Contextual Filtering of Rhetorical Relations for Discourse Structuring in Dialogue

Vladimir Popescu[*], Jean Caelen
Laboratoire d'Informatique de Grenoble, Grenoble Institute of Technology, France
{vladimir.popescu, jean.caelen}@imag.fr

Abstract

In this paper we propose an alternative to the Segmented Discourse Representation Theory (SDRT) mechanism for dealing with ambiguous rhetorical connections between utterances. Thus, starting from the observation that the SDRT mechanism for ordering discourse structures (the Maximize Discourse Coherence – MDC principle) relies on a fragile scalar interpretation of rhetorical relations in terms of their quality, we propose a reduction of the discourse structures to abstract attribute grammars (AAG) that will allow us to check the constraints they impose on the *paths* to the utterances, in the discourse structure. Thus, the non-unifiability of the restriction sets determined by the AAGs for each particular utterance represent a criterion for ruling out the rhetorical relations that contributed the AAGs with non-unifiable restriction sets. This "contextual filtering" approach is first presented, theoretically grounded and discussed with respect to several potential weak points. Then, an extended example illustrates the feasibility of the proposed method for dialogue situations.

1 Introduction

Determining the discourse structure of texts is a thoroughly-studied, yet not completely solved, issue, either from a theoretical standpoint, or guided by practical, computational goals. On the theoretical side, first discourse coherence was hinted at, abstracting away from the types of rhetorical connections between text spans; Discourse Representation Theory (DRT) is a typical illustration of this strand (Kamp and Reyle, 1993). However, although DRT could predict some phenomena, such as certain anaphoric chains, other aspects such as ellipsis (Asher et al, 2001) or illocutionary aspects (Vanderveken, 1990–1991) were left untackled. Hence, one has moved to a more fine-grained analysis of the nature of rhetorical connections between text spans. One of the first attempts in this respect is Rhetorical Structure Theory (RST) (Mann and Thompson, 1988), where an *open* set of several tens of rhetorical relation types are proposed, such as ELABORATION, CONCESSION, etc. However, RST misses the depth of the (utterance-level) analysis performed in DRT, thus failing to predict sub-sentential linguistic phenomena (Asher, 1993). This is why, the idea of combining insights from both DRT and RST was quite natural. One of the first and most representative attempts in this respect was Asher's SDRT (Asher, 1993), where the fine-grained predictive power of DRT was extended to multi-utterance discourses, by specifying the types of the rhetorical relations that connect spans of text. This early work of Asher (1993) was mainly concerned only with monologue, written, texts. An extension of this approach to dialogue situations was started with Asher and Lascarides (2003), and henceforth continued in Lascarides and Asher (2009), for dealing with more dialogue-specific issues, such as commitments modeling (Asher and Lascarides, 2008), or common ground (Traum, 1994).

However, it was evident even from early work of Asher (1993) and Asher and Lascarides (2003) that a thorny issue had not received enough attention an adequate account: the *underspecification* of the discourse relations (Reyle, 1993). This underspecification is due either to the lack of enough contextual elements for computing a unique appropriate rhetorical relation between two utterances or text spans, or to the inherent

[*] Now at Laboratoire Informatique d'Avignon, University of Avignon, France; e-mail: vladimir.popescu@univ-avignon.fr.

ambiguity of these very utterances, or of the (optional) discourse connector between them (e.g. "but yet"). One way of dealing with this issue was proposed by Hobbs et al (1993), and consisted in using abductive reasoning for imposing constraints on the appropriate rhetorical relations connecting utterances. Abduction is, however, a rather fragile approach, because, on the other hand, an extensive and fine-grained knowledge base is needed and, on the other hand, in a reasoning chain like "if $P \Rightarrow Q$ and Q are true, then, *normally*, P is true", there can be several potential premises for Q, and P actually covering a *set* of propositions. The problem stems from the need to decide which particular P in that set is the most relevant premise. Another way of dealing with underspecification was proposed by Egg and Redeker (2007), where *holes* are left for the unspecified rhetorical relations or for their scopings (i.e., the spans of text that are the arguments of these relations). These holes are then progressively filled when contextual information becomes available, e.g., via subsequent sentences or utterances in discourse.

On the practical side, several attempts have been made for building discourse parsers. Some of these were purely rule-based, such as early work of Marcu (1997), where constraints on discourse coherence, in the RST framework, were formalized in a first-order logic and driven by a mapping between discourse connectives and rhetorical relations. Other attempts at achieving robust discourse parsing were corpus-based, as in later work of Marcu (2000), where several metrics derived from corpora were linearly combined for ordering the RST discourse trees obtained for the same discourse, or statistical, like in Sporenader and Lascarides (2006), where different word-level Bayesian classifiers are combined and tested, with limited effectiveness, though. Another approach, proposed in Schilder (2001) combined theoretical insights from Asher's SDRT and Egg's Underspecified SDRT, with practical Information Retrieval (IR) methods for achieving a progressive specification of an initially fully underspecified initial Segmented Discourse Representation Structure (SDRS). Several clues were used in Schilder (2001) for progressively specifying the SDRSs: first, discourse connectives are used for constraining the set of rhetorical relations, then, information regarding the topicality of the utterances, as well as their position (in a properly formatted written text, such as a newspaper article) is used for further assessing the *relevance* (in an IR sense) of the utterances. Although, unlike, e.g., work by Asher and Lascarides (2003), Schilder's approach exhibits the crucial advantage of not relying on a domain encyclopedic knowledge base for deriving discourse structure, it also has some drawbacks. First, the mapping between discourse connectives and rhetorical relations is rather crude, simplistic: indeed, for instance it is said that "because" marks the rhetorical relation of *Explanation* between its arguments. However, as pointed out in Knott (1996) or Jayez and Rossari (1999), a discourse connective can mark a relation between the mere intensions of its arguments, or between the speech acts conveyed by these arguments, or even by the attitudes of the speakers towards these arguments. Moreover, it is shown in Jayez and Rossari (1999) that connectors actually have *semantic profiles*, that is, they are defined by a set of dynamic constraints they impose on states in the worlds that license the arguments of these connectives. Hence, the mapping between discourse connectives and rhetorical structures (SDRSs) is more intricate than implied by Schilder's proposal.

Thus, taking as a departure point Asher and Lascarides' SDRT as developed in Asher and Lascarides (2003), we propose an alternative way of tackling rhetorical relation ambiguity. More precisely, in SDRT the MDC principle is used for ordering SDRSs in terms of their relevance. For this, an essential ingredient is the *quality* of the rhetorical relations that connect a given pair of utterances or discourse constituents. The problem with this approach stems from the fragility of this scalar interpretation of the rhetorical relations: indeed, it is stated in Asher and Lascarides (2003) that, e.g., an *Elaboration* relation has a higher quality than *Narration*. This might not always be true, for instance, when the speaker is supposed to perform an enumeration of events that follow in time, without any other underlying connection (be it elaboration) holding between these events. This is why we propose in this paper to replace MDC with a rather "syntactical" notion of context, which allows us to put forward a rather precise criterion for ruling out alternative rhetorical relations between a pair of utterances. The context is achieved by starting from a local coherence analysis, as proposed in Asher and Lascarides (2003), for computing a set of semantically-valid rhetorical relations between two utterances; for this, only the logical forms of these utterances and the SDRT semantics of the rhetorical relations are used. Then, the semantics of each rhetorical relation that was deemed to hold between a pair of utterances, is approximated in a static first-order predicate logic, and then converted into a (Prolog-like) logical program, where the labels of the utterances are the variables. Such a program is converted in its turn into an AAG (Isakowits, 1991), defined by a set of production rules and a set of *restriction sets*, one restriction set per variable; such a restriction set represents represents the set of *paths* (Deransart and Małuzýnski, 1993) to each variable. Then, the grammars for all the rhetorical relations deemed to hold (on a local basis) in a discourse are merged in a discourse-level AAG. The coherence criterion is represented by the unifiability of the attribute sets associated to each variable. In other words, we check that no inconsistent predicates are deemed valid for any utterance (a.k.a. variable in the discourse-level AAG). Thus, we rule out any rhetorical relation that yields an AAG with a restriction set that is not unifiable with the other restriction sets in discourse, for the variable concerned.

After a brief review of the semantics of some SDRT rhetorical relations used and of the rationale behind these (Section 2.1), we show how each of these relations is converted into a logic program (Section 2.2), then from a logic program to an abstract attribute grammar (Section 2.3). In Section 3 we present the core of the paper, the contextual filtering mechanism (Section 3.1), along with the rationale behind it (Section 3.2) and with a discussion of particular topics in this regard (Section 3.3). In Section 4 we show how the computational cost of each rhetorical relation can be assessed and the numerical filtering performed. In Section 5, an extended example of contextual filtering is analyzed in detail. A set of conclusions end up the paper (Section 6).

2 From Discourse Relations to Abstract Attribute Grammars

2.1 First-order Semantics of SDRT Rhetorical Relations

Vanilla SDRT proposes a dynamic semantics framework for representing and updating discourse structures. More specifically, it uses a dynamic propositional logic (the Glue) for expressing the semantics of the rhetorical relations (Asher and Lascarides, 2003), and a first-order dynamic semantics, inherited from Kamp and Reyle's DRT (Kamp and Reyle, 1993), for representing the semantics of the utterances (more precisely, their intensional aspects). However, a direct conversion from a dynamic logic (be it propositional, like SDRT's Glue) to a statical representation, like a (Prolog-like) logic program, or an attribute grammar (Deransart and Małuzýnski, 1993) is not obvious, because this would impose the necessity of performing model consistency and variable unification checks that can be crippling, from a computational standpoint (Gallier, 1986), (Staudacher, 2005). For these reasons, we considered necessary to smoothen the transition between SDRT's dynamic rhetorical relations semantics, and statical representations, by approximating the SDRT semantics in a static, first-order predicate logic (Popescu et al, 2007).

In the sequel we give an outlook of this first-order emulation of (a fragment of) SDRT. Namely, we present the main ideas behind the first-order logical expressions for some rhetorical relations, showing that these semantics preserve the prescriptions in vanilla SDRT (Asher and Lascarides, 2003). The semantics of these SDRT rhetorical relations are expressed in terms of a set of discourse predicates and functions.

One first predicate, $\mathsf{question}/1$, is true if its argument is a question[1], which comes to its logical form containing non-initialized variables; at a semantic level, this is formalized as follows:

$\mathsf{question}(\alpha) ::= \exists\nu : \mathsf{MemberOf}(\nu, K(\alpha)) \wedge \neg\exists\omega : \mathsf{MemberOf}(\omega, \Omega) \wedge \mathsf{equals}(\nu, \omega)$.

A similar definition is given for the predicate $\mathsf{enounce}/1$, true if its argument is not a question. Thus:

$\mathsf{enounce}(\alpha) ::= \neg\mathsf{question}(\alpha)$.

The $\mathsf{confirmation}/2$ predicate is true if the sentence that occurs as its second argument confirms the sentence / utterance that occurs as its first argument, and the two utterances were produced by different agents. Thus,

$\mathsf{confirmation}(\alpha, \beta) ::= \mathsf{equals}(K(\beta), K(\alpha)) \wedge \neg\mathsf{equals}(\mathsf{emitter}(\beta), \mathsf{emitter}(\alpha))$.

The function $\mathsf{answer}/2$ returns the set of logical forms that represent answers to the utterance taken as argument. In technical terms,

$\mathsf{answer}(\beta, \alpha) ::= \mathsf{equals}(\mathsf{topic}(\beta), \mathsf{topic}(\alpha) \wedge \mathsf{greater}(t(\beta), t(\alpha)) \wedge \forall\nu : \mathsf{MemberOf}(\nu, K(\beta))\exists\omega :$
$\mathsf{MemberOf}(\omega, \Omega) \wedge \mathsf{equals}(\nu, \omega)$.

[1] Here, in include indirect questions as well. In this case, the non-initialized variable concerns the truth value of the predicate of a proposition.

The function $\mathsf{topic}/1$ worths more discussion, since it tries to substitute the "topic" discourse relation in vanilla SDRT ($\Downarrow$). Our function just retrieves a structure that contains the types and values of the initialized variables in the utterance it has as argument. Hence, in our emulation of SDRT the topic is an utterance-level feature, instead of being a relation between a couple of utterances or discourse constituents, as in vanilla SDRT. Technically, this is represented as:

$\mathsf{topic}(\alpha) ::= \mathsf{ExhaustiveDecomposition}(\nu, \omega) \wedge \mathsf{MemberOf}(\nu, K(\alpha)) \wedge \mathsf{MemberOf}(\omega, \Omega) \wedge \mathsf{equals}(\nu, \omega)$.

The function $SARG/1$ returns the "Speech Act-Related Goal" of the utterance taken as argument (Asher and Lascarides, 2003). In our approximation, $SARG(\alpha)$ returns the action predicates in the logical form of utterance α, that is, in $K(\alpha)$ (Popescu et al, 2007); these are extracted by using an encyclopedic or domain-specific knowledge base.

The predicates $\mathsf{good\_time}/1$ and $\mathsf{bad\_time}/1$ bare the same meanings as in the Verbmobil project (Schlangen et al, 2001): they are true if their argument is a moment in time that corresponds to another moment in time in a preceding utterance, that (partially) shares the same topic as the utterance concerned. Technically,

$\mathsf{good\_time}(\Delta t(\beta)) ::= \exists \alpha : \neg\mathsf{Disjoint}(\mathsf{topic}(\alpha), \mathsf{topic}(\beta)) \wedge \mathsf{smaller}(t(\alpha), t(\beta)) \wedge (\mathsf{SubclassOf}(\Delta t(\beta), \Delta t(\alpha))$
$\vee \mathsf{equals}(\Delta t(\alpha), \Delta t(\beta)));$

$\mathsf{bad\_time}(\Delta t(\beta)) ::= \neg\mathsf{good\_time}(\Delta t(\beta))$.

