

How to Build Explanations of Automated Proofs

A Methodology and Requirements on Domain Representations

Helmut Horacek

Department of Computer Science, Saarland University
P. O. Box 15 11 50, D-66041 Saarbrücken, Germany
horacek@ags.uni-sb.de

Abstract

There is ample evidence that results produced by problem-solving methods and ingredients suitable for human-adequate explanations may differ fundamentally, which makes documenting the behavior of intelligent systems and explaining the solutions they produce quite challenging. Focusing on the explanation of solutions found by the most general problem-solvers, automated theorem provers, we sketch what has emerged as a methodology over the past decade in our working group for building content specifications for these kind of explanations. This methodology is conceived as a stratified model with dedicated transformation processes bridging between adjacent strata. Our investigations have shown that explanation capabilities based on problem-solving knowledge only are limited in a number of ways, which motivates one to represent extra knowledge relevant for communication purposes.

Introduction

Explanation capabilities are generally considered important for knowledge-based systems, since the rationale behind the solutions they find may not always be sufficiently clear to humans. However, the provision of material that adequately serves the purpose of human users may be quite difficult, even for well-documented reasoning processes, because the demands of human cognition are rather different from those of machine-based reasoning. Therefore, a variety of conceptually different transformation processes are required to make the results produced by the machine easily accessible to humans. This is especially the case when the machine reasoning is done on a level that is quite different from that of human experts, which particularly holds for automated theorem provers (ATPs).

Various methods have been proposed for proof explanations, mostly direct translations. Conceptual and/or linguistic requirements have rarely been addressed in a principled manner. In our work, we have investigated a variety of issues that contribute to a human-oriented exposition of results produced by a theorem prover. In this paper, we describe how to orchestrate them in a systematic manner, and we elaborate demands on the knowledge representation of ATPs.

Copyright © 2007, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

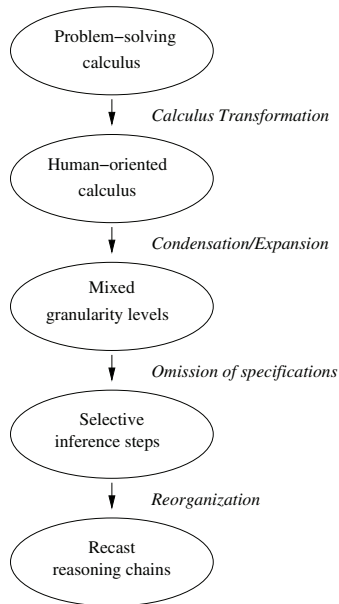
This paper is organized as follows. First, we motivate the goals underlying our approach. In the main part of the paper, we present the general framework, which comprises four subprocesses, and we describe their orchestration. Finally, we formulate requirements on the knowledge representation of theorem provers to further increase explanation capabilities.

Motivation

Whereas explanations in general may refer to a variety of issues, including the provision of background information and a focused elaboration of crucial arguments, we restrict our attention to the human-adequate description of complete solutions produced by an ATP. In order to be useful to humans, such explanations must be in accordance with their cognitive needs as much as possible. There is ample evidence that the needs of humans are quite different from representation needs for machine-based reasoning, so that it makes a huge difference in the exploitation of a system's capabilities whether or not the results produced by a machine are recast according to the needs of its users. Negative examples include chess endgame databases, such as king plus rook and pawn versus king plus rook. They can serve as a perfect look-up table, but even grandmasters seem to be puzzled about the underlying rationale. A positive example is the study by Di Eugenio *et al.* (2005), which shows significant performance improvements in learning, when using human-adequate presentations. In that study, the results of the expert system examined are quite cumbersome and unfocused when presented directly to its users as explanations. Through recasting the underlying facts according to commonalities, thereby enabling aggregation of facts sharing properties and rhetorically adequate organization of groups of facts, the authors achieved considerable presentation improvements that led to a significant performance gain.

In this paper, we argue that an even wider variety of measures with stronger impact on underlying representation structures are required to make solutions found by an ATP easily understandable to humans. We elaborate what we consider the major discrepancies between solution representations produced by an ATP and communicative needs of humans. These discrepancies motivate the transformation steps of our methodology, which we describe in the next section. They concern the following three properties:

Figure 1: Architecture for building content specifications



- *Uniformity versus Diversification*

Whereas reasoning within machines works best on a uniform level of representation, the resolution calculus being the most prominent example, humans frequently mix up levels of representation in their argumentation to meet the expected skills in understanding. Within the calculus of *Natural Deduction*, for instance, cognitively difficult reasoning patterns lead to incorrect conclusions more often than simple patterns. In fact, psychological experiments have shown substantial differences among the ease of understanding deductive syllogisms: approx. 90% correct inferences for modus ponens, as opposed to around 50% only for modus tollens and disjunction, according to Johnson-Laird and Byrne (1990). Thus, inferences following cognitively difficult patterns need more detailed descriptions to be understood properly (Walker 1996).

