The Semantics of Pragmatic Connectives in TAG. The French *donc* example

J. Jayez and C. Rossari

February 25, 1999

1 Introduction

Pragmatic connectives or *discourse markers* can be defined roughly as adverbs or conjunctions which link sentences or parts of discourse. Standard examples are oppositive connectives (*however*, *yet*, etc.), rephrasing connectives (*in other words*, *in short*, etc.), or consequence connectives (*so*, *therefore*, etc.). Quite a few studies have been devoted to them. See for example (Reichman 1985, Schiffrin 1987, Knott 1996) for English, (Lang 1984, Grote and Lenke 1995) for English and German, (Roulet et al. 1985, Jayez 1988a, Moeschler 1989) for French, (Rossari 1994) for French and Italian, (Ferrari 1995) for Italian. The background perspective of this paper is that of computer–assisted language learning (CALL, Holland et al. 1995). The role of examples is crucial in CALL. Typically, learners are presented with databases of examples, where patterning is dependent upon linguistic properties. Constructing such databases requires that intuitive linguistic analyses be couched, whenever possible, into an explicit and controllable format. Considerations of efficiency are less important than flexibility and accuracy, since information is in a sense 'known' to the designer who wants to check its coherence and adequacy. In this paper, we will be concerned with the subtask of integrating lexico–syntactic and semantic properties, which is of special importance in the case of connectives because syntactic cues function also as semantic cues.

We propose to describe the interrelation between syntax and semantics in a modified version of the formalism of synchronous TAGs (Shieber & Schabes 1990), which allows one (i) to factor the interacting constraints (a general property of TAGs), (ii) to express the interplay between syntax and semantics (a general property of synchronous TAGs), (iii) to take into account a richer, 'non–local' semantics (a property of the formalism we consider). As an illustration, we will present here the French connective *donc* in its so–called 'deductive' or 'inferential' use,[1] partially analogous to that of English *therefore* in sentences like 'It is cold, therefore I will stay at home'. We will first point out its basic syntactic and semantic properties (sections 2 and 3), before presenting our modification of the synchronous TAGs formalism and showing how to code some of the observations in it (section 4).

[1] We will not consider here other uses of *donc*.

2 Syntactic properties

The distinction between VP and sentence adverbs is currently used in the syntax of adverbials (see Ernst 1984 for English, Molinier 1984 for French). *Donc* is usually classified as a conjunction in traditional French grammar. Molinier (1984) considers that items like *donc* have a hybrid behavior: syntactically, they are adverbs, since, unlike conjunctions, they can occupy different positions in the sentence; semantically, they conjoin propositions. Accordingly, he uses the label *conjunctive adverbs* for this class.[2] This terminology is potentially misleading, for it can induce one to believe that items like *donc* have mixed syntactic properties, but it is so widespread that we will use it in this paper. The class of conjunctive adverbs includes items in various semantic categories, such as consequence or oppositive connectives like *alors* ('then') or *pourtant* ('yet'). The behavior of this class with respect to position is more complex than is generally acknowledged. When no other adverb intervenes, these connectives can occupy the initial or V–internal position. Other positions are also possible, generally when the connective is isolated by pauses. In the following, || notes a pause and Δ the position of *donc* in an ideal word–to–word translation.[3]

(1) a. Donc Marie est venue
Δ Mary has come
'Therefore Mary came'

b. Marie est donc venue
Mary has Δ come
'Therefore, Mary came'

c. Marie || donc || est venue
Mary || Δ || has come
'Therefore, Mary came'

d. Marie est venue || donc
Mary has come || Δ
'Mary came, therefore'

Some conjunctive adverbs, including *donc*, must be immediately to the right of the verbal flexion carrier (the verb or the auxiliary when it exists) . Some other conjunctive adverbs are less constrained:

(2) a. Marie a donc vu Pierre
Mary has Δ seen Peter
'Therefore, Mary saw Peter'

b.*Marie a vu donc Pierre
Mary has seen Δ Peter
'Therefore, Mary saw Peter'

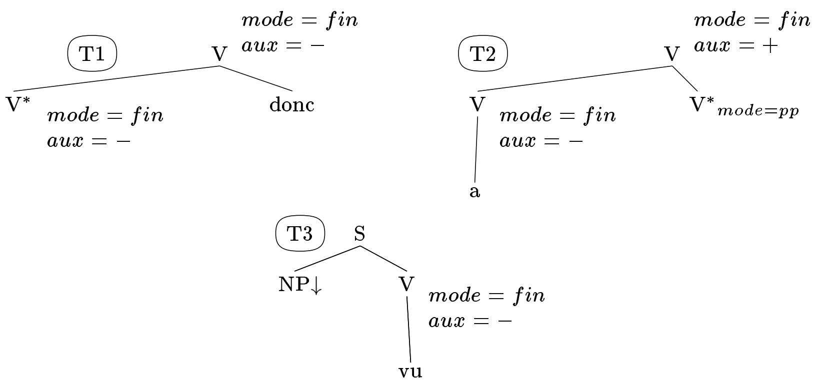
[2] Piot (1993) endorses the same principle of classification

[3] We will not in general use a word–to–word translation but rather a rough paraphrase. English approximate equivalents are sometimes provided for French connectives, for instance *then* for *alors*, etc.

c. Marie a vu aussi/ensuite Pierre
Mary has seen also/next the same Peter
'(Next,) Mary saw (also) Peter'

The restriction on internal position can be simulated in TAG by means of two constraints. First, as noted by Anne Abeillé (p.c.), we can dispose of the cases with auxiliaries and modals by requiring that *donc* be right–adjoined only to non–aux verbs, that is, verbs which are not constructed with any auxiliary or modal. This is done by a feature $aux = -$. It will prevent forms like **a vu donc* (= 'has seen Δ') or **peut voir donc* (= 'may/can see Δ') to occur. Second, it must be required that *donc* be right–adjoined only to finite verbs, to prevent forms such as **Marie est venue pour voir donc Pierre* (= 'Mary came to see Δ Peter') to occur. An open question is whether we can package these two different constraints into a more abstract, unique, constraint. The next three trees show how the two restrictions could be implemented. Past participles (*pp*), infinitivals and gerunds are subsumed by $\neg fin$.