The functions $\Delta t/1$ and $t/1$ deserve a particular discussion, since, although both are functions taking utterances as arguments and returning moments in time, the first function returns the time span conveyed by the utterance, whereas the second function simply returns the moment in time (with respect to the beginning of the current dialogue) where the utterance was produced. Thus, whereas the t function is computed by simply internal management of the utterances in the dialogue controller, the Δt function is computed by virtue of a semantics related to the utterance that the function is applied to. Hence,

$\Delta t(\beta) ::= (t_2 - t_1) : \exists t_1, t_2 : (\mathsf{MemberOf}(t_2 - t_1, K(\beta))) \vee (\mathsf{MemberOf}(t_2 - t_1, \mathsf{topic}(\beta)) \wedge \mathsf{MemberOf}(t_1, K(\beta))$
$\wedge \mathsf{MemberOf}(t_2, K(\beta)))$.

Then, we have several predicates for set relations: $\mathsf{MemberOf}/2$ for "$\in /2$", $\mathsf{SubclassOf}/2$ for "$\subset /2$", $\mathsf{Disjoint}/2$ for "$\cap/2 = \emptyset$", $\mathsf{ExhaustiveDecomposition}/2$ for "$\cup/2 = \mathsf{All}$"; for measures: $\mathsf{equals}/2$ for "$=/2$", $\mathsf{smaller}/2$ for "$< /2$", and $\mathsf{greater}/2$ for "$> /2$".

As for rhetorical relations, we first consider some dialogue relations, that will be used in the remainder of this paper; the complete outlook of the approximated SDRT rhetorical relations semantics can be found in Popescu (2008):

▷ *P-Elab* ("Plan Elaboration"): it is a subordinating relation that connects two utterances or discourse constituents such that the second utterance elaborates on a plan for reaching the goal stated in the first utterance. An example is shown below:

U_1: Could I have this book for next week?$^\alpha$

U_2: In order to have it you must go to our headquarters on Martin Street and reserve it, from the beginning of next week$^\beta$.

The semantics of *P-Elab* is shown below:

$P - Elab(\alpha, \beta) ::= \mathsf{good\_time}(\Delta t(\beta)) \wedge \neg\mathsf{Disjoint}(\Delta t(\beta), SARG(\alpha)) \vee \mathsf{bad\_time}(\Delta t(\beta)) \wedge$
$\neg\mathsf{equals}(SARG(\alpha) \setminus \Delta t(\beta), \emptyset)$.

▷ $Background_q$: it connects two utterances or discourse constituents such that the second utterance is a question to which any answer is in a *Background* relation (see below) with the first utterance. An example is shown below:

U_1: This book has already been lended by a customer yesterday$^\alpha$.

U_2: Are there any other customers that have looked for this book?$^\beta$

The semantics of $Background_q$ is shown below:

$Background_q(\alpha, \beta) ::= \mathsf{enounce}(\alpha) \wedge \mathsf{question}(\beta) \wedge (\forall \chi : \mathsf{answer}(\chi, \beta) \Rightarrow Background(\chi, \alpha))$.

▷ *Elab*$_q$ ("Elaboration$_q$"): it is a subordinating relation that connects two utterances or discourse constituents such that the second utterance is a question to which any answer is in a *Elab* relation (see below) with the first utterance. An example is shown below:

U_1: We have got new books on your field of interest$^\alpha$.

U_2: Could you give me some titles?$^\beta$

The semantics of *Elab*$_q$ is shown below:
$Elab_q ::= \mathsf{enounce}(\alpha) \wedge \mathsf{question}(\beta) \wedge (\forall\chi : \mathsf{answer}(\chi, \beta) \Rightarrow Elab(\alpha, \chi))$.

▷ *QAP* ("Question-Answer Pair"): it is a subordinating relation that connects two utterances or discourse constituents such that the first utterance is a question and the second utterance is a direct answer to this question[2]. An example is shown below:

U_1: Where can I find book number '11608208'?$^\alpha$

U_2: The second floor, to the right, on the third shelf$^\beta$.

The semantics of *QAP* is shown below:
$QAP(\alpha, \beta) ::= \mathsf{question}(\alpha) \wedge \mathsf{enounce}(\beta) \wedge \neg\mathsf{Disjoint}(\mathsf{topic}(\alpha), \mathsf{topic}(\beta)) \wedge \mathsf{answer}(\beta, \alpha)$.

▷ *ACK* ("Acknowledgement"): it is a subordinating relation as well, that connects two utterances or discourse constituents, such that the second utterance is a direct confirmation (by a speaker different from the producer of the first utterance) of the first utterance. An example is shown below:

U_1: Is this book OK for you?$^\alpha$

U_2: Yes$^\beta$.

The semantics of *ACK* is shown below:

$$\begin{aligned} ACK(\alpha, \beta) ::= {} & (\mathsf{question}(\alpha) \vee \mathsf{enounce}(\alpha)) \wedge \mathsf{enounce}(\beta) \wedge \neg\mathsf{Disjoint}(\mathsf{topic}(\alpha), \mathsf{topic}(\beta)) \wedge \\ & \mathsf{confirmation}(SARG(\alpha), \beta). \end{aligned}$$

As for the monologue relations, we present below some of them, used in the remainder of the paper; the full list of first-order approximations of the SDRT rhetorical relations can be found in Popescu (2008); the same notational conventions as for the dialogue relations are followed:

▷ *Background*: it is a coordinating relation that connects two utterances or discourse constituents such that the first utterance specifies immutable contextual information for the second utterance. An example is shown below:

U: I will lend you the book "Back to America" by Michel Ardan$^\alpha$.

U: You are allowed to borrow up to three books$^\beta$.

The semantics of *Background* is specified below:
$Background(\alpha, \beta) ::= \mathsf{enounce}(\alpha) \wedge \mathsf{enounce}(\beta) \wedge \mathsf{entails}(\alpha, \beta)$.

▷ *Consequence*: it is a coordinating relation that corresponds to the dynamic implication (⇒); here, it will be simulated through the static implication (see below). An example is shown below:

U: If we lent you Michel Ardan's "Back to America"$^\alpha$,

U: then the library would run out of instances of this book!$^\beta$

The semantics of *Consequence* is shown below:
$Consequence(\alpha, \beta) ::= \mathsf{enounce}(\alpha) \wedge \mathsf{enounce}(\beta) \wedge (K(\alpha) \Rightarrow K(\beta))$.

▷ *Elaboration*: it is a subordinating relation that connects two utterances or discourse constituents such that the second utterance elaborates on the theme of the first utterance and that the two utterance have the same topic. An example is shown below:

[2] Here, by 'direct answer' we mean an answer that allows only for the variables not-initialized in the question, to be initialized, not providing further information, not asked for in the question.

U: I cannot lend you Michel Ardan's "Back to America"!$^{\alpha}$

U: It is the only exemplar in the library!$^{\beta}$

The semantics of *Elaboration* is given below:
$Elaboration(\alpha, \beta) ::= \mathsf{enounce}(\alpha) \wedge \mathsf{enounce}(\beta) \wedge (\mathsf{SubclassOf}(\mathsf{topic}(\beta),\ \mathsf{topic}(\alpha)) \vee$
$\mathsf{equals}(\mathsf{topic}(\alpha),\ \mathsf{topic}(\beta)).$

▷ *Narration*: it is a coordinating relation that connects two utterances that share the topic and are consecutive in time. An example is shown below:

U: You go to the corridor with Scandinavian Literature$^{\alpha}$.

U: You look for the shelf named "Andersen"$^{\beta}$.

The semantics of *Narration* is given below:
$Narration(\alpha, \beta) ::= \mathsf{enounce}(\alpha) \wedge \mathsf{enounce}(\beta) \wedge \neg\mathsf{Disjoint}(\mathsf{topic}(\alpha),\ \mathsf{topic}(\beta)) \wedge \mathsf{smaller}(\Delta t(\alpha), \Delta t(\beta)).$

2.2 Discourse Relations as Logic Programs

The conversion from the semantics of rhetorical relations to logic programs having utterance labels as input variables is rather straightforward from the manner whereby we defined the semantics of the relations, gradually rendering explicit several predicates used. Thus, for a general account, let us first assume that we have a rhetorical relation ρ, expressed in terms of predicates p_1 to p_n (α and β designate the utterances):

$\rho(\alpha, \beta) ::= p_1\xi_1 p_2\xi_2 ... \xi_{n-1} p_n,$

where ξ_i for i from 1 to $n-1$ designate predicate connectors in the set $\{\wedge, \vee, \neg, \Rightarrow\}$, along with quantifiers in the set $\{\exists, \forall\}$. In its turn, each of the predicates p_i is expressed in terms of other functors t^i_j, with j from 1 to n_i. We also assume that the arguments of the predicates p_i are terms of the form s^i_k and that some of these terms take α and / or β as arguments. By re-numbering the inferior indexes for the s terms, the semantics above can be rephrased as:

$\rho(\alpha, \beta) ::= p_1(s^1_1(\alpha, ...), s^1_2(\beta, ...), ...)\xi_1 ... \xi_{n-1} p_n(s^n_1(\alpha, ...), s^n_2(\beta, ...), ...).$

Similarly, assuming that among the t terms there exist some that take α and β as arguments and re-indexing these terms, we have the following predicate definition, for each p_i, that further extends, recursively, until facts in the knowledge base are reached (here, ξ^i_j are a re-indexing of ξ_i, therefore they are connectors or quantifiers as well):

$p_i(s^i_1(\alpha, ...), s^i_2(\beta, ...), ...) ::= t^i_1(\alpha ...)\xi^i_1 t^i_2(\beta, ...)\xi^i_2 ... \xi^i_{n_i-1} t^i_{n_i-1}(...).$

In order to turn these specifications into Prolog programs, we have to replace $::=$ by $:-$ and to perform a mapping μ from the set of connectors and quantifiers, to the set of legal Prolog constructs:

$\mu : \{\wedge \vee \neg \Rightarrow \exists\forall\} \rightarrow \{, ; \backslash+\}.$

For the first three elements in these sets, the mapping is straightforward ($\mu(\wedge) =,$, $\mu(\vee) =;$ and $\mu(\neg) = \backslash+$), whereas for the last three, denoting by $\uparrow$ the concatenation operator, we have that $\mu(\Rightarrow) = \mu(\neg) \uparrow \mu(\vee) = \backslash+ \uparrow ;$[3]. As for the quantifiers, taking into account Prolog behavior (Gallier, 1986), $\exists$ can be eliminated, and $\forall$ replaced with the $\mathsf{setof/3}$ predicate[4].

Hence, we will present below a concrete example of logic program obtained from the semantics of the *QAP* rhetorical relation, assuming that the utterances are input variables.

We start from the semantics of *QAP*, that we reprise here for the ease of reading:

$QAP(\alpha, \beta) ::= \mathsf{question}(\alpha) \wedge \mathsf{enounce}(\beta) \wedge \neg\mathsf{Disjoint}(\mathsf{topic}(\alpha), \mathsf{topic}(\beta)) \wedge \mathsf{equals}(\beta, \mathsf{answer}(\alpha));$

Then, the program is shown below, expanding each discourse predicate or function up to built-in Prolog predicates or to facts.

(1) $QAP(\alpha, \beta) : -\mathsf{question}(\alpha), \mathsf{enounce}(\beta), \backslash + \mathsf{Disjoint}(\mathsf{topic}(\alpha), \mathsf{topic}(\beta)), \mathsf{answer}(\beta, \alpha).$

(2) $\mathsf{enounce}(\alpha) : -\mathsf{memberOf}(X, K(\alpha)), \mathsf{memberOf}(Y, \Omega), X == Y.$

[3] This is based on the fact that, logically, the clause $\alpha \Rightarrow \beta$ is equivalent to $\neg\alpha \vee \beta$.

[4] More precisely, a clause of the form $\forall X : p(X)$ can be replaced with $\mathsf{setof}(X, p(X), \mathsf{Set})$, where Set collects all the instantiations of X such that $p(X)$.

(3) $\mathsf{question}(\alpha) : -\backslash + \mathsf{enounce}(\alpha).$

(4) $\mathsf{memberOf}(X, Y) : -\mathsf{member}(X, Y).$

(5) $K(X) : -X = ..L, L.$

(6) $\mathsf{Disjoint}(X, Y) : -\mathsf{memberOf}(\_X, X), memberOf(\_Y, Y), \_X = \backslash = \_Y.$

(7) $\mathsf{topic}(X) : -\mathsf{ExhaustiveDecomposition}(\phi(\_i), \omega(\_j)), \mathsf{memberOf}(\phi(\_i), K(X)),$
$\mathsf{memberOf}(\omega(\_j), \Omega), \phi(\_i) == \omega(\_j).$

(8) $\mathsf{ExhaustiveDecomposition}(X, Y) : -\mathsf{memberOf}(\_X, X), \mathsf{memberOf}(\_Y, Y),$
$\mathsf{memberOf}(\_X, \Omega),$
$\mathsf{memberOf}(\_Y, \Omega), \mathsf{memberOf}(\_\omega, \Omega),$
$(\mathsf{memberOf}(\_\omega, X);$
$\mathsf{memberOf}(\_\omega, Y)), (\mathsf{memberOf}(\_X, X)- >$
$\backslash + \mathsf{memberOf}(\_X, Y); \mathsf{memberOf}(\_Y, Y)- >$
$\backslash + \mathsf{memberOf}(\_Y, X)).$

(9) $\mathsf{answer}(X, Y) : -\mathsf{topic}(X) == \mathsf{topic}(Y), t(X) > t(Y), \mathsf{setof}(\_\nu, \mathsf{member}(\_\nu, K(\beta)), Set),$
$\mathsf{member}(\_\omega, \Omega), \mathsf{member}(\_\nu, Set), \_\nu == \_\omega.$

(10) $\mathsf{emitter}(X) : -\backslash + \mathsf{memberOf}(u, K(X))- > X = u; \backslash + \mathsf{memberOf}(m, K(X))- > X = m.$

The set Ω represents the concepts in the domain knowledge base. It can be asserted as a Prolog entity:
:- $\mathsf{assert}(\Omega = [$/*Concepts list*/$].)$.

