D-STAG: a Formalism for Discourse Analysis based on SDRT and using Synchronous TAG

D-STAG

Laurence Danlos

Université Pars 7, IUF, ALPAGE

25 août 2010

New formalism : D-STAG

For parsing discourse

Formalism which extends a sentential syntax-semantics interface to the discourse level.
It computes the "discourse structure" of the input discourse : discourse relations link together the meanings conveyed by discourse segments

Discourse theory

SDRT = Segmented Discourse Theory
(Asher 1993, Asher and Lascarides 2003)

Parsing formalism

STAG = Synchronous Tree Adjoining Grammar
(Shieber and Shabes 1990, Nesson and Shieber 2006)
syntax-semantics interface and machine translation

Architecture of D-STAG

Sentential analyzer

which provides the syntactic and semantic analyses of each sentence in the discourse given as input,

Sentence-discourse interface

which is a mandatory component if one wants not to make any change to the sentential analyzer,

Discursive analyzer

which computes the discourse structure.

Discursive Linguistic Data

Discourse connective

A discourse relation is expressed by a "discourse connective" :
- subordinating and coordinating conjunctions (*because*, *or*) and discourse adverbials (*next*, *therefore*)
- empty connective ϵ (*Fred fell.* ϵ *Max tripped him up.* of the form C_1. ϵ C_2. the discourse relation must be inferred)

Discourse relation

semantic predicate
lexicalized by a discourse connective (possibly empty)
two arguments : discursive semantic representations of (continuous) discourse segments.

Discourse structures as dependency graphs

RST (Mann and Thomson 1988) and D-LTAG (Webber et al. 2003)

dependency graphs are tree-shaped

Myth

Wolf and Gibson (2006)
Danlos (2004, 2007)
SDRT home-made graphs converted as dependency graphs

DAGs

not necessarily tree-shaped
respect strong constraints

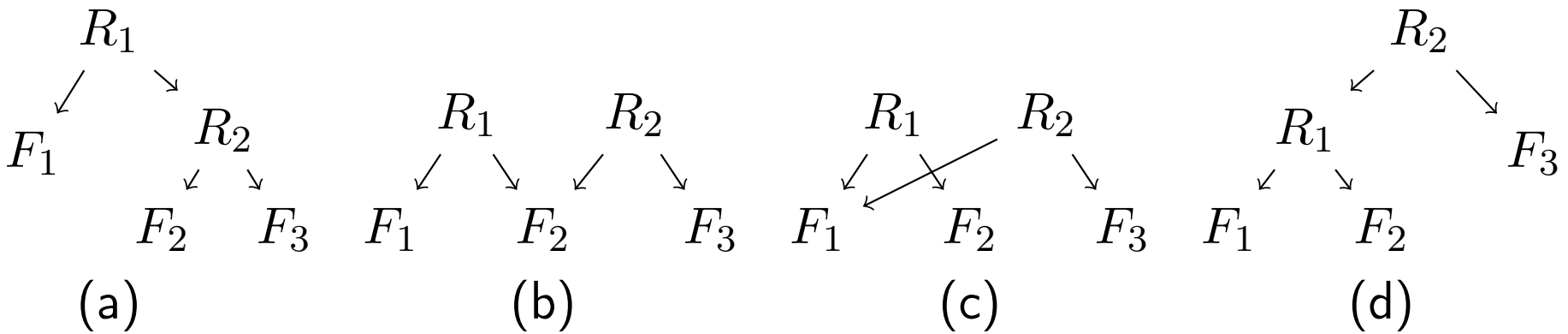
Discourses of the form C_1 *because* C_2. Adv_2 C_3.

(1)a. Fred is in a bad mood because he lost his keys. Moreover, he failed his exam.

b. Fred is in a bad mood because he didn't sleep well. He had nightmares.

c. Fred went to the supermarket because his fridge is empty. Then, he went to the movies.

d. Fred is upset because his wife is abroad for a week. This shows that he does love her.



D-STAG can produce non tree-shaped dependency graphs

Introduction to TAG (Joshi 1985)