- *Explicitness versus Selectivity*

For machine-based reasoning, all pieces of information for drawing inferences must be represented explicitly. For humans, presentations in this form are frequently perceived as redundant and boring, so that people tend to forget the essential information found between the descriptions perceived as redundant. In contrast, humans prefer arguments containing only some part of the underlying reasoning pattern (“modus brevis” according to Sadock (1977)), even with intermediate arguments not mentioned at all, when they can be recovered on the basis of contextual expectations (Thüring and Wender 1985). Another aspect is the use of presentation means that exploit commonalities across arguments, similar to the approach by Di Eugenio *et al.* (2005), provided compact natural language forms to express common parts are available.

Figure 2: The definition of the Steamroller problem

Axioms:

1. Wolves, foxes, birds, caterpillars, and snails are animals, and there are some of each of them.
2. Also there are some grains, and grains are plants.
3. Every animal either likes to eat all plants or all animals much smaller than itself that like to eat some plants.
4. Caterpillars and snails are much smaller than birds, which are much smaller than foxes, which in turn are much smaller than wolves.
5. Wolves do not like to eat foxes and grains.
6. Birds like to eat caterpillars, but not snails.
7. Caterpillars and snails like to eat some plants.

Theorem:

8. Therefore, there is an animal that likes to eat a grain-eating animal.

- *Technical versus Conceptual/linguistic references*

References to subarguments in a reasoning structure, as virtually all internal references in computers, are realized in a straightforward manner by some sort of pointers. Obviously, this measure is inaccessible to humans, and it is naturally replaced by some sort of naming reference to pieces of domain knowledge that make up an argument. The more these references are elaborated, and the more background knowledge is taken into account for their realization, the more easily understood are explanations in which such references are embedded.

In the following, we will describe techniques for building the content specifications of explanations that take into account the discrepancies outlined above, so that representation differences are bridged as well as possible.

Building the Content of Explanations

In the scope of this paper, we only address methods for building the content of explanations, not techniques for expressing these specification in natural language. In principle, presentations based on the content representation of automated proofs could be realized in full-fledged natural language, in some pattern-based pseudo natural language or even in terms of some sort of graphical and textual display.

We propose a stratified process to tailor the content of explanations originating from problem solving results produced by an ATP. The architecture is depicted in Figure 1, with individual strata enclosed by ellipses. As a running example, we use the Steamroller problem (Stickel 1986), see the definition in Figure 2. In a nutshell, the proof runs along the following lines: through applying the central axiom (3.) three times, it is first derived that birds eat grain, then that foxes do not eat grain and, finally, that foxes eat the smaller grain-eating birds, the last being the witness needed to prove the theorem (8.) to be true.

Figure 3: Expanding an *Assertion Level* step in the solution to the Steamroller problem to the *Partial Assertion Level*

$$\begin{array}{l}
 1. \quad \frac{Eats(w, g) \vee ((Eats(f, g) \wedge (f < w)) \rightarrow Eats(w, f)) \quad \neg Eats(w, g) \quad \neg Eats(w, f) \quad f < w}{\neg Eats(f, g)} \text{Assertion} \\
 \quad \quad \quad \downarrow \\
 2. \quad \frac{\frac{Eats(w, g) \vee ((Eats(f, g) \wedge (f < w)) \rightarrow Eats(w, f)) \quad \neg Eats(w, g)}{(Eats(f, g) \wedge (f < w)) \rightarrow Eats(w, f)} \text{DE} \quad \neg Eats(w, f)}{\neg Eats(f, g) \vee \neg(f < w)} \text{MT} \quad f < w}{\neg Eats(f, g)} \text{DE}
 \end{array}$$

Reaching a Human-Oriented Reasoning Level

In this phase, representations in a machine-oriented calculus – typically resolution graphs – are transduced into a system of human-oriented inferences – mostly based on *Natural Deduction*. In principle, a resolution proof itself could also be subject to an explanation, but the conceptual means to tailor its content according to human needs would be limited. An important representation form based on *Natural Deduction* is the *Assertion Level* (Huang 1994), which constitutes a representation on the level of domain axioms. It is an abstraction based on *Natural Deduction* – one *Assertion Level* step comprises several *Natural Deduction* steps. There exist a variety of procedures that build *Natural Deduction* representations on the basis of resolution proofs. Moreover, *Assertion Level* representations can be built out of *Natural Deduction* proofs and even directly out of resolution proofs (Meier 2000). Hence, accomplishing this transformation step in the overall process can be considered as solved.

