D-STAG: Parsing discourse with Synchronous TAG and SDRT background

Laurence DANLOS

Université Paris 7, IUF, ALPAGE

Constraints on Discourse III – July 30st, 2008

We propose a new formalism for parsing discourse, called D-STAG, Discourse Synchronous TAG, which is

- inspired by SDRT as a discourse theory,
- akin to D-LTAG as a formalism which extends a sentential TAG syntax/semantic interface to the discourse level.

SDRT relies upon *discourse relations* of two types:

- a *coordinating (multinuclear)* relation links two Nuclei,
- while a *subordinating (nucleus-satellite)* relation links a Nucleus (head) and a Satellite (modifier).
- This allows the construction of hierarchical discourse structures richly annotated with coordinating and subordinating relations.

In D-STAG, discourse analyses are hierarchical structures which can deterministically be converted into SDRT discourse structures. As a consequence, D-STAG can take advantage of the results brought by this *discourse theory*.

- D-STAG is like D-LTAG – Discourse Lexicalized TAG (Forbes, Webber et al. 2006) – in that the two formalisms extend a sentential syntax/semantic interface to the discourse level.
- Idea behind D-LTAG and D-STAG is to build a complete integrated text understanding parsing system which incorporates the same mechanisms for the sentence and discourse levels.
- However, there exist crucial differences between D-LTAG and D-STAG (section 5)

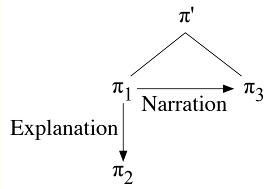
1. Introduction to SDRT
2. Introduction to TAG
3. Introduction to Synchronous TAG
4. D-STAG
   - Syntactic elementary trees for discourse connectives
   - D-STAG Parsing
   - D-STAG at the semantic level
   - Illustration of the D-STAG parser
5. Comparison between D-STAG and D-LTAG
6. Conclusion and future research

SDRT graphs for discourse structures

- Two kinds of nodes (leaving aside topic nodes)
  - π_i atomic nodes that are labels of logical forms for atomic clauses,
  - π primed nodes (π', π'') that are scope nodes which immediatly outscope atomic nodes, e.g $i - outscope(\pi', \pi_1)$. Outscoping relations are represented with dashed lines.
- Right frontier: the last atomic node and any node that dominates it via a series of outscoping and/or subordinating relations.

SDRT graph for discourse (1)

(1) John went to the supermarket because his fridge was empty. Next, he went to the movies.

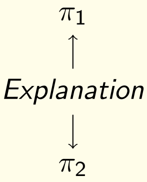


Right Frontier Constraint: The discourse constituents on the right frontier are the only available nodes for attachment of new information.

Slight modifications to SDRT graphs

- SDRT focuses mainly on how to infer discourse relations which are not made explicit through a discourse connective
- Therefore, SDRT graphs don't include nodes for discourse connectives/relations (discourse relations are just labels for arrows)
- Not standard in semantic dependency formalisms, although SDRT graphs are similar to semantic dependency graphs
- So we propose slight modifications to SDRT graphs so as to obtain pure semantic dependency graphs

Slight modifications to SDRT graphical representation of subordinating and coordinating relations

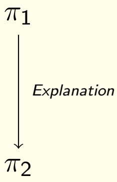


π_1

Explanation

π_2

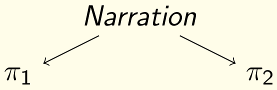
INSTEAD of



π_1

Explanation

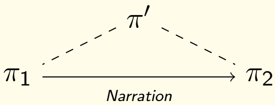
π_2



Narration

π_1 π_2

INSTEAD of



π'

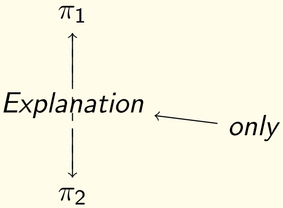
π_1 π_2

Narration

Advantages of our slight modifications to SDRT graphs

- Modification of a discourse connective/relation

(2) Fred is upset only because his wife is abroad for a week.

