Planning Argumentative Texts

Xiaorong Huang
Fachbereich Informatik, Universität des Saarlandes
66041 Saarbrücken, Germany, email: huang@cs.uni-sb.de

Abstract

This paper presents *PROVERB* a text planner for argumentative texts. *PROVERB's* main feature is that it combines global hierarchical planning and unplanned organization of text with respect to local derivation relations in a complementary way. The former splits the task of presenting a particular proof into subtasks of presenting subproofs. The latter simulates how the next intermediate conclusion to be presented is chosen under the guidance of the local focus.

1. Introduction

This paper presents a text planner for the verbalization of natural deduction (ND) style proofs [Gen35]. Several similar attempts can be found in previous work. Developed before the era of NL generation, the system EXPOUND of D. Chester [Che76] can be characterized as an example of *direct translation*: Although a sophisticated linearization is applied on the input ND proofs, the steps are then translated locally in a template driven way. ND proofs were tested as input to an early version of the MUMBLE system of D. McDonald [McD83], the main aim however, was to show the feasibility of the architecture. A more recent attempt can be found in THINKER [EP93], which implements several interesting but isolated proof presentation strategies, without giving a comprehensive underlying model.

Our computational model can therefore be viewed as the first serious attempt at a comprehensive computational model that produces adequate argumentative texts from ND style proofs. The main aim is to show how existing text planning techniques can be adapted for this particular application. To test its feasibility, this computational model is implemented in a system called *PROVERB*.

Most current NL text planners assume that language generation is planned behavior and therefore adopt a hierarchical planning approach [Hov88, Moo89, Dal92, Rei91]. Nonetheless there is psychological evidence that language has an unplanned, spontaneous aspect as well [Och79]. Based on this observation, researchers have exploited organizing text with respect to some local relations. Sibun [Sib90] implemented a system generating descriptions for objects with a strong domain structure, such as houses, chips and families. Once a discourse is started, local structures suggest the next objects available. Instead of planning globally, short-range strategies are employed to organize a short segment of text. From a computational point of view, a hierarchical planner elaborates recursively on the initial communicative goal until the final subgoals can be achieved by applying a primitive operator. A text generator based on the local organization, in contrast, repeatedly chooses a part of the remaining task and carries it out.

The macroplanner of *PROVERB* combines *hierarchical planning* with *local organization* in a uniform planning framework. The hierarchical planning is realized by so-called top-down presentation operators that split the task of presenting a particular proof into subtasks of presenting subproofs. While the overall planning mechanism follows the RST-based planning approach [Moo89, Rei91], the planning operators more closely resemble the schemata in *schema-based* planning [McK85, Par88]. Bottom-up presentation operators are devised to simulate the unplanned aspect, where the next intermediate conclusion to be presented is chosen under the guidance of the local focus mechanism in a more spontaneous way. Since top-down operators embody explicit communicative norms, they are always given a higher priority. Only when no top-down presentation operator is applicable, will a bottom-up presentation operator be chosen.

This distinction between planned and unplanned presentation leads to a very natural segmentation of the discourse into an *attentional hierarchy*, since, following the theory of Grosz and Sidner [GS86], there is a one-to-one correspondence between the intentional hierarchy and the attentional hierarchy. This attentional hierarchy is used to make *reference choices* for inference methods and for previously presented intermediate conclusions. The inference choices are the main concern of the microplanner of *PROVERB*(see [Hua94b]).

2. Context of Our Research

The text planner discussed in this paper is the macroplanner of *PROVERB*, which translates machine-found proofs in several steps into natural language. *PROVERB* adopts a *reconstructive* approach: Once a proof in a machine oriented formalism is generated in the proof development environment Ω-MKRP, a new proof that more resembles those found in mathematical textbooks is reconstructed [Hua94a]. The reconstructed proof is a *proof tree*, where *proof nodes* are derived from their children by applying an inference method (also called a justification). Most of the steps are justified by the application of a definition

$$\cfrac{\cfrac{u(u_1, 1_u, *), u_1 \in U}{\boxed{2} : u_1 * 1_u = u_1}\text{Du}, \quad \cfrac{u_1 \in U, \cfrac{sgr(U, F)}{U \subset F}\text{Dsubgr}}{\boxed{3} : u_1 \in F}\text{Ds}, \quad \cfrac{U \subset F, \cfrac{u(U, 1_u, *)}{1_u \in U}\text{Du}}{\boxed{4} : 1_u \in F}\text{Ds}, \quad \cfrac{gr(F, *)}{segr(F, *)}\text{Dg},}{\boxed{1} : Solution(u_1, u_1, 1_u, F, *)}\text{Tsol}$$

Figure 1: An Example Input Proof

or a theorem, the rest are justified by inference rules of the natural deduction (ND) calculus, such as the "Case" rule. Figure 1 is an example of a segment of a possible input proof, where some nodes are labeled for convenience.
The justifications "Du", "Dsubgr", "Ds", "Dg", and "Tsol" stand for the definitions of unit element, of subgroup, of subset, of group, and the theorem about solution, respectively.

