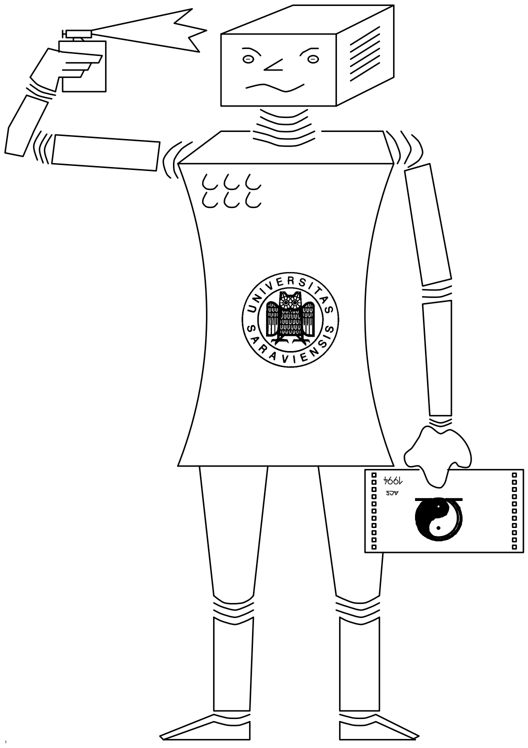
UNIVERSITÄT DES SAARLANDES
FACHBEREICH INFORMATIK
D-66041 SAARBRÜCKEN
GERMANY
WWW: http://jswww.cs.uni-sb.de/pub/seki/



Presenting Machine-Found Proofs

Xiaorong Huang and Armin Fiedler

Published as: Proc. of 13th International Conference on Automated Deduction

Presenting Machine-Found Proofs

Xiaorong Huang Armin Fiedler

Fachbereich Informatik, Universität des Saarlandes
Postfach 15 11 50, D-66041 Saarbrücken, Germany
{huang,afiedler}@cs.uni-sb.de

Abstract. This paper outlines an implemented system named *PROVERB* that transforms and abstracts machine-found proofs to natural deduction style proofs at an adequate level of abstraction and then verbalizes them in natural language. The abstracted proofs, originally employed only as an intermediate representation, also prove to be useful for proof planning and proving by analogy.

1 Introduction

This paper outlines an implemented system named *PROVERB* that presents and verbalizes machine-found natural deduction proofs in natural language. Apart from its practical use, we hope *PROVERB* will also address some cognitive aspects of proof representation and proof presentation.

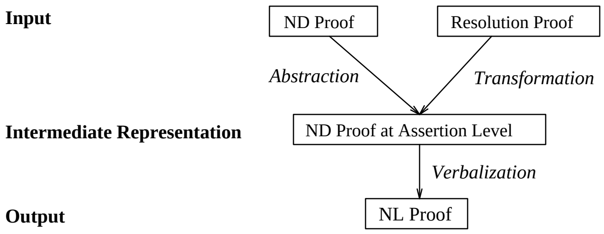
Efforts have been made to transform proofs from machine-oriented formalisms into a more natural formalism [And80, Pfe87, Lin90]. As the target formalism, usually a variation of the natural deduction (ND) proof proposed by Gentzen [Gen35] is chosen. The resulting ND proofs are then used as inputs by natural language generators [Che76, EP93]. In general, the presentation of ND proofs has been performed through *ordering*, *pruning*, and *augmentation*.

All of these verbalizations suffer from the same problem: The derivations they convey are exclusively at the level of the inference rules of the ND calculus. In contrast to the informal proofs found in standard mathematical textbooks, such proofs are composed of derivations familiar from elementary logic, where the focus of attention is on syntactic manipulations rather than on the underlying semantic ideas. The main problem, we believe, lies in the absence of intermediate structures in ND proofs, that allow atomic justifications at a higher level of abstraction.

To incorporate the more abstract justifications given in mathematical textbooks, we have defined the concept of *assertion level* inference rules [Hua94b]. Derivations justified by these rules can be understood intuitively as the application of definitions or a theorems (collectively called *assertions*). In this paper, we illustrate how *PROVERB* transforms machine-found proofs into this intermediate representation, and how up-to-date techniques of natural language processing can be used to produce coherent text.

2 Overview of *PROVERB*

The entire architecture of *PROVERB* when used as a stand-alone system is sketched out below.



The input to the first version of *PROVERB* was restricted to ND proofs. Within the proof development system Ω–MKRP [HKK+94], such input is prepared by other components that translate proofs in machine-oriented formalisms like resolution into ND proofs. The ND proofs are first raised to a more adequate level of abstraction by the *Abstraction* module, before techniques of natural language generation are used by the *Verbalization* module to produce the final proof in natural language (NL).

The output of those transformation components in Ω–MKRP is often not satisfactory. Moreover, we have established a correspondence between resolution proofs and ND proofs at a more abstract level (see [Hua96]), namely in terms of the application of assertions. We are incorporating a new *Transformation* component into *PROVERB*. Currently, we have integrated a preliminary version of our algorithm which transforms resolution proofs directly into ND proofs at the assertion level.