For the Steamroller problem, the proof graph obtained by applying the theorem prover OTTER (McCune 1994) and transforming the result to the *Assertion Level* consists of 51 nodes. The proof presentation system PROVERB (Huang and Fiedler 1997) produces a full page of text from this proof graph, which we believe is pretty much comparable in length to what other systems would do. From the perspective of human needs, the presented content is quite disappointing. On the one hand, it is too verbose, because it contains many trivially inferable reasoning steps, mostly categorical generalizations and instantiations. On the other hand, it is too compact in some essential parts, since the *Assertion Level* representations do not expose cognitively difficult inference steps in detail. In the following two subsections, we illustrate how these problems are handled by our methodology.

Condensation or Expansion of Inference Steps

This is the first transformation step that takes into account human needs for diversification of presentation contents. Depending on whether the input representation at this stage is fine-grained (*Natural Deduction*) or coarse-grained (*Assertion Level*), applying condensation or expansion operations is advisable. By and large, *Natural Deduction* is considered too detailed in general, while the *Assertion Level* is a good approximation of the representations underlying hu-

man argumentation. However, the *Assertion Level* falls short for cases of cognitively difficult inference patterns, since presentations at this level of granularity do not adequately support understanding of the underlying reasoning. For example, applying transitivity in reverse direction involves a modus tollens embedded in a single *Assertion Level* step, so that it is hard to understand the inference presented in this compact form. To handle such problematic situations, we have proposed the *Partial Assertion Level* (Horacek 1999). It is mostly an *Assertion Level* representation, but inference steps containing cognitively difficult *Natural Deduction* steps are expanded to a composition of these more fine-grained inference steps. Since building representations at this level is a variant of building representations at the *Assertion Level*, this transformation step in the overall process can also be considered as solved.

In the Steamroller problem, performing the transformation from the *Assertion Level* to the *Partial Assertion Level* is quite essential. It applies to three involved *Assertion Level* steps, which constitute the major subproofs of the whole problem: 1) birds eat grain, 2) foxes do not eat grain, and 3) foxes eat birds, on the basis of 1) and 2). In Figure 3, we show the expansion of subproof 2): The involved *Assertion Level* step is decomposed into two disjunction eliminations (DE), surrounding a modus tollens (MT). The variables f , g , and w stand for fox, grain, and wolf, respectively.

In natural language terms, using references to meat eater and plant eater (see the overnext subsection), the two representation variants can be paraphrased by:

1. From applying axiom 3 to the wolf and the fox, it follows that foxes are not plant eaters, since wolves are not plant eaters, are bigger than foxes, and do not eat foxes.
2. From applying axiom 3 to the wolf and the fox, it follows that wolves are meat eaters, since they are not plant eaters. Moreover, foxes are not plant eaters or not smaller than wolves, since wolves do not eat foxes. Since foxes are not smaller than wolves, they cannot be plant eaters.

Whereas the underlying reasoning in the more compact variant requires considerable afterthought to be understood properly, if possible at all, the second variant exposes the underlying reasoning steps explicitly in detail, so that the whole subproof can be followed rather easily.

Figure 4: A rule that prunes portions of the proof tree

$$\begin{array}{c}
 \text{Aware} - \text{of}(User, Axiom(T)) \\
 \wedge \\
 \text{Able} - \text{Infer}(User, Premises(T), Conclusion(T)) \\
 \rightarrow \\
 \text{Omit} < \text{References to Axiom } T \text{ in the inference step} >
 \end{array}$$

Omission of Inferable Inference Components

In this transition phase, adopting selectivity on the fully explicit proof representation is addressed. This measure aims at building “truncated” arguments, thereby going beyond Sadock’s (1977) “modus brevis” forms. To present an argument, which is mostly the application of a domain axiom in modus ponens, not only the instantiated form of the axiom is omitted, which is a simple measure used in many methods. In addition to that, appropriate applications of three categories of omissions contribute to the incremental compactification of an originally large proof derivation tree:

1. *Omitting a premise*
Premises considered to be trivial should be omitted.
2. *Omitting the reference to an axiom*
The reference to the axiom applied may be omitted, in case the premises and the conclusion are such that it is apparent to the addressee which axiom was used. This is frequently the case in mathematical reasoning and also in real world situations (see Thüring and Wender (1985)).
3. *Omitting an inference step altogether*
Even a whole inference step may be omitted, if its contribution to understand the course of reasoning is negligible. This may be the case for inferences considered trivial, or for such inference steps which conform to contextual expectations. For example, the application of commutativity, or the access to components of a conceptual definition (ρ is an equivalence relation [Therefore, it is reflexive, symmetric, and transitive. ... Therefore, it is symmetric.] Hence, ...) are considered to be trivial by humans and appear therefore redundant in an argumentation.