2.3 A Grammatical View of Discourse Relations

The correspondence between logic programs and attribute grammars is a thoroughly studied subject (Deransart and Małuzýnski, 1993). The difference between attribute grammars and abstract attribute grammars resides in the variable assignments that accompany each production rule: whereas in attribute grammars there is an *equality assignment* for each variable, to a corresponding value, in abstract attribute grammars variables are specified through *restriction sets* that contain all variables that unify into the same value (known or not). This difference is important, since attribute grammars need a prior splitting of variables into *inherited* (i.e. input) and *synthesized* (i.e. output) variables, whereas abstract attribute grammars (hence their name) do not need that splitting. This is particularly useful for a "grammatical account" on programs for which no query has been formulated (and hence, no input/output variable splitting exists). This can be assimilated to the situation where a logic program that corresponds to a rhetorical relation is handled for any possible pair of utterances (i.e. utterances are kept as variables and not instantiated).

In the general case, a transformation from a logic program (that corresponds to the semantics of a rhetorical relation) to an abstract attribute grammar supposes the construction of a production rule and of a set of restriction sets (one such set for each variable in the original Prolog rule). Thus, for a rule of the form (notations in Section 2.1 are kept):

$\rho(\_\alpha, \_\beta) : -p_1(s_1^1(\_\alpha, ...), s_2^1(\_\beta, ...), ...)\mu(\xi_1)...\mu(\xi_{n-1})p_n(s_1^n(\_\alpha, ...), s_2^n(\_\beta, ...), ...),$

we obtain the following production rule, in an AAG:

$\rho \rightarrow p_1\tau(\mu(\xi_1))...\tau(\mu(\xi_{n-1}))p_n,$

where τ is a mapping function, from Prolog connectors to BNF ("Backus-Naur Form") notation: $\tau : \{, ; \backslash +\} \rightarrow \{, |\}$, and $\tau(\backslash + p) = q$, so that $q \equiv \neg p$.

In order to construct the restriction sets for the variables (aka attributes) that occur in the original Prolog rule, we define the *selector* functions λ_m^f: for a functor f of arity M, π_m^f returns the m-th argument of f, if $m \leq M$, and $\emptyset$ otherwise. Thus, for example, if we have $f(t_1, ..., t_m, ..., t_M)$, then $\lambda_m^f = t_m$. These selector functions can be composed: if in its turn t_m is of arity N we can select its n-th argument by $\lambda_n^{t_m}$, for $n \leq N$: if we have $t_m(s_1, ..., s_n, ..., s_N)$, then $\lambda_m^f \circ \lambda_n^{t_m} = s_n$. Here, the composition relation $\circ$ is defined by $f \circ g(x) = g(f(x))$ (Deransart and Małuzýnski, 1993).

Thus, following a construction similar to Isakowits (1991), we denote the elements in the restriction sets for a rule by $p.i(j)$, where p is the predicate where the variable occurs, i is the index (the position) of the variable in predicate p, and j designates the *depth* of the predicate: j equals ϵ if the predicate p is to the left side of the production rule, 1 if the predicate is just to the right of the $\rightarrow$ symbol of the production rule and t if there are $t-1$ terms (separated by connectors) between the → symbol and predicate p. Moreover, we denote by id the identity function: $id(X) = X$, whatever that X is (i.e. term or atom). Hence, for the general Prolog rule considered above, the following restriction sets corresponding to variables $\_\alpha$ and $\_\beta$ attach to the production rule shown above:

$\_\alpha : \{\rho.1(\epsilon), \lambda_1^{s_1^1} \circ id(p_1.1(1)), ...\lambda_1^{s_1^n} \circ id(p_n.1(n))\}$; $\_\beta : \{\rho.2(\epsilon), \lambda_1^{s_2^1} \circ id(p_1.2(1)), ...\lambda_1^{s_2^n} \circ id(p_n.2(n))\}$.

We will illustrate the approach on an example, concerning the QAP rhetorical relation, for which the semantics and logic program were shown in previous sections. For each line of the logic program there will be a line containing a production rule, and a "line" containing the restriction sets for the variables involved. First, we start with the production rules:

(1′) $QAP \rightarrow$ enounce, enounce, neg, confirmation
(2′) enounce → memberOf, memberOf, ==
(3′) memberOf → member
(4′) $K \rightarrow$ = ..
(5′) Disjoint → memberOf, memberOf, neg
(6′) topic → ExDec, memberOf, memberOf, ==
(7′) ExDec → memberOf, memberOf, memberOf, memberOf, memberOf, memberOf, neg | neg | neg | neg
(8′) confirmation → ==, neg
(9′) emitter → Disjoint | = | Disjoint | =.

The non-terminal neg denotes the application of the τ function on a construction of the type $\backslash + p$, to yield a q such that $q \equiv \neg p$. In these production rules, we consider that built-in predicates in Prolog and facts in the knowledge bases available are terminals (e.g. in the rules above, the predicates = ../2, ==/2, =/2 and member/2 are terminals).

As for the restriction sets that result from the variables present in the original Prolog rules, we list below a few (those associated to the first two production rules):

(1″) $\_\alpha : \{id(QAP.1(\epsilon)), id(\mathsf{enounce}(1)), \lambda^{\mathsf{neg}} \circ \lambda_1^{\mathsf{Disjoint}} \circ id(\mathsf{topic}(3)), \lambda_1^{\mathsf{confirmation}} \circ id(SARG(4))\}$;
$\_\beta : \{id(QAP.2(\epsilon)), id(\mathsf{enounce}(2)), \lambda^{\mathsf{neg}} \circ \lambda_2^{\mathsf{Disjoint}} \circ id(\mathsf{topic}(3)), id(\mathsf{confirmation.2(4)})\}$
(2″) $\_\alpha : \{id(\mathsf{enounce}(\epsilon)), \lambda_2^{\mathsf{memberOf}} \circ id(K(1)), id(\mathsf{memberOf.}1(1)), id(== .1(2))\}$;
$\_X : \{id(\mathsf{memberOf.}1(1)), id(== .1(2))\}$; $\_Y : \{id(\mathsf{memberOf.}2(1)), id(== .2(2))\}$.

3 Contextual Filtering of Rhetorical Relations

3.1 Filtering Procedure

One way of leaning on these semantics can consist in trying to assess the relevance of a certain rhetorical relation with respect to the overall discourse context, not only the pair of utterances connected by the relation. One manner of doing this is described in this section, and consists in viewing a discourse structure as *coherent* only if it can be *computed*, via the semantics of the rhetorical relations and of the utterances. This brings us to the following:

Definition 1. *A discourse structure (SDRS) is* ***coherent*** *if it can be computationally derived, via a logic program, using the semantics of the rhetorical relations.*

From a structural point of view, a given SDRS at an updating step consists in:

- utterances or discourse constituents: π_1, ..., π_N;
- rhetorical relations: ρ_1, ..., ρ_M;
- structure: $SDRS_{MN} = \bigcup_{i=1}^{M} \bigcup_{j,k=1}^{N} \rho_i(\pi_j, \pi_k)$; moreover, there might exist $k \in \{1, ..., N\}$ such that there exist $l \in \{1, ..., M\}$ and $m, n \in \{1, ..., N\}$ such that $\pi_k = \rho_l(\pi_m, \pi_n)$.

If an utterance, labeled π_{N+1} is to be added to $SDRS_{MN}$, we assume that SDRT-based local coherence analysis yielded $R \geq 1$ rhetorical relations such that:

$\forall i \in \{1, ..., R\}\ \forall r_i \in \{1, .., N\} \wedge |\langle r_i \rangle| = R : \rho_i(\pi_{r_i}, \pi_{N+1}) = \mathsf{TRUE}$,

where $\langle x \rangle$ means the set of possible x satisfying certain conditions priorly specified.

In this setting, contextual filtering essentially consists in the following stages:

1. Conversion of $SDRS_{MN}$ into a logic program, by converting each rhetorical relation in the corresponding LP and then *concatenating* these LPs:

   $SDRS_{MN} = \bigcup_{i=1}^{M} \bigcup_{j,k=1}^{N} \rho_i(\pi_j, \pi_k) \longmapsto \uparrow_{i=1}^{M} \uparrow_{j,k=1}^{N} LP_{\rho_i}(\pi_j, \pi_k) = LP_{MN}$.

   The mapping $\rho \mapsto LP_\rho$ is performed as shown in Section 2.2: if

   $\rho_i(\pi_j, \pi_k) ::= p_1(s_1^1(\pi_j, ...), s_2^1(\pi_k, ...), ...)\xi_1 ... \xi_{I-1} p_I(s_1^I(\pi_j, ...), s_2^I(\pi_k, ...), ...)$, then the corresponding clause in LP is $\rho_i(\pi_j, \pi_k) : -p_1(s_1^1(\pi_j, ...), s_2^1(\pi_k, ...), ...)\mu(\xi_1) ... \mu(\xi_{I-1}) p_I(s_1^I(\pi_j, ...), s_2^I(\pi_k, ...), ...)$;

2. Conversion of the logic program LP_{MN} corresponding to the discourse structure, into an abstract attribute grammar:

   $LP_{MN} = \uparrow_{i=1}^{M} \uparrow_{j,k=1}^{N} LP_{\rho_i}(\pi_j, \pi_k) \mapsto \uparrow_{i=1}^{M} \uparrow_{j,k=1}^{N} AAG_{\rho_i}(\pi_j, \pi_k) = AAG_{MN}$.

   The mapping $LP_\rho \mapsto AAG_\rho$ is performed as shown in Section 2.3: for an LP as shown at step 1, we have rules of the form:

   $(1)\ \rho_i \rightarrow p_1 \tau(\mu(\xi_1)) ... \tau(\mu(\xi_I)) p_I$,

   $\vdots$

   $(K)\ p_I \rightarrow t_1^I \tau(\mu(\xi_1^I)) ....$

   The corresponding restriction sets for the (*global* – see below) variables π_j and π_k are of the form:

   $(1')\ \pi_j : \{\rho_i.1(\epsilon), \lambda_1^{p_1} \circ id(s_1^1.1(1)), ..., \lambda_1^{p_I} \circ id(s_1^I.1(I))\}$; $\pi_k : \{\rho_i.2(\epsilon), \lambda_2^{p_1} \circ id(s_2^1.1(1)), ..., \lambda_2^{p_I} \circ id(s_2^I.1(I))\}$;

   $\vdots$

   $(K')\ \pi_j : \{\lambda_1^{p_I} \circ id(s_1^I.1(\epsilon)), \lambda_{...}^{t_1^I} \circ \ldots, ...\}$; $\pi_k : \{\lambda_2^{p_I} \circ id(s_2^I.1(\epsilon)), \lambda_{...}^{t_1^I} \circ \ldots, ...\}$.

   Here, a global variable means a logic variable that neither changes its meaning, nor resets its value, from one rule in an AAG to another.

3. Addition of $AAG_{\rho_i}(\pi_{r_i}, \pi_{N+1})$, with $i = 1, ..., R$, to AAG_{MN}:

   $AAG_{M+1,N+1}(i) = AAG_{MN} \uparrow AAG_{\rho_i}(\pi_{r_i}, \pi_{N+1})$.

   We thus obtain R abstract attribute grammars that correspond to R *possible* discourse structures, each one with the same $N+1$ utterances, but with different sets of $M+1$ rhetorical relations[5].

4. Unifiability check, for each of the R discourse-level AAGs: that is, for each global variable (i.e. utterance) in each such grammar, we have to *unify* all the restriction sets for that variable, in order for the logic program that generated the grammar to be consistent. This means that for the LP, from which the AAG was generated, to be consistent (i.e. it does not contain mutually incompatible rules – rules that cannot be all true under the same variable assignments) we have to ensure that all the elements in all the restriction sets for each global variable in each grammar $AAG_{M+1,N+1}(i)$ can be unified. For instance, for the global variable π_j shown at step 2, we should have that, if the current rhetorical relation i is such that $r_i = j$ (this in not guaranteed to be true, but if it is not, we choose a j for which it is true), an element of the type $\rho_o.1(\epsilon)$ (where ρ_o denotes an "older" rhetorical relation, that is, already present in $SDRS_{MN}$) can be unified with an element of the type $\lambda_k^{p_J} \circ id(s_k^J.m(J))$. Hence, this means that the equation $\rho_o.1(\epsilon) = \lambda_k^{p_J} \circ id(s_k^J.m(J))$ should have solutions. In Section 5 we show, via an extensive example, that sometimes this is true, sometimes it is not.

[5] These sets of rhetorical relations differ by one element, precisely the last rhetorical relation added.