FIG. I – *Position of* donc



Adjoining T1 to T3 produces a tree with two V nodes marked by $aux = -$, which precludes any adjunction of T2. Adjoining T2 to T3 produces a tree with three V nodes, of which only the *avoir* node is open to adjunction of T1. The upper V node blocks adjunction by $aux = +$ and the *voir* node does so by *mode* = *pp*. The initial *donc* is considered as a sentence modifier. Other positions are disposed of by incorporating pauses in the representation.[4]

When other adverbs cooccur with *donc*, the mutual positions are determined by the adverbial hierarchy, which is responsible for contrasts like the following.

[4] In the treatment we sketch here, we consider pauses to be syntactic informations. A complete treatment should analyze the pauses as fully–fledged intonational entities, or *breaks*, for instance along the lines of (Beckman and Ayers 1994).

(3) a. Marie a (donc fréquemment / *fréquemment donc) interrogé Pierre
(Mary has (Δ frequently / frequently Δ) questionned Peter)
b. Donc, indiscutablement / *Indiscutablement, donc) Pierre s'est trompé
(Δ, indisputably / indisputably, Δ) Peter was mistaken)

In some GB approaches (Shlyter 1977, Laenzlinger 1998, Cinque, to appear), the possibility of having different 'surface' positions for an adverb is explained by movement rules. A treatment in agreement with the general philosophy of TAG would consist of constraining the relative position of adverbs by propagating features which mark the possibility /impossibility of adverbs from a given category to the right/left of some given adverb.[5] We will not discuss this point here.

3 Semantic properties

3.1 Modal strength

Since connectives are essentially linking devices, they raise two questions:
(i) which kind of entities does the linking involve?
(ii) which kind of semantic relation does the linking correspond to?
We will assume for simplicity that connectives link propositions, although the exact situation is much more complex (see Jayez and Rossari 1997 for a discussion). Intuitively, for most speakers, *donc* introduces a more definitive conclusion than *alors* ('then'). Iatridou (1994) proposes to correlate two semantic conditions to a form *if* X *then* Y:
(a) The speaker asserts that Y holds whenever X holds.
(b) The speaker presupposes that there exists at least one situation where X is false and Y is false.
The second condition has to do with the causal or implicative relevance of a premise with respect to a given conclusion. We will call the condition (a) a *modal* condition, since it amounts to use the standard necessary implication of modal logic (Chellas 1980).

If *donc* is modally stronger than *alors*, we should relax Iatridou's condition on *then* for *alors*, for instance by substituting (a') for (a):
(a') The speaker asserts that Y holds in some situations where X holds.
If we adopt condition (b), it is still possible to distinguish *alors* from *et* (*and*), which would presumably satisfy the condition (a') (but not the condition (b)). However, to distinguish (a) from (a') with respect to the pair *donc* vs *alors*, we must show have linguistic pairs where one of the connectives is clearly clumsier than the other. Some dialogue pairs support the distinction.[6]

[5] For examples of this and similar techniques, see Abeillé 1991 upon inserting French clitics in their standard order, Hockey and Mateyak 1998 for English determiners and Abeillé and Godard 1997 for negative adverbs in an HPSG framework.

[6] There are also non–dialogue examples, such as *Je vais prouver que mon client n'était*

(4) a. A – Tu es libre ce soir ? (Are you free tonight ?)
B – Oui (Yes)
A – Alors / # Donc, allons au cinéma (Then / Therefore let's go to the movies)

A's reply with *donc* is perfectly normal if, for instance, the issue of going to the movies has already been discussed, and the dialogue participants have agreed that they would go to the movies if they had the time. This suggests that *donc* will be preferably used in situations in which the link between the two propositions p and q, in a form p *donc* q, is in some sense 'public'. Convergent observations on *donc* and *alors* (Hybertie 1996) suggest that *donc* presupposes that the consequence relation from p to q is shared by the participants in the linguistic exchange (in particular the addressee). A similar distinction seems to hold for the English pair *so* (or *then*) vs *therefore.* We propose a formal account of this distinction in (Jayez and Rossari 1997). We will reflect it here in a very simple way, by using a feature `modvalue`. For connectives like *donc* and *par conséquent*, the value of `modvalue` is **strong**, while it is **weak** for *alors*, *de ce fait* and *du coup.* The **strong** vs **weak** distinction is crucial when consequence connectives interact with conditional sentences.

3.2 Interaction with conditional sentences

In this section, we show that *donc* is sensitive to the type of reasoning of which it introduces the conclusion. Specifically, consider a sequence of sentences S_1, S_2, ... *if* S_{n-1} DONC(S_n), where DONC(S_n) denotes the conclusion in which *donc* occurs. It turns out that *donc* will be more or less natural according to which kind of reasoning the sequence $S_1 \ldots S_n$ exemplifies. This shows that the use of *donc* is controlled in some cases by *non–local* factors, that is, factors which may not be detected in the sentence where *donc* occurs. Such factors characterize sequences of sentences.

Following a long tradition in logic, we will call *enthymematic* any reasoning where some premises are lacking. The manifest premises are called *explicit*, while any reconstructed premise, necessary for the reasoning to be conclusive, is *implicit.* Most reasonings of everyday life are enthymematic. This is due to their non-monotonic character, which is probably due in turn to the impossibility of describing causality in an exact way. In most current situations, we retain only a part of the causal factors which are involved in bringing about the situation. We often neglect negative factors, which could prevent the situation.[7] Some enthymematic reasonings can be extended to perfectly monotonic reasonings, when the combination of

pas chez lui le jour du meurtre. Donc/??Alors j'établirai qu'il est innocent ('I am going to prove that my client was not at home on the day of the murder. Therefore/Then I will demonstrate that he is not guilty'). If a lawyer utters this sentence, she unduly weakens the essential conclusion (that his client is innocent) by using *alors* .