- 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

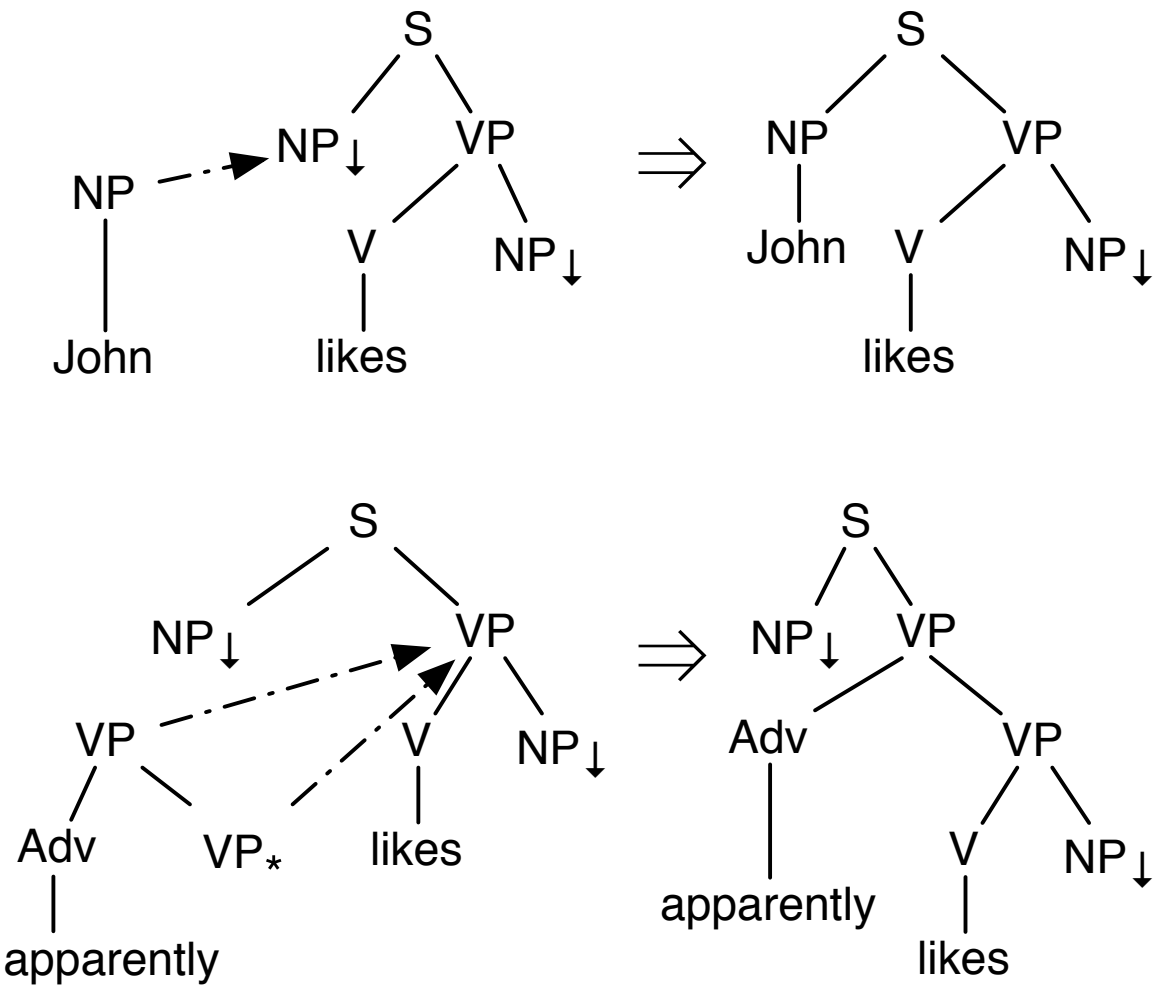


FIGURE: Example TAG substitution and adjunction operations (From Nesson and Shieber, 2006)

Introduction to STAG (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.

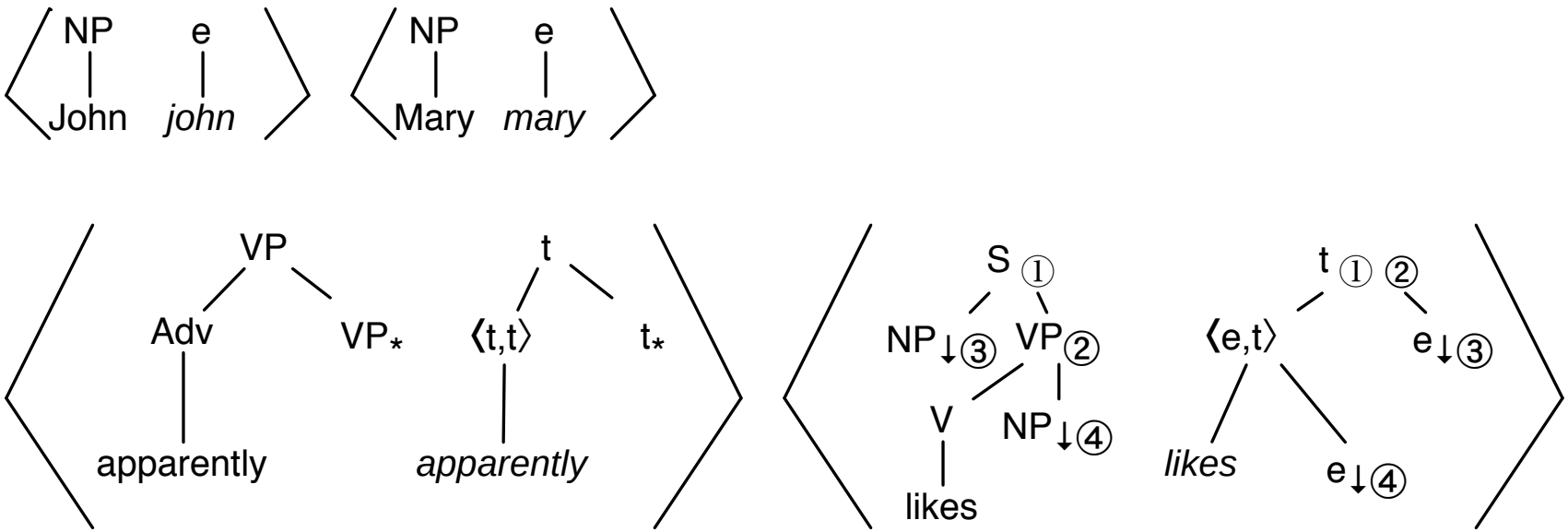


FIGURE: An English syntax/semantics STAG fragment for *John apparently likes Mary.* (From Nesson and Shieber 2006)

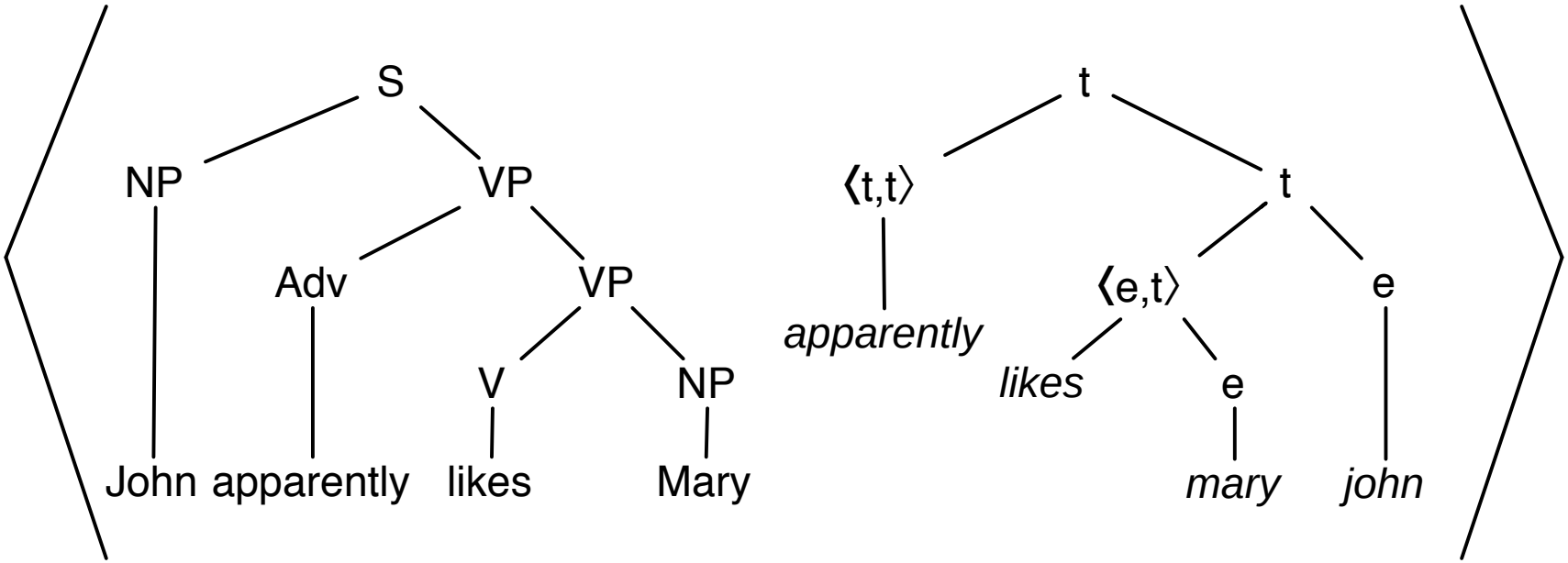


FIGURE: Derived tree pair for *John apparently likes Mary.*

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 : $apparently(likes(john, mary))$