The input proof tree is also augmented with an ordered list of nodes, being roots of subproofs planned in this order. The proof in Figure 1 is associated with the list: ([2], [3], [4], [1]).

3. The Framework of the Macroplanner

The macroplanner of *PROVERB* elaborates on communicative goals, selects and orders pieces of information to fulfill these goals. The output is an ordered sequence of ***proof communicative act intentions*** (PCAs). PCAs can be viewed as ***speech acts*** in our domain of application.

Planning Framework

PROVERB combines the two above mentioned presentation modes by encoding communication knowledge for both top-down planning and bottom-up presentation in form of operators in a uniform planning framework. Since top-down presentation operators embody explicit communicative norms, they are given a higher priority. A bottom-up presentation is chosen only when no top-down presentation operator applies. The overall planning framework is realized by the function `Present`. Taking as input a subproof, `Present` repeatedly executes a basic planning cycle until the input subproof is conveyed. Each cycle carries out one presentation operator, where `Present` always tries first to choose and apply a top-down operator. If impossible, a bottom-up operator will be chosen. The function `Present` is first called with the entire proof as the presentation task. The execution of a top-down presentation operator may generate subtasks by calling it recursively. The discourse produced by each call to `Present` forms an attentional unit (compare the subsection below).

The Discourse Model and the Attentional Hierarchy

The discourse carried out so far is recorded in a *discourse model*. Rather than recording the semantic objects and their properties, our discourse model consists basically of the part of the input proof tree which has already been conveyed. The discourse model is also segmented into an *attentional hierarchy*, where subproofs posted by a top-down presentation operators as subtasks constitute attentional units. The following are some notions useful for the formulation of the presentation operators:

- *Task* is the subproof in the input proof whose presentation is the current task.
- *Local focus* is the intermediate conclusion last presented, while the semantic objects involved in the local focus are called the *focal centers*.

Proof Communicative Acts

PCAs are the primitive actions planned during the macroplanning to achieve communicative goals. Like speech acts, PCAs can be defined in terms of the communicative goals they fulfill as well as their possible verbalizations. Based on an analysis of proofs in mathematical textbooks, each PCA has as goal a combination of the following subgoals:

1. Conveying a step of the derivation. The simplest PCA is the operator `Derive`. Instantiated as below:

   ```
   (Derive Reasons:  (a ∈ S1, S1 ⊆ S2)
           Intermediate-Results: nil
           Derived-Formula:  a ∈ S2
           Method: def-subset)
   ```

   depending on the reference choices, a possible verbalization is given as following:

   > "Because a is an element of S_1 and S_1 is a subset of S_2, according to the definition of subset, a is an element of S_2."

2. Updates of the global attentional structure. These PCAs sometimes also convey a partial plan for the further presentation. Effects of this group of PCAs include: creating new attentional units, setting up partially premises and the goal of a new unit, closing the current unit, or reallocating the attention of the reader from one attentional unit to another. The PCA

   ```
   (Begin-Cases Goal:  Formula
                Assumptions:  (A B))
   ```

   creates two attentional units with A and B as the assumptions, and *Formula* as the goal by producing the verbalization:

   > "To prove *Formula*, let us consider the two cases by assuming A and B."

Thirteen PCAs are currently employed in *PROVERB*. See [Hua94b] for more details.

Structure of the Planning Operators

Although top-down and bottom-up presentation activities are of a conceptually different nature, the corresponding communication knowledge is uniformly encoded as *presentation operators* in a planning framework, similar to the plan operators in other generation systems [Hov88, Moo89, Dal92, Rei91]. In general, presentation operators map an original presentation task into a sequence of subtasks and finally into a sequence of PCAs. All of them have the following four slots:

- *Proof*: a proof schema, which characterizes the syntactical structure of a proof segment for which this operator is designed. It plays the role of the goal slot in the traditional planning framework.
- *Applicability Condition*: a predicate.
- *Acts*: a procedure which essentially carries out a sequence of presentation acts. They are either primitive PCAs, or are recursive calls to the procedure `Present` for subproofs.
- *Features*: a list of features which helps to select the best of a set of applicable operators.