[7] See (Geffner 1992) on causal analysis and non-monotony.

implicit and explicit premises is equivalent to a logical inference, that is, to an inference licensed by some logical system. The two following examples illustrate this case.

(5) a. A is superior to B, B is superior to C, so A is superior to C
    b. If John is not the culprit, it is Sam

In (5a), we have a logical reasoning which does not require that we provide additional factual premises.[8] In (5b), if we add the implicit premise 'the culprit is necessarily John or Sam', we obtain again a logical reasoning, which exemplifies the deduction rule called *disjunctive syllogism*: from $A \vee B$ and $\neg A$, infer B. In contrast, in many cases we find non–logical reasonings, where the conclusion holds non–monotonically. Consider (6a):

(6) a. If the weather is fine, I'll take a walk
    b. Every equilateral triangle is equi–angled, if every equi–angled triangle has three 60 degrees angles, every equilateral triangle has three 60 degrees angles

In (6a), we have a typical non-monotonic reasoning: while the connection between weather and walks is intuitive, it is highly defeasible. One can easily imagine reasons for not inferring 'I'll have a walk' from 'the weather is fine'. For instance, the speaker may be tired or engrossed in some urgent work. Positing a special rule like 'I always have a walk when the weather is fine' will not turn the non–monotonic reasoning into a rigid one, because such a rule is not a hard fact nor a logical law, but rather a generic or habitual proposition, which is open to exceptions.

We will distinguish between these two types of reasoning by means of logical labels or reasoning *types*. Logical reasonings will get the label **logreas**, non–logical ones the label **empreas** (for *empirical reasoning*). The interaction of *donc* with conditional sentences can be summarized by two general observations.

1. Logical reasonings (reasonings of type **logreas**) license the internal *donc*. They seem to tolerate the initial *donc* when the conclusion and the non–conditional premise are of the same generality type (they are both a fact or both a rule). In (5a) and (5b), the conclusion is a fact and the non–conditionnal (explicit or implicit) premise is also a fact. In (6b), the conclusion is a rule and the non–conditional explicit premise is also a rule. *donc* would be (more or less marginally) allowed in such cases.

In contrast, in the following example, the conclusion is a fact while the non–conditional explicit premise is a rule.

(7)
  *Tout triangle rectangle a un angle droit. Si ce triangle est rectangle, donc il a un angle droit

[8]However, note that the reasoning requires that we provide the law of transitivity for 'superior to', to be correct. In general, reasonings which sound 'logical' lack only obvious general axioms schemas or inference rules, such as transitivity or *modus ponens*.

(Every rigth–angled triangle has a 90 degrees angle. If this triangle is right–angled, therefore it has a 90 degrees angle)

In (Jayez and Rossari 1997), it is proposed that these variations are caused by the dynamic informational structure of the reasoning. By and large, reasonings which *resemble* logical reasonings will host the initial *donc* more naturally. By varying and combining parameters such as the nature of the reasoning, its degree of explicitness, the nature of the premises (facts or rules), it is possible to devise a very flexible system. However, calibrating the set of parameters would require a psycholinguistic investigation. Since our aim in this paper is to illustrate a general technique, we just assume that the initial *donc* is licensed in logical reasonings where the non–conditional premise and the conclusion are of the same nature (rule or fact). Reasonings of this kind will be called *homogeneous* logical reasonings and will get the type **homlogreas**, which is a subtype of **logreas**.

2. Empirical reasonings license the internal *donc* only when the main non–conditional premise is explicit. Compare:

(8) a.*S'il fait beau, j'irai donc me promener
(If the weather is fine, I will therefore have a walk)
b. Je vais toujours me promener quand il fait beau. S'il fait beau, j'irai donc me promener
(I always have a walk when the weather is fine. If the weather is fine, I will therefore have a walk)

Such observations suggest that the internal *donc* may link propositions more freely than the initial *donc*: in a structure *si* X *donc* Y, the implicational link provided by the initial *donc* holds between X and Y, while it can hold between an implication $X \Rightarrow Y$ (corresponding to the *si*) and some external (set of) proposition(s) if the *donc* is internal. So, the meaning of (8b) can be paraphrased by 'I always have a walk when the weather is fine, therefore, if the weather is fine, I will have a walk'. Observations where there is no implicational link between the *si*–sentence (X) and the last sentence (Y) lend support to this hypothesis. In such cases *donc* may not be accounted for by assuming that it expresses an implicational link between X and Y, since such a link is simply missing. So, we must assume that *donc* draws its premises from a source different from X. This is the case in the following example.

(9)

Marie est cachée dans la maison. Si Jean arrive, il ne la verra donc pas
(Mary is hidden in the house. If John comes up, he will therefore not see her)

We have no implicational connection between 'John comes up' and 'John will not see Mary', as evidenced by the oddity of 'John came up. Therefore he did not see Mary'. So, we must assume that the meaning of (9) is roughly: 'Mary is hidden in the house. Therefore, if John comes up, he will not see her'. In short, we propose that, in a structure D *if* X *donc* Y, where *donc* is internal and D is a discourse sequence, there is always the possibility of a reading D, *donc if* X Y. When there is no explicit D, the sentence is acceptable if the reasoning obtained by reconstructing a premise is logical.[9] Dealing with the reviewed data require that we make a difference between 2 and 3–propositions reasonings. Logical reasonings of form premise–conclusion (resp. premise1–premise2–conclusion) will be labelled **2logreas** (resp. **3logreas**). A similar distinction holds for empirical reasonings for which we distinguish between **2empreas** and **3empreas**.[10] When the premise before the conclusion is not conditional *donc* requires simply that the reasoning be at least of type **2logreas** or **2empreas**. Of course, more explicit reasonings (**3logreas** or **3empreas**) will make *donc* even more natural.

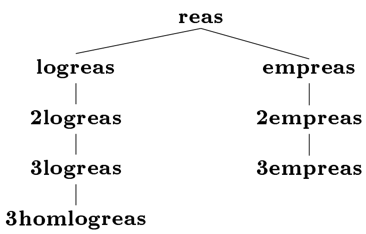
Summarizing, in a structure D–DONC(p), where D is some sequence of sentences or *discourse*, and p a final sentence introduced by *donc*, we have the following cases.