3 The Assertion Level as an Intermediate Representation

3.1 Abstraction to the Assertion Level

If we examine a mathematical textbook carefully, it is not difficult to see that most inference steps are justified in terms of the application of an assertion (i.e., a definition, an axiom, or a theorem). For instance, $a_1 \in F_1$ can be inferred from $U_1 \subset F_1$ and $a_1 \in U_1$ by the application of the definition of subset. This atomic step, however, is often given in the input as a compound ND proof segment like following one:

$$\frac{\dfrac{\dfrac{\dfrac{\dfrac{\forall S_1, S_2{\scriptstyle\blacksquare}\, S_1 \subset S_2 \Leftrightarrow (\forall x{\scriptstyle\blacksquare}\, x \in S_1 \Rightarrow x \in S_2)}{U_1 \subset F_1 \Leftrightarrow (\forall x{\scriptstyle\blacksquare}\, x \in U_1 \Rightarrow x \in F_1)}\forall E}{U_1 \subset F_1 \Rightarrow (\forall x{\scriptstyle\blacksquare}\, x \in U_1 \Rightarrow x \in F_1)}\Leftrightarrow E, \quad U_1 \subset F_1}{\forall x{\scriptstyle\blacksquare}\, x \in U_1 \Rightarrow x \in F_1}\Rightarrow E}{a_1 \in U_1 \Rightarrow a_1 \in F_1}\forall E, \quad a_1 \in U_1}{a_1 \in F_1}\Rightarrow E$$

In [Hua94b], the intuitive notion of the application of an assertion was formalized. Using this formalization, *PROVERB* usually substantially shortens input

ND proofs by *abstracting* them to the assertion level. This is achieved by replacing compound proof segments like the above one by atomic derivations that are justified by assertion level rules like the following one.

$$\frac{U_1 \subset F_1, a_1 \in U_1}{a_1 \in F_1} \textit{Def-Subset.}$$

3.2 From Resolution Directly to ND Proof at the Assertion Level

The quality of proofs produced by the *Abstraction* component depends heavily on the ND proofs transformed from other proof formalisms. Unfortunately, the quality of these proofs not always satisfactory. In [Hua96], we showed that a resolution proof of a certain structure (*SSPU*-resolution, which stands for *u*nit *r*esoultion for a *s*imple *s*tructured *p*roblem) is basically a sequence of applications of assertions.

The set of initial clauses:
C1 $= \{+(a * a^{-1} = e)\}$ C2 $= \{+(e * a^{-1} = a^{-1})\}$
C3 $= \{-(x \in S), -(y \in S), -(x * y^{-1} = z), +(z \in S)\}$
C4 $= \{+(a \in S)\}$ C5 $= \{-(a^{-1} \in S)\}$
The resolution steps:
C3,1 & C4,1 : add R1: $\{-(y \in S), -(a * y^{-1} = z), +(z \in S)\}$
R1,1 & C4,1 : add R2: $\{-(a * a^{-1} = z), +(z \in S)\}$
R2,1 & C1,1 : add R3: $\{+(e \in S)\}$
C3,2 & C4,1 : add R4: $\{-(x \in S), -(x * a^{-1} = z), +(z \in S)\}$
R4,2 & C2,1 : add R5: $\{-(e \in S), +(a^{-1} \in S)\}$
R3,1 & R5,1 : add R6: $\{+(a^{-1} \in S)\}$
R6,1 & C5,1 : add R7: $\square$

For instance, the *SSPU*-resolution above contains two applications of C3, which is one of the group criteria. First the sequence R1, R2, R3 derives $e \in S$ from the premises $a \in S$ and $a * a^{-1} = e$. Second the sequence R4, R5, R6 derives $a^{-1} \in S$ using as premises $a \in S$ and $e * a^{-1} = a^{-1}$. The *Transformation* component of *PROVERB* produces the following ND proof at the assertion level, where line 6 and line 7 correspond to the application steps above. Note that C3 is the CNF of line 1 below.

No	Hyps		Formula	Reason
1.	;1	$\vdash$	$\forall x, y, z_\blacksquare\, x \in S \land y \in S \land x * y^{-1} = z \Rightarrow z \in S$	(GrpCrit)
2.	;2	$\vdash$	$a * a^{-1} = e$	(Hyp)
3.	;3	$\vdash$	$e * a^{-1} = a^{-1}$	(Hyp)
4.	4;	$\vdash$	$a \in S$	(Hyp)
5.	2,1,4;	$\vdash$	$e \in S$	(GrpCrit 4 4 2)
6.	2,3,1,4;	$\vdash$	$a^{-1} \in S$	(GrpCrit 5 4 3)

Since resolution proofs are not always a *SSPU*-resolution, the *Transformation* component often has to split an arbitrary input resolution proof into *SSPU-refutable* subproofs, and then reorder them into *SSPU*-resolution proofs [Hua96]. Note that the splitting will always produce a collection of *SSPU*-resolution proofs, since resolution proofs consisting of only unit clauses are degenerate forms of *SSPU*-resolution.