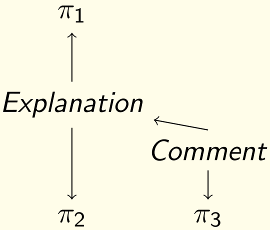


See slide 44

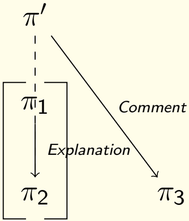
Advantages of our slight modifications to SDRT graphs (bis)

- Avoiding scope nodes and brackets (which are not needed in dependency formalisms)

(3) [Fred is upset because his wife is abroad for a week]. This proves that he does love her. (Danlos 2008)



INSTEAD of



Disadvantages of our slight modifications to SDRT graphs

- Discourse relations not lexicalized by a discourse connective
- We posit the existence of an empty (adverbial) connective noted ϵ (Harris 1966)
- As an illustration, we lay down that the discourse

John fell. Max pushed him.

is of the form

$S_1. \epsilon S_2.$

and by misuse of language, we say that the empty connective "conveyed" Explanation

TAG data structures and operations

- Set of elementary tree structures: initial or auxiliary trees
- Two operations to combine these structures: substitution and adjunction
- Use of the diacritic $\downarrow$ on a frontier node indicates that it is a *substitution node*
- Auxiliary trees are elementary trees in which the root and a frontier node, called the *foot node* and distinguished by the diacritic $*$, are labeled with the same nonterminal

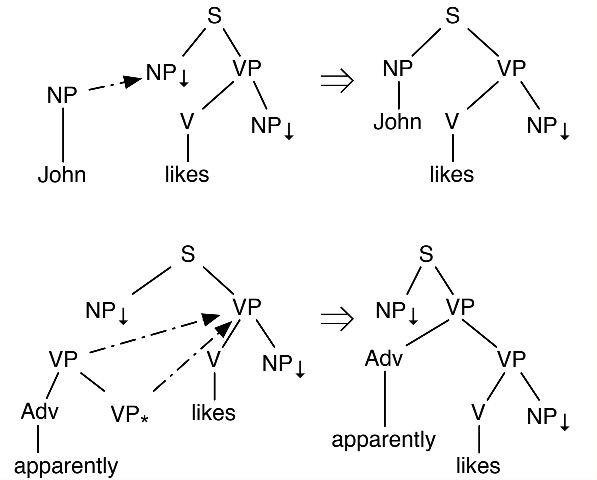
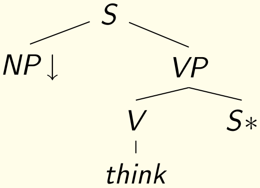


Figure: Example TAG substitution and adjunction operations (From Nesson and Shieber, 2006)

Two kinds of auxiliary trees

- Modifier auxiliary trees, e.g auxiliary tree anchored by *apparently*
- Predicative auxiliary tree, e.g. auxiliary tree anchored by *think*



Long distance dependencies

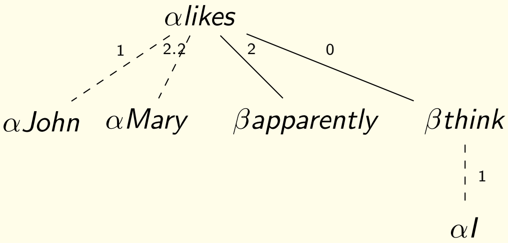
(4)a. The woman John apparently likes

b. The woman *I think* John apparently likes

Derivation trees

Which operations have been used to obtain the syntactic analysis (called syntactic derived tree)

(5) I think John apparently likes Mary.