FIG. II – *Donc and inferential types*

	q DONC(r)	p, si q DONC(r)	si q DONC(r)
initial *donc*	**logreas** $\vee$ **empreas**	**3homlogreas**	bad
internal *donc*	**logreas** $\vee$ **empreas**	possibly = p DONC(si q, r)	**logreas**

We interpret a type as **2logreas** as the type of logical reasonings which have *at least* one explicit premise. Types like **3homlogreas** are the conjunction of the types **3logreas** and **homlogreas**.[11] Under this interpretation, we have the following hierarchy of types.

FIG. III – *The hierarchy of inferential types*



[9] We conjecture that this is caused by the nature of the reconstructed premise. In a **logreas** this premise is either a fact, like 'either John or Sam did it' in (5b), or a logico–mathematical rule, true in virtue of a definition or of a proof. Such premises are easier to contemplate than empirical, generic or habitual, truths, like 'I always have a walk when the weather is fine', etc., which are by nature defeasible.

[10] In fig. X below, we use also **3reas**, which means **3empreas** $\vee$ **3logreas**.

[11] Recall that this type is assigned to premise1–premise2–conclusion logical reasonings where the first premise and the conclusion are both facts or rules.

So, the types to which *donc* is sensitive may not be assigned to isolated sentences and we must find a way of noting them in the representation, at least if we want to capture some differences exhibited by conditional sentences.[12] This is the topic of the next section.

4 The coding of semantic constraints

The most natural idea for coding reasonings is to assign types such as **logreas** or **empreas** to predicative skeletons of sentence sequences, keeping only, for instance, the predicate–argument structure and the tenses. The representation of (8b) along these lines would be.

FIG. IV – *Information structure*

$$\begin{bmatrix} \text{RELN} & \textit{implication} \\ \text{PREMISE} & \begin{bmatrix} \text{RELN} & \textit{fine} \\ \text{ARG} & \textit{weather} \end{bmatrix} \\ \text{CONSEQ} & \begin{bmatrix} \text{RELN} & \textit{have a walk} \\ \text{ARG} & \textit{speaker} \end{bmatrix} \\ \text{TENSE} & \text{present, generic} \end{bmatrix} \begin{bmatrix} \text{RELN} & \textit{fine} \\ \text{ARG} & \textit{weather} \\ \text{TENSE} & \text{present} \end{bmatrix} \begin{bmatrix} \text{RELN} & \textit{have a walk} \\ \text{ARG} & \textit{speaker} \\ \text{TENSE} & \text{futur} \end{bmatrix}$$

It would be interesting in computer–assisted language teaching to be able to store and maintain a database of such textual skeletons. It would also be desirable to take into account simple semantic entailments and pragmatic implicatures. For instance, an entailment such as *x has a walk* $\Rightarrow$ *x goes out* would license the following typing entailment:
If the representation of *I always go out when p, D* gets the type σ, the representation of *I always have a walk when p, D* gets the same type σ.
Implicatures like *I feel like walking* $\rightsquigarrow$ *I'll have a walk* can also be useful in deriving some type assignments. Generally speaking, the more we gain in flexibility and control over abstract semantic structures, the more we are able to vary examples and to make the effect of semantic shades apparent. If we want to keep an eye on syntax and to preserve a certain amount of compositionality, a natural candidate is a synchronization mechanism, in which syntactic composition and semantic construction are paired, whenever possible and desirable.

4.1 Synchronous TAGs

TAGs offer the possibility of synchronizing syntactic and semantic operations in the formalism of *synchronous* TAGs (Shieber and Schabes 1990, Shieber 1994, Rambow and Satta 1996). However, there are two general limits to the mechanism of synchronous TAGs, whatever version of it we choose.

[12]The problem is actually more general, since such variations exist for many *presuppositional* conjunctions, e.g., *puisque* (*since*), *comme* (*as*), *vu que* (*given that*). Such conjunctions assert the proposition they introduce and presuppose a causal/explanatory link between this proposition and other propositions.

First, TAGs in general and hence synchronous TAGs, do not not incorporate a typing mechanism. The feature-value pairs used in unification when adjunction or substitution operations take place can be 'propagated' (by sharing) across several nodes, but they do not characterize the information in a tree in a global way. In typed feature structures (Carpenter 1992), types are used to summarize the identity of attributes and values that an entity of a given type may possess. More importantly, they are strongly connected with subsomption. If D is an information structure of type σ containing a substructure D', the result of substituting D" for D' in D produces a new structure of type σ whenever D' *subsumes* D", that is contains less information than D". For example, replacing *going out* by *having a walk*, as in the example above amounts to replace the substructure [RELN *going out*] by the (subsumed) substructure [RELN *having a walk*]. More sophisticated examples of lexical hierarchies and inheritance suggest that it will be difficult to capture interesting generalizations without some form of typing (Davis 1996). The second problem is not a direct consequence of the formal definition(s) of TAGs, but is rather associated with their dominant use. Usually, efficiency considerations are essential in TAG–based NLP application and feature handling is limited to what is necessary. Splitting the information structure into more complex feature structures is theoretically possible. However, this would lead to hybrid entities, which behave as trees, when viewed from the dominance perspective, and as feature structures, when one looks at the complex features hosted by the tree nodes. This is not conceptually very satisfactory.

This prompted us to dispense with the trees altogether in the right projection of the pairs used by synchronous TAGs. Instead of having $\langle tree_1, tree_2 \rangle$ pairs, we have $\langle tree, D \rangle$, where D is a *description* in the sense of (Carpenter 1992). Adjunctions and substitutions are redefined so as to apply to descriptions. This does not buy us any computational advantage, since we keep adjunction/substitution on one side and unification on the other, but, as noted in the introduction, in the domain of CALL, clarity and flexibility are, in some circumstances, more important than speed.

