Generating Inference-Rich Discourse Through Revisions of RST-Trees

Helmut Horacek

Universität des Saarlandes, FB 14 Informatik
Postfach 1150
D-66041 Saarbrücken, Germany
horacek@cs.uni-sb.de

Abstract

The majority of generation systems to date are able to communicate information only by uttering it explicitly. Rhetorical Structure Theory (RST), one of the most frequently used discourse theories for text planning in natural language generation, does not support more flexibility either, because it ignores implicit rhetorical relations and accepts only one prominent relation between clauses. In formal systems, however, the underlying information is represented in a very detailed way, which requires easily inferable parts to be left implicit for producing natural and comprehensible discourse. In order to improve the quality of texts generated from fine-grained semantic specifications, we present an approach that successively revises an explicit text plan by introducing addressee dependent short-cuts and communicatively justified reorganizations. Text plan revisions include the compactification of state-action and reasoning sequences, the omission of redundant conditions, and the reorganization of arguments for presentation purposes. Our techniques enable us to generate shorter and better understandable texts from detailed representations, as in formal systems, especially in deduction systems.

Introduction

The majority of generation systems to date are able to communicate information intended to be conveyed only by uttering it explicitly. Especially Rhetorical Structure Theory (RST) (Mann and Thompson 1987a), one of the most frequently used discourse theories for text planning, does not support more flexibility either. It ignores implicit rhetorical relations and accepts only one prominent relation between clauses. In contrast to that, the underlying information is typically represented in a semantically fine-grained way in formal systems, especially in deduction systems, which requires easily inferable parts to be left implicit for producing natural and comprehensible discourse. In such applications, RST provides an insufficient basis for generating fluent texts from facts underlying a situation.

In order to improve the quality of texts generated from semantically fine-grained background information, as available in formal systems, we present an approach that successively revises an explicit and detailed text plan by introducing addressee dependent short-cuts and communicatively justified reorganizations. Text plan revisions include the compactification of state-action and reasoning sequences, the omission of redundant conditions, and the reorganization of arguments for purposes of presentation.

This paper is organized as follows. We first review approaches that attempt to convey information in discourse indirectly, especially extensions to RST. Then we present the functionality of our method, realized by a small set of presentation rules that initiate revisions of RST trees. Finally, we demonstrate the improvements obtained for presenting mathematical proofs in natural language.

Planning Inference-Rich Discourse

Even at an early stage of NL generation research, (Mann and Moore 1981) have stated that information stemming from any reasonably complete body of knowledge needs to be condensed and generated selectively. Their knowledge filter is extremely simple in marking pieces of knowledge known to the addressee for omission in the output, coherence permitting. (Mellish and Evans 1989) apply algebraic simplification rules to reduce plan constructs, such as consequences of actions, and (McDonald 1992) exploits domain-specific conditions to reduce the information conveyed explicitly. In contrast to these early systems, recent methods are better motivated in linguistic or in cognitive terms. In TECH-DOC (Stede 1993), state-action transitions are modeled for multi-lingual instruction generation so that compactly expressing them becomes a matter of lexical choice. In WISHFUL-II (Zukerman and McConachy 1993), concise explanations are produced by using a constraint-based optimization mechanism addressing the user's inferences. (Green and Carberry 1994) consult coherence rules for the generation of indirect answers to accomplish complementary discourse goals by modeling potential obstacles that prevent intended achievements.

Within the framework of RST, (Moore and Pollack 1992) have shown that relations on the informational as well as on the intentional level may be relevant for understanding and for producing discourse in a principled way. However, their work does not provide an operational model. (Maier 1995) has developed a text planner that can handle some co-occurrence patterns between ideational and textual relations, even with divergent semantics.

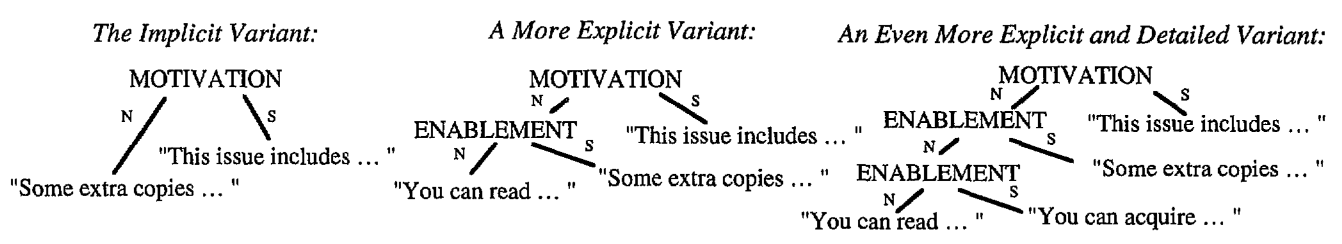

Figure 1: A MOTIVATION relation in RST, in varying degrees of explicitness and detail

(Horacek 1997) aims at capturing inference relations between generic and referential pieces of knowledge through rules expressing aspects of conversational implicature. The rules operate on an RST tree and annotate contextually inferable parts that can be left out when a text is generated.