α*likes*

1 2.2 2 0

α*John* α*Mary* β*apparently* β*think*

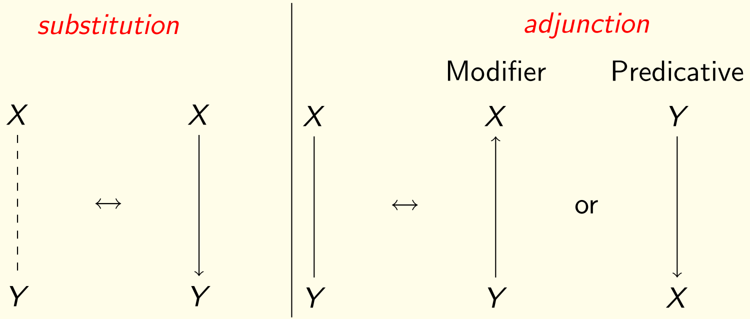
1

α*I*

Substitutions are notated with a dashed line, adjunctions with a solid line.

Derivation trees as semantic dependency graphs

Conversion rules (Candito, Kahane, 1998)



substitution

adjunction

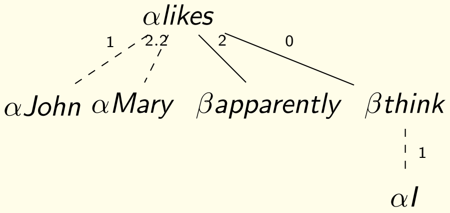
Modifier

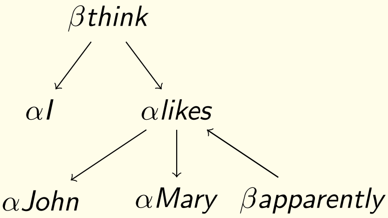
Predicative

or

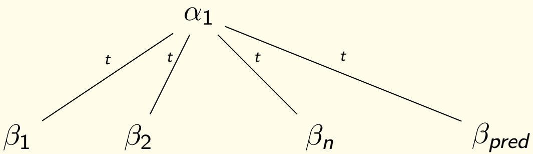
- TAG derivation trees as semantic dependency graphs
- SDRT graphs as semantic dependency graphs

Conversion of the derivation tree for (5)





Multiple adjunctions to the same node at the same address t



- Modifier trees β_i (independent modifiers) are ordered *(red roasted pepper, roasted red pepper)*
- At most one predicative tree β_{pred}
- Outermost predication constraint: the adjunction of the predicative tree must come after the adjunction of the modifiers trees
- Equivalent to Right Frontier Constraint for attachment (slide 37)

STAG data structures (Shieber 1994)

- Synchronous TAG (STAG) extends TAG by taking the elementary structures to be pairs of TAG trees with links between particular nodes in those trees.
- An STAG is a set of triples, $\langle t_L, t_R, \frown \rangle$ where t_L and t_R are elementary TAG trees and $\frown$ is a linking relation between nodes in t_L and nodes in t_R
- Derivation proceeds as in TAG except that all operations must be paired. That is, a tree can only be substituted or adjoined at a node if its pair is simultaneously substituted or adjoined at a linked node.
- Links are notated by using circled indices (e.g. ①) marking linked nodes.

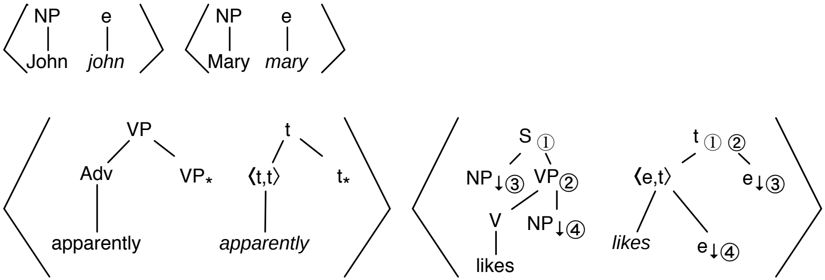


Figure: An English syntax/semantics stag fragment for *John apparently likes Mary.* (From Nesson and Shieber 2006)