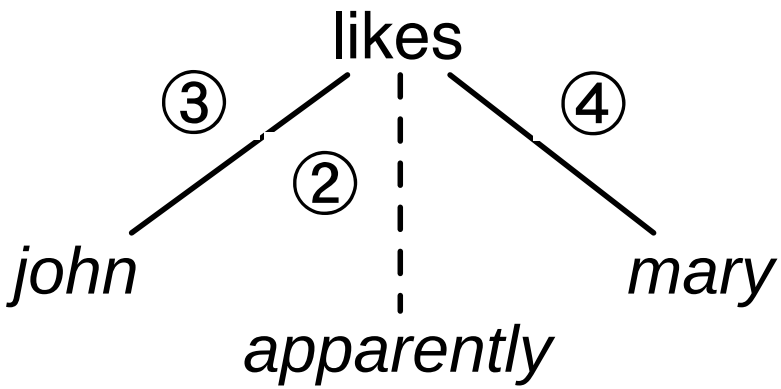


FIGURE: Derivation tree for *John apparently likes Mary.* (From Nesson and Shieber 2006)

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

Sentence-Discourse Interface

Why this interface is necessary ?

The idea in D-STAG : extend a sentential analyzer to the discourse level **without making any change to it**.

Mismatches between the arguments of a connective at the discourse level and its arguments at the sentence level

- an adverbial connective has compulsorily **two** arguments at the discourse level, whereas it has only **one** argument at the sentence level
- a subordinating conjunction can have an argument at the discourse level which **crosses a sentence boundary** whereas this is out of the question at the sentence level

The sentence-discourse interface gives sentence boundaries the simple role of punctuation signs and allows us to re-compute the (two) arguments of a connective.

Discourse Normalized Forms (DNFs)

sequence of "discourse words"

a connective, an identifier C_i for a clause (without any connective), a punctuation sign, ...

Fred went to the movies. As he was in a bad mood, he didn't enjoy it. He then went to a bar because he was dead thirsty.

$C_1.\ \epsilon\ as\ C_2,\ C_3.\ then^{vp}\ C_4\ because\ C_5.$

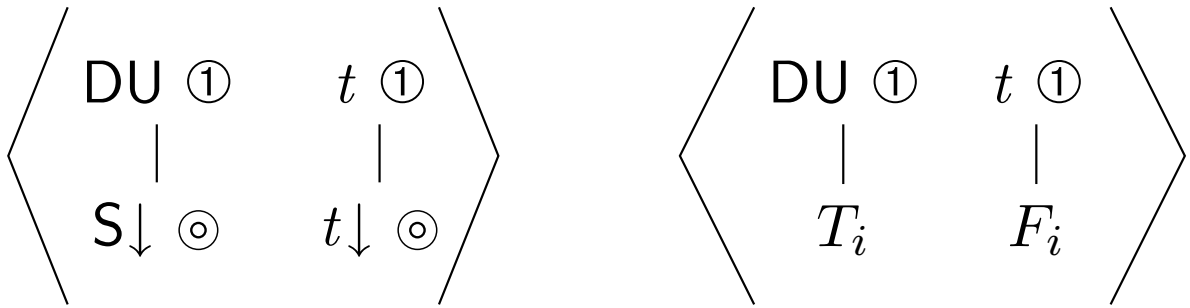
Regular grammar

A DNF without any preposed conjunction follows the pattern
$C_1\ Conn_2\ C_2 \ldots Conn_{n-1}\ C_n$
disregarding punctuation signs.

Plug sentential analyses into discursive ones

For a clause C_i

its syntactic tree rooted in S and noted T_i,
its semantic tree rooted in t and noted F_i,
and its derivation tree noted τ_i.



(a) τ_i'

FIGURE: Tree pairs α*S-to-D* and τ_i'

DU : Discourse unit

Discursive Component of D-STAG

STAG grammar for connectives

Basic principle

When a given connective $Conn_i$ lexicalizes a single discourse relation R_i, tree pair $Conn_i \div R_i$ whose syntactic tree is anchored by $Conn_i$ and whose semantic tree is anchored by (a lambda-term associated with) R_i.

When a connective is ambiguous, it anchors as many syntactic trees as discourse relations it lexicalizes.

However, ambiguity issues are not in the scope of this paper.