While there is no doubt that these approaches advance generation capabilities, the kinds of implicitness they can cope with are still too limited – especially when presenting or explaining results obtained by formal systems.

Implicit Relations behind RST Trees

In this section, we examine implicit relations underlying RST representations, and we categorize them. We illustrate this discussion by examples from everyday knowledge and by text portions from the domain of mathematical proofs. For the former, we assume underlying relations in a knowledge base; for the latter, we refer to proof graphs of theorem provers. Let us start with the following text. In

(1) "Some extra copies of the Spring 1984 issue of AI Magazine are available in the library. This issue includes a "Research in Progress" report on AI research at ISI." (Matthiessen and Thompson 1987)

a MOTIVATION relation is conveyed (see Figure 1, left side). However, the action the MOTIVATION refers to, is not the mere availability of copies, but the potentiality to acquire and read such a copy. This justifies the extended relations (see the middle and right parts of Figure 1), which we assume to be present in an underlying knowledge base. Given such a representation, a good text planner should be able to produce concise and natural texts from it rather than mechanically expressing all relations explicitly.

In the domain of mathematical proofs, a similar case is the occurrence of nested causality: for example, if '$0 < a$' is a direct cause for some proposition p, '$1 < a$' as an assumption in the underlying proof is usually preferred for justifying proposition p, because the truth of '$0 < a$' easily follows from '$1 < a$'. Because avoiding a redundancy here leads to omission of intermediate reasoning or action sequence steps, we call this case *Compactification.*

Sentence (2) (Mann and Thompson 1983) relies on background knowledge to be understood correctly. However, adding a clause to it as in (2)' would lead to an apparent redundancy or somewhat bizarre markedness (Mann and Thompson 1985), see the left and middle parts of Figure 2.

(2) "I'm hungry. Let's go to the Fuji Gardens."

(2)' "I'm hungry. Let's go to the Fuji Gardens. My going to the Fuji Gardens would contribute significantly to solving the problem of my hunger."

While explicitly expressing some of the intermediate steps in a *Compactification* case is not uncommon and may even be preferable in knowledge-intensive domains, this is unacceptable in (2)'. In theorem proving systems, this sort of redundancy is a central part of their reasoning mechanism: the pattern in the right part of Figure 2 (only one relation name is different) is the classical modus ponens inference. Because avoiding a redundancy here leads to omission of an implication (a rule), we call this case *Rule-Cutting.*

Besides these omissions, there may be structural changes (see (Maier 1995)). They prominently occur in the context of sequences or enumerations, which are best represented without expansions. Further details precede or follow the entire sequence or enumeration in the text, although they are in fact related to one of the elements of that construct.

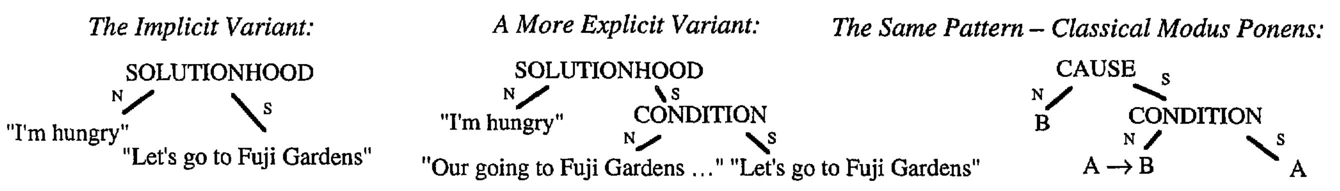

Figure 2: RST relation in varying degrees of redundancy for texts from different domains