In (Horacek 1999), we have proposed a technique for manipulating an inference graph in such a way that components of the inference structure are omitted if they appear redundant to the audience on the basis of assumptions about its cognitive capabilities; Horacek (1998) describes the same method, but manipulating a rhetorical structure instead of an inference graph. In both versions, a tree structure is shrunk to its “essential arguments”, by omitting leaf nodes, and by replacing intermediate nodes by its successors, when the nodes omitted or replaced are associated with arguments assumed to be easily reconstructible in the given context.

Taking the inferential capabilities of the addressee into account to omit portions of content specifications assumed to be inferable is quite rare in proof explanations and in the field of natural language generation in general. For instance, even the elaborate natural language explanation techniques by Moore and Swartout (1989) do not address this

Figure 5: Successively pruning portions of a proof tree

$$\begin{array}{c}
 \frac{0 < 1 \quad 1 < a \quad \text{Transitivity}(<)}{0 < a} \quad \frac{\quad}{\text{Trichotomy}(<, \neq)} \\
 \frac{\quad}{0 \neq a} \\
 \text{Omitting} \quad \downarrow \quad \text{a trivial premise} \\
 \frac{1 < a \quad \text{Transitivity}(<)}{0 < a} \quad \frac{\quad}{\text{Trichotomy}(<, \neq)} \\
 \frac{\quad}{0 \neq a} \\
 \text{Omitting} \quad \downarrow \quad \text{axiom references} \\
 \frac{1 < a}{0 < a} \\
 \frac{0 < a}{0 \neq a} \\
 \text{Omitting} \quad \downarrow \quad \text{an intermediate step} \\
 \frac{1 < a}{0 \neq a}
 \end{array}$$

issue. Among the few noteworthy exceptions are a model for building indirect answers by Green and Carberry (1999) and an approach that specifically addresses conciseness in discourse by Zukerman and McConachy (1997).

In Figure 4, we show a rule (category 2 from the above list). It applies to the attainment of a conclusion from an instantiated rule (T) and some of its premises. If the user is aware of that rule ($Axiom(T)$) and can infer the instantiated form of this rule from the premises considered not inferable and from the conclusion, then he can be assumed to understand the inference without reference to the axiom.

We illustrate the result of several applications of the rules responsible for obtaining omissions in proof trees, when applied to the derivation tree in Figure 5 (the Steamroller problem is inappropriate for showing the effects of all rules handling omissions). In order to derive $a \neq 0$ from the premise $1 < a$, a theorem prover needs a (trivial) assertion ($0 < 1$) plus the application of transitivity and trichotomy axioms. In contrast to that, only stating $1 < a$ as a justification is a perfect explanation, even for mathematically untrained persons, First, $0 < 1$ is omitted because it is considered to be a trivial fact (*Omitting a premise*). Assuming the user is aware of transitivity and trichotomy, and further assuming that he is able to understand each of these inference step on the basis of the premises and the conclusion, references to these axioms are omitted in the derivation tree (*Omitting the reference to an axiom*). Finally, since the two inference steps are considered to be “coherent”, the intermediate conclusion is omitted as well (*Omitting an inference step altogether*). At the end, only $1 < a \rightarrow 0 \neq a$ remains, which are excellent content specifications for human-oriented presentations.

For the Steamroller problem, applying these rules leads to a considerable reduction of the whole proof tree. Thereby, all instantiations, such as “ w is a wolf”, and the associated categorical generalizations, such as “ w is a wolf, hence w is an animal”, are omitted (all of these inferences are *Assertion*

Figure 6: Examples of conceptual equivalence definitions

-
1. $Eating - habits(x) ::=$
 $\forall y(Plant(y) \rightarrow Eats(x, y)) \vee$
 $\forall y \exists z((Plant(z) \wedge Animal(y) \wedge$
 $Eats(y, z) \wedge (y < x)) \rightarrow Eats(x, y))$
 2. $Plant - eater(x) ::=$
 $\forall y(Plant(y) \rightarrow Eats(x, y))$
 3. $Meat - eater(x) ::=$
 $\forall y \exists z((Plant(z) \wedge Animal(y) \wedge$
 $Eats(y, z) \wedge (y < x)) \rightarrow Eats(x, y))$
-

Level steps). Inferences in this category make up more than half of the text produced from the original proof graph.