Adverbial Connectives and Postposed Conjunctions

Syntactic trees

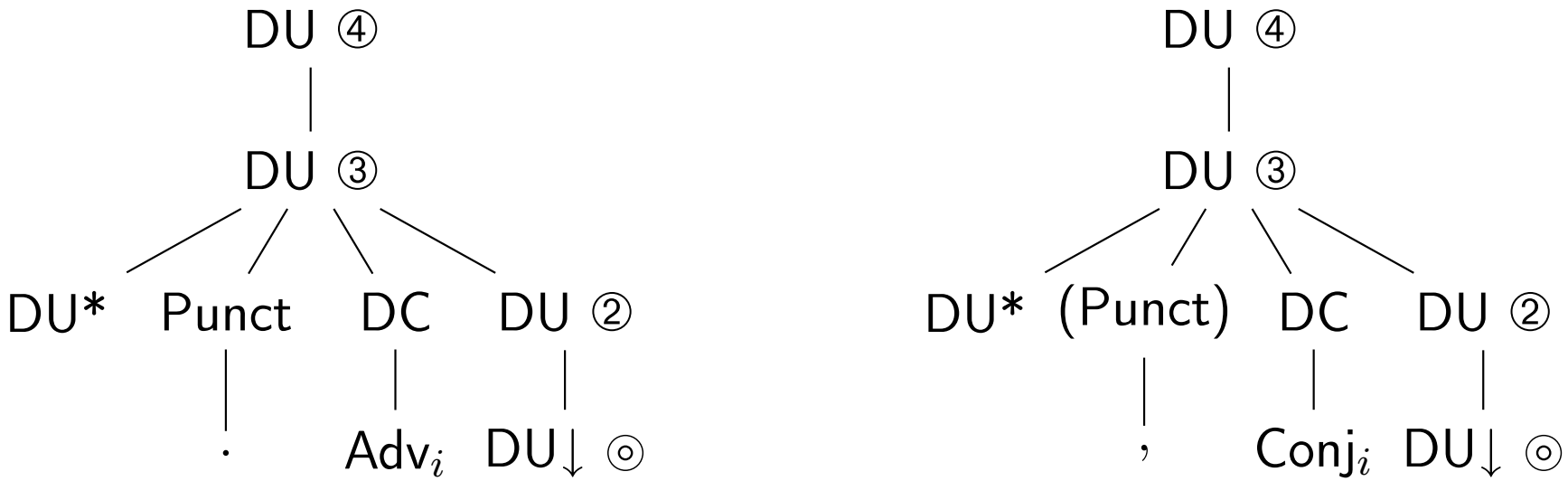


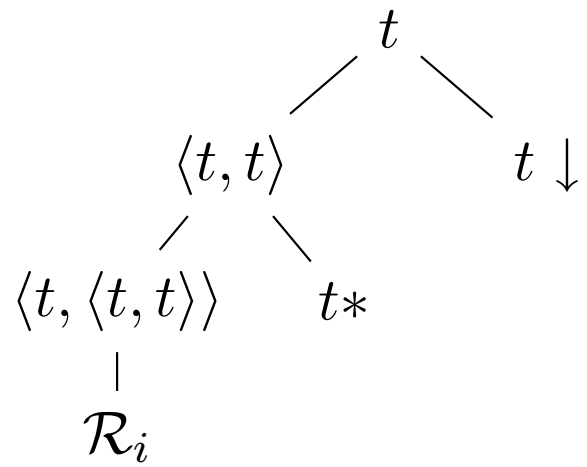
FIGURE: Syntactic trees for adverbial connectives and postposed conjunctions

Auxiliary trees with a foot node $DU*$ and a substitution node $DU\downarrow$

Semantic trees

First idea

Simple functor $\mathcal{R}_i$ (associated to the discourse relation R_i) with $t*$ and $t\downarrow$



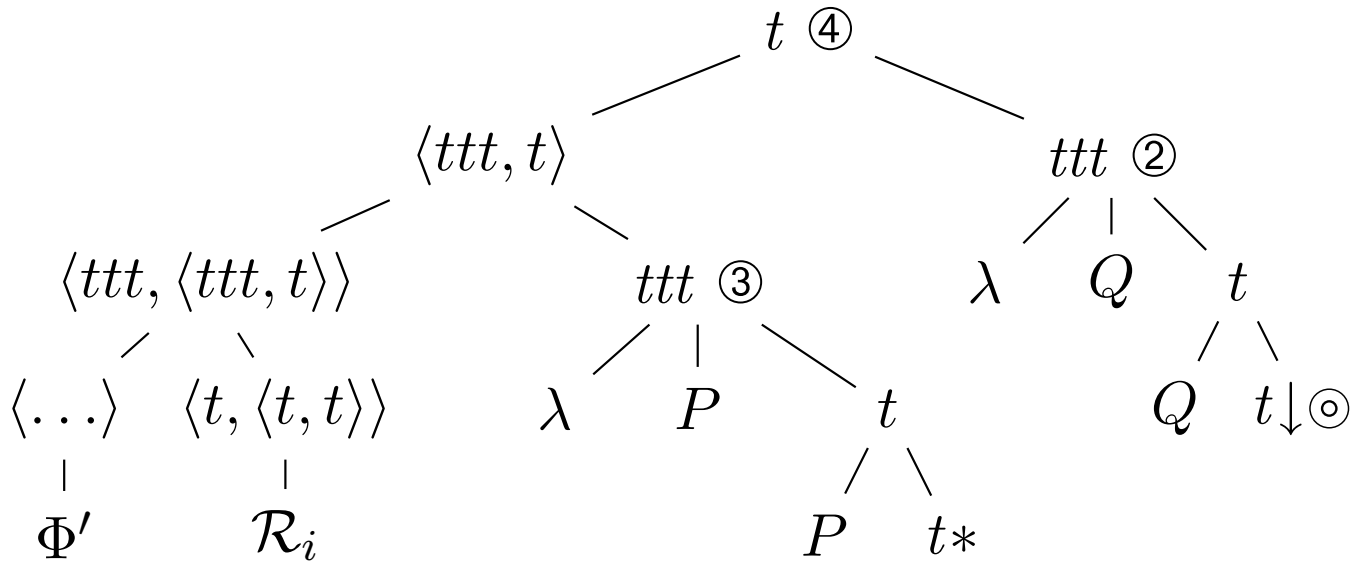
Too simple

Only appropriate for DNFs with two clauses

No conjunction of formulae (needed for DNFs with three (or more) clauses)

Functor $\Phi'(\mathcal{R}_i) = \mathcal{R}'_i$

$\Phi' = \lambda\mathcal{R}_i XY.X(\lambda x.Y(\lambda y.\mathcal{R}_i(x,y)))$
$\Phi'(\mathcal{R}_i) = \mathcal{R}'_i = \lambda XY.X(\lambda x.Y(\lambda y.\mathcal{R}_i(x,y)))$
with $X, Y : ttt = \langle\langle t, t\rangle, t\rangle$ and $x, y : t$