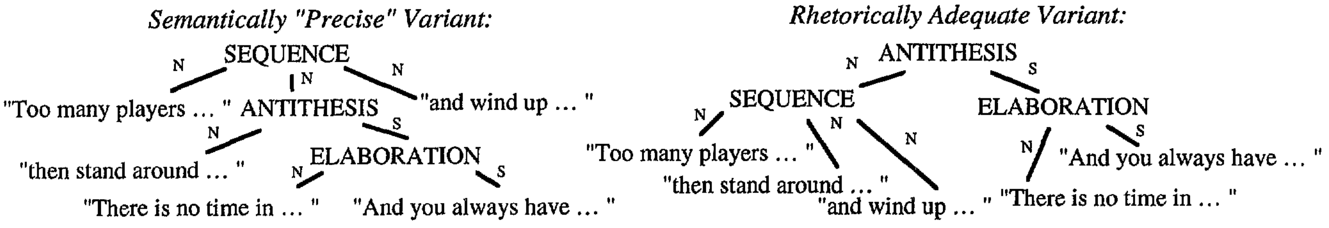

Figure 3: RST relation in varying degrees of argumentative organization

In text (3) (Mann and Thompson 1987b), the antithesis semantically relates to the second clause of the sequence (see Figure 3). Moreover, if the same text span appears in multiple places, lifting it may help in improving the text.

(3) "Too many players hit an acceptable shot, then stand around admiring it and wind up losing the point. There is no time in an action game like tennis to applaud yourself and still get in position for the next shot. And you always have to assume there will be a next shot."

In the domain of mathematical proofs, this situation occurs frequently, motivated by special presentation modes (series of inequations (Fehrer and Horacek 1997b)), or by multiply used assumptions. Because reorganizing argumentation involves structural changes, we call this case *Restructuring.*

Formalizing the Revision Model

In this section, we illustrate how RST trees that express relations implicitly as discussed in the previous section can be generated from underlying detailed representations. This is achieved by the aid of *presentation* rules that capture the cases identified above: *Rule-Cutting*, *Compactification*, and *Restructuring* rules. These rules operate on an RST tree and revise it by adding more direct or lifted relations. These additions are beneficially kept as alternatives for a number of reasons (the first one is specific to our application domain): (1) special presentation modes such as building chains of inequations may not work for some short-cuts, (2) the cognitive accessibility of antecedents is unpredictable at the local level where the *Restructuring*-rule operates, so that resulting changes may not be realizable globally, (3) simultaneously available alternatives prove valuable for presentations to addressees with divergent knowledge and in interactive environments. In the following, we first illustrate the model of the addressee's knowledge and inferential capabilities which constrain the application of the presentation rules. Then we give formalizations of these rules, and we describe their application.

For the model of mental capabilities, we distinguish the following categories of knowledge and communicative competence attributed to the audience:

- *knowledge per se*, taxonomic and referential knowledge,
- the *attentional state* of the addressee,
- *taxonomic* inferences,
- *logical* inferences, and
- *communicatively motivated* inferences.

Communicatively motivated inferences concern the capability to augment logically incomplete pieces of information in a given context, which is expressed by the presentation rules themselves. Assumptions about the addressee's taxonomic and referential knowledge are represented by simple stereotypes (see (Fehrer and Horacek 1997a)), and the dialog memory and focus mechanisms of our proof presentation system PROVERB (Huang and Fiedler 1997), respectively. The remaining categories need more explanation.

The attentional state is modeled by attributing the user with *awareness* concerning the facts needed to mentally reconstruct implicit information, which is more than just being acquainted with these facts. For generic knowledge, that is, domain regularities, the user is considered to be aware of those parts of his/her knowledge that belong to 'basic' world knowledge or to special knowledge about the issue presented – this criterion expresses coherence and expectation of some sort. For referential knowledge, the user is credited with being aware of all facts mentioned or implied so far within the current discourse segment – this criterion expresses human memory limitations, as intensively examined in (Walker 1996). For the domain of mathematical proofs, the underlying categorization of knowledge is oriented on relevant mathematical theories: basic axioms and axioms within the particular subarea of the proof presented are considered to be part of the user's awareness, and discourse segments essentially follow the structure of the proof (e.g., subproof, case analysis).

The remaining inference categories, taxonomic and logical inferences, are understood here as elementary clues, either relying on purely taxonomic knowledge, or following a standard logical inference pattern such as podus ponens. The mechansim that computes these inferences operates on pieces of domain knowledge and takes into account memory limitations. To assess these skills in our domain, we have categorized the reasoning steps in the derivation of machine-generated proofs, and we have examined the complexity of the substitutions involved, in a number of text book examples and in machine-found proofs. Typical reasoning steps are generalizations of terms and relations, and abstractions from individuals, embedded in modus ponens derivarions, with substitutions intro-