5. If there exists a global variable (utterance, actually) in $AAG_{M+1,N+1}(i)$ for which all its restriction sets cannot be unified (i.e. there exist at least two elements in two distinct restriction sets that cannot be unified), this means that $LP_{MN} \uparrow LP_{\rho_i}(\pi_{r_i}, \pi_{N+1})$ is not consistent, which means, by virtue of Lemma 1 (see below), that $SDRS_{M+1,N+1}(i) = SDRS_{MN} \bigcup \{\pi_{N+1}, \rho_i(\pi_{r_i}, \pi_{N+1})\}$ is not coherent. In this case, that rhetorical relation $\rho_i(\pi_{r_i}, \pi_{N+1})$ is discarded.

A point very important to be emphasized is that, unlike it might have been suggested in Section 2.3, in contextually filtering the rhetorical relations we need not consider the abstract attribute grammar associated to the *entire* derivation of the rhetorical relation, by expanding each discourse predicate. This is the case, since in SDRT-based local coherence analysis each rhetorical relation has already been validated, hence each discourse predicate in its structure has been found to hold. In contextual filtering, these predicates can be taken as *facts* and all we need to do is check that the elements in the restriction sets for each global variable are unifiable; each rhetorical relation is thus represented through only *one* derivation rule (along with the restriction sets associated to the global variables that occur in the rule) in the abstract attribute grammar of the entire discourse, namely the derivation rule that corresponds to the Prolog rule that implements the semantics of that rhetorical relation. Therefore, at step 2 in the procedure shown above, only rules (1) along with attribute sets $(1')$ are relevant; rules and attribute sets from (2) to (K), respectively from $(2')$ to (K') should be ignored.

Until now, nothing was said about the role of the derivation rules in the abstract attribute grammar of the discourse, in filtering the rhetorical relations. Actually, these rules play a minor part in the filtering process; they can serve only as a way of verifying the usage of the semantics of the rhetorical relations: if several (production) rules have identical heads, they must also have identical bodies, since each such rule is an abstraction of the Prolog clause that implements the semantics of a rhetorical relation, thus each head of a rule represents a type of rhetorical relation; by consequence, the bodies should represent the semantics of the rhetorical relations.

A key extension that the framework described here brings with respect to previous work reported in Deransart and Małuzýnski (1993) and Isakowits (1991) concerning the conversion from logic programs to abstract attribute grammars consists in the notion of *global variable* in a logic program. In the description of the contextual filtering of the rhetorical relations we glossed over this notion, but here we mention that making certain variables "global", that is, so that they keep their meaning and assignment from one rule of the abstract attribute grammar to another, we open the path towards a formalization of a "syntactic" consistency of the grammar. Thus, the unifiability of all the restriction sets that correspond to each global variable in the grammar is an expression of the consistency of this grammar. This is true, because each element in a restriction set actually represents a *path* to that variable; and if the variable keeps its value from one rule to another, then the elements in its restriction sets should be unifiable. Then, by making global variables representing utterances in discourse, unifiability of their associated restriction sets basically boils down to not having contradictory assertions (a.k.a. discourse predicates) concerning these utterances (a.k.a. global variables).

For example, we cannot have at the same time predicates such as: $\mathsf{equals}(\chi, \mathsf{topic}(\alpha))$ and $\mathsf{equals}(\chi, SARG(\alpha))$ for the global variables χ and α, since these predicates generate the following "paths" towards α in the restriction sets: $\lambda_2^{\mathsf{equals}} \circ id(\mathsf{topic}(*))$ and $\lambda_2^{\mathsf{equals}} \circ id(SARG(*))$ respectively, where $*$ stands for the "depth" of the predicates in the rules that contain them; since these depths are *not* relevant in the unification process, we have substituted them with a wildcard. The unification between the two "paths" is not possible since they are equivalent to $\mathsf{topic}(\lambda_2^{\mathsf{equals}})$ and $SARG(\lambda_2^{\mathsf{equals}})$ and these cannot be equal, since $\mathsf{topic}/1$ returns a *set* of initialized variables (in the utterance taken as argument), whereas $SARG/1$ returns a *clause* that contains the action communicated in the utterance. An apparent issue with this argument is that if, for the example above, χ is not a global variable anymore, then, although the restriction set of α is the same, it is perfectly valid that $\mathsf{topic}(\alpha)$ unifies with an instance of the local variable χ, while $SARG(\alpha)$ unifies with another instance of the local variable χ, in another rule. The answer is that, when χ is local, given the fact that the predicate $\mathsf{equals}/2$ is just a renaming of the Prolog predicate $== 2$, $\lambda_2^{\mathsf{equals}}$ is *by default* bound to $\lambda_1^{\mathsf{equals}}$ and this in its turn is local, then there is no constraint in constructing the restriction set for α.

Finally, we should remark that in the case of rhetorical relations whose semantics contain the $\vee$ operator (hence, whose logic programs contain ';' and abstract grammars contain |), the attribute sets computed (and due to be unified) correspond *only* to the "branches" in the disjunction that were actually deemed valid by the SDRT-based local coherence analysis.

Via the procedure sketched above we operate a contextual (since all the discourse structure is taken into account) filtering (since the output of the procedure consists *at most* of the input set of rhetorical relations) of potentially valid rhetorical relations between a pair of utterances in dialogue; what we need to do now is to show that the approach is theoretically relevant for rhetorical structure updating in language generation for dialogue.

3.2 Theoretical Foundations

In this section we will show that the contextual filtering procedure previously presented is formally correct, that is, it is sound and complete. In other words, we will show that any discourse structure that is coherent (according to Definition 1) has an abstract attribute grammar that is "unifiable" (i.e. where all the restriction sets associated to every global variable can be unified) – completeness, and that any discourse structure that has an "unifiable" attribute grammar is coherent – soundness. In order to prove the theoretic correctness of the contextual filtering algorithm proposed, we first need the following:

Lemma 1. *For a segmented discourse structure to be coherent, it is necessary that the logic program obtained by concatenating the semantics of the rhetorical relations according to the discourse structure, is consistent.*

Proof. We prove the lemma by assuming the contrary, i.e. that exists a coherent discourse structure such that its corresponding logic program is not consistent.

If this logic program is not consistent, then it cannot succeed, hence the discourse structure cannot be computed as a consequence of the program.

But this contradicts Definition 1 and the assumption that the discourse structure is coherent. □

This lemma is helpful in proving the main result concerning contextual filtering:

Proposition 1. *The contextual filtering procedure is sound and complete.*

Proof. We first prove completeness (i.e. that any coherent discourse structure has a "unifiable" abstract attribute grammar).

For this, we assume the contrary, namely that there exists a coherent discourse structure that has an abstract attribute grammar which contains a global variable whose restriction sets cannot be unified. In this case, it results from the construction of the abstract attribute grammar from the logic program (see Section 2.3) (and from Lemma 2 – see below) that the logic program that the grammar originated from is not consistent.

Hence, we have a coherent discourse structure that has an inconsistent logic program associated, which contradicts Lemma 1.

For proving soundness (i.e. that any discourse structure that has a "unifiable" abstract attribute grammar is coherent), we proceed by assuming the contrary, namely that there exists a non-coherent discourse structure with a "unifiable" abstract attribute grammar.

From Definition 1, this means that this discourse structure is not computable via a logic program, using the semantics of the rhetorical relations. But, on the other hand, the abstract attribute grammar that corresponds to this discourse structure is "unifiable", thus the corresponding logic program is consistent.

Therefore, we have a consistent logic program that, however, does not succeed for a particular assignment to its global variables (that is, the actual utterances). This means that there is at least one (Prolog) rule in this program that fails, which boils down to the existence of at least one predicate applied to a global variable that (for a particular assignment of this variable) fails. This implies that the rhetorical relation whose semantics contains this predicate cannot hold for the particular assignment of the variable concerned.

On the other hand, this rhetorical relation resulted as valid according to the SDRT-based local coherence analysis, hence this rhetorical relation holds for the particular assignment of the (global) variable concerned. Thus, we have a contradiction.

In conclusion, the fact that SDRT-based local coherence analysis precedes the contextual filtering ensures the soundness of the latter. □

The proof of Proposition 1 tacitly assumed that there is a surjective mapping from the set of logic programs to the set of attribute grammars, i.e. that for any logic program there exists a unique abstract grammar (the functional character of the mapping from logic programs to abstract attribute grammars) and several logic programs can lead to the same attribute grammar (the surjectivity of the mapping from logic programs to abstract attribute grammars). From this surjective mapping we have that an abstract attribute grammar can be viewed as an *equivalence class* (modulo input / output or, equivalently and respectively "inherited"/"synthesized" variable assignments) for a set of logic programs. From this fact, we can infer that the non-unifiabiliy of any set of restriction sets for any global variable in an abstract attribute grammar yields the inconsistency of all the logic programs in the same equivalence class induced by the abstract attribute grammar under discussion.

Hence, we have to prove that the mapping from logic programs to abstract attribute grammars is a surjective function, which is the result of:

Lemma 2. *The mapping from logic programs to abstract attribute grammars is a surjective function.*

Proof. We can assume, without restraining the generality of the problem, that a logic program consists in a clause of the form:
$q^0(q_1^1(q_1^2(...(q_1^D(X_1,...,X_{N_{D_1}})...)...)...)...q_N^1(...q_M^1(...q_P^E(X_1,...,X_{M_{E_P}})...)...)...) :-$
$p_1^0({}_1p_1^1({}_1p_1^2(...({}_1p_1^F(X_1,...,X_{N_{F_1}})...)...)...)...)\mu(\xi_1)...$
$\mu(\xi_{Q-1})p_Q^0({}_Qp_1^1(...{}_Qp_M^2(...(...{}_Qp_P^G(X_1,...,X_{M_{G_P}})...)...)...)...).$

In this formula, the upper indexes $0, 1, 2, ..., D, E, F, G$ denote the nesting depths of the predicates and variables, while X_i denote the variables in the program.

From this (Prolog) rule we can derive an abstract attribute grammar, that contains one rule and a set of restriction sets, one restriction set for each variable. Hence, these elements are shown below:

(i) the rule: $q^0 \rightarrow p_1^0\tau(\mu(\xi_1))...\tau(\mu(\xi_{Q-1}))p_Q^0$;

(ii) the restriction sets for the variables (there exist integers T and R that satisfy the following expressions and $i \in \{1,...,\max(N_{D_1},...,M_{E_P},...,N_{F_1},...,M_{G_P})\}$): $X_i : \{\lambda_1^{q^0} \circ \lambda_1^{q_1^1} \circ \ldots \circ \lambda_1^{q_1^{D-1}} \circ id(q_1^D.i(\epsilon)),...,\lambda_N^{q^0} \circ \lambda_M^{q_M^1} \circ \ldots \circ \lambda_T^{q_T^{E-1}} \circ id(q_P^E.i(\epsilon)), \lambda_1^{p_1^0} \circ \lambda_1^{{}_1p_1^1} \circ \ldots \circ \lambda_1^{{}_1p_1^{F-1}} \circ id({}_1p_1^F.i(1)),...,\lambda_1^{{}_Qp_1^1} \circ \lambda_M^{{}_Qp_M^2} \circ \ldots \circ \lambda_R^{{}_Qp_R^{G-1}} \circ id({}_Qp_P^G.i(Q))\}$.

From the construction above, it is evident that, structurally, the abstract attribute grammar is uniquely determined by the rule and the family of restriction sets, therefore, for the given logic program, it is unique. Thus, the first part of the lemma has been proved, namely that the mapping from logic programs to abstract attribute grammars is a function.

As for the surjective character of the mapping, we need to show that there can be at least two logic programs that have the same abstract attribute grammars and for any logic program there exists a grammar. The last part of this assertion is evident, from the fact that the abstract attribute grammar is mechanically constructed from the logic program, regardless of the particular nature of the latter. For the first part of the assertion, let us have two logic programs, consisting, each one, of one rule of the type above. For the ease of representation, we assume that the programs are LP and $\overline{LP}$, where LP is exactly as shown above, and $\overline{LP}$ has the form:
$\overline{q^0}(\overline{q_1^1}(\overline{q_1^2}(...(\overline{q_1^D}(X_1,...,X_{\overline{N_{D_1}}})...)...)...)...\overline{q_N^1}(...\overline{q_M^1}(...\overline{q_P^E}(X_1,...,X_{\overline{M_{E_P}}})...)...)...) :-$
$\overline{p_1^0}(\overline{{}_1p_1^1}(\overline{{}_1p_1^2}(...(\overline{{}_1p_1^F}(X_1,...,X_{\overline{N_{F_1}}})...)...)...)...)\mu(\overline{\xi_1})...$
$\mu(\overline{\xi_{Q-1}})\overline{p_Q^0}(\overline{{}_Qp_1^1}(...\overline{{}_Qp_M^2}(...(...\overline{{}_Qp_P^G}(X_1,...,X_{\overline{M_{G_P}}})...)...)...)...).$

The corresponding abstract attribute grammar,$\overline{AAG}$ consists of:

(i) the rule: $\overline{q^0} \rightarrow \overline{p_1^0}\tau(\mu(\overline{\xi_1}))...\tau(\mu(\overline{\xi_{Q-1}}))\overline{p_Q^0}$;

(ii) the restriction sets for the variables: $X_i : \{\lambda_1^{\overline{q^0}} \circ \lambda_1^{\overline{q_1^1}} \circ \ldots \circ \lambda_1^{\overline{q_1^{D-1}}} \circ id(\overline{q_1^D}.i(\epsilon)), ..., \lambda_N^{\overline{q^0}} \circ \lambda_M^{\overline{q_M^1}} \circ \ldots \circ \lambda_T^{\overline{q_T^{E-1}}} \circ id(\overline{q_P^E}.i(\epsilon)), \lambda_1^{\overline{p_1^0}} \circ \lambda_1^{\overline{{}_1p_1^1}} \circ \ldots \circ \lambda_1^{\overline{{}_1p_1^{F-1}}} \circ id(\overline{{}_1p_1^F}.i(1)), ..., \lambda_1^{\overline{{}_Qp_1^1}} \circ \lambda_M^{\overline{{}_Qp_M^2}} \circ \ldots \circ \pi_R^{\overline{{}_Qp_R^{G-1}}} \circ id(\overline{{}_Qp_P^G}.i(\overline{Q}))\}$.