4. Top-Down Planning

This section elaborates on the communicative norms concerning how a proof to be presented can be split into *subproofs*, as well as how the hierarchically-structured subproofs can be mapped onto some *linear* order for presentation. In contrast with operators employed in RST-based planners that split goals according to the rhetorical structures, our operators encode standard schemata for presenting proofs, which contain subgoals. The top-down presentation operators are roughly divided into two categories:

- schemata-based operators encoding complex schemata for the presentation of proofs of a specific pattern (twelve of them are currently integrated in *PROVERB*),
- general operators embodying general presentation norms, concerning splitting proofs and ordering subgoals.

$$\cfrac{\cfrac{\cfrac{\cdots}{}\\ \vdots}{?L_4 : \Delta \vdash F \vee G} \quad \cfrac{\cfrac{F \vdash F}{}\\ \vdots}{?L_2 : \Delta, F \vdash Q} \quad \cfrac{\cfrac{G \vdash G}{}\\ \vdots}{?L_3 : \Delta, G \vdash Q}}{?L_1 : \Delta \vdash Q}\ \text{CASE}$$

Figure 2: A Schema Involving Cases

Let us first look at an operator devised for proof segments containing cases. The corresponding schema of such a proof tree is shown in Figure 2. Under two circumstances a writer may recognize that he is confronted with a proof segment containing cases. First, when the subproof that has the structure of Figure 2 is the current presentation task, tested by (task $?L_1$)[1]. Second, when the disjunction $F \vee G$ has just been presented in the bottom-up mode, tested by (local-focus $?L_4$). Under both circumstances, a communication norm motivates the writer to first present the part leading to $F \vee G$ (in the second case this subgoal has already been achieved), and then to proceed with the two cases. It enforces also that certain PCAs be used to mediate between parts of proofs. This procedure is exactly captured by the presentation operator below.

Case-Implicit

- Proof: as given in Figure 2
- Applicability Condition: ((task $?L_1$) $\vee$ (local-focus $?L_4$)) $\wedge$ (not-conveyed ($?L_2$ $?L_3$))
- Acts:
  1. if $?L_4$ has not been conveyed, then present $?L_4$ (subgoal 1)
  2. a PCA with the verbalization: "First, let us consider the first case by assuming F."
  3. present $?L_2$ (subgoal 2)
  4. a PCA with the verbalization: "Next, we consider the second case by assuming G."
  5. present $?L_3$ (subgoal 3)
  6. mark $?L_1$ as conveyed
- features: (top-down compulsory implicit)

The feature values can be divided into two groups: those characterizing the style of the text this operator produces, and those concerning other planning aspects. "Implicit" is a stylistic feature value, indicating that the splitting of the proof into the three subgoals is not made explicit. In its explicit dual **Case-Explicit** a PCA is added to the beginning of the Acts slot, which produces the verbalization:

> "To prove Q, let us first prove $F \vee G$, and consider the two cases separately."

The feature value "compulsory" indicates that if the applicability condition is satisfied, and the style of the operator conforms to the global style the text planner is committed to, this operator should be chosen. Two weaker values also reflect the specificity of plan operators: "specific" and "general".

General presentation operators perform a simple task according to some general text organization principles. They either

- enforce a linearization on subproofs to be presented, or
- split the task of the presentation of a proof with ordered subproofs into subtasks.

[1] Labels stand for the corresponding nodes

The first ordering operator operationalizes a general ordering strategy called *minimal load principle*. This principle predicates that a writer usually presents shorter branches before longer ones. The argument of Levelt is rather simple: When one branch is chosen to be described first, the writer has to have the *choice node* flagged in his memory for return. If he follows the shorter branch first, the duration of the load will be shorter. The concrete operator is omitted.

Note that, the subproofs being ordered are subproofs conceptually planned while the corresponding proof is constructed. There are two other ordering operators based on general ordering principles: the *local focus* principle and the *proof time order* principle [Hua94b].

The invocation of an ordering operator is always followed by the invocation of a splitting operator, which actually posts subgoals by calling the function `Present` with the ordered goals subsequently.