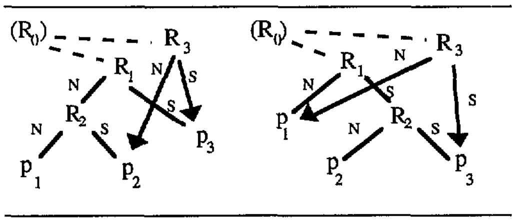

Figure 4: Patterns for tree condensation additions

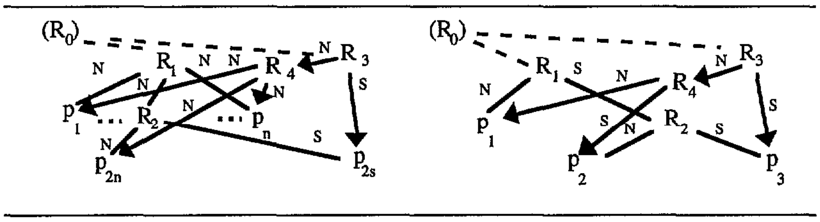

Figure 5: Patterns for tree restructuring additions

ducing one additional operator, at most. Hence, if a reasoning step concluding p from q remains within certain complexity limits, we say that '$q \mid\!\sim p$' holds ('$\mid\!\sim$' denotes the 'human' calculus), and the addressee is assumed to be able to make that inference on his/her own, so that it can be left out in a presentation.

A further issue involved is the composition of inference steps. According to psychological experiments, leaving out intermediate steps in a chain of argumentation should still be understood as a 'direct' cause, while 'indirect' causes negatively affect the reasoning effort (Thüring and Wender 1985) – this criterion needs also be modeled for the domain at hand. In a previous approach to expert system explanations, this aspect has been modeled by requiring purposes of domain rules involved to be identical (Horacek 1997). For proofs, we try to capture it by a structural similarity between intermediate and final conclusions.

Details about these categories of knowledge and inferential skills are formalized by the following predicates:

AWARE-OF (User, p) – the user's attentional state, which subsumes that the user knows p.

GENERIC (p) – replaces all variables in p by their categories (according to the definitions in the proof).

ABLE-INFER (User, φ_1, φ_2) – inferential skill: $\varphi_1 \mid\!\sim \varphi_2$.

SIMILAR(p_1, p_2) holds in our domain, if the propositions p_1 and p_2 are structurally identical.

MISFIT(p_1, p_2, R) holds between propositions p_1 and p_2 when being connected by relation R, that is, some of the conditions associated with R are violated.

PREFER((p, E), (p, F)) means that p is better presentable with expansion E than with expansion F, such as (<equation>, { }) and (<equation>, F) for $F \neq \{\}$.

MULT(E,{T}) – text span E is an element of set {T}.

TREE(T) yields the set of nodes rooting in node T.

The rules' funcionality in terms of changes made in the text structure tree is illustrated by the diagrams in Figures 4 and 5, whose applicability is restricted by the conditions listed in Figure 6. These figures can essentially be seen as overlays of the explicit and implicit tree representation variants, as exemplified in Figures 1 to 3. In Figures 4 and 5, full lines represent tree portions that need to be matched against subtrees of the text structure tree under consideration, and lines with an arrow represent the additions to be made, applicability conditions permitting. The dashed lines in these Figures indicate that, unless the top level relation of a pattern matches the root node of the text structure tree, the top relations of the nodes added and of the existing nodes have the same predecessor node (R_0). All nodes are associated with variables, distinguishing relations (R) from propositions (p), which may or may not be expanded further in the text structure tree examined. These variables are referred to from the application conditions in Figure 6.

The application conditions as shown in Figure 6 verify the applicability of these rules by matching them against the model of the mental capabilities of the addressee. As the formalizations demonstrate, the presentation rules are expressed in a widely domain-independent way; domain-specific aspects can and to some extent must be integrated through, e.g., operationalizations of the calculus '$\mid\!\sim$', and interpretations of the predicates SIMILAR and PREFER.

For the *Compactification* rule, there are variants for nucleus expansion and for satellite expansion, which refer to tree patterns on the left and right side of Figure 4, correspondingly. Compactification is licensed if the user is credited with the ability to infer the original relation pattern from the communicated relation and its associated facts ($R_1(R_2(p_1,p_2),p_3)$ from $R_3(p_2,p_3)$ or $R_1(p_1,R_2(p_2,p_3))$ from $R_3(p_1,p_3)$, respectively). Whether this ability can be ascribed to the user depends on the expansion direction. For satellite expansions, we are confronted with chains of arguments. In this case, the proposition left implicit and the 'remaining' proposition must be 'similar' – to a certain extent, this criterion is tailored to our domain. For nucleus extensions, a certain mismatch between the facts explicitly conveyed and the relation linking them on the surface must be present in order to motivate the hearer to mentally search for the information left implicit (see the left pattern in Figure 1 – a MOTIVATION relation between these two facts cannot be taken as a motivation proper, which suggests the insertion of an ENABLEMENT relation).