The two grammars, AAG (for LP) and $\overline{AAG}$ (for $\overline{LP}$) are identical if and only if their rules are identical *and* their attribute sets can be unified, for each variable. Identical rules, in the case considered here, boil down to the identity of functors and arities for the predicates of depth 0, and to the identities of the connectors, that is, for example, adopting a Prolog syntax, functor($\overline{q^0}$, _f, _a) and functor(q^0, _f, _a) are simultaneously true for the same bindings for _f and _a, and $\overline{\xi_i} \equiv \xi_i$, for i from 1 to $Q-1$. Unifiable attribute sets mean that each pair of terms in the union of the attribute sets of each variable and each of the two programs can be unified. However, the construction of each abstract attribute grammar and the assumption that each logic program is consistent ensure that the interesting unification cases to be considered concern pairs of elements such that each one is in a different restriction set for one variable. More precisely, denoting by $RS(X_i)$ the restriction set of variable X_i in the grammar AAG and by $\overline{RS}(X_i)$ the restriction set of the same variable in the grammar $\overline{AAG}$, we should inspect only equations of the form $Y = Z$, where $Y \in RS(X_i)$ and $Z \in \overline{RS}(X_i)$. For example, we should have that[6]:

$\lambda_N^{q^0} \circ \lambda_M^{q_M^1} \circ \ldots \circ \lambda_T^{q_T^{E-1}} \circ id(q_P^E.i(\epsilon)) = \lambda_N^{\overline{q^0}} \circ \lambda_M^{\overline{q_M^1}} \circ \ldots \circ \lambda_T^{\overline{q_T^{E-1}}} \circ id(\overline{q_P^E}.i(\epsilon))$, which is equivalent to (see Section 2.3 for the definition of the composition of selector functions):

$q_P^E.i(\lambda_T^{q_T^{E-1}}(...(\lambda_M^{q_M^1}(\lambda_N^{q^0}(\epsilon))))...) = \overline{q_P^E}.i(\lambda_T^{\overline{q_T^{E-1}}}(...(\lambda_M^{\overline{q_M^1}}(\lambda_N^{\overline{q^0}}(\epsilon))))...)$.

Thus, it is only necessary that the compositions of several functions are identical; this need not determine that all functions involved in the compositions are respectively identical. If all the functions involved in the compositions were respectively identical, then the two logic programs would have been identical; since this is not necessarily the case, we have a class of programs that lead to the same abstract attribute grammar. □

Lemma 2 opens the path towards investigating the conditions that a set of logic programs should accomplish in order to be in the same equivalence class, i.e. to have the same abstract attribute grammar derived from them. In the literature (Deransart and Małuzýnski, 1993), (Isakowits, 1991) it is usually stated that an abstract attribute grammar accounts for a class of syntactically identical programs, but all possible input/output variable assignments. Obviously, this is true, but, as we have seen from the proof of Lemma 2, this is not necessary. Therefore, the identity of the logic programs, except for the "inherited"/"synthesized" variable splitting is a sufficient, but non-necessary, condition for a set of logic programs to lead to the same abstract attribute grammar. Hence, finding necessary conditions for two programs to lead to the same abstract attribute grammar is a topic of further research.

3.3 Discussion

The contextual filtering of rhetorical relations in discourse structure updating, previously presented, has been rather extensively described and theoretically grounded. However, there are some aspects that worth further discussion.

First of all, we should investigate what happens to the contextual filtering process when no prior dialogue history is available. More precisely, what is the effect of contextual constraints on the rhetorical relations between two utterances, when the latter are the first two in dialogue? The (obvious) answer is that contextual filtering has no effect in this case, because (i) the context is reduced to the two utterances under discussion, (ii) the only rule in each discourse-level abstract attribute grammar consists in each rhetorical relation that had been deemed as valid by the SDRT-based local coherence analysis (hence, the attribute sets of the "global" variables concerned – actually, the two utterances, are, by necessity, unifiable).

However, a less evident answer to this question is that contextual filtering can play a verification role, in that if one finds non-unifiable elements in the restriction sets of the two global variables, this means that the SDRT-based local coherence analysis was not performed correctly; assuming that the correctness

[6] Actually, from the manner whereby the abstract attribute grammars are derived and from the assumption that the two logic programs are consistent, it suffices to show that we can unify only *one* pair of elements in restriction sets that belong to different grammars.

of the inferential process is ensured (e.g. by using Prolog), the incorrectness of the SDRT-based local coherence analysis can hint for inconsistencies in the encyclopedic knowledge base. Nevertheless, given that the discourse predicates are independent of the particular domain, we can conclude that contextual filtering can serve in detecting flaws in the particular domain knowledge base considered.

Moreover, since, in the case where no dialogue context is actually available, contextual filtering has no effect on the set of relevant rhetorical relations between a pair of utterances, the first two utterances are essentially connected (except for possible inconsistencies in the knowledge base – see above) through the rhetorical relations yielded by the SDRT-based local coherence analysis, therefore, through, possibly, more than one discourse relation.

A second and subtler issue regarding contextual filtering concerns another "limit" discourse update situation, where SDRT-based local coherence analysis yielded at least one valid rhetorical relation between a pair of utterances (or discourse constituents), contextual filtering yields no (relevant) rhetorical relation between these utterances and, consequently, the last utterance in discourse is not connected to any previous utterance (or constituent) in that discourse.

In this case, if one adopts the heuristics of penalizing "shorter" discourse structures (i.e. with a lower number of utterances) in favor of "longer" ones, then, if a "hiatus" occurs in the discourse structure (i.e. we have an utterance not connected to any previous utterance in the same discourse), we can *backtrack* towards the last previous "point" in discourse (by "point" in discourse, we mean a pair of rhetorically-connected utterances or discourse constituents) where more than one rhetorical relation had passed contextual filtering.

The rhetorical relation that had been actually chosen at that "point" in discourse (before the backtracking phase) was the one with the lowest computational cost (i.e. that passed numerical filtering – see below in Section 4); now, if backtracking is needed (by the "hiatus" mentioned above), then, at that "point" in discourse where several alternative rhetorical relations were available, we choose the rhetorical relation with the second lowest computational cost and we *re-apply* contextual filtering on all the utterances from the "point" under discussion, until the utterance where the "hiatus" occurred.

An example of dialogue with a hiatus, between two speakers U_1 and U_2 in the context of reserving a book in a library, is (we have marked by $\vdots$ the "hiatus"):

U_1: Hello, what can I do for you?

U_2: Well, I would like to have a book on Astronomy...

U_1: You want an advanced book, or rather an introductory text?

$\vdots$

U: What's the weather like on Mars?

Now, three possibilities arise:

(i) the last utterance occurring in discourse gets connected to some of the previous ones and the "hiatus" is thus ruled out;

(ii) the hiatus is maintained at the same "point" in discourse, i.e. involving the last utterance due to be connected to the discourse structure;

(iii) another "hiatus" occurs earlier in discourse, that is, for an utterance produced earlier than the last utterance in the whole discourse (where the "hiatus" originally occurred), but later than the last utterance in the discourse "point" where we backtracked to.

The second situation shows that backtracking and modifying a discourse "point" does not bring any advantage to the discourse updating process, whereas the third one shows that backtracking worsens things, yielding even shorter non-interrupted discourse structures. Thus, if situation (ii) occurs, then we keep the modification operated at that discourse "point" and we backtrack even further to an earlier discourse "point" where several alternative rhetorical relations had passed contextual filtering, and to this "point" we change the rhetorical relation, to the second last costly relation, then we re-apply contextual filtering to the rest of utterances and rhetorical relations from this "point" on, and so on.

If situation (iii) occurs, then we *discard* the modification made at the latest discourse "point" earlier than the last utterance and either try another alternative discourse relation at the same point and re-apply contextual filtering from that "point" on, or, if there is no more alternative discourse relation, we backtrack to an even earlier discourse "point" where alternative rhetorical relations exist and we proceed as in situation (ii). If situation (i) occurs, then, obviously, this is what we wanted and we can proceed with the discourse structure updating.

For a more precise account of this "hiatus coping" strategy, we assume that in a discourse structure we have several utterances, denoted by π_i, where i varies from 1 to N, the most recently produced utterance. Furthermore, let us assume that, except for the initial pair of utterances (where we recall that contextual filtering does not result in any actual filtering on rhetorical relations), we have two discourse "points" where alternative rhetorical relations are deemed to hold, according to the SDRT-based local coherence analysis and to the contextual filtering procedure; let us mark these "points" by utterances π_i and π_j (the first "point") and utterances π_l and π_m (for the second "point"). Let us further assume, without restraining the generality of the account, that at each of these two discourse "points" we have two alternative rhetorical relations that passed contextual filtering: ρ_k and ρ'_k for the first discourse "point", and ρ_n and ρ'_n for the second discourse "point". Moreover, we will mark by $\rho_{\dots}$ rhetorical relations whose particular identities (types) are not relevant for the current discussion, by N the index of the last utterance in discourse, by Δ, a natural number greater than or equal to 2 so that $N - \Delta > m$, and by ρ_M a discourse relation that is eventually (in situation (i)) deemed valid between utterances π_{N-1} and π_N.

With these notations, we illustrate the "hiatus coping" strategy in Figure 1, where we marked by dotted arrows alternative rhetorical relations not yet selected, by continuous arrows, rhetorical relations currently selected, by crossed continuous arrows rhetorical relations previously selected, but discarded, by non-arrow arcs accompanied by "?" we mean that there could be some rhetorical relations between the utterances concerned, but this has to be computed, and by da-dotted arrows we mean rhetorical relations that might not hold anymore, due to changes in rhetorical relations at previous discourse "points". In Figure 1 we thus represented the "hiatus coping" strategy, emphasizing the initial "hiatus" situation, two backtracking stages, for situations (ii) and (iii), and the "ideal" situation (i). We remark that if, after a backtracking one does not mitigate the "hiatus" and if previous discourse "points" with alternative rhetorical relations are available, subsequent backtracking processes take place, aiming to arrive in situation (i).

However, this situation of "hiatus" (that is, where an utterance that naturally is part of a given dialogue cannot be rhetorically connected to any previous utterance or discourse constituent in that dialogue) seems to seldom occur in practice; in the vast majority of *real* dialogues investigated at present, we actually did not find any utterance that could not be rhetorically connected to previous utterances (or discourse constituents) in the dialogue it is part of. Nevertheless, the discussion above is important when very ambiguous utterances occur in dialogues, or utterances strongly accompanied by multimodal support (such as gestures), or even ironical utterances coming from the user. For example, let us assume the following dialogue between two speakers (a librarian – U_2 and a customer in the library – U_1):

U_1: Hello, may I have some books on algebraic geometry?

U_2: Hello, sir, is it an intermediary or an advanced book that you like?

U_1: Well, let's start with an intermediary one and then we'll see...

U_2: OK, I can give you the book "Foundations of Algebraic Geometry" by A. River, it is a very thorough and formal, yet accessible introduction to the field.

U_1: $*$ Well, do we often get drown there?

Obviously, the last utterance of U_1 is difficult to interpret, thus it is difficult to rightfully find a rhetorical relation that connects it to the rest of the dialogue. Actually, it strongly depends on the domain encyclopedic