4.2 Modification of synchronous TAGs

Before turning to the modification, let us give an intuitive picture of what we are after. Let us assume that a set of information structures corresponding to sequences of sentences is available. E.g., there is an information structure which could be expressed by the surface sequence *Il fait beau. J'irai me promener* ('The weather is fine. I'll have a walk'), another which could be expressed by the sequence *Je vais toujours me promener quand il fait beau. Il fait beau. J'irai me promener* ('I always have a walk when the weather is fine. The weather is fine. I'll have a walk'), and so forth. Simple sentences, such as *Il fait beau* ('The weather is fine') may also have representatives in this set. At least one inferential type is assigned to any information structure. This set of assignments forms what we call an *inferential database*, or an IDB for short. In addition to an IDB, we have sequences of sentences or simple sentences which are associated with a derivation tree. The idea is to rate the 'goodness' of these (sequences of) sentence(s) against the IDB, for instance by observing that, up to some level or node in the derivation, the (sequence of) sentence(s) is (are) ok, but further derivation yields an anomalous object which may not be labelled by one of the assignments of the IDB. For space reasons, we focus only in this paper on the possible mismatch between the IDB and the trees which form the partial results of a derivation. Many other aspects are ignored, such as the coding of the derivation trees or the operations on the IDB.

Synchronous TAGs match pairs of *trees*. A pair of trees $\langle T_{i_\alpha}, T_{i_\beta} \rangle$ operates on another pair of trees $\langle T_{j_\alpha}, T_{j_\beta} \rangle$ whenever T_{i_α} and T_{i_β} operate simultaneously on T_{j_α} and T_{j_β} respectively, according to the constraints imposed by the model. In the present approach, the basic entities are pairs $\langle T_i, D_i \rangle$, where T_i is a syntactic tree as in synchronous TAGs, while D_i is an untyped feature structure description in the sense of (Carpenter 1992). Untyped feature structures are attribute–value matrices. Feature structure descriptions or simply 'descriptions' are finite sets of constraints which characterize a (possibly empty) *class* of feature structures. Using descriptions provides for the required flexibility but keeps us within the bounds of unification techniques.

More precisely, let π, π', etc., be finite sequences of attributes or 'paths', over a fixed alphabet of attributes, and Σ a set of types. We define recursively the notion of description.

Descriptions

- any $\sigma \in \Sigma$ is a description,
- if π and π' are paths, $\pi \doteq \pi'$ is a description,
- if D is a description and π a path, $\pi : D$ is a description,
- if D and D' are descriptions, $D \wedge D'$, $D \vee D'$ and $\neg D$ are descriptions,
- if D and D' are descriptions, the list $\langle D, D' \rangle$ is a description
- if π is a path and X a variable over descriptions, $\pi : X$ is a description.

The *terms* of a description D are defined standardly as D and D', if D is $D \wedge D'$ or $D \vee D'$, and as D itself in any other case. An expression *occurs* in a description iff it is a term of this description. A *subdescription* in a description D is D itself or any description D' such that $\pi : D'$ occurs in a subdescription of D for some π.

There are *a priori* many possible kinds of synchronization rules. We limit ourselves to those which are strictly necessary in the present framework, leaving a general discussion of the possibilities to another work.[13]

[13] In contrast with (Kikui 1994), we keep a strong notion of synchronization in order to make the pairing between syntax and semantics non–directional.

Synchronization rules
Let $\langle T_i, D_i\rangle$ and $\langle T_j, D_j\rangle$ be two tree–description pairs. We assume that a node is linked to a variable iff it is linked to each occurrence of this variable. When an expression is substituted for a variable in a description, it is substituted for the variable in all the subdescriptions where this variable occurs.

1. If T_j is substituted in T_i at node n_i to get T_i', and node n_i is linked to the variable X in D_i, D_j is substituted for X in D_i to get D_i'. The resulting pair is $\langle T_i', D_i'\rangle$.
2. If T_j is adjoined to T_i at node n_i to get T_i', node n_i is linked to the subdescription D_i' in D_i, and the footnode of T_j is linked to the variable X in D_j, D_i' is substituted for X in D_j to get D_j'. The resulting pair is $\langle T_i', D_j'\rangle$

4.3 The use of inferential typing

In the grammar described in the next section, the semantic content is captured as a set of attributes. For each simple sentence, we store:
– the inferential type in an attribute `infty`,
– the type expected in subsequent unifications, in an attribute `reqty`,
and, in `cont` (for *semantic content*),
– the grammatical tense of the sentence in an attribute `tense`,
– The predicative structure in an attribute `predstr`, which points to the relation (`reln`) and its arguments (`arg1`, `arg2`, etc.).
When sentences are combined their semantic contents are merged in the upper `cont` by the $\bullet$ operator defined by:
– $L_1 \bullet L_2 = L_2$ if L_1 is the void list,
– $L_1 \bullet L_2 = \langle L_1, L_2\rangle$ otherwise.[14]
Abstractly conceived, the elements of an IDB are descriptions. We will not consider here the possible solutions for compiling the IDB information under a more economic format. Let D be a description constructed in agreement with the two synchronization rules given above. D contains various information, in particular semantic information, stored as the value of the attribute `cont`. Every description D has a feature `infty` which holds the result of comparing D with the assignments in the IDB. We will note the existence of this comparison by the (pseudo) equation `infty` = **IDBcheck**, where **IDBcheck** is a function defined by the following condition.

IDBcheck
For a given description D and an IDB, **IDBcheck** returns $\sigma_1 \vee \ldots \vee \sigma_n$, for each type $\sigma_i\ _{(1 \leq i \leq n)}$ such that there is an assignment $\langle D', \sigma_i\rangle$ in the

[14] $\bullet$ is used here instead of standard concatenation in order to save internal groupings, which can be assigned inferential types.

IDB and D' subsumes[15] the value of `cont` in D. If there is no such type, **IDBcheck** returns the type **unknown**.