The *Rule-Cutting* rule – the right side in Figure 4 shows the underlying pattern – applies to a CONDITION relation (R_2) within the context of some other relation (R_1). If R_1 is a CAUSE relation, the added relation (R_3) is always a REASON relation because some argument is left out in the precise and logically complete CAUSE relation; otherwise, R_1 and R_3 are identical. In order for this rule to be licensed, the user must be acquainted with the principle underlying the condition (the generic form of the rule, GENERIC(p_2)), and he/she must be 'aware' of that rule in the sense that the rule coherently 'fits' in the context of the CONDITION relation. In addition, the user must be able to infer that

Compactification rule (nucleus expansion)

$\mathrm{MISFIT}(p_2, p_3, R_3) \wedge$
$\mathrm{ABLE\text{-}INFER}(\mathrm{User}, (p_2 \wedge p_3 \wedge R_3(p_2,p_3)),$
$\varphi \wedge R_1(R_2(\varphi,p_2),p_3) \wedge (\varphi \Rightarrow p_1))$

The underlying pattern appears in Figure 4, left side, with $R_1 = R_3$, e.g., MOTIVATION, and R_2 mostly ENABLEMENT.

Compactification rule (satellite expansion)

$\mathrm{SIMILAR}(p_1, p_2) \wedge$
$\mathrm{ABLE\text{-}INFER}(\mathrm{User}, (p_1 \wedge p_3 \wedge R_3(p_1,p_3)),$
$(\varphi \wedge R_1(p_1,R_2(\varphi,p_3)) \wedge (\varphi \Rightarrow p_2)))$

The underlying pattern appears in Figure 4, right side, mostly for a chain of arguments, with $R_1 = R_3$. R_1 is typically a REASON, as is R_2 in case of chained arguments.

Rule-Cutting rule

$\mathrm{AWARE\text{-}OF}(\mathrm{User},\mathrm{GENERIC}(p_2)) \wedge$
$\mathrm{ABLE\text{-}INFER}(\mathrm{User}, (p_1 \wedge p_3 \wedge R_3(p_1,p_3)),$
$(R_1(p_1,R_2(\varphi,p_3)) \wedge (\varphi \Rightarrow p_2)))$

The underlying pattern appears in Figure 4, right side, with $R_1 = R_3$, unless R_1 is a CAUSE - then R_3 is a REASON, while R_2 is always a CONDITION.

Restructuring rule (multi-nucleus expansion)

$\mathrm{AWARE\text{-}OF}(\mathrm{User},p_{2s}) \wedge$
$(\mathrm{PREFER}((p_{2n},\{\}),(p_{2n}, R_2(p_{2n},p_{2s}))) \vee$
$\mathrm{MULT}(p_{2s}, \mathrm{TREE}(\mathrm{ROOT})))$

The underlying pattern appears in Figure 5, left side, with $R_1 = R_4$ and $R_2 = R_3$, R_1 is typically SEQUENCE or JOINT.

Restructuring rule (satellite expansion)

$\mathrm{AWARE\text{-}OF}(\mathrm{User},p_3) \wedge$
$(\mathrm{PREFER}((p_2,\{\}),(p_2, R_2(p_2,p_3))) \vee$
$\mathrm{MULT}(p_3, \mathrm{TREE}(\mathrm{ROOT})))$

The underlying pattern appears in Figure 5, right side, with
$R_1 = R_4$ and $R_2 = R_3$

Figure 6: Application conditions for presentation rules

some regularity (φ) is needed to fully explain the relation R_3 between p_1 and p_3, and that φ entails p_2.

For the *Restructuring* rule, finally, there are also two variants, one for multi-nucleus and one for nucleus-satellite relations (for the underlying patterns, see the left and the right parts of Figure 5, correspondingly). The proposition p_3 (or p_{2s} for the multi-nucleus case) expanding R_2 is lifted according to the underlying patterns if the user is aware of this proposition (that is, it is in the current focus of attention), and if there are presentation preferences in favor of the lifting operation (expressed by the predicate PREFER), or if p_3 has co-references in the entire text structure tree (expressed by the predicate MULT).

The application of the presentation rules in context is organized by building an initial RST tree, invoking the rules in the specific order described below, and expressing the resulting tree in natural language. The initial RST tree is built on the basis of the proof graph obtained by a theorem prover: logical derivation steps are transduced into modus ponens constructs, conjoined preconditions are linked by a JOINT relation, and multiple references are resolved by copying the head node and attaching expansions to the first occurrence in the tree.