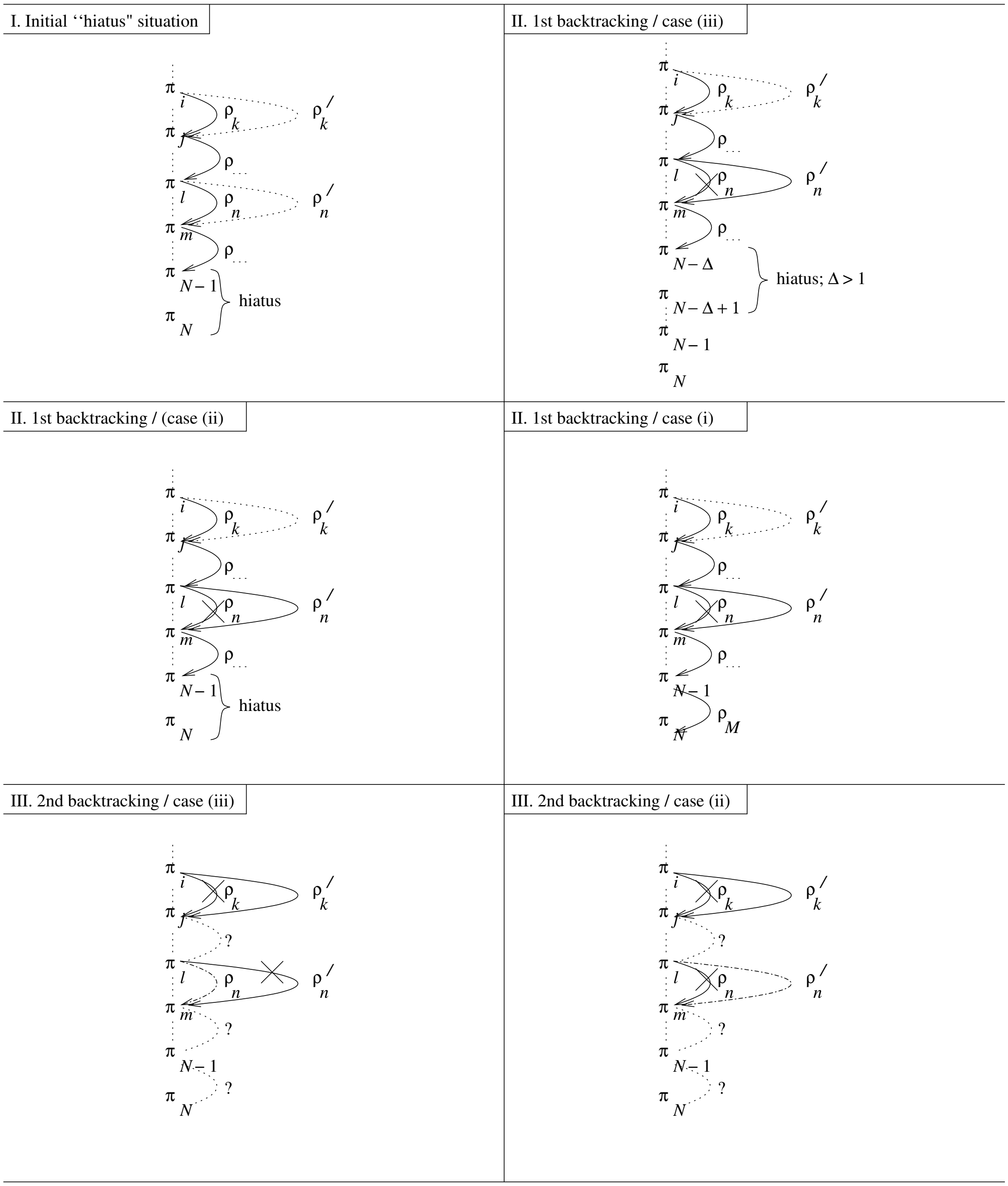
I. Initial ‘‘hiatus" situation
II. 1st backtracking / case (iii)
II. 1st backtracking / (case (ii)
II. 1st backtracking / case (i)
III. 2nd backtracking / case (iii)
III. 2nd backtracking / case (ii)
π i
π j
π l
π m
π N − 1
π N
π N − Δ
π N − Δ + 1
ρ k
ρ′ k
ρ n
ρ′ n
ρ ...
ρ M
hiatus
hiatus; Δ > 1
?

Figure 1: Discourse “hiatus coping” strategy.

knowledge base (Asher and Lascarides, 2003) if we can do that or not; for instance, if we take this utterance as a blatant aggressive irony related to the name of the author, then we look at this utterance either as a confirmation of acceptance of the book, or, on the contrary, as a sign that the book is of little interest with respect to, for instance, the person of the librarian.

Thus, apparently U_2 could infer either one of these two interpretations, or discard the utterance or, furthermore, take it as an encoded question on, for example, the number of exercises included in the book (if we often get drown there, then there are many exercises). If the last utterance from U_1 is to be interpreted as a blatant irony with respect to the person of the librarian for instance, then the first utterance should be seen rather in a *P-Corr* relation with the second utterance (whereby the librarian tries to dissuade the customer from behaving ironically), than in a $Elab_q$ relation with the same (second) utterance, as it might seem at a first sight. Moreover, instead of looking at the pair between the first utterance of U_2 and the second utterance of U_1 as connected through a *QAP* or *P-Elab* relation, we can look at them as connected through a *P-Corr* relation, whereby U_1 enforces its irony with respect to the librarian. However, then, the librarian does not seem to have interpreted the customer's utterance in this way, since it tries to produce an utterance (the second one of U_2) in a *P-Elab* relation with respect to the second utterance of U_1.

In this setting, U_2 can now interpret the last utterance of U_1 as in a (monologue) relation of *Elaboration* with the second utterance from the same speaker, instead of looking at it as in a $Background_q$ relation with the last utterance of U_2. Hence, we see that sometimes the exploration (and backtracking on) alternative rhetorical relations between utterances might help enforce the coherence that a dialogue participant is aware of, for a given (undergoing) dialogue. However, perhaps in real-life situations such a dialogue could have never taken place, since the irony of U_1 would have been conspicuous, due to multimodal factors, such as look, gesture and prosody.

Finally, we should point out a minor but technically important detail regarding the "hiatus coping" strategy described above: in choosing among several rhetorical relations at a discourse "point" we rely on the numerical filtering algorithm; the precise manner whereby we do this resides in *sorting* in increasing manner the alternative rhetorical relations in each discourse "point", with respect to the computational score, computed as shown in Section 4. Thus, even if the numerical filtering stage is not very important *per se*, it helps in guiding the exploration of alternative rhetorical relations when backtracking on discourse "points" in order to attempt to mitigate "hiatuses" in discourse.

4 Numerical Filtering of Rhetorical Relations

To each abstract attribute grammar obtained from the semantics of a rhetorical relation we associate a computational cost, trying to assess the load required to compute that relation between two utterances (available, remember, in logical form). Thus, for a given rhetorical relation ρ, we compute the following quantities:

- $|AAG_\rho|$ ::= total number of production rules in the AAG associated to ρ;
- $|AAG_\rho^{(N)}|$ ::= total number of production rules that contain non-terminals in the right-hand side;
- for any rule (r) in the AAG associated to ρ, $|Var_\rho(r)|$ ::= total number of distinct variables (i.e. of attribute sets) in rule (r);
- for any rule (r) in the AAG associated to ρ, and for any distinct variable $\_\nu$ in $Var_\rho(r)$, $|Attr_\rho^{(r)}(\_\nu)|$ ::= size (i.e. number of elements) of the restriction set associated to variable $\_\nu$ in rule (r) in the AAG associated to rhetorical relation ρ;

These quantities are not used in order to derive an absolute "value" for the cost, but, instead, allow one to assess *relative* loads, that is, by comparing two or several rhetorical relations, applied to the same pair of utterances or discourse constituents. Here, we will only informally illustrate the intuitions behind the idea; in the next section an algorithm in this respect will be formally specified. Thus, considering that we have two rhetorical relations ρ and σ connecting the same pair of utterances, we have the following *comparison* procedure:

(i) if $|AAG_\rho| = |AAG_\sigma|$, then move on to the next comparison step; else, choose the relation with the minimal number of derivation rules;

(ii) if $\frac{|AAG_\rho^{(N)}|}{|AAG_\rho|} = \frac{|AAG_\sigma^{(N)}|}{|AAG_\sigma|}$, then move on to the next step; else choose the relation with the minimum ratio between the number of rules with non-terminals in the right-hand side and the total number of rules; the intuition behind this step is that rules with non-terminals in the right-hand side need further derivations in order to "succeed", thus they induce a certain *future* supplementary load;

(iii) if for all the rules (r) in AAG_ρ and (s) in AAG_σ, we have that $|Var_\rho(r)| = |Var_\sigma(s)|$, then move on to the next comparison step; else, choose the rhetorical relation whose AAG has the property that $\sum_{(r)\in AAG} |Var(r)|$ is minimal; the intuition behind this step is that more variables in a rule yield more unification processes due to be executed, which induce a further load;

(iv) if for all the rules (r) in AAG_ρ and (s) in AAG_σ and all the variables $\_\nu$ in $Var_\rho(r)$ and $\_\omega$ in $Var_\sigma(s)$, we have that $|Attr_\rho^{(r)}(\_\nu)| = |Attr_\sigma^{(s)}(\_\omega)|$, then randomly choose one of the two rhetorical relations[7]; else, choose the relation whose AAG has the property that $\sum_{(r)\in AAG} \sum_{\_\nu \in Var(r)} |Attr^{(r)}(\_\nu)|$ is minimal; the intuition behind this is that larger restriction sets yield a larger number equalities (between the members of these sets) to be checked for, hence a larger number of unification processes, that determine a supplementary computational load.

The inputs to the procedure for assessing the computational cost of the rhetorical relations consist in: (i) a pair of utterances (specified in logical form), connected via strictly more than one rhetorical relation; (ii) the types (names) of the rhetorical relations that connect the pair of utterances. The output of the algorithm consists in strictly one (type of) rhetorical relation, included in the set given as input (ii).

The algorithm numerically filters rhetorical relations between each pair of utterances or discourse constituents; thus, for each new utterance, one computes the set of rhetorical relations that hold between previous utterances (or discourse constituents) and itself.

5 Extensive Example

As an example, we will analyze a fragment from Scene six, Act I from Eugène Labiche's vaudeville "The Jackpot" (≪ La cagnotte ≫)[8]. The text listed below is a translation from the original in the French language; Blanche and Léonida are the characters in this dialogue. The analysis will assess the contextual and numerical filtering procedures described in this paper. Thus, assuming that the SDRT apparatus computes a set of rhetorical relations[9] between each pair of utterances[10] in a dialogue, we apply contextual filtering in order to rule out ambiguous situations where several relations had been computed between a pair of utterances. If, after this filtering, ambiguous situations still exist, that is, if there exists a pair of utterances with at least two rhetorical relations that connect them, then the numerical filtering algorithm presented in this paper is applied, in order to rule out some of these relations, so that each pair of utterances is connected through at most one rhetorical relation and there exists a path in the graph whose nodes are the utterances and whose edges are the rhetorical relations between these utterances.

The dialogue is listed below:

Blanche: Oh! My aunt!$^{\pi_1^1}$ If you only knew how happy I am!$^{\pi_1^2}$

Léonida: Right...$^{\pi_2^1}$

Blanche: Mr. Felix just asked me from my dad.$^{\pi_3^1}$ And dad told him to keep hope...$^{\pi_3^2}$

Léonida: What, you love Mr. Felix?!?$^{\pi_4^1}$

We shall present in the sequel a trace on the discourse updating process, as each new dialogue turn becomes available to the rhetorical relations filtering procedure. For this, we "model" both characters, in that rhetorical structuring is performed for speech turns coming from both Blanche and Léonida. Thus, we actually show how discourse structure updating is dynamically performed. Here, we will show: 1) the rhetorical relations having been yielded by the SDRT-based coherence analysis, and 2) the *details* involved in contextual filtering.

First, Blanche produces a speech turn composed of utterances π_1^1 and π_1^2. The SDRT-based local coherence analysis, by trying each rhetorical relation in terms of its semantics (see Section 2.1), and of the semantics of the utterances, retains $Elaboration(\pi_1^1, \pi_1^2)$ and $Consequence(\pi_1^1, \pi_1^2)$. Contextual filtering should theoretically not impose any further constraint on the two rhetorical relations that connect these first two

[7] However, from the way the semantics of the rhetorical relations were defined and from the splitting between monologue and dialogue rhetorical relations – only relations in one group are checked for a given pair of utterances, this situation does not occur.

[8] The electronic version of the play was downloaded from `http://fr.wikisource.org`.

[9] This set can also be void in some cases.

[10] In this paper, each time the construction "pair of utterances" is used, it is meant that one or both members of such a pair can be either an utterance or a discourse constituent, i.e., an SDRS.

utterances, since no previous dialogue history is available. However, contextual filtering will be run and its functioning will be shown in detail, in order to detect the possible flaws in the SDRT-based local coherence analysis (see Section 3.3). Thus, the restriction sets for π_1^1 and π_1^2 are, for *Elaboration*:[11]

$Elaboration(\pi_1^1, \pi_1^2)$:

π_1^1: $\{id(Elab.1(\epsilon)), id(\mathsf{enounce}(1)), \lambda_2^{\mathsf{SubclassOf}} \circ id(\mathsf{topic}(3))\}$;

π_1^2: $\{id(Elab.2(\epsilon)), id(\mathsf{enounce}(2)), \lambda_1^{\mathsf{SubclassOf}} \circ id(\mathsf{topic}(3))\}$.

We can see that all the elements in each restriction set are unifiable, hence the SDRT-based local coherence analysis was not flawed. For the *Consequence* relation between these utterances, the restriction sets are:

$Consequence(\pi_1^1, \pi_1^2)$:

π_1^1: $\{id(Consequence.1(\epsilon)), id(\mathsf{enounce}(1)), \lambda^{\mathsf{neg}} \circ id(K(3))\}$;

π_1^2: $\{id(Consequence.2(\epsilon)), id(\mathsf{enounce}(2)), id(K(3))\}$.

We can see that no bogus situation is signaled neither, therefore the *Consequence* relation holds as well.

Secondly, Léonida produces utterance π_2^1. SDRT-based local coherence analysis yields as valid the dialogue rhetorical relations $ACK(\pi_1^1, \pi_2^1)$ and $ACK(\pi_1^2, \pi_2^1)$. These relations should be further validated by the contextual filtering procedure; they introduce new restriction sets for π_1^1, π_1^2 and π_2^1:

$ACK(\pi_1^1, \pi_2^1)$ and $ACK(\pi_1^2, \pi_2^1)$:

π_1^1 and π_1^2: $\{id(ACK.1(\epsilon)), id(\mathsf{enounce}(1)), \lambda_1^{\neg\mathsf{Disjoint}} \circ id(\mathsf{topic}(3)), \lambda_1^{\mathsf{confirmation}} \circ id(SARG(4))\}$;

π_2^1: $\{id(ACK.2(\epsilon)), id(\mathsf{enounce}(2)), \lambda_2^{\neg\mathsf{Disjoint}} \circ id(\mathsf{topic}(3)), id(\mathsf{confirmation.2}(4))\}$.