Altogether, the transformation subprocess dealt with in this subsection is much more delicate than the two previous ones, since it is not a mapping based on purely logical grounds, but it rather depends on a variety of assumptions about the cognitive capabilities of the audience. So far, our elaborations on which inferences can be assumed to be understood by some audience are rather limited, especially concerning contextual influences by embedding sequences of inferences. Being “aware” of an axiom is simply modeled by assuming that the user is aware of all axioms relevant for the subtheory to which the problem at hand belongs. Moreover, the user is assumed to be “able to infer” the use of such an axiom if the substitutions used to obtain the instantiated form are elementary – expressions with at most one operator. Finally, “coherence” between adjacent inference steps is judged by structural similarity of the expressions that constitute subsequent conclusions. While these measures seem to work quite nicely for a small set of examples of a complexity comparable to that of the Steamroller problem, the associated achievements are admittedly on an anecdotal level still. We believe that progress can be expected by studying presentations in mathematical textbooks, thereby reconstructing underlying assumptions about the audience, so that levels of explicitness observed can be mimicked.

Reorganization of the Inference Structure

This final stage of transformation addresses aspects of diversification, which may be populated by a variety of recasting operations and uses of references to inference substructures. Unless the ultimate presentation is a form of graphical display, explanations as communication in natural language in general, follows a linear interaction channel. In order to better meet the associated presentation needs, some portions of the inference structure built so far may be recast in a logically equivalent form which is in some places more compact and pursues a more linear line of argumentation with fewer embedded substructures. The aggregation of similar components, which also plays a significant role in (Di Eugenio *et al.* 2005) is a prominent feature of this process.

In our environment, we encountered a similar potential for reorganization in connection with categorizing algebraic

Figure 7: A text paraphrasing the final content specifications

-
1. We assume that birds are meat eaters. Since snails are plant eaters and smaller than birds, it would follow that birds eat snails, which contradicts the axiom that birds do not eat snails. Since the above assumption fails, birds are not meat eaters. Hence, they must be plant eaters.
 2. Since wolves are not plant eaters, they must be meat eaters. Since they do not eat foxes, foxes must either be no plant eaters, or not smaller than wolves. Since they are in fact smaller than wolves, they cannot be plant eaters.
 3. Therefore, foxes are meat eaters. Since birds are plant eaters and smaller than foxes, foxes eat birds.
-

structures built on the basis of residue classes. Examinations of properties required within this task are undertaken for each residue class separately, but the results obtained exhibited a number of commonalities that suggest the incorporation of aggregation methods. In some cases, the results of an operation applied to all residue classes turned out to be identical but for the specific residue class examined, so that the case distinction can be omitted altogether. Moreover, the frequent use of case distinctions in this investigation, specifically nested embeddings, inspired us to elaborate a method for recasting argumentation structures which contain multiple case distinctions (Horacek 2006). The operations applied include a lifting of an embedded case distinction into the embedding one, and the aggregation of cases identical to each other except to the reference to the case value. In a number of instances, the skilful combination of these operations led to a compactification of the reasoning structure, also stressing linearity in the presentation.

When addressing references to subproofs as arguments, we encountered a couple of cases where simple naming of axioms and using these names in references falls short. One relevant situation relates to semantically complex axioms, when components of such an axiom serve as arguments, but there are only names associated with the axiom as a whole. This problem can be addressed by increasing the repertoire of naming associations, giving also names to axiom components that have some chance to serve as arguments and which are conceptually meaningful. For example, referential access to the top level components of the involved axiom in the Steamroller problem (3.) would enable a system to explain the solution to this problem in a much more compact and elegant fashion. Thereby, considerably complex descriptions of the axiom’s components would be replaced by “meat eater” and “plant eater” (see Horacek (2001) for details). Figure 6 shows definitions that express the equivalence between components of the axiom at hand and terms that can be associated with names. Applying reification definitions involves non-trivial matching, to identify instantiations of these conceptual definitions as justifications of proof steps. Some limited forms of logical equivalence can be recognized by the matching procedure described in (Horacek

Figure 8: Content specifications for the proof to the Steamroller problem obtained after all transformations

$$\begin{array}{c}
 \frac{\frac{[Meat - eater(b)]}{Meat - eater(b)} AI \quad Plant - eater(s) \quad s < b}{Eats(b, s)} \quad \frac{}{CND} \quad \frac{}{\neg Eats(b, s)} TND \\
 \hline
 1. \text{ Deriving what birds eat} \quad \frac{Meat - eater(b) \rightarrow FALSE}{\neg Meat - eater(b)} MT \\
 \hline
 \frac{}{Plant - eater(b)} DE \\
 \hline
 2. \text{ Deriving what foxes do not eat} \quad \frac{\frac{\neg Plant - eater(w)}{Meat - eater(w)} DE \quad \frac{}{\neg Eats(w, f)} MT}{\neg Plant - eater(f) \vee \neg(f < w)} \\
 \hline
 \frac{}{f < w} DE \\
 \hline
 \frac{}{\neg Plant - eater(f)} DE \\
 \hline
 3. \text{ Deriving that foxes eat birds} \quad \frac{\frac{\neg Plant - eater(f)}{Meat - eater(f)} DE \quad Plant - eater(b) \quad b < f}{Eats(f, b)} \quad \frac{}{CND}
 \end{array}$$

