the length of the string to be analyzed.

FIG. 6 – Shared forest derivation inference rules