Hence, first assuming that *Elaboration* holds between π_1^1 and π_1^2, we see that, since π_1^1 and π_2^1 have the same restriction set as yielded by the ACK relation to π_2^1 but, on the other hand, the restriction set of these two utterances should unify with their restriction sets as yielded in the preceding discourse updating process, that is by the *Elaboration* relation between them. This implies that restriction sets of these utterances, as yielded by the *Elaboration* relation between them, should unify, therefore, $\lambda_1^{\mathsf{SubclassOf}} \circ id(\mathsf{topic}(3))$ should unify with $\lambda_2^{\mathsf{SubclassOf}} \circ id(\mathsf{topic}(3))$, which boils down to the fact that *both* arguments of the $\mathsf{SubclassOf}/2$ predicate should be identical, which contradicts the semantics of this predicate which, however, was deem to hold by the SDRT-based local coherence analysis. Therefore, at least one of the relations $ACK(\pi_1^1, \pi_2^1)$ and $ACK(\pi_1^2, \pi_2^1)$ does not hold. Either relation we exclude, we ensure the syntactic coherence of the current discourse structure (because the restriction sets become unifiable – there is no problem in unifying the elements in the restriction sets of either one of π_1^1 and π_1^2 as yielded by the *Elaboration* relation between them, with either one of the restriction sets of π_1^1 and π_1^2, as yielded by the ACK relation between either one of these utterances and π_2^1); thus, contextual filtering does not further guide which choice to make. In that case, we adopt the heuristics that penalizes connections to older utterances, in favor of connections to more recent ones: if an utterance can be connected (with no further restrictions) to either one of two utterances (but not to the two of them simultaneously), we choose the connection to the most recently-produced one; thus, in our case, we retain $ACK(\pi_1^2, \pi_2^1)$.

Moreover, if we assume that the *Consequence* relation holds[12] between utterances π_1^1 and π_1^2, then, following the same argument as above, if we accepted both $ACK(\pi_1^1, \pi_2^1)$ and $ACK(\pi_1^2, \pi_2^1)$, then we would have to unify $id(K(3))$ with $\lambda^{\mathsf{neg}} \circ id(K(3))$, which is equivalent to unifying $K(3)$ with $K(\lambda^{\mathsf{neg}}(3))$, which means that one should find an entity whose semantic content coincides with the semantic content of the negation of the same entity, which, in our setting, is impossible. Hence, one of the two ACK relations has to be ruled out; then, we follow the same argument as above.

Thirdly, Blanche produces utterances π_3^1 and π_3^2. SDRT-based local coherence analysis yields the following valid relations: $Elaboration(\pi_1^2, \pi_3^1)$, $Consequence(\pi_3^1, \pi_3^2)$ and $Narration(\pi_3^1, \pi_3^2)$. First, we check whether $Elaboration(\pi_1^2, \pi_3^1)$ is contextually relevant; the restriction sets for π_1^2 and π_3^1, as determined by this rhetorical relations, are:

$Elaboration(\pi_1^2, \pi_3^1)$:

π_1^2: $\{id(Elab.1(\epsilon)), id(\mathsf{enounce}(1)), \lambda_2^{\mathsf{SubclassOf}} \circ id(\mathsf{topic}(3))\}$;

[11] The rules in the abstract attribute grammar are not shown, since we assume that the semantics of the rhetorical relations are used in a consistent manner, i.e. we do not associate several semantic expressions to a given rhetorical relation.

[12] Actually, the numerical filtering procedure yields the *Consequence* relation, since it has a lower computational cost than *Elaboration*, mainly because of the expansion of the *topic*/1 function in *Elaboration*.

π_3^1: $\{id(Elab.2(\epsilon)), id(\mathsf{enounce}(2)), \lambda_1^{\mathsf{SubclassOf}} \circ id(\mathsf{topic}(3))\}$.

We thus have to unify the restriction set of π_1^2 shown above, with the restriction set of the same utterance, as yielded by previous rhetorical relations (namely, $ACK(\pi_1^2, \pi_2^1)$, and *Elaboration*(π_1^1, π_1^2)); the only elements that apparently seem problematic are $\lambda_2^{\mathsf{SubclassOf}} \circ id(\mathsf{topic}(3))$ as yielded by *Elaboration*(π_1^2, π_3^1), and $\lambda_1^{\mathsf{SubclassOf}} \circ id(\mathsf{topic}(3))$, as yielded by *Elaboration*(π_1^1, π_1^2), due to the asymmetry of the SubclassOf/2 predicate. But this only apparent, since, unlike in the situation above (when $ACK(\pi_1^1, \pi_2^1)$ was ruled out), here the argument "3" of the predicate topic/1 in the two elements in the restriction sets of π_1^2 does not have a global scope, because the predicates occur in *different* rules (that correspond to different rhetorical relations) in the abstract attribute grammar of the discourse and "3" is not global. In the situation with the ACK above, the issue was that the two elements due to be unified were yielded by the same discourse relation (and thus associated to the same rule in the grammar of the discourse), hence "3" had to maintain the same binding for the two elements; here, this is not the case, since the elements result from different rules in the grammar, which is equivalent to the fact that, here topic(3) behaves like a local variable which can be appropriately and *differently* instantiated, so that unification is possible. Another way of seeing things was that in the ACK situation above, the unification of the elements in the attribute sets stemmed to a reciprocal strict inclusion between two topics of utterances, whereas here we only have that there exist two other utterances such that the topic of the same (third) utterance is included in the topic of the first utterance and includes the topic of the second utterance, which does not lead to a contradiction. Therefore, we can conclude that *Elaboration*(π_1^2, π_3^1) passes contextual filtering.

We now have to apply contextual filtering on the locally valid (as yielded by SDRT-based local coherence analysis) relations *Narration* and *Consequence* between utterances π_3^1 and π_3^2. *Narration* yields the following restriction sets for these two utterances:

$Narration(\pi_3^1, \pi_3^2)$:

π_3^1: $\{id(Narration.1(\epsilon)), id(\mathsf{enounce}(1)), \lambda_1^{\neg\mathsf{Disjoint}} \circ id(\mathsf{topic}(3)), \lambda_1^{\mathsf{smaller}} \circ id(\Delta t(4))\}$;

π_3^2: $\{id(Narration.2(\epsilon)), id(\mathsf{enounce}(2)), \lambda_2^{\neg\mathsf{Disjoint}} \circ id(\mathsf{topic}(3)), \lambda_2^{\mathsf{smaller}} \circ id(\Delta t(4))\}$.

The restriction set of π_3^1 as shown above should unify with the restriction set of the same utterance, as yielded by the *Elaboration*(π_1^2, π_3^1) relation; but $\lambda_1^{\mathsf{SubclassOf}} \circ id(\mathsf{topic}(3))$ can be unified with $\lambda_1^{\neg\mathsf{Disjoint}} \circ id(\mathsf{topic}(3))$, since "3" has local significance as well, and these two elements can be written as $\mathsf{topic}(\lambda_1^{\mathsf{SubclassOf}}(3))$ and, respectively, $\mathsf{topic}(\lambda_1^{\mathsf{Disjoint}}(\lambda^{\mathsf{neg}}(3)))$ and nothing forbids two different bindings for "3" (let us denote them by 3_1 and 3_2) such that $\lambda_1^{\mathsf{Disjoint}}(\lambda^{\mathsf{neg}}(3_1)) = \lambda_1^{\mathsf{SubclassOf}}(3_2)$. Hence, *Narration* is deemed to hold between π_3^1 and π_3^2.

As for the alternative *Consequence* relation between the same two utterances, the restriction sets that it yields are:

$Consequence(\pi_3^1, \pi_3^2)$:

π_3^1: $\{id(Consequence.1(\epsilon)), id(\mathsf{enounce}(1)), \lambda^{\mathsf{neg}} \circ id(K(3))\}$;

π_3^2: $\{id(Consequence.2(\epsilon)), id(\mathsf{enounce}(2)), id(K(3))\}$.

Now, we have to check for two things:

(i) that the *Consequence* relation can hold between utterances π_3^1 and π_3^2 as an alternative to the *Narration* relation, between the same two utterances. Like proceeded before, for *Narration*, we obtain that *Consequence* holds as well, for the moment assuming that *Narration* is not included in the discourse structure.

(ii) that the *Consequence* relation can hold between the same two utterances, at the same time with *Narration*, that is, that *Consequence* and *Narration* can be simultaneously valid; if this is true, then numerical filtering should be applied, in order to select one of these two relations. Thus, we have to check that the restriction sets of π_3^1 and π_3^2, as yielded by *Narration* and by *Consequence*, can be unified. A problematic case is, for π_3^1, the unification of $\lambda_1^{\neg\mathsf{Disjoint}} \circ id(\mathsf{topic}(3))$ (or, equivalently, $\lambda^{\mathsf{neg}} \circ \lambda_1^{\mathsf{Disjoint}} \circ id(\mathsf{topic}(3))$) and $\lambda^{\mathsf{neg}} \circ id(K(3))$. This actually boils down to unifying $\mathsf{topic}(\lambda_1^{\mathsf{Disjoint}}(\lambda^{\mathsf{neg}}(3)))$ with $K(\lambda^{\mathsf{neg}}(3))$ but, since topic/1 returns a set, whereas $K/1$ returns a logical form[13], this is impossible. Hence, *Narration* and *Consequence* cannot hold simultaneously, but only as exclusive alternatives; it is even pointless to check

[13] Actually, the difference between these two unary functions resides in that K returns functors, logical connectors, atoms and an ordering relation between them, whereas topic returns only atoms.

whether the restriction sets for π_3^2 can be unified or not. Thus, this discourse "point" (utterances π_3^1 and π_3^2) is a good instance of a possible backtracking point in the "hiatus coping" strategy if a "hiatus" occurs later in discourse (see Section 3.3). Thus, between *Narration* and *Consequence*, the latter is chosen, due to lower computational cost involved in computing it (see above).

Then, Léonida produces the fourth speech turn in this dialogue; it consists in utterance π_4^1. SDRT-based local coherence analysis yields no monologue rhetorical relation between utterances π_2^1 and π_4^1; as for the dialogue relations, only $Background_q(\pi_1^2, \pi_4^1)$ and $Elab_q((\pi_1^2; \pi_3^1; \pi_3^2), \pi_4^1)$ are yielded by the SDRT-based local coherence analysis, where $(\pi_1^2; \pi_3^1; \pi_3^2)$ represents the discourse constituent formed with the utterances mentioned between brackets, along with the rhetorical relations that connect them, thus: $(\pi_1^2; \pi_3^1; \pi_3^2) ::= Elaboration(\pi_1^2, \pi_3^1) \wedge Consequence(\pi_3^1, \pi_3^2)$. We now apply contextual filtering on these two rhetorical relations. First, $Background_q(\pi_1^2, \pi_4^1)$ yields the following restriction sets for the two utterances[14]:

$Background_q(\pi_1^2, \pi_4^1)$:
π_1^2: $\{id(Background_q.1(\epsilon)), id(\mathsf{enounce}(1)), id(\mathsf{entails}.2(6))\}$;
π_4^1: $\{id(Background_q.2(\epsilon)), id(\mathsf{question}(2)), \lambda_2^{\mathsf{setof}} \circ id(\mathsf{answer}.2(3))\}$.

This is true because the semantics of $Background_q(\pi_1^2, \pi_4^1)$ was expressed as the following logic program:

$$Background_q(\pi_1^2, \pi_4^1) : -\mathsf{enounce}(\pi_1^2), \mathsf{question}(\pi_4^1), \mathsf{setof}(\chi, \mathsf{answer}(\chi, \pi_4^1), Set),$$
$$\mathsf{memberOf}(\_\chi, Set), \mathsf{enounce}(\_\chi), \mathsf{entails}(\_\chi, \pi_1^2).$$

As it is apparent from π_1^2's restriction set, it can readily be unified with previous restriction sets of this utterance, as yielded by $Elaboration(\pi_1^1, \pi_1^2)$, $ACK(\pi_1^2, \pi_2^1)$ and $Elaboration(\pi_1^2, \pi_3^1)$. Thus, we can conclude that $Background_q$ holds between utterances π_1^2 and π_4^1.