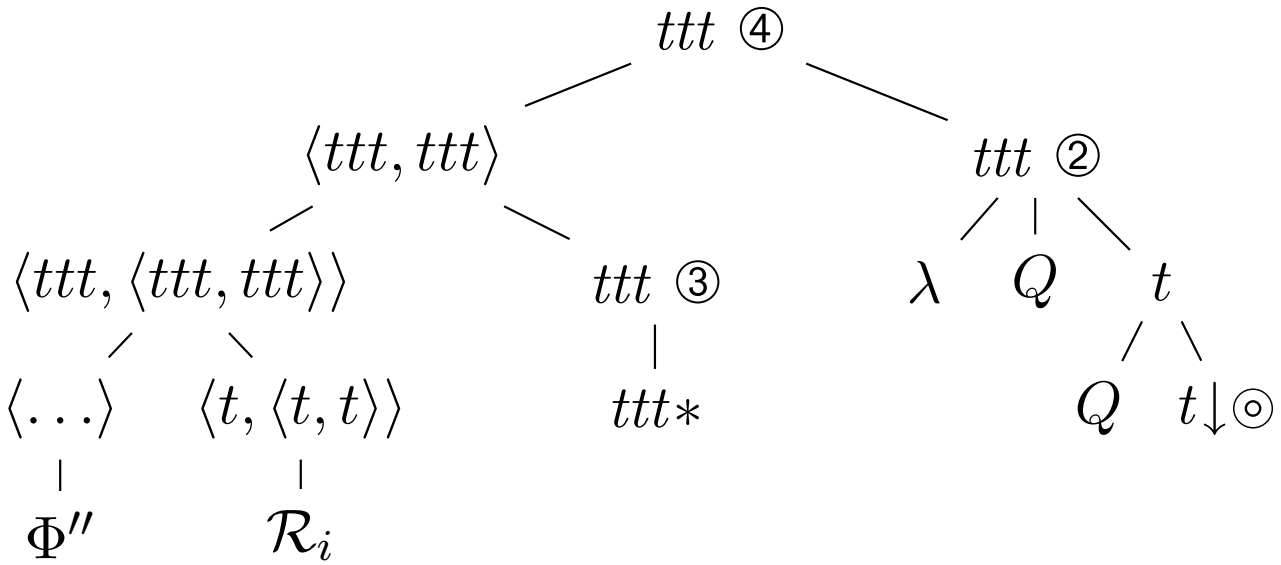


Result

$R_i(F_1, F_2)$

Functor $\Phi''(\mathcal{R}_i) = \mathcal{R}''_i$

$\Phi'' = \lambda \mathcal{R}_i XY P.X(\lambda x.Y(\lambda y.\mathcal{R}_i(x,y) \wedge P(x)))$
$\Phi''(\mathcal{R}_i) = \mathcal{R}''_i = \lambda XY P.X(\lambda x.Y(\lambda y.\mathcal{R}_i(x,y) \wedge P(x)))$
with $X, Y : ttt = \langle\langle t,t\rangle, t\rangle$, $P : \langle t,t\rangle$ and $x, y : t$



Result

$\lambda P.(R_i(F_1, F_2) \wedge P(F1))$

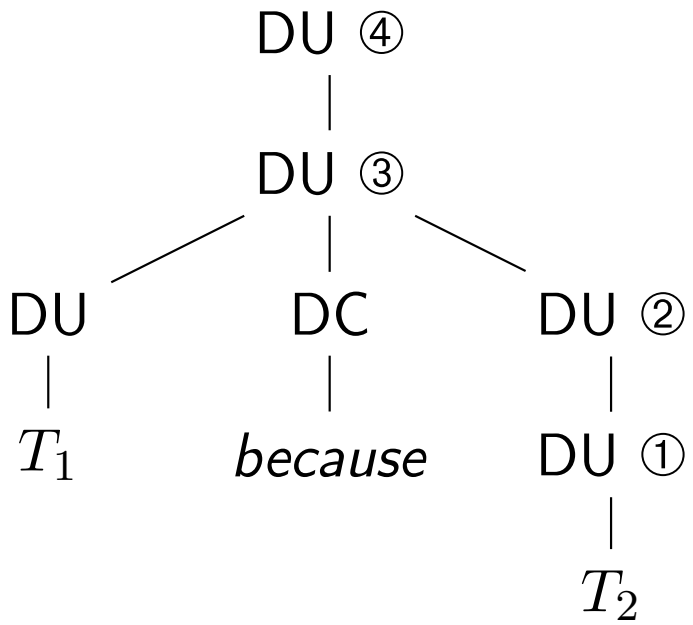
Analysis of DNFs with three clauses

Four types of interpretations

illustrated on examples (1) of the form C_1 *because* C_2. Adv_2 C_3.

β_1 : the tree pair $because_{post} \div Explanation$

β_2 : the tree pair $Adv_2 \div R_2$.



Right frontier

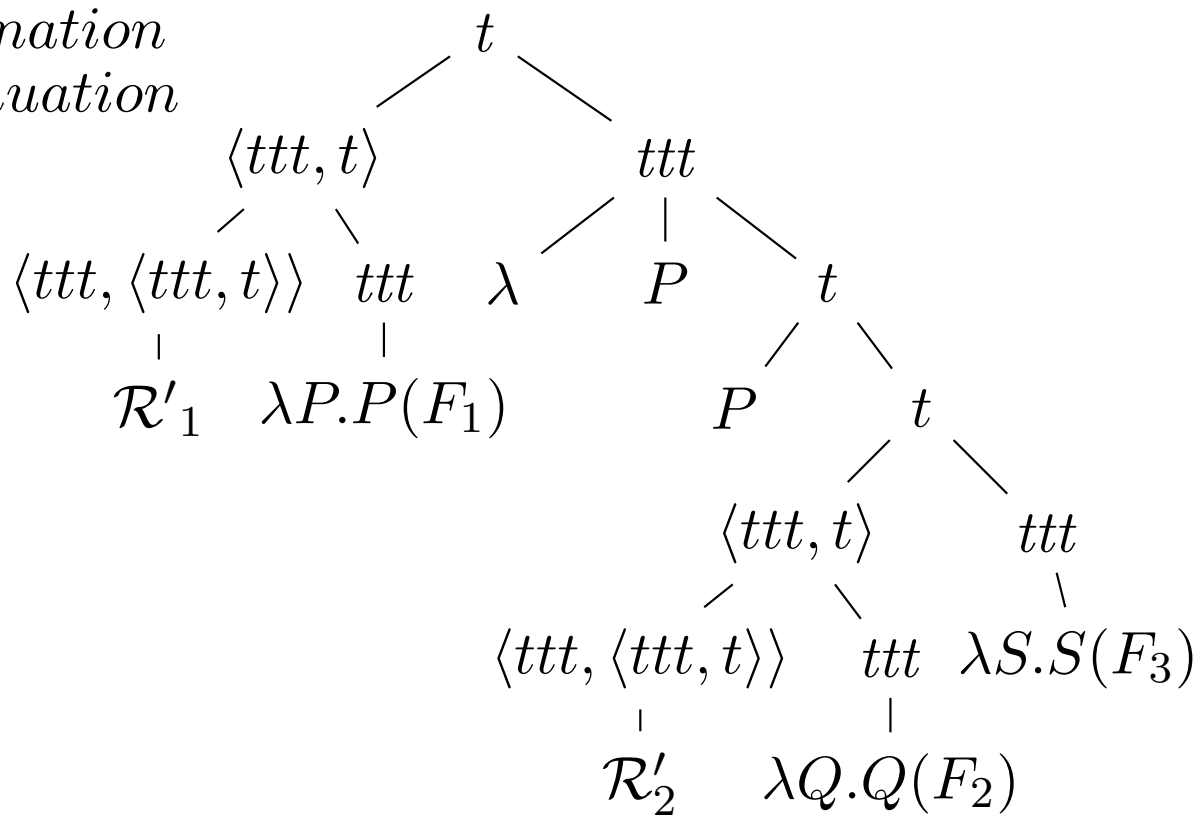
Four nodes DU with different links

Analysis of (1a)

Fred is in a bad mood because he lost his keys. Moreover, he failed his exam.

