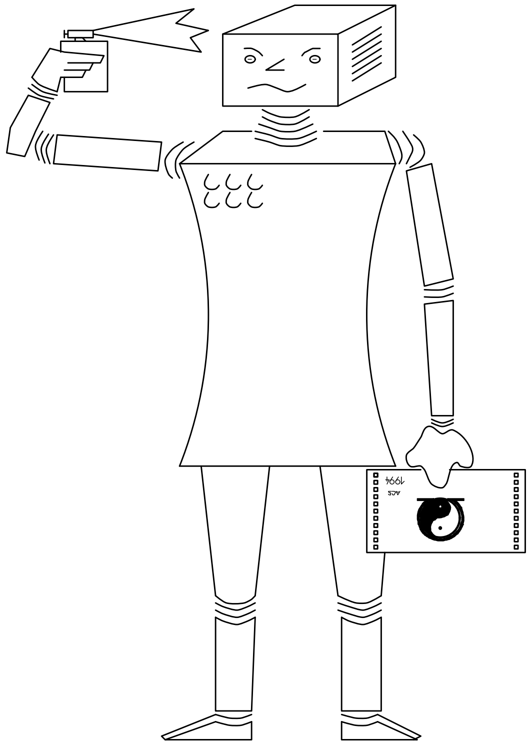
OMEGA Publication



Proof Verbalization in *PROVERB*

Xiaorong Huang and Armin Fiedler

Internet Copy

Published as: Proceedings of the First International Workshop on Proof Transformation and Presentation

Proof Verbalization in *PROVERB*

Xiaorong Huang   Armin Fiedler

Extended Abstract

This paper outlines the linguistic part of an implemented system named *PROVERB*[3] that transforms, abstracts, and verbalizes machine-found proofs in natural language. It aims to illustrate, that state-of-the-art techniques of natural language processing are necessary to produce coherent texts that resemble those found in typical mathematical textbooks, in contrast to the belief that mathematical texts are only schematic and mechanical.

The verbalization module consists of a *content planner*, a *sentence planner*, and a *syntactic generator*. Intuitively speaking, the *content planner* first decides the order in which proof steps should be conveyed. It also some messages to highlight global proof structures. Subsequently, the *sentence planner* combines and rearranges linguistic resources associated with messages produced by the content planner in order to produce connected text. The *syntactic generator* finally produces the surface text.

Content Planning

Mainly two kinds of knowledge are incorporated into the content planner in the form of presentation operators. The *hierarchical planning* splits the task of presenting a particular proof into subtasks of presenting subproofs. *Local navigation* operators simulates the unplanned aspect, where the next conclusion to be presented is chosen under the guidance of a local focus mechanism. The two kinds of *planning operators* are treated differently. Since hierarchical planning operators embody explicit communicative norms, they are given a higher priority. Only when none of them is applicable, will a local navigation operator be chosen. The output of the content planner is an ordered sequence of *proof communicative acts* (PCAs), structured in a attentional hierarchy.

PCAs are the primitive actions planned during the content planning to achieve communicative goals. They can be defined in terms of the communicative goals they fulfill as well as in terms of their possible verbalizations. Based on an analysis of proofs in mathematical textbooks, there are mainly two types of goals a PCA is generated to achieve:

Conveying a step of derivation: In terms of rhetorical relations, PCAs in this category represent a variation of the rhetorical relation *derive*. Below is an example of the simplest PCA of this sort called `Derive`.

```
(Derive Reasons:  (a ∈F, F ⊆G)
        Method: def-subset
        Conclusion:  a ∈G)
```

Depending on the reference choices, a possible verbalization is “Since a is an element of F and F is a subset of G, a is an element of G by the definition of subset.”

Updating the global attentional structure: These PCAs either convey a partial plan for the forthcoming discourse or signal the end of a subproof.

The PCA

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

produces the verbalization: “To prove *Formula*, let us consider the two cases by assuming A and B.”

See [1] for further details.

Sentence Planning

The task of sentence planning comprises, among others, making reference choices; choosing between linguistic resources for functions, predicates and various types of derivations; and combining and reorganizing such resources into paragraphs and sentences.

Many of the first natural language generation systems link their information structure to the corresponding linguistic resources either through predefined templates or via careful engineering for a specific application. Therefore their expressive power is restricted. First experiments with *PROVERB* using a simplistic sentence planning mechanism resulted in fairly mechanical texts. According to our analysis, there are at least two linguistic phenomena that call for appropriate sentence planning techniques.

First, naturally occurring proofs contain paraphrases of rhetorical relations, as well as of logical functions or predicates. For instance, the derivation of B from A can be verbalized as "Since A, B." or as "A leads to B."

The logic predicate $para(C1, C2)$, also, can be verbalized as "Line $C1$ parallels line $C2$." or as "The parallelism of the lines $C1$ and $C2$."

Second, with only a simple sentence planner *PROVERB* generates text structured exactly mirroring the information structure of the proof and the formulae. This means that every step of derivation is translated into a separate sentence, and formulae are recursively verbalized. As an instance of the latter, the formula $Set(F) \wedge Subset(F, G)$ is verbalized as "F is a set. F is a subset of G." although the following is much more natural: "The set F is a subset of G."

We obtained this flexibility by introducing an intermediate level of representation called *Text Structure*. In *PROVERB*, the Text Structure is organized as a tree, in which each node represents a constituent of the text. A typing mechanism ensures that the planner only build expressible Text Structures. For instance, if tree A should be expanded at node n by tree B, the resulting type of B must be compatible to the type restriction attached to n. The sentence planner essentially maps PCAs as well as the functions and predicates in the PCAs into Text Structure subtrees in a two-staged way and combines and rearranges them into a single Text Structure. See [2] for further details.

The Text Structure serves as linguistic specification and is passed on to the syntactic generator, which finally produces the surface text.

Example

In this section, we present a short example of *PROVERB*'s output. The input is a machine-found proof for a theorem taken from a mathematical textbook. *PROVERB*'s output is as follows:

Theorem:

Let F be a group, let U be a subgroup of F, and let 1 and 1_U be unit elements of F and U. Then 1_U equals 1.

Proof:

Let F be a group, let U be a subgroup of F, and let 1 and 1_U be unit elements of F and U.

Because 1_U is an unit element of U, $1_U \in U$. Therefore, there is x such that $x \in U$.

Let $u1$ be such an x. Since $u1 \in U$ and 1_U is an unit element of U, $u_1 * 1_U = u_1$. Since F is a group, F is a semigroup. Since U is a subgroup of F, $U \subset F$. Because $U \subset F$ and $1_U \in U$, $1_U \in F$. Similarly, because $u_1 \in U$ and $U \subset F$, $u_1 \in F$. Then, 1_U is a solution of $u_1 * x = u_1$.

Because $u_1 \in F$ and 1 is an unit element of F, $u_1 * 1_U = u_1$. Since 1 is an unit element of F, $1 \in F$. Then, 1 is a solution of $u_1 * x = u_1$.

Therefore, 1_U equals 1. This conclusion is independent of the choice of $u1$. ∎

References

[1] Xiaorong Huang. Planning argumentative texts. In *Proceedings of 15th International Conference on Computational Linguistics*, 1994.

[2] Xiaorong Huang and Armin Fiedler. Paraphrasing and aggregating argumentative texts using text structure. In *Proceedings of the 8th International Natural Language Generation Workshop*, 1996.

[3] Xiaorong Huang and Armin Fiedler. Presenting machine-found proofs. In M.A. McRobbie and J.K Slaney, editors, *Proceedings of the 13th Conference on Automated Deduction*, 1996.