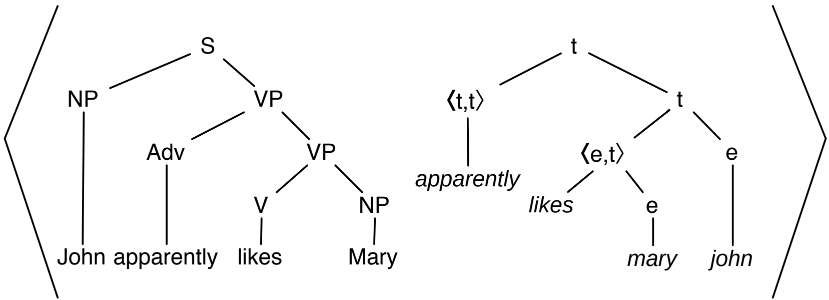


Figure: Derived tree pair for *John apparently likes Mary.*

Resulting semantic representation can be read off the semantic derived tree by treating the leftmost child of a node as a functor and its siblings as its arguments: *apparently*(*likes*(*john*, *mary*))

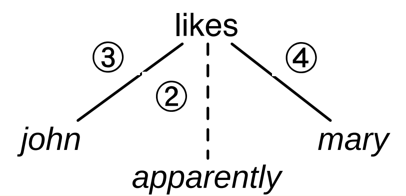


Figure: Derivation tree for *John apparently likes Mary.* (From Nesson and Shieber 2006)

- Each link in the derivation tree specifies a link number in the elementary tree pair.
- Only one derivation tree for both the syntactic and semantic representations.

Conventions

In the rest of the talk, we assume that the syntax and semantic analyses of clauses are generated by an STAG grammar, and we use the following symbols:

- T_i represents the syntactic analysis of clause S_i (a tree rooted S),
- F_i its semantic analysis (a tree rooted t),
- τ_i its derivation tree.

Arguments of discourse connectives/relations

Discourse connectives/relations always have two arguments

Adverbials (Adv)

- First argument: DU (Discourse Unit) on the left of the host sentence of the adverbial and belonging to the right frontier
- Second argument : DU identical to or starting at the host sentence of the adverbial
- S_1 $Conj_a$ S_2. Adv_b S_3. ...
    - First argument of Adv_b: S_1 or S_2, or S_1 $Conj_a$ S_2 (if $Conj_a$ conveys a subordinating relation)
    - Second argument of Adv_b: S_3 or $[S_3 \ldots]$

Arguments of discourse connectives/relations

Subordinating conjunctions (Conj): postposed position

- First argument: DU on the left of the host sentence *without crossing a sentence boundary* (a period) and belonging to the right frontier
- Second argument : DU identical to or starting at the host sentence
- S_1. Adv_a S_2 $Conj_b$ S_3 $Conj_c$ S_4. ...
    - First argument of $Conj_c$: S_2 or S_3, or S_2 $Conj_b$ S_3 (if $Conj_b$ conveys a subordinating relation); not S_1 (Adv_a)
    - Second argument of $Conj_c$: S_4 or $[S_4 \ldots]$

Arguments of discourse connectives/relations

Subordinating conjunctions (Conj): preposed position

- First argument: DU identical to or starting at the matrix clause
- Second argument : DU identical to the subordinate clause
- $Conj_a\ S_1,\ S_2 \ldots.$
  - First argument of $Conj_a$: S_2 or $[S_2 \ldots]$
  - Second argument of $Conj_a$: S_1

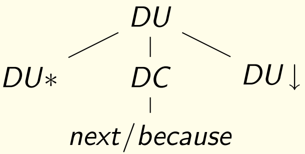
Coordinating conjunctions: left aside

D-STAG syntactic elementary trees anchored by a discourse connective (discourse level)

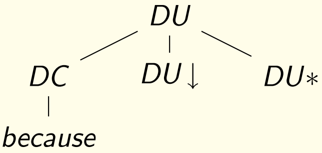
DCs always anchor auxiliary trees with *two arguments*:

- a foot node $DU*$ for the first argument (on the left of the DC)
- a substitution node $DU\downarrow$ for the host sentence