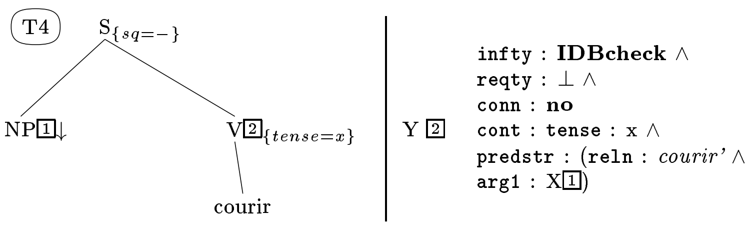
The feature `reqty` stores the expected type, to be propagated upwards in a combination of sentences. E.g., according to table II, a sentence of form *donc* S demands that the sequence it terminates be of type **3homlogreas**.

The fact that a sentence contains a connective is registered by putting a feature `conn` in the description. This feature points to a subdescription which contains information on (i) the position, *via* a feature `pos`, which can have values such as **init** (for the sentential initial position) or **int** (for the internal position), (ii) the category of the connective, *via* a feature `connsrt` which holds the value **cons** in the case of consequence connectives like *donc*, (iii) the modal strength, as explained juste before section 3.2 and (iv) the type expected by the connective (the value of `reqty`). So a description value for the feature `conn` has always the form (`pos` $= \ldots \wedge$ `connsrt` $= \ldots \wedge$ `modvalue` $= \ldots \wedge$ `reqty` $= \ldots$).

At the present stage, we have (ideally) at our disposal: (i) an IDB, that is, a set of assignments which determines, for each information structure (formally, a description), the value of its attribute `infty`, a skeleton for the descriptions to be used in the synchronous TAGs grammar, that is, the requirement that these descriptions contain (at least) certain attributes (`cont`, etc.), (iii) synchronization rules. It remains to provide the grammar itself, that is, the set of pairs $\langle tree, description \rangle$. We will present it in a stepwise fashion. In the schemas to come, tags $\boxed{i}$ note the linkings between the syntactic tree and the description. **unknown** is subsumed by any type. $\perp$ is the most general type, which subsumes any other type (in particular **unknown**).

4.3.1 Simple sentences

FIG. V – *Simple sentences*



Simple sentences are non sequential ($sq = -$). The basic predicative structure is introduced at this level. The required type does not matter at the level of simple sentences, since the corresponding feature is discourse

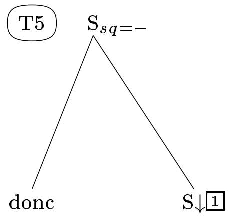
[15]Recall that a description D_i subsumes a description D_j iff any feature structure realizing D_j realizes D_i. We extend this definition to lists: L subsumes L' iff L and L' have the same length and each element of L at position k subsumes the corresponding element at position k in L'.

sensitive, not sentence sensitive, whence the all-purpose value ⊥. Simple sentences have no connective (conn has the boolean value **no**).

4.3.2 Connectives

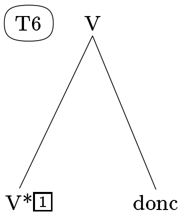
Connectives are introduced in simple sentences or in VPs. The type expected by the sentence, i.e., the value of reqty at the top of the description, is the same as the type expected by the connective. Since *donc* is most naturally hosted by empirical or logical reasonings it expects the disjunction of these two types. The connective position, internal or initial, is added at substitution or adjunction.[16] Features such as su1, su2, etc., contain different pieces of information or 'semantic units'. There is no semantic difference between the two *donc*. They are differentiated by the value of the feature pos. The structure for conditionals (fig. X) is sensitive to the value of pos. This reflects the fact that the final type expectations are triggered conjointly by the connective and by the conditional structure.

FIG. VI – *The sentential* donc



su1 : X1 ∧ reqty ≐ conn|reqty ∧
infty ≐ su1|infty ∧
cont|tense ≐ su1|cont|tense ∧
cont|predstr ≐ su1|cont|predstr ∧
conn |pos : **init** ∧
conn |connsrt : **cons** ∧
conn |modvalue : **strong** ∧
conn |reqty : **empreas** ∨ **logreas**

FIG. VII – *The internal* donc



su1 : X1 ∧ reqty ≐ conn|reqty ∧
infty ≐ su1|infty ∧
cont|tense ≐ su1|cont|tense ∧
cont|predstr ≐ su1|cont|predstr ∧
conn |pos : **int** ∧
conn |connsrt : **cons** ∧
conn |modvalue : **strong** ∧
conn |reqty : **empreas** ∨ **logreas**

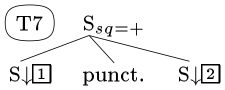
4.3.3 Sequences of sentences

We distinguish between a sequence of sentences in general, and a *text*, which is sequence of sentences with an inferential type different from **unknown**. Sequences of sentences are assigned a required inferential type (reqty) which is the unification of the required inferential types of their component sentences or sequences of sentences, but this operation has not to yield a special type. When a sequence has no match in the IDB, it gets the inferential type **unknown**. This does not prevents it to be embedded in a larger sequence with a 'correct' (i.e., ≠ **unknown**) match. Consider example (9). The sequence 'If John comes up, he will therefore not see

[16]For simplicity, we assume that sentences have at most one connective.

Mary' is semantically ill–formed, because checking the IDB for the type to be assigned to the concatenation of the semantic contents (values of `cont`) of the two sentences will return nothing. The sequence will then get the inferential type **unknown**. However, when one adds 'Mary is hidden in the house', the new sequence gets the inferential type **3empreas** which licenses the internal *donc*. For simple sequences, we have the following.[17]

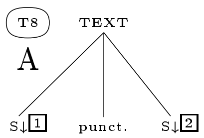
FIG. VIII – *Simple sequences*



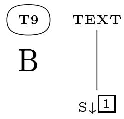
$\texttt{su1} : X\boxed{1} \wedge \texttt{su2} : Y\boxed{2} \wedge$
$\texttt{reqty} \doteq \texttt{su1|reqty} \wedge \texttt{reqty} \doteq \texttt{su2|reqty} \wedge$
$\texttt{cont} : (\texttt{su1|cont} \bullet \texttt{su2|cont})$

Turning to texts, we simply exclude the type **unknown**. The `cont` value is obtained by chaining the `cont` values of the two terms. The inferential type must unify with the types required by the left and right sequences and must be different from **unknown**.