2001), which takes into account some simple axioms, such as commutativity and De Morgan rules. An identification of all potential opportunities for using an equivalence definition is, of course, impossible, since this would amount to solve the problem of logical form equivalence.

There are crucial differences between recasting operations and the use of conceptual equivalence definitions. Whereas recasting operations always maintain logical equivalence between original and modified representations, applying a conceptual definition does not only make use of an extra piece of inferential knowledge, this application may also introduce some sort of abstraction, such as replacing a two-place predicate by a one-place predicate that abstracts from one of the two variables. In technical terms, the recasting operations achieve similar results as refactoring in software engineering (Fowler 1999), aiming at some sort of simplification. For purposes of proof explanation, however, logical simplification of some expression is only one presentation goal. Another motivation is the transformation into rhetorically more adequate structures, which are not necessarily simpler as well. These transformations include building flatter structures out of embedded ones, and the reorganizations of case distinctions in such a way that simple cases are handled first. Such structures support the use of conditionals in natural language presentations (If $\langle Case1 \rangle$ then $\langle Conclusion1 \rangle$. Otherwise ...), so that presenting the case distinction explicitly can be avoided.

In the Steamroller problem, the content specifications obtained after all transformations, reductions, and substitutions have been applied are given in Figure 8 – the intermediate results (1.) and (2.) are referred to in the final subproof (3.). The labels for the inference rules AI, TND, and CND stand for Assumption Introduction, Tertium Non Datur, and Composed Natural Deduction, and the variables b , f , s , and w stand for bird, fox, snail, and wolf. We have chosen to maintain the form of a derivation tree to represent these specifi-

cations, although this is formally inaccurate due to the effects of the omission operations. Moreover, references to $Meat - eater$ and $Plant - eater$ result from the application of conceptual equivalence definitions. When deriving $Plant - eater$ from $\neg Meat - eater$, the reference to $Eating - habits$ is omitted, assuming the audience can infer that this is the underlying justification. This final representation can serve the purpose of communicating the content of the proof even without natural language verbalization – it exposes the essential reasoning steps in a compact form, with the use of intuitive predicates and simple compositions of terms. A text expressing these content specifications in a reasonably fluent but rather direct way is given in Figure 7.

So far, we have investigated only a few measures that address reorganization. In addition to the ones reported in this section, we have also developed a method for building and presenting chains of inequations (Fehrer and Horacek 1999). Nevertheless, a variety of discrepancies between structures underlying automated proofs and human-adequate presentations are still waiting to be bridged, for instance, reference to components of structured mathematical objects. Again, studying mathematical textbooks and analyzing structural differences between the line of argumentation observed, in comparison to those underlying machine-found proofs is required to enhance the limited repertoire of recasting operations we have defined so far.

The Overall Organization

The stratified organization of the transformation measures described above is in some sense inspired by the “consensus” architecture for natural language generation systems (Reiter 1994), which is adopted in virtually all applications, despite a number of theoretical drawbacks – dependencies between subtasks across phases in the stratified model cannot always be dealt with in a satisfactory manner. In our model, which covers only parts of what could arguably be

subsumed under natural language generation subtasks, thus also reducing the number of dependencies, this drawback is much less severe. The second and third phase can be perceived as content selection, the task of applying omission operations being depending on choices made regarding levels of granularity, but not vice-versa. The last phase, which can be perceived as part of content organization, contains internal dependencies, for instance, among case reorganization operations. Consequently, adequate handling depends on the repertoire of operations incorporated in this phase. If internal dependencies exist, a simple control structure, which we propose in view of experience with approaches to natural language generation, is easily manageable and it is likely to lead to good though possibly not optimal results.

Requirements on Representations

The architecture proposed for building the content of explanations indicates the need for a substantial amount of extra information, that is, information not needed for the proper problem-solving process (as also argued by Kittredge *et al.* (1991) for natural language presentations in general). In addition to that, *motivations* underlying problem-solving steps, which are not available in ATP representations, would be quite helpful in supporting human understanding. Motivations are also missing in presentation of other kinds of machine-oriented procedures, such as plans, which becomes apparent in their presentations (Mellish and Evans 1989).