Adverbials
and
Postposed conjunctions

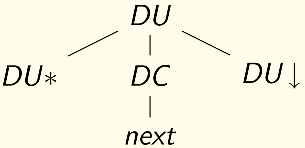


Preposed conjunctions



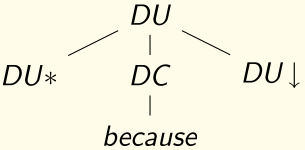
Right Frontier Constraint (RFC) for attachment

- If a DC lexicalizes a *subordinating* relation, it anchors a *modifier* auxiliary tree
- If a DC lexicalizes a *coordinating* relation, it anchors a *predicative* auxiliary tree



$DR = Narration$

$Predicative = +$



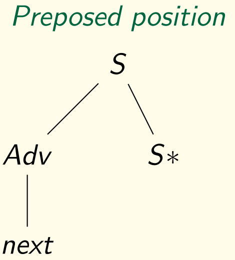
$DR = Explanation$

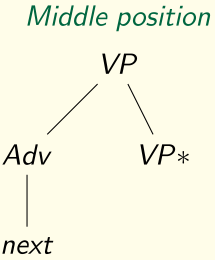
$Predicative = -$

Outermost predication constraint simulates RFC (slide 37)

TAG syntactic elementary trees anchored by an adverbial discourse connective (sentence level)

Adverbials: auxiliary trees with only one argument





A Sentence/Discourse interface is compulsorily needed

D-STAG Parsing

- Sentence level: standard S-TAG parsing of each sentence
- Sentence/Discourse interface (inspired from D-LTAG):
  - Extract the clausal derivations and the discourse connectives
  - Introduce the empty connective ϵ if necessary
  - Abstract away from any middle position of the adverbials
- Input of the discourse parser: string of words which are either clauses or discourse connectives (in a preposed position)

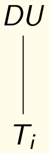
$$S_1.\ Adv_a\ S_2\ Conj_b\ S_3.\ \epsilon\ S_4.$$

For each S_i, STAG has produced its syntactic analysis T_i, its semantic analysis F_i and its derivation tree τ_i

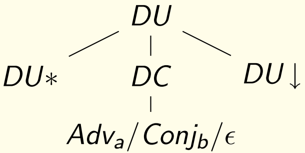
Discourse parser

- Input: string of words (S_1. Adv_a S_2 $Conj_b$ S_3. ϵ S_4.)
- One (several) elementary trees anchored by each word:

Initial trees
for each clause S_i



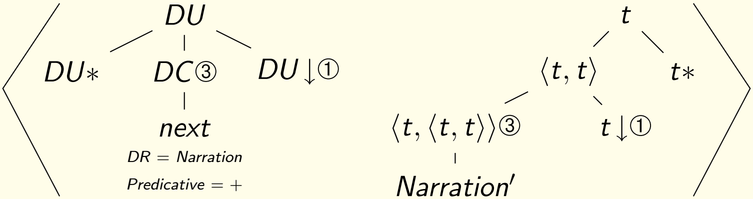
Auxiliary trees
for each DC



- Standard (simple) TAG parsing, with one special rule:
  - subordinating conjunction constraint: in
    S_1. Adv_a S_2 $Conj_b$ S_3. ϵ S_4.
    the first argument of $Conj_b$ is S_2 (cannot be S_1)

D-STAG at the semantic level

- A pair of D-STAG elementary trees consists of a tree anchored by a DC linked to a tree anchored by a functor associated to the discourse relation conveyed by the DC



DU

$DU*$ $DC③$ $DU\downarrow①$

$next$

$DR = Narration$

$Predicative = +$

t

$\langle t, t \rangle$ $t*$

$\langle t, \langle t, t \rangle\rangle ③$ $t\downarrow①$

$Narration'$

- $Narration'$ functor: $\lambda pq.\phi_{Narration(^\wedge p, ^\wedge q)}$ with $p, q : t$
- $\phi_{Narration(^\wedge p, ^\wedge q)}$ are “the special semantic constraints pertinent to the particular discourse relation $Narration(^\wedge p, ^\wedge q)$” (Asher and Lascarides 2003)

Illustration of the D-STAG parser on discourses in (6)

(6)a. Fred is in a bad mood because [he didn't sleep well. He is also worried about his exams.]

b. [Fred went to the supermarket]because his fridge is empty. Next, he went to the movies.

c. [Fred is upset because his wife is abroad for a week]. This proves that he does love her.