5. Bottom-up Presentation

The *bottom-up presentation* process simulates the unplanned part of proof presentation. Instead of splitting presentation goals into subgoals according to standard schemata, it follows the local derivation relation to find a next proof node or subproof to be presented. In this sense, it is similar to the local organization techniques used in [Sib90]. When no top-down presentation operator applies, *PROVERB* chooses a bottom-up operator.

The Local Focus

The node to be presented next is suggested by the mechanism of *local focus*. Although logically any proof node having the local focus as a child could be chosen for the next step, usually the one with the greatest semantic overlapping with the *focal centers* is preferred. As mentioned above, focal centers are semantic objects mentioned in the proof node which is the local focus. This is based on the observation that if one has proved a property about some semantic objects, one tends to continue to talk about these particular objects before turning to new objects. Let us examine the situation when the proof below is awaiting presentation.

$$\frac{\dfrac{[1]: P(a,b)}{[2]: Q(a,b)}, \quad \dfrac{[1]: P(a,b), \ [3]: S(c)}{[4]: R(b,c)}}{[5]: Q(a,b) \wedge R(b,c)}$$

Assume that node [1] is the local focus, the set $\{a, b\}$ are the focal centers, [3] is a previously presented node and node [5] is the current task. [2] is chosen as the next node to be presented, since it does not (re)introduce any new semantic object and its overlap with the focal centers ($\{a, b\}$) is larger than those of [4] ($\{b\}$).

The Bottom-Up Presentation Operators

Under different circumstances the derivation of the next-node is also presented in different ways. The corresponding presentation knowledge is encoded as bottom-up presentation operators. The one most frequently used presents one step of derivation:

Derive-Bottom-Up

- Proof: $\frac{?Node_1, \ldots, ?Node_n}{?Node_{n+1}} ?M$
- Applicability Condition: $?Node_{n+1}$ is suggested by the focus mechanism as the next node, and $?Node_1$, ..., $?Node_n$ are conveyed.
- Acts: a PCA that conveys the fact that $?Node_{n+1}$ is derived from the premises $?Node_1$, ..., $?Node_n$ by applying $?M$.
- Features: (bottom-up general explicit detailed)

If the conclusion $?Node_{n+1}$, the premises and the method $?M$ are instantiated to $a \in S_1$, ($a \in S_2$, $S_1 \in S_2$), *def-subset* respectively, the following verbalization can be produced:

> "Since a is an element of S_1, and S_1 is a subset of S_2, a is an element of S_2 according to the definition of subset."

A *trivial* subproof may be presented as a single derivation by omitting the intermediate nodes. This *next subproof* is also suggested by the local focus. This is simulated by a bottom-up operator called **Simplify-Bottom-Up**. Currently seven bottom-up operators are integrated in *PROVERB*.

6. Verbalization of PCAs

Macroplanning produces a sequence of PCAs. Our microplanner is restricted to the treatment of the *reference choices* for the inference methods and for the previously presented intermediate conclusions. While the former depends on static salience relating to the domain knowledge, the latter is similar to subsequent references, and is therefore sensitive to the context, in particular to its segmentation into attentional hierarchy. Due to space restrictions, we only show the following piece of a *preverbal message* as an example, being a PCA enriched with reference choices for reasons and method by the microplanner [Hua94b, Hua94b].

```
(Derive Reasons: (((ELE a U) explicit)
                  ((SUBSET U F) omit))
        Conclusion: (ELE a F)
        Method: (Def-Subset omit))
```

Our surface generator TAG-GEN [Kil94] produces the utterance:

> "Since a is an element of U, a is an element of F."

Notice, only the reason labeled as "explicit" is verbalized.

Finally, to demonstrate the type of proofs currently generated by *PROVERB*, below is the complete output for a proof constructed by Ω-MKRP:

Theorem: Let F be a group and U a subgroup of F, if 1 and 1_U are unit elements of F and U respectively, then $1 = 1_U$.

Proof:

Let F be a group, U be a subgroup of F, 1 be a unit element of F and 1_U be a unit element of U. According to the definition of unit element, $1_U \in U$. Therefore there is an X, $X \in U$. Now suppose that u_1 is such an X. According to the definition of unit element, $u_1 * 1_U = u_1$. Since U is a subgroup of F, $U \subset F$. Therefore $1_U \in F$. Similarly $u_1 \in F$, since $u_1 \in U$. Since F is a group, F is a semigroup. Because $u_1 * 1_U = u_1$, 1_U is a solution of the equation $u_1 * X = u_1$. Since 1 is a unit element of F, $u_1 * 1 = u_1$. Since 1 is a unit element of F, $1 \in F$. Because $u_1 \in F$, 1 is a solution of the equation $u_1 * X = u_1$. Since F is a group, $1_U = 1$ by the uniqueness of solution. This conclusion is independent of the choice of the element u_1.