Presentation rules are applied to yield all feasible shortcuts by traversing the RST tree from its leaf nodes to the root node, without returning to direct or indirect ancestor nodes. This strategy assures that the application conditions of a rule once it has been applied successfully remain unchanged throughout further processing. This procedure is carried out in two cycles. In the first one, the *Rule-Cutting* rule and then the *Compactification* rule are applied to each node; in the second cycle, the *Restructuring* rule is applied.

For communicating the RST tree in natural language, propositions associated with the nodes are handed over to the surface generator, by starting from the root node down to all reachable leaf nodes. Due to the alternative routes available via short-cuts, a number of nodes and even some leaf nodes may be left out in this path construction process. In our application, alternatives among the available relations are chosen according to the following preferences: (1) hypotheses and lemmas are expressed at the earliest opportunity, (2) the number of REASON relations to be verbalized is minimized (by following short-cuts whenever possible), and (3) compact notations forms, such as chains of inequations, are applied wherever possible (criterion (3) may overrule the other two criteria).

An Example from Proof Presentation

In order to illustrate the combined functionality of our presentation rules, we demonstrate the proof of theorem 1.11:

Theorem 1.11 (Lüneburg 1981): Let K be an ordered field. If $a \in K$, then $1 < a$ implies $0 < a^{-1} < 1$, and vice-versa.

Lemma 1.10 (Lüneburg 1981): Let K be an ordered field. If $a \neq 0 \in K$, then $0 < a$ implies $0 < a^{-1}$, and vice-versa.

Proof (for $1 < a \Rightarrow 0 < a^{-1} < 1$): Let $1 < a$. According to Lemma 1.10 we then have $a^{-1} > 0$. Therefore $a^{-1} = 1a^{-1} < aa^{-1} = 1$.

Proof presentation is invoked with the RST tree shown in Figure 7 which originates from a proof graph on the assertion level (Huang 1994) that is obtained by transforming the result produced by the theorem prover OTTER (McCune 1994). Individual nodes are labeled by numbers to ease references in the text. The audience is assumed to know basic axioms for ordered fields, such as unit and inverse elements, and other axioms such as transitivity, monotony, etc. Awareness about these axioms is assumed, too. In this environment, revisions are invoked as follows:

First, the *Rule-Cutting* rule is applied to the subtree rooted in node *27*, adding a REASON node *27'* as satellite

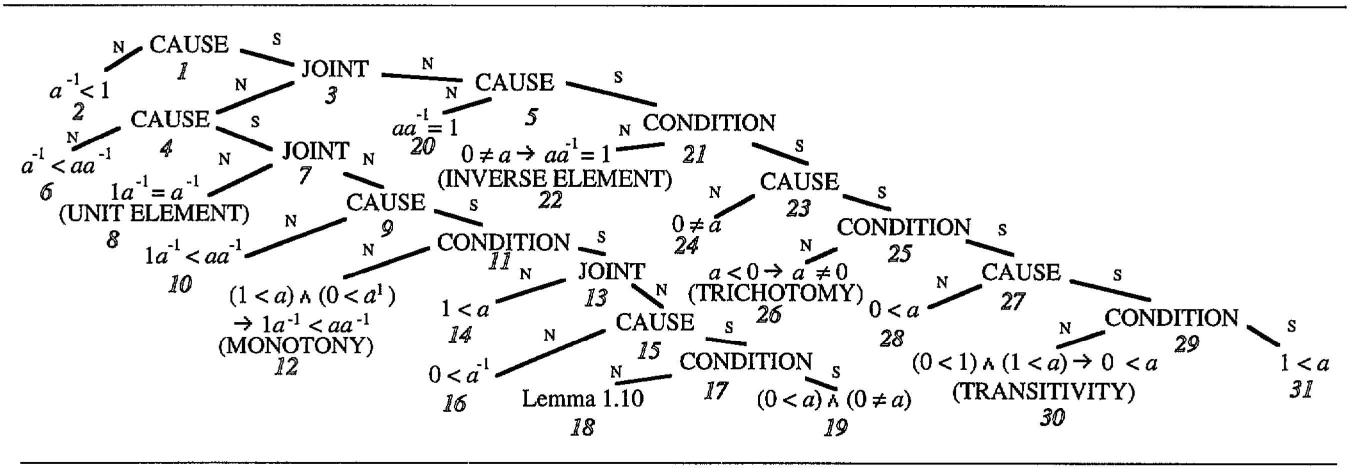

Figure 7: Initial RST tree for the proof of theorem 1.11 in (Lüneburg 1981).