Then, we should inspect whether $Elab_q((\pi_1^2; \pi_3^1; \pi_3^2), \pi_4^1)$ holds as well. In order to do this, we have to show how *nested* rhetorical relations are handled, at a syntactic level: actually, they are processed as any other nested predicate (considered as a true fact if such a nested rhetorical relation had already been yielded by contextual filtering), via selector functions that are applied to them. The only differences are that, in our case, the $Elab_q$ rhetorical relation yields four restriction sets, for each global variable (that is, each of the utterances involved), and that the conjunction operator "connecting" discourse "points" in a discourse constituent is distributed over the predicates where this constituent occurs as argument. For instance, if we have a predicate like $p(\Pi)$, where Π is a discourse constituent of the form $\Pi ::= \rho_i(\pi_j, \pi_k) \wedge ... \wedge \rho_l(\pi_m, \pi_n)$, then $p(\Pi)$ can be written as: $p(\Pi) = p(\rho_i(\pi_j, \pi_k)) \wedge ... \wedge p(\rho_l(\pi_m, \pi_n))$. Therefore, $Elab_q((\pi_1^2; \pi_3^1; \pi_3^2), \pi_4^1)$ can be written as the following logic program:

$$Elab_q((\pi_1^2; \pi_3^1; \pi_3^2), \pi_4^1) : -\mathsf{enounce}(Elaboration(\pi_1^2, \pi_3^1)), \mathsf{enounce}(Consequence(\pi_1^3, \pi_3^2)), \mathsf{question}(\pi_4^1),$$
$$\mathsf{setof}(\chi, \mathsf{answer}(\chi, \pi_4^1), Set), \mathsf{memberOf}(\_\chi, Set), \mathsf{enounce}(\_\chi),$$
$$(\mathsf{SubclassOf}(\mathsf{topic}(\_\chi), [\mathsf{topic}(Elaboration(\pi_1^2, \pi_3^1)),$$
$$\mathsf{topic}(Consequence(\pi_3^1, \pi_3^2))]); \mathsf{equals}(\mathsf{topic}(\_\chi), [\mathsf{topic}(Elaboration(\pi_1^2, \pi_3^1)),$$
$$\mathsf{topic}(Consequence(\pi_3^1, \pi_3^2))])).$$

In this program, we put in a (Prolog-like) list the result of the function $\mathsf{topic}/1$ applied on the conjunction of rhetorical relations (in the discourse constituent), because actually this function returns a list, hence the effect of distributing the conjunction operator over $\mathsf{topic}/1$ is just the concatenation in a list of the results that this function returns on each member of the conjunction. However, in this situation one question arises: how do the selector functions operate on lists? The answer is that the list is just treated a multidimensional argument, whose components are selected just as the arguments of a multinary functor. Hence, the attribute sets for the utterances involved are shown below:

$Elab_q((\pi_1^2; \pi_3^1; \pi_3^2), \pi_4^1)$:
π_1^2: $\{\lambda_1^{Elab_q} \circ id(Elaboration.1(\epsilon)), \lambda^{\mathsf{enounce}} \circ id(Elaboration.1(1)), \lambda_2^{\mathsf{SubclassOf}}.1 \circ \lambda^{\mathsf{topic}} \circ id(Elaboration.1(7)),$
$\lambda_2^{\mathsf{equals}}.1 \circ \lambda^{\mathsf{topic}} \circ id(Elaboration.1(8))\}$;
π_1^3: $\{\lambda_1^{Elab_q} \circ id(Elaboration.2(\epsilon)), \lambda_1^{Elab_q} \circ id(Consequence.1(2)), \lambda^{\mathsf{enounce}} \circ id(Elaboration.2(1)),$
$\lambda^{\mathsf{enounce}} \circ id(Consequence.1(2)), \lambda_2^{\mathsf{SubclassOf}}.1 \circ \lambda^{\mathsf{topic}} \circ id(Elaboration.2(7)),$
$\lambda_2^{\mathsf{SubclassOf}}.2 \circ \lambda^{\mathsf{topic}} \circ id(Consequence.1(7)), \lambda_2^{\mathsf{equals}}.1 \circ \lambda^{\mathsf{topic}} \circ id(Elaboration.2(8)),$

[14] In applying the semantics of $Background_q$, we expanded the semantics of the *Background* relation, that appears as constituent of the former relation.

$\lambda_2^{\mathsf{equals}}.2 \circ \lambda^{\mathsf{topic}} \circ id(Consequence.1(8))\}$;

π_3^2: $\{\lambda_1^{Elab_q} \circ id(Consequence.2(\epsilon)), \lambda^{\mathsf{enounce}} \circ id(Consequence.2(2)), \lambda_2^{\mathsf{SubclassOf}}.2 \circ \lambda^{\mathsf{topic}} \circ$ $id(Consequence.2(7)), \lambda_2^{\mathsf{equals}}.2 \circ \lambda^{\mathsf{topic}} \circ id(Consequence.2(8))\}$;

π_4^1: $\{id(Elab_q.2(\epsilon)), id(\mathsf{question}(3)), \lambda_2^{\mathsf{setof}} \circ id(\mathsf{answer}.2(4))\}$.

Now, if there are at least two restriction sets for at least one of the first three global variables that cannot be unified, then the $Elab_q$ relation can be ruled out. Let us try for instance to unify two restriction sets of π_1^2; for this, we try to unify $\lambda_2^{\mathsf{SubclassOf}}.1 \circ \lambda^{\mathsf{topic}} \circ id(Elaboration.1(7))$ in the restriction set shown above, with $\lambda_2^{\mathsf{SubclassOf}} \circ id(\mathsf{topic}(3))$, in the restriction set yielded by $Elaboration(\pi_1^2, \pi_3^1)$. This comes to solving the equation:

$Elaboration.1(\lambda^{\mathsf{topic}}(\lambda_2^{\mathsf{SubclassOf}}.1(7))) = \mathsf{topic}(\lambda_2^{\mathsf{SubclassOf}}(3))$. Even if we can find an instantiation for "3" and "7" so that $\lambda_2^{\mathsf{SubclassOf}}.1(7)$ unifies with $\lambda_2^{\mathsf{SubclassOf}}(3)$, the problem is that $\mathsf{topic}/1$ returns a set, whereas *Elaboration* is equivalent to a logical form that, in the best case, just *strictly* includes $\mathsf{topic}/1$ (there is no semantics of a rhetorical relation that is made only of (a conjunction of) $\mathsf{topic}/1$ functions). Therefore, these two terms cannot be unified in our setting; hence, $Elab_q((\pi_1^2; \pi_3^1; \pi_3^2), \pi_4^1)$ is ruled out.

In conclusion, the (segmented) discourse structure that remains valid, after SDRT-based local coherence analysis and contextual filtering, consists in $Elaboration(\pi_1^1, \pi_1^2)$ or $Consequence(\pi_1^1, \pi_1^2)$ (if numerical filtering is applied, then the latter is retained), $ACK(\pi_1^2, \pi_2^1)$, $Elaboration(\pi_1^2, \pi_3^1)$, $Consequence(\pi_3^1, \pi_3^2)$ or $Narration(\pi_3^1, \pi_3^2)$ (if numerical filtering is applied, then the former is retained), and $Background_q(\pi_1^2, \pi_4^1)$.

6 Conclusions and Further Work

In this paper we have proposed two filterings of rhetorical relations in dialogue interpretation. The first filtering process takes into account the whole structure of discourse, however from a rather "syntactic" perspective; the filtering criterion resides in the unifiability of all the restriction sets that correspond to a global variable in the abstract attribute grammar of the whole discourse. The second filtering process is based on assessing the relative cost of computing a certain rhetorical relation between a pair of utterances or discourse constituents. This cost is seen as a direct reflection of the complexity of the abstract attribute grammar that corresponds to each rhetorical relation.

The two filterings presented in this paper can be seen as an alternative to MDC in vanilla SDRT. The advantage of the proposed filtering mechanism with respect to MDC resides in that we do not need a scalar interpretation of the quality of the rhetorical relations (e.g., an *Elaboration* has a higher quality than a *Narration*). Such a scalar interpretation is problematic, because it strongly depends on the context of discourse: for instance, when a speaker simply tells a story, narrating some facts, a *Narration* can be more appropriate than an *Elaboration*, although local coherence analysis (based on the semantics of the rhetorical relations and of the pair of utterances connected) might authorize both relations. Our contextual filtering tackles such ambiguous situations by inspecting the overall discourse context and ensuring that no contradictory predicates are validated on utterances. As for numerical filtering, it is just a rather artificial way of tackling situations that remain ambiguous even after contextual filtering. In this case, it might have also been appropriate to use a decision mechanism based on the frequencies of the rhetorical relations in domain-specific dialogue corpora. However, such a strategy is expensive and, in our view, fragile, especially for dialogues that are not easily classifiable in one domain or another.

In conclusion, our approach to rhetorical structuring in dialogue accounts for updates to the discourse structure, following utterances produced by the participants in dialogue, in a sequential process: first, given a pair of utterances due to be rhetorically connected, we assess, according to the SDRT apparatus, the semantic validity of the rhetorical relations (dialogue or monologue relations, according to the producers of the utterances), and we retain a number of *potentially relevant* relations; then, for the same pair of utterances and all previous utterances and discourse relations that connect them, we check, through contextual filtering, which relies on a conversion from rhetorical relations semantics to abstract attribute grammars, the rhetorical relations that are relevant according to the whole discourse structure. We thus retain an even smaller number (usually containing at most one element) of *contextually-relevant* rhetorical relations. Finally, if, for a given pair of utterances, the set of relevant rhetorical relations has more than one element, numerical filtering is applied, in order to retain only one discourse relation, that is the easiest to compute.

The main advantage of our approach, with respect to other discourse structuring mechanisms proposed, based on SDRT (such as in Schilder (2001)), or not (such as the RST-based structuring component of Marcu (2000)) resides in its *incremental* character: we do not need to have the whole dialogue available in order to compute the discourse structure; this structure is updated and *refined* progressively, as utterances are produced. This is a matter of key importance for interactive (conversational) systems, where the machine should compute a discourse structure after each speech turn in dialogue. The machine could then use this rhetorical structure for guiding its own production of answers (Popescu and Caelen, 2008).

The reduction of discourse relations to logical programs and abstract attribute grammars is particularly suited for practical systems, because a rather simple and computationally-feasible (essentially, first-order) formalism is used. However, for achieving practical value, the method should handle a current crippling issue: the reliance on an encyclopedic (or domain-specific, for restricted tasks) knowledge base, which is difficult to build and to maintain. For this, we plan to use an approach partly similar, in spirit, to Schilder (2001), in that we start from a fully underspecified SDRS for the *current* discourse[15]. Then, discourse connectives are used for partially specifying the SDRSs. However, we intend to rely on Jayez's account on discourse connectives as defined by dynamic semantic profiles, which could henceforth be *projected* into SDRT's Glue logic (in a manner akin to how semantic transfer is performed from DRT utterance-level semantics to Glue), in order to obtain an SDRS graph characterization of each connector. The remainder of unspecified relations could then be tackled via the contextual filtering proposed in this paper. We should thus be able to avoid the reliance on document formatting clues (such as topicality and position of the utterances (Schilder, 2001)), which are mostly irrelevant in spoken dialogue contexts.

References

Asher N (1993) Reference to Abstract Objects in Discourse. Kluwer Academic Publisher, Dordrecht, Netherlands

Asher N, Lascarides A (2003) Logics of Conversation. Cambridge University Press, UK

Asher N, Lascarides A (2008) Making the right commitments in dialogue. In: Fall 2008 Workshop in Philosophy and Linguistics, University of Michigan

Asher N, Hardt D, Busquets J (2001) Discourse parallelism, ellipsis, and ambiguity. Journal of Semantics 18(1)

Deransart P, Małuzýnski J (1993) A Grammatical View of Logic Programming. MIT Press

Egg M, Redeker G (2007) Underspecified discourse representation. In: Constraints in Discourse, vol 1, John Benjamins

Gallier J (1986) Logic for Computer Science. Wiley

Hobbs J, Stickel M, Appelt D (1993) Interpretation as abduction. Artificial Intelligence 63:69–142

Isakowits T (1991) Can We Transform Logic Programs into Attribute Grammars? Technical Report, New York University, NY

Jayez J, Rossari C (1999) Pragmatic connectives as predicates. the case of inferential predicates. In: Predicative Structures in Natural Language and Lexical Knowledge Bases, Kluwer, pp 285–319

[15] We re-emphasize the fact that our approach is incremental, and we believe that this feature should be maintained, especially in the context of dialogue systems.

Kamp H, Reyle U (1993) From Discourse to Lexicon: Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory. Kluwer Academic Publishers, Dordrecht, Netherlands

Knott A (1996) A Data-driven Methodology for Motivating a Set of Coherence Relations. Ph. D. Thesis, University of Edinburgh

Lascarides A, Asher N (1993) Temporal interpretation, discourse structure and commonsense entailment. Linguistics and Philosophy 16:437–493

Lascarides A, Asher N (2009) Grounding and correcting commitments in dialogue. Journal of Semantics (to appear)

Mann WC, Thompson SA (1988) Rhetorical structure theory: Toward a functional theory of text organization. Text 8(3):243–281

Marcu D (1997) The Rhetorical Parsing, Summarization and Generation of Natural Language Texts. Ph. D. Thesis, University of Toronto

Marcu D (2000) The rhetorical parsing of unrestricted texts: A surface-based approach. Computational Linguistics 26(3):395–448

Maudet N, Muller P, Prévot L (2006) Social constraints on rhetorical relations in dialogue. In: Proceedings of the 2nd SIGGen Workshop Constraints in Discourse, ACL, Maynooth, Ireland, pp 133–139

Popescu V (2008) Formalisation des contraintes pragmatiques pour la génération des énoncés en dialogue homme-machine multi-locuteurs. Ph. D. Thesis, Grenoble Institute of Technology

Popescu V, Caelen J (2008) Argumentative ordering of utterances for language generation in multi-party human-computer dialogue. Argumentation [DOI: 10.1007/s10503-008-9122-y](Online first):33p

Popescu V, Caelen J, Burileanu C (2007) Logic-based rhetorical structuring for natural language generation in human-computer dialogue. Lectures Notes on Artificial Intelligence 4629:309–317

Reyle U (1993) Dealing with ambiguities by underspecification: construction, representation, and deduction. Journal of Semantics 10:123–179

Schilder F (2001) Robust discourse parsing via discourse markers, topicality and position. Natural Language Engineering 8:235–255

Schlangen D, Lascarides A, Copestake A (2001) Resolving underspecification using discourse information. In: Proceedings of the Bi-Dialog Workshop on the Semantics and Pragmatics of Dialogue, Bielefeld

Sporenader C, Lascarides A (2006) Using automatically labelled examples to classify rhetorical relations: an assessment. Natural Language Engineering 14(3):369–416

Staudacher M (2005) SDRT Reformulated using DPL. Term Paper, Bielefeld University, Bielefeld

Traum D (1994) A Computational Theory of Grounding in Natural Language Conversation. Ph. D. Thesis, University of Rochester

Vanderveken D (1990–1991) Meaning and Speech Acts. Cambridge University Press, United Kingdom