$\beta_2 = moreover \div Continuation$
interpretation $R_1(F_1, R_2(F_2, F_3))$
adjunction of β_2 at link ① of τ'_2

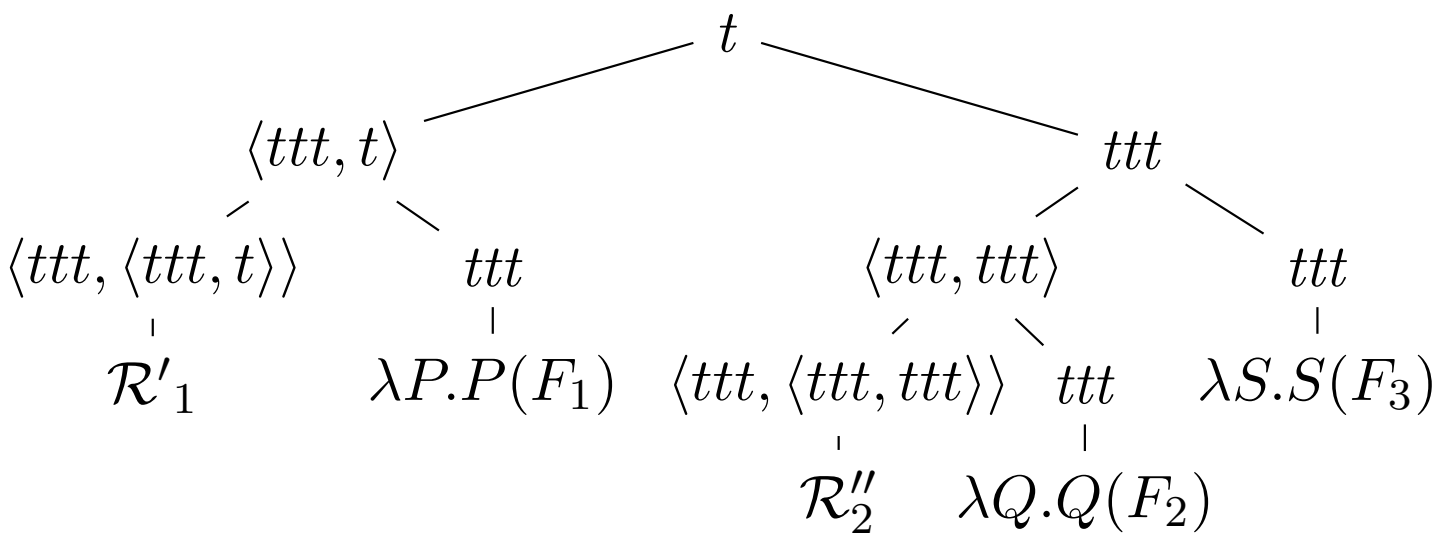
$R_1 = Explanation$
$R_2 = Continuation$



Analysis of (1b)

Fred is in a bad mood because he didn't sleep well. He had nightmares.

$\beta_2 = \epsilon \div Explanation$
interpretation $Explanation(F_1, F_2) \wedge Explanation(F_2, F_3)$
adjunction of β_2 at link ② of β_1



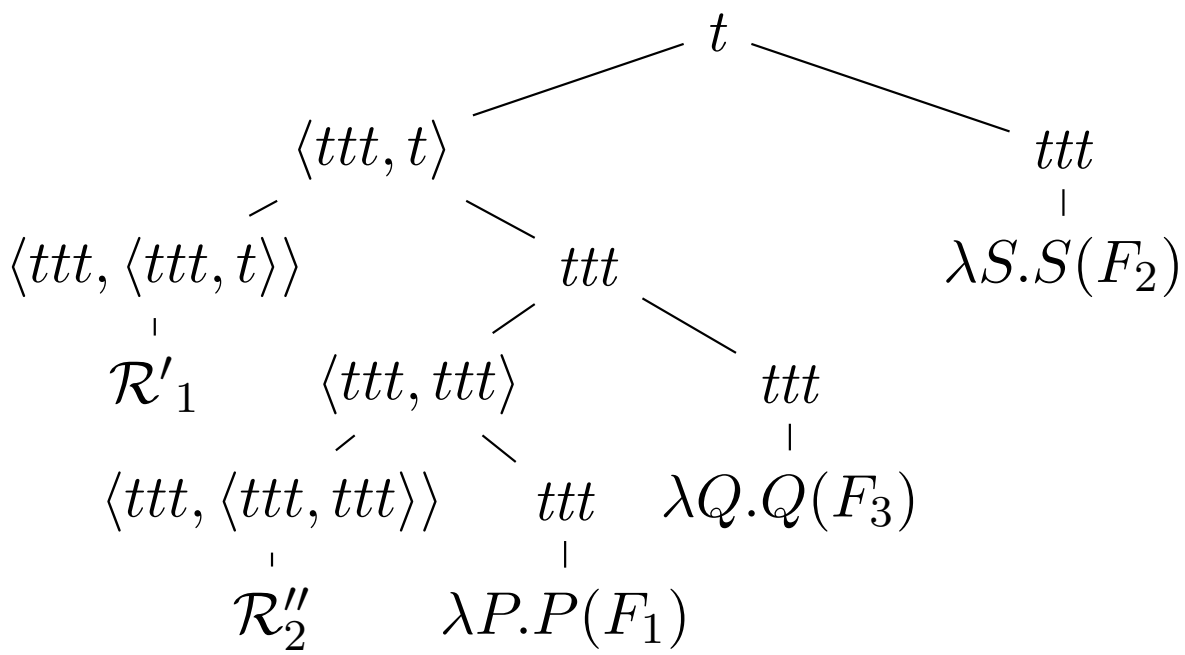
Analysis of (1c)

Fred went to the supermarket because his fridge is empty. Then, he went to the movies.