of node *25*, with node *28* as nucleus and node *31* as satellite. Then the *Compactification* rule is applied, which introduces a direct connection from node *25* to node *31*, so that the new REASON node *27'* is not needed for the final presentation. This operation sequence is identically repeated with the nodes *23*, *23'*, *21*, *24*, and *31*, correspondingly. Moreover, the *Rule-Cutting* rule is applied to the subtree rooted in node *5*, adding a REASON node *5'* with node *20* as nucleus and node *31* as satellite. Unlike in previous cases, the *Compactification* rule is not applicable to the resulting subtree, because the predicate SIMILAR fails for $aa^{-1} = 1$ and $1 < a$. Altogether, the right branch of the JOINT relation (node *3*) can be fully expressed by the short-cut via the REASON relation in node *5'* (see Figure 8, left side; newly created relations are printed in italics, and links as dashed lines with arrows). In the left branch of the RST tree, coreference is exploited for the satellite of the CONDITION of lemma 1.10 by adding a link from node *17* to a copy of node *31*, node *31'* as its satellite. From there, the *Rule-Cutting* rule is applied to the subtree rooted in node *9*, yielding a REASON node *9'*, making node *10* its nucleus and *13* its satellite (see the left side of Figure 8).

Applications of the *Restructuring* rule concern the nodes *14*, *31*, *31'*, and *17*, that is, the assumption $1 < a$ (three times), and the proposition $0 < a^{-1}$, together with its extension, Lemma 1.10. $1 < a$ is lifted to the top of the RST tree because it is referred to from several relations, even from both main branches of the RST tree. $0 < a^{-1}$ is also lifted near the top of that tree because it expands a node that is presented in the special form within a series of inequations, together with the other nodes below node *4*. The right part of Figure 8 shows the relevant parts of the final result. A most concise text expressing it is:

> Let $1 < a$. $0 < a^{-1}$ follows from $1 < a$ and Lemma 1.10. Therefore $a^{-1} = 1a^{-1} < aa^{-1} = 1$ holds.

which pretty much resembles the book proof, and is much better than the verbalization *without* presentation rules:

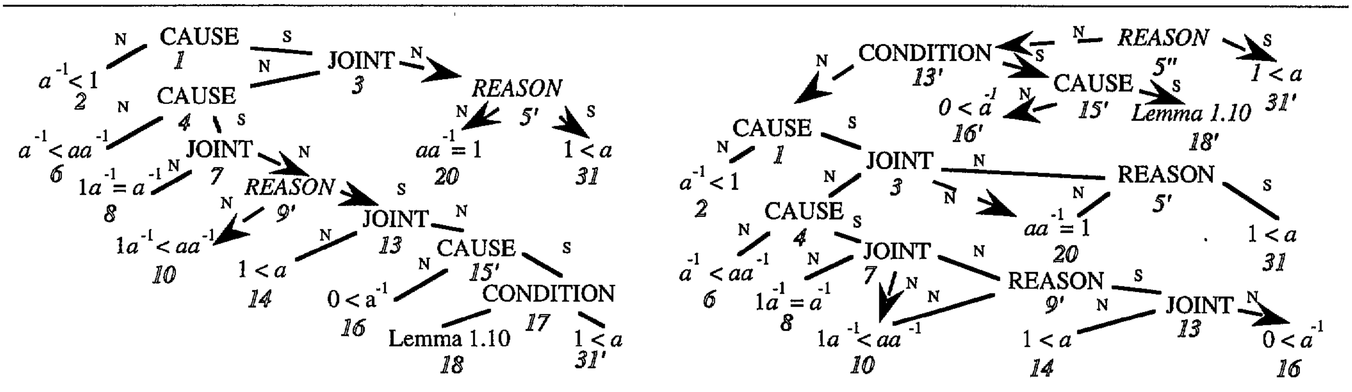

Figure 8: Fragments of the revised RST tree – parts of an intermediate state (left side) and parts of the final state (right side)

Proof (PROVERB): Let $1 < a$. Then $0 < a$, because '<' is transitive and $0 < 1$. $0 \neq a$ follows from the trichotomy of '<'. Lemma 1.10 implies $0 < a^{-1}$. Since '<' is monotone and $1 < a$, $1a^{-1} < aa^{-1}$. $a^{-1} < aa^{-1}$ because of the definition of the unit element of K. $aa^{-1} = 1$ because of the definition of the inverse element of K for $a \neq 0$. Hence $a^{-1} < 1$.