7. Conclusion and Future Work

This paper puts forward an architecture that combines several established NL generation techniques adapted for a particular application, namely the presentation of ND style proofs. We hope that this architecture is also of general interest beyond this particular application.

The most important feature of this model is that hierarchical planning and unplanned spontaneous presentation are integrated in a uniform framework. Top-down hierarchical planning views language generation as planned behavior. Based on explicit communicative knowledge encoded as schemata, hierarchical planning splits a presentation task into subtasks. Although our overall presentation mechanism has much in common with that of RST-based text planners, the top-down planning operators contain mostly complex presentation schemata, like those in schema-based planning. Since schemata-based planning covers only proofs of some particular structure, it is complemented by a mechanism called bottom-up presentation. Bottom-up presentation aims at simulating the unplanned part of proof presentation, where a proof node or a subproof awaiting presentation is chosen as the next to be presented via the local derivation relations. Since more than one such node is often available, the local focus mechanism is employed to single out the candidate having the strongest semantic links with the focal centers. The distinction between planned and unplanned behavior enables a very natural segmentation of the discourse into an attentional hierarchy. This provide an appropriate basis for a discourse theory which handles reference choices [Hua94b].

Compared with proofs found in mathematical textbooks, the output of *PROVERB* is still too tedious and inflexible. The tediousness is largely ascribed to the lack of ***plan level*** knowledge of the input proofs, which distinguishes crucial steps from unimportant details. Therefore, sophisticated plan recognition techniques are necessary. The inflexibility of text currently produced is partly inherited from the schemata-based approach, for which a fine-grained planning in terms of single PCAs might be a remedy. It is also partly due to the fixed lexicon choice, which we are currently reimplementing.

References

[Che76] D. Chester. The translation of formal proofs into English. *Artificial Intelligence*, 1976.

[Dal92] R. Dale. *Generating Referring Expressions*. MIT Press, 1992.

[EP93] A. Edgar and F. J. Pelletier. Natural language explanation of natural deduction proofs. In *Proc. of the first Conf. of the Pacific Assoc. for Comp. Linguistics*, 1993.

[Gen35] G. Gentzen. Untersuchungen über das logische Schließen I. *Math. Zeitschrift*, 1935.

[GS86] B. J. Grosz and C. L. Sidner. Attention, intentions, and the structure of discourse. *Computational Linguistics*, 1986.

[Hov88] E. H. Hovy. *Generating Natural Language under Pragmatic Constrints*. Lawrence Erlbaum Associates, Hillsdale, 1988.

[Hua94a] X. Huang. Reconstructing proofs at the assertion level. In *Proc. of 12th CADE*, 1994, forthcoming.

[Hua94b] X. Huang. Planning Reference Choices for Argumentative Texts. In *Proc. of the 7th International Workshop on Natural Language Generation*, 1994, forthcoming.

[Hua94b] X. Huang. *A Reconstructive Approach to Human Oriented Proof Presentation*. PhD thesis, Universität des Saarlandes, Germany, 1994, forthcoming.

[Kil94] A. Kilger. Using UTAGs for incremental and parallel generation. *Computational Intelligence*, forthcoming, 1994.

[McD83] D. D. McDonald. Natural language generation as a computational problem. In *Brady/Berwick: Computational Models of Discourse*. MIT Press, 1983.

[McK85] K. R. McKeown. *Text Generation*. Cambridge University Press, 1985.

[Moo89] J. D. Moore. *A Reactive Approach to Explanation in Expert and Advice-Giving Systems*. PhD thesis, Univ. of California, 1989.

[Och79] E. Ochs. Planned and unplanned discourse. *Syntax and Semantics*, 1979.

[Par88] C. Paris. Tailoring object descriptions to a user's level of expertise. *Computational Linguistics*, 1988.

[Rei91] N. Reithinger. *Eine parallele Architektur zur inkrementeller Dialogbeiträge*. PhD thesis, Universität des Saarlandes, 1991.

[Sib90] P. Sibun. The local organization of text. In K.R. McKeown etal, editors, *Proc. of the 5th International Workshop on Natural Language Generation*, 1990.