FIG. IX – *Two–sequence and one-sequence texts*



$\texttt{su1} : X\boxed{1} \wedge \texttt{su2} : Y\boxed{2} \wedge$
$\texttt{cont} : (\texttt{su1|cont} \bullet \texttt{su2|cont}) \wedge$
$\texttt{infty} : \textbf{IDBcheck} \wedge$
$\texttt{infty} \doteq \texttt{su1|reqty} \wedge$
$\texttt{infty} \doteq \texttt{su2|reqty} \wedge$
$\neg(\texttt{infty} : \textbf{unknown})$



$\texttt{su1} : X\boxed{1} \wedge$
$\texttt{cont} \doteq \texttt{su1|cont} \wedge$
$\texttt{infty} : \textbf{IDBcheck} \wedge$
$\texttt{infty} \doteq \texttt{su1|reqty} \wedge$
$\neg(\texttt{infty} : \textbf{unknown})$

To illustrate this, consider possible derivations of a two–sentence text like *Il fait beau. J'irai donc me promener* ('The weather is fine. Therefore I'll take a walk'). We start with two trees Ta and Tb corresponding to *Il fait beau* and *J'irai me promener*. We have two possibilities.

a. We substitue Ta and Tb in T8 (fig. IX). At this stage, the value of `infty` is fixed by **IDBcheck**. Since Ta and Tb correspond to simple sentences, which put no specific requirement on types, the value of `infty` will be determined by **IDBcheck**. However, the constraint `infty`≐`su2|reqty` will restrict the possible updates on the value of `su2|reqty`. Such an update occurs in particular when we adjoin T6 to the subtree corresponding to *J'irai me promener*. The value of `reqty` must be unifiable with that of `conn|reqty`, that is **empreas** ∨ **logreas**. So, now, the value obtained by **IDBcheck** must be unifiable with **empreas** ∨ **logreas**. This is the case because the initial sentence (without *donc*) would get the type **empreas**, among other possibilities.

b. We adjoin T6 to *J'irai me promener*. The disjunctive value **empreas** ∨ **logreas** is thereby assigned to `reqty`. It will matched against the result of **IDBcheck** when *J'irai donc me promener* enters the substitution process to construct a text of form T8.

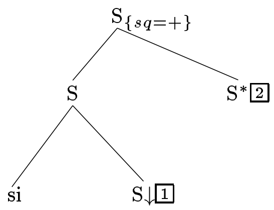
[17]To keep the trees lexicalized, we use the punctuation as an anchor, according to the suggestion of a reviewer.

Note that it is not possible in this model to redeem a sentence pair which gets **unknow** by putting a connective in the second sentence. The types compatible with the connective must already be supported by the sentence pair without the connective.

4.3.4 Conditional sentences

FIG. X – *Conditional sentences*[18]

T10



su1 : X$\boxed{1}$ ∧ su2 : Y$\boxed{2}$ ∧
cont : (su1|cont • su2|cont) ∧
$[_1$ su2|conn : **no** ∧
reqty ≐ **condtype** ⊔ su1|reqty ⊔
su2|reqty $]_1$
∨
$[_2$ su2|conn|connsrt : **cons** ∧
su2|conn|modvalue : **weak** ∧
reqty ≐ **condtype** ⊔ **reas**
⊔ su1|reqty ⊔ su2|reqty $]_2$
∨
$[_3$ su2|conn|connsrt : **cons** ∧
su2|conn|modvalue : **strong** ∧
(($_\alpha$ su2|conn|pos : **int** ∧
(⟨reqty ≐ **logreas** ⊔ su1|reqty ⊔ su2|reqty⟩ ∨
⟨reqty ≐ **3reas** ⊔ su1|reqty ⊔ su2|reqty⟩) $)_\alpha$ ∨
($_\beta$ su2|conn|pos : **init** ∧
reqty ≐ **3homlogreas** ⊔ su1|reqty ⊔ su2|reqty) $_\beta$) $]_3$

The description of this pair contains (i) general compositional information (the first two lines of the description) and (ii) a disjunction of descriptions, corresponding to three different cases. The compositional information simply connects the first and the second clause of the conditional form *si* X, Y to two different semantic units (su1 and su2) and performs the usual chaining of semantic contents. At this stage, there is no information pertaining to the conditional form itself or to the occurrence of some connective. The disjunction of descriptions of form $[_1\ C_1\ ]_1 \vee [_2\ C_2] \vee [_3\ C_3]_3$, where the C_i are conjunctions, takes care of this.
The first description holds when su2 has no connective. The type required by the conditional form *si* X, Y should then unify with **condtype**, which is assumed to be the type for conditional forms.[19] The required type must also unify with the types required by the two clauses of the conditional

[18] To make this structure more compact, we abbreviate conjunctions of equations $\pi \doteq \pi_1 \wedge \pi \doteq \pi_2$ by $\pi \doteq \pi_1 \sqcup \pi_2$. We also write things like $\pi \doteq$ **type** $\sqcup\, \pi' \sqcup \ldots$, where **type** is a type.

[19] We do not address here the problem of defining appropriate semantic types for conditional forms. For simplicity, we have assumed that all the possible types for such forms are subsumed by the general type **condtype**. This is clearly an idealization, since, for instance, one would want to put so–called *austinian* conditionals (*If you are thirsty, there is beer in the frig*) into a radically different category from that of other conditionals. It would then be appropriate to have non–unifiable types rather than one general type.

form. If the content obtained by chaining the contents of `su1` and `su2` gets an appropriate type from the IDB, the conditional form yields a wellformed text, according to fig. IXB.
The second description detects consequence connectives with a *weak* modal value (such as *alors*, *du coup*, *de ce fait*). The required type must then unify with the type for conditionals *and* with the type for consequence connectives, which we assume to be **reas** of fig. III. The required types of the subclauses are also propagated. The resulting pair will be handled directly by IXB, or by IXA if some sentences are added to the conditional form.
The third description deals with strong consequence connectives. Two cases ($()_\alpha$ and $()_\beta$) are distinguished: if the connective is internal, there are two (non mutually exclusive) possibilities. First, the required type can be logical. IXA will then license examples such as (5b). Second, the required type can be **3reas**, which means that the conditional structure has to be completed by another sentence which will provide the first premise of a premise1–premise2–conclusion reasoning (empirical or logical). This simulates the possibility for the internal *donc* to license a shift from *si* X DONC(Y) to Z, DONC(*if* X, Y), where Z is some expected extra premise. Last, when the position is initial, we simply require that the expected type be **3logreas**. Needless to say, many other aspects (coordinations, conditional connectives other than *si*, use of the imparfait after *si* in French etc.) have been deliberately ignored.