Conclusion

In this paper, we have presented an approach to generate natural and concise argumentations from semantically fine-grained representations typically available in formal systems. We achieve this result by revising RST trees, through compactifying state-action and reasoning sequences, omitting redundant conditions, and reorganizing arguments for purposes of presentation. In comparison to previous approaches, we can deal with significantly more structural differences between an underlying knowledge representation and RST trees representing text structures – these differences lead to the introduction of new relations, additions of links, and regrouping of subtrees. We have motivated our approach by texts from everyday discourse.

We have elaborated the application conditions of these operations for the domain of mathematical proofs and integrated them into a dedicated presentation system. Since the associated modeling is and must be domain-specific to a certain degree, we have also made clear which parts in the formalization need to be adapted. Our method constitutes an important step in making the results obtained by reasoning systems better accessible to its users. In addition to the quality improvements in generating inference-rich discourse in general, our method enhances also the contribution of RST for generating these kinds of texts.

References

Fehrer, D.; and Horacek, H. 1997a. Exploiting the Addressee's Inferential Capabilities in Presenting Mathematical Proofs. In Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence, 556-560. Menlo Park, Calif.: International Joint Conferences on Artificial Intelligence, Inc.

Fehrer, D.; and Horacek, H. 1997b. Presenting Inequations in Mathematical Proofs. In Proceedings of JCIS'97, Special Section on Logical Methods for Computational Intelligence, North Carolina.

Green, N.; and Carberry, S. 1994. A Hybrid Reasoning Model for Indirect Answers. In Proceedings of ACL-94, Las Cruces, New Mexico.

Horacek, H. 1997. A Model for Adapting Explanations to the User's Likely Inferences. *User Modeling and User Adapted Interaction* 7:1-55.

Huang, X. 1994. Reconstructing Proofs at the Assertional Level. In Proceedings of the 12th International Conference on Automated Deduction, 738-752, Springer-Verlag.

Huang, X.; and Fiedler, A. 1997. Proof Verbalization as an Application of NLG. In Proceedings of the 15th International Joint Conference on Artificial Intelligence, 965-971, Menlo Park, Calif.: International Joint Conferences on Artificial Intelligence, Inc.

Lüneburg, H. 1981. *Vorlesungen über Analysis*. BI Wissenschaftsverlag.

Maier, E. 1995. Textual Relations as Parts of Multiple Links Between Text Segments. In Trends in Natural Language Generation – An Artificial Intelligence Perspective, 68-87, Springer-Verlag.

Mann, W.; and Moore, J. 1981. Computer Generation of Multiparagraph English Text. *Computational Linguistics* 7(1):17-29.

Mann, W.; and Thompson, S. 1983. Rhetorical Structure Theory: A Theory of Text Organization. Technical Report, ISI/RR-83-115, ISI at University of Southern California.

Mann, W.; and Thompson, S. 1985. Assertions from Discourse Structure. Technical Report, ISI/RR-85-155, ISI at University of Southern California.

Mann, W.; and Thompson, S. 1987a. Rhetorical Propositions in Discourse. The Structure of Discourse. Polanyi L. ed., Norwood, Ablex.

Mann, W.; and Thompson, S. 1987b. Antithesis: A Study in Clause Combining and Discourse Structure. Technical Report, ISI/RR-87-171, ISI at University of Southern California.

Matthiessen, C.; and Thompson, S. 1987. The Structure of Discourse and Subordination. Technical Report, ISI/RR-87-183, ISI at University of Southern California.

McCune, W. 1994. Otter 3.0 Reference Manual and Guide, Technical Report, ANL-94/6, Argonne National Lab.

McDonald, D. 1992. Type-Driven Suppression of Redundancy in the Generation of Inference-Rich Reports. In Aspects of Automated Natural Language Generation, 72-88, Springer-Verlag.

Mellish, C.; and Evans, R. 1989. Natural Language Generation from Plans. *Computational Linguistics* 15(4):233-249.

Moore, J., and Pollack, M. 1992. A Problem for RST. The Need for Multi-Level Discourse Analysis. *Computational Linguistics* 18(4):537-544.

Stede, M. 1993. Lexical Choice Criteria in Language Generation. In Proceedings of the 6th Conference of the European Chapter of the Association for Computational Linguistics, 454–459.

Thüring, M., and Wender, K. 1985. Über kausale Inferenzen beim Lesen. *Sprache und Kognition* 2:76-86.

Walker, M. 1996. The Effect of Resource Limits and Task Complexity on Collaborative Planning in Dialogue. *Artificial Intelligence* 85:181-243.

Zukerman, I.; and McConachy, R. 1993. Generating Concise Discourse that Addresses a User's Inferences. In Proceedings of the 13th International Joint Conference on Artificial Intelligence, 1202-1207, Menlo Park, Calif.: International Joint Conferences on Artificial Intelligence, Inc.