Unfortunately, representing such motivations and making them accessible for presentation purposes is extremely hard for ATPs, due to their specific reasoning methods. Unlike for knowledge-based systems, where inferencing is closer to human reasoning, it is not sufficient to attach explanation-relevant information on the level of the machine calculus, on the lines of the approach to *Explainable Expert Systems* (Swartout *et al.* 1991). Such a measure might be useful for higher-level reasoning specifications, such as the “methods” by Melis and Siekmann (1999). However, in order to be useful as explanation material, the underlying knowledge in a method must be separated from the domain-specific interpretation, which is the form of knowledge needed for finding solutions. For example, the method “complex-estimate” used in the domain of limit theorems would need to be re-represented as a more general method “partitioning into simpler subproblems”, with a specific interpretation suitable for polynomial expressions. Apparently, building such a representation would require more effort than to represent only what is needed for pure problem-solving purposes. Requirements on representations fall into three categories:

- *Technical requirements:*

This category relates to the provision of representations at different levels of granularity. This is an issue that is met on the basis of logics only. Apart from the quasi-standard *Natural Deduction* level, more abstract levels are desirable, such as the *Assertion Level*. Even more abstract levels, referring to “methods” (for instance, (Melis and Siekmann 1999)), would be desirable, but using representations on this level requires more communication knowledge, since these representation are hand-tailored.

- *Cognitive requirements:*

This category relates to the cognitive role that pieces of domain knowledge play in human reasoning. It imposes a kind of classification on domain rules, that is, the axioms, on a scale from easy to difficult, with regard to monitoring their application. For instance, commutativity is considered very easy to handle, while complex axioms, such as the main axiom in the Steamroller problem, are considered increasingly difficult. These classifications can be expressed as annotations attributing the likely cognitive load to each piece of domain knowledge. It would also be advisable to provide for a repertoire of such annotation sets, depending on user categories or individual users, when the incorporation of a user model is an option.

- *Linguistic requirements:*

This category relates to the expressibility of ingredients of an explanation, that is, references to pieces of domain knowledge. This issue comprises an obligatory and an optional part. The obligatory part lies in associating names and descriptions with pieces of domain knowledge that are applied in the course of a proof. This enables one to use compact references in places where unnecessarily complex formulas would appear otherwise. The optional part lies in associating such references also to pieces of domain knowledge that are not directly used in a proof, but are in some sense compositions of elements in the proof. Prior to installing linguistic associations for them, these pieces of knowledge need to be defined on the basis of more elementary ones, which amounts to formulating reifications (such as identifying specific portions of the major axiom in the Steamroller problem, which are then associated with descriptions *Meat – eater* resp. *Plant – eater*, whose precise meanings within the axiom needs to be explained once, so that they can be used whenever such a portion of the axiom is referred to). Moreover, expressing generalization in the knowledge base has the potential of increasing the repertoire of available referring expressions. When investigating the use of referring expressions in mathematical textbooks, Horacek *et al.* (2004), found a number of cases which require some extra ontological definitions in order to be accessible to a system. Examples include aggregates (“group properties”), and superordinates (“algebraic structures”).

Apart from explaining a full proof in the detail required to understand the underlying course of reasoning, there are a number of related communicative purposes that have an impact on the content selection for explanations. Such variations in explaining a proof include summaries (Horacek 2000), and interactive adaptations (Fiedler 2001), where a user can choose among varying degrees of detail, and ask for expositions of different parts of a proof. Most recently, we have looked into explanations meeting tutorial purposes (Horacek and Wolska 2007). As far as requirements on representation are concerned, only the tutorial setting poses new demands. Specifically, incomplete and vague references to domain concepts, which are still understandable to humans, that is, tutors, must be handled in a reasonable manner. For this purpose, we have introduced an extra layer of knowl-

edge which represents an intuitive, informal view on the domain concepts (Horacek and Wolska 2006).

Conclusion

In this paper, we have presented a methodology for building content specifications of explanations, on the basis of representations of solutions produced by an ATP. This methodology is perceived as a stratified model with dedicated transformation processes bridging between adjacent strata. Our investigations have shown considerable success, but also limitations of existing explanation capabilities, which motivates the representation of extra communication-relevant knowledge not required for pure problem-solving purposes.