5 Conclusion

In this paper, we have presented a system to express inferential constraints, usually triggered by pragmatic connectives, in a TAG–based framework which integrates synchronization and a flexible, although minimal, semantics. This allows one to link local syntactic information and non–local, textual and inferential information. Extending this approach to other (classes of) connectives requires that (i) we should provide a more complete analysis of the scope of connectives, possibly transferring the present constraints of section 2 to the semantic domain, (ii) we should elaborate the description in section 3 to integrate more refined distinctions about the inferential relation and the semantic types of entities it associates. The main advantage of TAG is to represent directly the incremental building of syntactic and semantic representations. Their main weakness is to offer only a limited typing mechanism. The strategy followed here is to keep the incrementality and to augment the typing potentialities by resorting to genuine type–based representations.

References

Abeillé, Anne. 1991. *Une grammaire lexicalisée d'arbres adjoints pour le français. Application à l'analyse automatique.* Doctoral dissertation, Université de Paris 7, France.

Abeillé, Anne, and Danièle Godard. 1997. The syntax of French negative adverbs. In *Negation and Polarity: Syntax and Semantics*, ed. D. Forget, P. Hirschbuller, and F. Martineau. Amsterdam: John Benjamins.

Beckman, Mary E., and George Ayers. 1994. *Guidelines for ToBI labelling (version 2.0).* Ohio State University.

Carpenter, Bob. 1992. *The Logic of Typed Feature Structures.* Cambridge: Cambridge University Press.

Chellas, Brian F. 1980. *Modal Logic. An Introduction.* Cambridge: Cambridge University Press.

Cinque, Guglielmo. 1998. *Adverbs and Functional Heads: A Cross–Linguistic Perspective.* Oxford: Oxford University Press. to appear.

Davis, Antony. 1996. *Linking and the Hierarchical Lexicon.* Ph.d. thesis, Stanford University.

Ernst, Thomas B. 1984. *Towards an Integrated Theory of Adverb Position in English.* Doctoral dissertation, Indiana University, Bloomington.

Ferrari, Angela. 1995. *Connessioni. Uno studio integrato della subordinazione averbiale.* Genève: Slatkine.

Geffner, Hector. 1992. *Default Reasoning: Causal and Conditional Theories.* Cambridge: The MIT Press.

Grote, B., and N. Lenke. 1995. Ma(r)king concessions in English and German. In *Proceedings of the Fifth European Workshop on Natural Language Generation, May 1995, Leiden, Netherlands.*

Hockey, Beth, and Heather Mateyak. 1998. Determining determiner sequencing: a syntactic analysis for English. this volume.

Holland, V.Melissa, Jonathan D. Kaplan, and Michelle R. Sams (ed.). 1995. *Intelligent Language Tutors: Theory Shaping Technology.* Mahwah: Lawrence Erlbaum Associates.

Hybertie, Charlotte. 1996. *La conséquence en français.* Paris: Ophrys.

Iatridou, Sabine. 1994. On the Contribution of the Conditional *then*. *Natural Language Semantics* 2(3):171–199.

Jayez, Jacques. 1988. *L'inférence en langue naturelle. Le problème des connecteurs.* Paris: Hermès.

Jayez, Jacques, and Corinne Rossari. 1997. Pragmatic Connectives as Predicates. The Case of Inferential Connectives. In *Predicatives Structures in Natural Language and Lexical Knowledge Bases*, ed. P. Saint-Dizier. Dordrecht: Kluwer Academic Publisher. To appear.

Kikui, Gen-Ichiro. 1994. A Semantic Head–Driven Generation System for Feature Structure Tree–Adjoining Grammars. *Computational Intelligence* 10(4):604–618.

Knott, Alistair. 1996. *A Data-Driven Methodology for Motivating a Set of Coherence Relations.* Ph.d. thesis, University of Edinburgh.

Laenzlinger, Christopher. 1998. *Comparative Studies in Word Order Variation. Adverbs, Pronouns and Clause Structure in Romance and Germanic.* Amsterdam: John Benjamins.

Lang, Ewald. 1984. *The Semantics of Coordination.* Amsterdam: John Benjamins. Translation from the German original *Semantik der koordinativen Verknüpfung.*

Molinier, Christian. 1984. Etude syntaxique et sémantique des adverbes de manière en –ment. Thèse d'Etat. Université de Toulouse–Le–Mirail, France.

Piot, Anne-Marie. 1993. Les connecteurs du français. *Linguisticae Investigationes* 17:141–160.

Rambow, Owen, and Giorgio Satta. 1996. Synchronous Models of Language. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, 116–123.

Reichman, Rachel. 1985. *Getting Computers to Talk Like You and Me.* Cambridge: MIT Press.

Rossari, Corinne. 1994. *Les opérations de reformulation.* Berne: Peter Lang.

Roulet et al., Eddy. 1985. *L'articulation du discours en français contemporain.* Berne: Peter Lang.

Schiffrin, Deborah. 1987. *Discourse Markers.* Cambridge: Cambridge University Press. Second edition.

Shieber, Stuart M. 1994. Restricting the weak–generative capacity of synchronous tree–adjoining grammars. *Computational Intelligence* 10(4):371–385.

Shieber, Stuart M., and Yves Schabes. 1990. Synchronous Tree–Adjoining Grammars. In *13th International Conference on Computational Linguistics*, 253–258.