$\beta_2 = then \div Narration$

$Explanation(F_1, F_2) \wedge Narration(F_1, F_3)$

adjunction of β_2 at link ③ of β_1

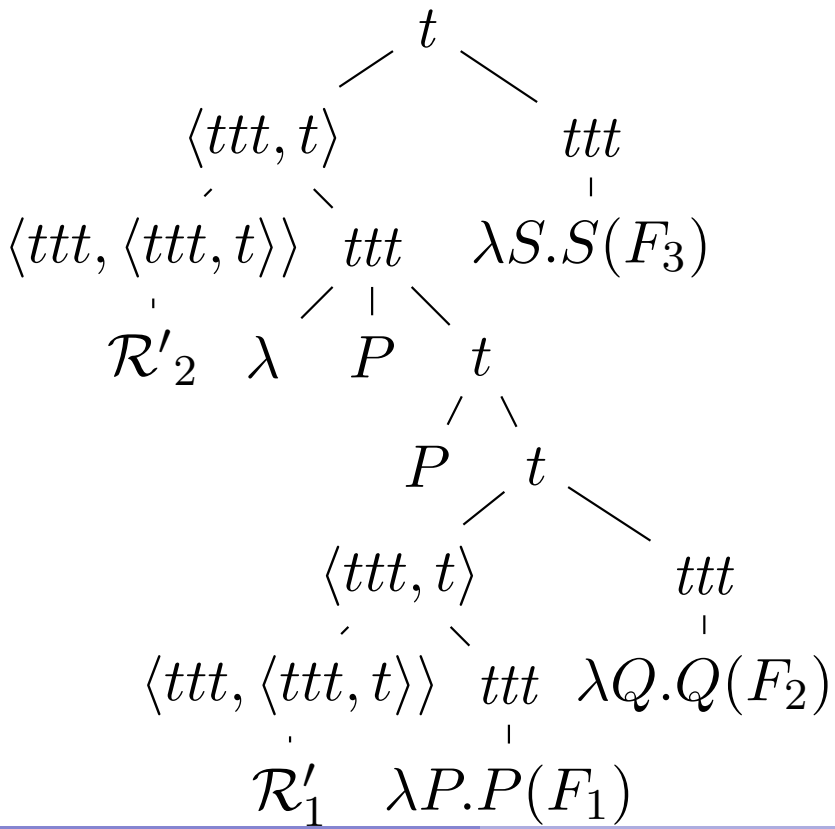


Analysis of (1d)

Fred is upset because his wife is abroad for a week. This shows that he does love her.

$\beta_2 = \epsilon \div Commentary$
interpretation $Commentary(Explanation(F_1, F_2), F_3)$
adjunction of β_2 at link ④ of β_1

$R_1 = Explanation$
$R_2 = Commentary$



Implementation of the Right Frontier Constraint (RFC)

SDRT

Distinction between two types of discourse relations : coordinating ($Narration$, $Continuation$) versus subordinating ($Explanation$, $Commentary$) ones

RFC : it is forbidden to attach new information to the first argument of a coordinating relation

D-STAG

Distinction between coordinating and subordinating relations $\Rightarrow$ two copies of the semantic trees which differ in a top feature $[coord = \pm]$ decorating their foot node.

RFC $\Rightarrow$ any adjunction at link ③ of the copies whose foot node is decorated with the feature $[coord = +]$ is forbidden

The interpretation $R_1(F_1, F_2) \wedge R_2(F_1, F_3)$ is excluded when R_1 is coordinating

Preposed Conjunctions

"Framing adverbial"

(2) When he was in Paris, Fred went to the Eiffel Tower. Next, he visited The Louvre.