References

- Di Eugenio, B.; Fossati, D.; Yu, D.; Haller, S.; and Glass, M. 2005. Aggregation Improves Learning: Experiments in Natural Language Generation for Intelligent Tutoring Systems. In *Proceedings of the 43rd Meeting of the Association for Computational Linguistics (ACL-05)*, 50-57.
- Fiedler, A. 2001. Dialog-Driven Adaptation of Explanations of Proofs. In *Proceedings of 17th International Joint Conference on Artificial Intelligence (IJCAI-2001)*, 1295-1300, Morgan Kaufman.
- Fehrer, D.; and Horacek, H. 1999. Presenting Inequations in Mathematical Proofs. *Information Sciences* 116:3-23.
- Fowler, M. 1999. Refactoring, Improving the Design of Existing Code. Addison-Wesley.
- Green, N.; and Carberry, S. 1999. Interpreting and Generating Indirect Answers. *Computational Linguistics* 25(3):389-435.
- Horacek, H. 1998. Generating Inference-Rich Discourse Through Revisions of RST Trees. In *Proceedings of 18th National Conference on Artificial Intelligence (AAAI-98)*, 814-820. Menlo Park, Calif.: AAAI Press.
- Horacek, H. 1999. Presenting Proofs in a Human-Oriented Way. In *Proceedings of CADE-16 (Conference on Automated Deduction)*, 142-156. Springer LNAI 1632.
- Horacek, H. 2000. Tailoring Inference-rich Descriptions through Making Compromises between Conflicting Cooperation Principles. *Int. J. Human-Computer Studies* 53:1117-1146.
- Horacek, H. 2001. Expressing References to Rules in Proof Presentations. In *Proceedings of the 1st International Joint Conference on Automated Reasoning (IJCAR-2001 (short paper))*, 76-85.
- Horacek, H.; Fiedler, A.; Franke, A.; Moschner, M.; Pollet, M.; and Sorge, V. 2004. Representing of Mathematical Concepts for Inferencing and for Presentation Purposes. *Cybernetics and Systems'2004*, 683-688.
- Horacek, H. 2006. Handling Dependencies in Reorganizing Content Specifications. *Research on Language and Computation* 4:111-139.
- Horacek, H.; and Wolska, M. 2006. Interpreting Semi-Formal Utterances in Dialogs about Mathematical Proofs. *Data & Knowledge Engineering* 58(1):90-106.
- Horacek, H.; and Wolska, M. 2007. Generating Responses to Formally Flawed Problem-Solving Statements. In *Proceedings of the 13th International Conference on Artificial Intelligence in Education (AIED-07)*. Forthcoming.
- Huang, X. 1994. Reconstructing Proofs at the Assertional Level. In *Proceedings of CADE-12 (Conference on Automated Deduction)*, 738-752. Springer LNAI 814.
- Huang, X.; and Fiedler, A. 1997. Proof Verbalization as an Application of NLG. In *Proceedings of 15th International Joint Conferences on Artificial Intelligence (IJCAI-97)*. 965-970, Morgan Kaufman.
- Johnson-Laird, P.; and Byrne, R. 1990. *Deduction*. Ablex Publishing.
- Kittredge R.; Korelsky, T.; and Rambow, O. 1991. On the Need for Domain Communication Knowledge. *Computational Intelligence* 7(4):305-314.
- McCune, W. 1994. Otter 3.0. Reference Manual and Guide. Technical Report ANL-94/6, Argonne National Laboratory.
- Meier, A. 2000. TRAMP: Transformation of Machine-Found Proofs into Natural Deduction Proofs at the Assertion Level. In *Proceedings of CADE-17 (Conference on Automated Deduction)*, 460-464. Springer LNAI 1831.
- Melis, E.; and Siekmann, J. 1999. Knowledge-Based Proof Planning. *Artificial Intelligence* 115(1):65-105.
- Mellish, C.; and Evans, R. 1989. Natural Language Generation from Plans. *Computational Linguistics* 15(4):233-249.
- Moore, J.; and Swartout, B. 1989. A Reactive Approach to Explanations. In *Proceedings of 11th International Joint Conference on Artificial Intelligence (IJCAI-89)*, 1504-1510. Morgan Kaufman.
- Reiter, E. 1994. Has a Consensus NL Architecture Appeared, and is it Psycholinguistically Plausible? In *Proceedings of the Seventh International Workshop on Natural Language Generation*, 163-170.
- Sadock, J. 1977. Modus Brevis: The Truncated Argument. In *Papers from the 13th Regional Meeting*, Chicago Linguistic Society, 545-554.
- Stickel, M. 1986. Schubert's Steamroller Problem: Formulations and Solutions. *Journal of Automated Reasoning* 2(1):89-101.
- Swartout, W.; Paris, C.; and Moore, J. 1991. Design for Explainable Expert Systems. *IEEE Expert* 6(3):59-64.
- 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 13th International Joint Conference on Artificial Intelligence (IJCAI-93)*, 1202-1207. Morgan Kaufman.