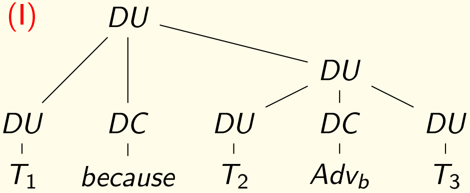
- For every discourse in (6), input to the discourse parser:

$$S_1 \textit{ because } S_2.\ Adv_b\ S_3.$$

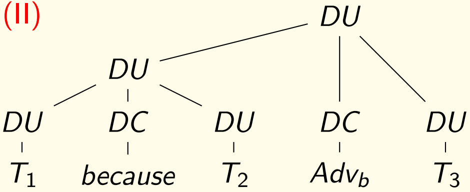
- with $Adv_b = also$ in (6a), *Parallel* (coordinating)
- with $Adv_b = next$ in (6b), *Narration* (coordinating)
- with $Adv_b = \epsilon$ in (6c), ?

Syntactic discourse derived trees for discourses in (6)

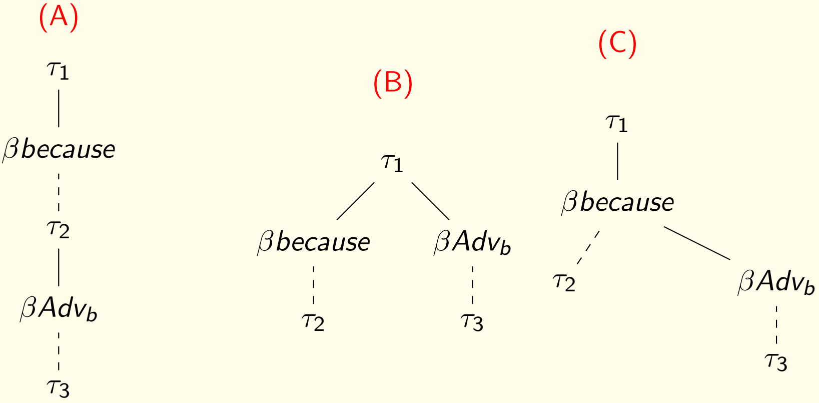
(I)



(II)



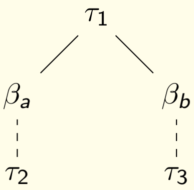
Derivations trees for discourses in (6)



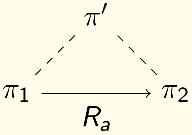
- (A) corresponds to syntactic tree (I)
- (B) and (C) correspond to syntactic tree (II)

Outermost predication constraint and RFC for attachment

Outermost predication constraint: this derivation tree is excluded if B_a/R_a is coordinating



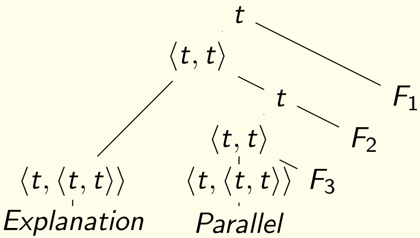
R_a coordinating



τ_1 not on the right frontier:
attachment of
β_b/R_b to π_1
is excluded

Disambiguation process for discourses in (6)

- SDRT machinery based on (extra)linguistic considerations
- and/or probabilistic data from annotated corpora
- (6a): derivation tree (A)



$Explanation(\pi_1, Parallel(\pi_2, \pi_3))$

- (6b): derivation tree (B) $Explanation(\pi_1, \pi_2) \wedge Narration(\pi_1, \pi_3)$
- (6c): derivation tree (C) $Comment(Explanation(\pi_1, \pi_2), \pi_3)$

Problematic case

(7) John broke his leg (because) he fell down in the stairs (because) he was drunk.