Interpretation : $Circumstance(Narration(F_2, F_3), F_1)$

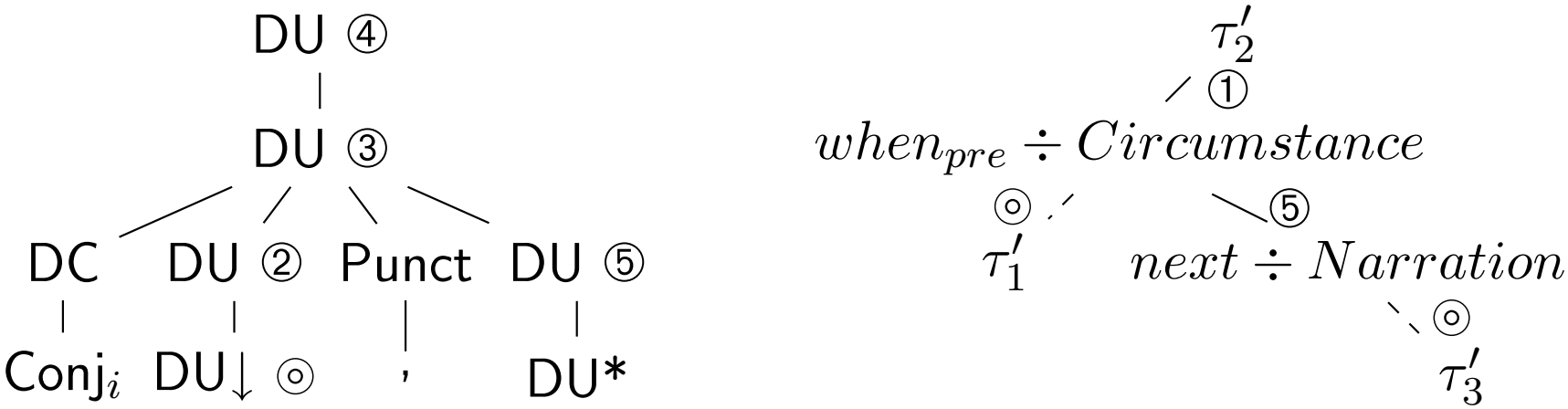


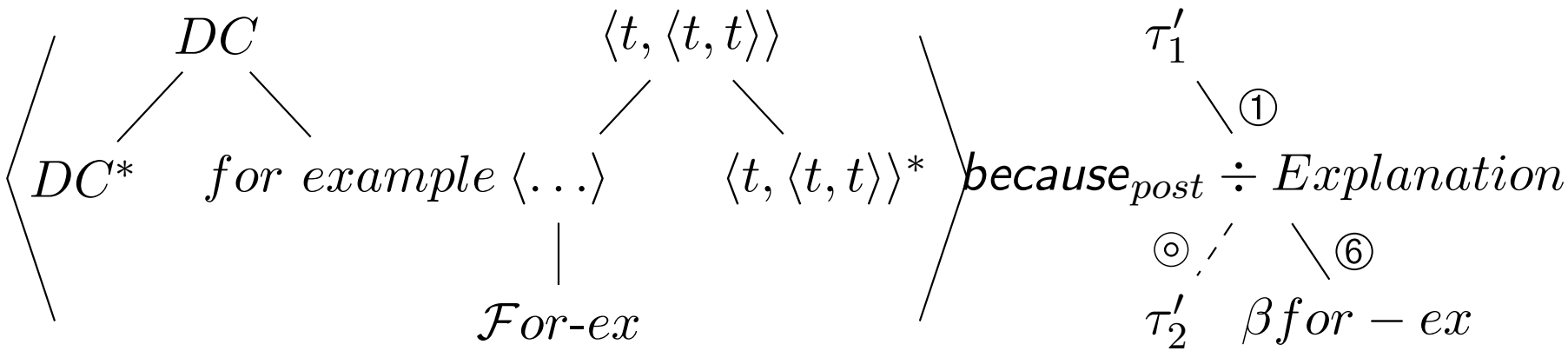
FIGURE: Tree anchored by a preposed conjunction and derivation tree for (2)

The syntactic discursive grammar of D-STAG is not a TIG (Tree Insertion Grammar)

Modifiers of Discourse Connectives/Relations

(3)a. Fred is in a bad mood *only/even/except* when it is sunny.
b. You shouldn't trust John because, *for example*, he never returns what he borrows. (Webber at al. 2003)

$\mathcal{F}$*or-ex* $= \lambda \mathcal{R}_i pq.Exemplification(q, \lambda r.\mathcal{R}_i(p, r))$
with $\mathcal{R}_i : \langle t, \langle t, t \rangle \rangle$, $p, q : t$.



Interpretation of (3b)

$Exemplification(F_2, \lambda r.Explanation(F_1, r))$ with $r : t$.

Correlative constructions with *neither ... nor*, *either ... or*

(4)a. Fred is pleased *neither* when it is sunny *nor* when it is rainy.
b. Fred will come *either* if it is sunny *or* if it is rainy.

D-STAG : *neither* and *nor* (adverbial) modifiers of the subordinating conjunctions on their right.
For (4a), interpretation $\neg Condition(F_1, F_2) \wedge \neg Condition(F_1, F_3)$

For (4b), interpretation $Condition(F_1, F_2) \vee Condition(F_1, F_3) = \neg(\neg Condition(F_1, F_2) \wedge \neg Condition(F_1, F_3))$ requires a multi-component semantic tree for *either* (only case)

Comparison between D-STAG and D-LTAG (Weber et al. 2003)

First, D-LTAG makes little use of discourse relations and ignores the distinction between coordinating versus subordinating relations. In short, it doesn't build on discourse theories, contrarily to D-STAG which build on SDRT.

Second, in D-LTAG, most adverbial connectives anchor trees with only **one** argument, whereas they all have **two** arguments in D-STAG. Differences in syntactic and semantic representations.

Third, D-LTAG provides only structures which correspond to tree shaped dependency graphs, contrarily to D-STAG.

Multiple connectives

(5) John ordered three cases of Barolo. But he had to cancel the order *because then* he discovered he was broke. (Webber et al. 2003)

In D-STAG, the DNF for (5) is automatically converted into
C_1 *but* C_2 *because* $\overline{C_3}$ *then* C_3

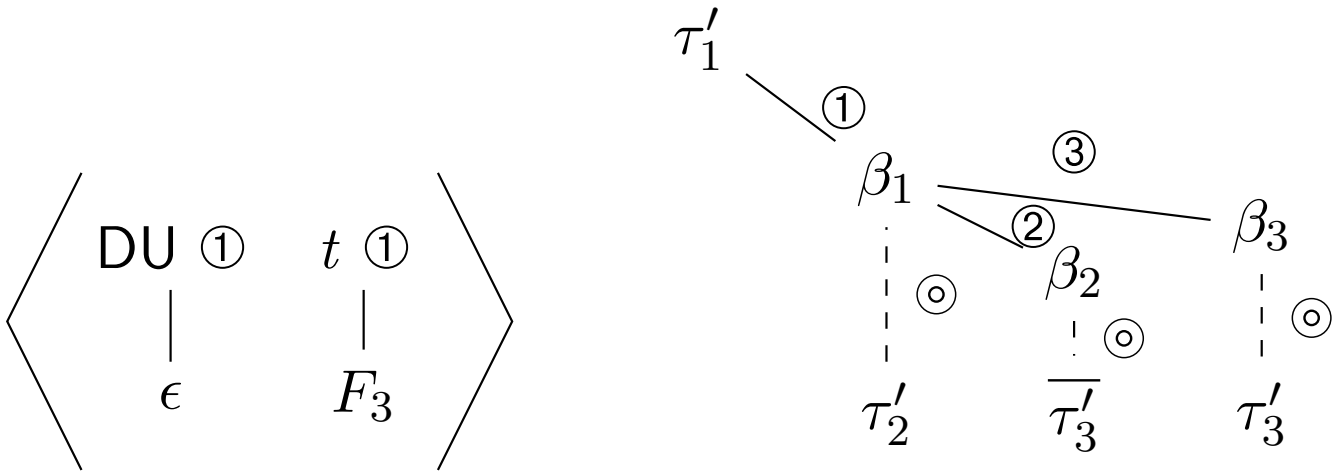


FIGURE: Tree pair $\overline{\tau'_3}$ and derivation tree for (5)

$Contrast(F_1, F_2) \wedge Explanation(F_2, F_3) \wedge After(F_1, F_3)$

Conclusion

D-STAG

Formalism which extends a sentential syntax-semantics interface to the discourse level

It computes the discourse structures of SDRT which are built with STAG

Sentence-discourse interface

It provides a DNF which is a sequence of discourse words which follows a regular grammar

So far : connectives, modifier of discourse connectives/relations, correlative constructions, multiple connectives

TODO

attribution and quotations

Implementation of D-STAG in a French discourse analyzer

Work in progress

LEXCONN : lexical database for French connectives (350 item) (Roze 2009)
PROTENOR : production of DNFs (Détrez 2009)
Discursive component : using ACG (P. de Groote and S. Pogodolla)

Output of D-STAG

The analyzer will produce a forest of dependency trees which represents the set of possible analyses.
The extraction of the best analysis (or the n best analyses) will require to build probabilistic disambiguation models based on the French annotated corpus Annodis.