4 From ND Proof to NL Proof

This section aims to illustrate, to the automated reasoning community, why state-of-the-art techniques of natural language processing are necessary to produce coherent texts that resemble those found in typical mathematical textbooks. Readers are referred to [Hua94a, HF96] for technical details. The *Verbalization* module consists of a *content planner* and a *sentence planner*. Intuitively speaking, the *content planner* first decides the order in which proof steps should be conveyed. It also produces proof communicative acts (PCAs), which highlight global proof structures. Subsequently, the *sentence planner* combines and rearranges linguistic resources associated with subsequent PCAs in order to produce more connected text.

4.1 Content Planning

Mainly two kinds of knowledge are incorporated into the content planner in the form of presentation operators. The *top-down presentation* operators split the task of presenting a particular proof into subtasks of presenting subproofs. *Bottom-up* presentation operators, on the other hand, 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 this paper, we will look at only one top-down presentation operator, which embodies a communicative norm concerning proofs in terms of case analysis. The corresponding schema of such a proof tree is shown below,

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

where the subproof rooted at $?L_4$ leads to $F \vee G$, while the subproofs rooted at $?L_2$ and $?L_3$ are the two cases that prove Q by assuming F and G, respectively. In *PROVERB*, there is a presentation operator that essentially suggests that the system present first the part leading to $F \vee G$ and then to proceed with the two cases. This operator also requires that certain PCAs be used to mediate between parts of a proof. The concrete operator is omitted because of space restrictions. The user may define a global style that will choose among competing operators.

4.2 Sentence Planning

The task of sentence planning comprises, among other subtasks, those of combining and reorganizing of linguistic resources associated with functions and predicates, and various types of derivations [HF96]. The first version of *PROVERB*, for example, generates one sentence for every step of a derivation. The below corresponds to two inference steps:

"We can derive $\sigma \subset \sigma^*$ by the definition of transitive closure. Since $(x, y) \in \sigma$ and $\sigma \subset \sigma^*$, $(x, y) \in \sigma^*$ by the definition of subset."

From the same input, the present version of *PROVERB* now produces a more connected text:

"We can derive $\sigma \subset \sigma^*$ by the definition of transitive closure, which gives us $(x, y) \in \sigma^*$ by the definition of subset, since $(x, y) \in \sigma$."

Another combination concerns conjunctive formulae. Instead of

"F is a set. F is a subset of G."

PROVERB now produces the following sentence:

"The set F is a subset of G."

The current version of *PROVERB* produces the following natural language proof from the resolution proof given in section 3.2:

Proof:

Let $a * a^{-1} = e$, $e * a^{-1} = a^{-1}$ and $a \in S$. Then $e \in S$ by the group criterion. Similarly, $a^{-1} \in S$. ■

5 Current State and Future Work

The components described in this paper are implemented within the proof development environment Ω–MKRP. The system runs fully automatically. On the linguistic side, however, we are still working on a more comfortable interface that will help the user with the edition of linguistic resources.

References

[And80] Peter B. Andrews. Transforming matings into natural deduction proofs. In *Proc. of the 5th CADE*, 1980.

[Che76] Daniel Chester. The translation of formal proofs into English. *AI*, 7:178–216, 1976.

[EP93] Andrew Edgar and Francis Jeffry Pelletier. Natural language explanation of natural deduction proofs. In *Proc. of the first Conference of the Pacific Association for Computational Linguistics*, 1993.

[Gen35] Gerhard Gentzen. Untersuchungen über das logische Schließen I. *Math. Zeitschrift*, 39:176–210, 1935.

[HF96] Xiaorong Huang and Armin Fiedler. Paraphrasing and aggregating argumentative text using text structure. In *Proc. of 8th International Workshop on Natural Language Generation*, 1996.

[HKK+94] Xiaorong Huang et al. Ω–MKRP – a proof development environment. In *Proc. of 12th CADE*, 1994.

[Hua94a] Xiaorong Huang. Planning argumentative texts. In *Proc. of 15th International Conference on Computational Linguistics*, 1994.

[Hua94b] Xiaorong Huang. Reconstructing proofs at the assertion level. In *Proc. of 12th CADE*, 1994.

[Hua96] Xiaorong Huang. Translating machine-generated resolution proofs into nd-proofs at the assertion level. In *Proc. of PRICAI-96*, 1996.

[Lin90] Christoph Lingenfelder. *Transformation and Structuring of Computer Generated Proofs.* PhD thesis, Universität Kaiserslautern, 1990.

[Pfe87] Frank Pfenning. *Proof Transformation in Higher-Order Logic.* PhD thesis, Carnegie Mellon University, 1987.