- S_2 explains $S1$ and is explained by S_3
- $Explanation(\pi_1, \pi_2) \wedge Explanation(\pi_2, \pi_3)$
- Derivation trees (A), (B) and (C) don't straightforwardly lead to this semantic analysis
- Solution: give a special semantic tree to β_b when used in a vertical derivation tree such as (A) (Danlos 1987)

Comparison between D-STAG and D-LTAG

- D-STAG is like D-LTAG in that the two formalisms extend a sentential TAG syntax/semantic interface to the discourse level.
- The crucial difference between D-STAG and D-LTAG is that D-LTAG ignores discourse relations and their coordinating or subordinating type.
- As a consequence, D-LTAG has nothing in common with SDRT (neither with RST) and doesn't use rhetoric or pragmatic knowledge.

Another difference between between D-STAG and D-LTAG

- in D-STAG, discourse connectives always anchor elementary trees with two arguments
- in D-LTAG, they can anchor elementary trees with just one argument which is *structurally* retrieved, the other one being provided *anaphorically* (Webber et al.)

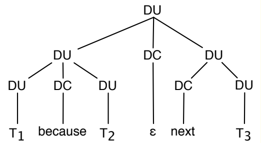


Figure: D-LTAG syntactic tree for discourse (1)

(1) John went to the supermarket because his fridge was empty. Next, he went to the movies.

Arguments structurally retrieved

- In D-LTAG, the arguments of a subordinating conjunction are said to be structurally retrieved (matrix and subordinate clauses).
- Examples in (8) are counterexamples to this claim.

(8)a. Fred is in a bad mood because [he didn't sleep well. He is also worried about his exams]. (6a)

b. While he was in Paris, [John visited Le Louvre. Next, he went to the Eiffel Tower . . .]

c. Fred is in a bad mood because [Mary played tuba when he was taking a nap].

- I don't understand

Modification of discourse relations

Moreover, D-STAG can easily benefit of the adjunction operation for modification of discourse relations.

(9)a. You should not trust John because, for example, he never returns what he borrows.

b. You should not trust John only because he never returns what he borrows.

Semantic representation of (9a) (Forbes et al.):

$$\mathit{Exemplify}({}^\wedge F_2, \lambda p.\phi_{\mathit{Explanation}({}^\wedge F_1, {}^\wedge p)}) \text{ with } p : t$$

Modification of discourse relations (bis)

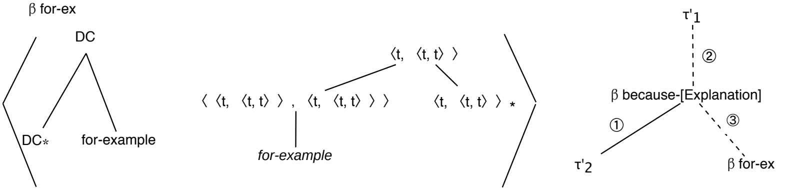


Figure: Derivation tree for discourse (9a)

$\lambda Rpq.Exemplify(\wedge p, \lambda r.R(^\wedge q, ^\wedge r))$ with $R : \langle t, \langle t, t\rangle\rangle$, and $p, q, r : t$

On the other hand, in D-LTAG, *for example* in (9a) is considered as a discourse connective, and the computations made to obtain the semantic representation of this discourse are heavy, see (Forbes et al. 2006)

Conclusion

D-STAG is designed to:

- use S-TAG for processing the syntactic and semantic sentence levels,
- extend this syntax/semantic interface to the discourse level while being based on a discourse theory, namely SDRT,
- take semantic dependency graphs as a pivot between TAG (derivation trees) and SDRT (discourse structure graphs).
- Implementation: within ALPAGE (TAG grammar for French)

Future research

Interleave the sentence and discourse levels not only to get an homogeneous process from a discourse to its interpretation (efficiency reasons) but also to handle pairs such as (10) (theoretical reasons).

(10)a. John held out a bone to the dog. She caught it quickly.
b. John held out a bone to the dog who caught it quickly.

Topic of a German-French project with Manfred Stede, Nicholas Asher and Laurence Danlos (submitted)