Argumentation in Explanations to Logical Problems

Armin Fiedler and Helmut Horacek

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

Abstract. Explaining solutions to logical problems is one of the areas where argumentation in natural language plays a prominent role. One crucial reason for the difficulty in pursuing this issue in a systematic manner when relying on formal inference systems lies in the discrepancy between machine-oriented reasoning and human-adequate argumentation. Aiming at bridging these two divergent views, we present a model for producing human-adequate argumentation from machine-oriented inference structures. Ingredients of our method are techniques to build representations to suitable degrees of abstraction and explicitness, and a module for their interactive and adaptive exploration. The presented techniques are not only relevant for the interactive use of theorem provers, but they also have the potential to support the functionality of dialog-oriented tutorial systems.

1 Introduction

Explaining solutions to logical problems is one of the areas where argumentation in natural language plays a prominent role. One crucial reason for the difficulty in pursuing this issue in a systematic manner when relying on formal inference systems lies in the discrepancy between machine-oriented reasoning and human adequate argumentation. The associated differences, however, do not manifest themselves so much in variations in format and syntax, but more substantially in the way how the underlying information is organized.

Aiming at bridging these two divergent views, we present a model for producing human-adequate argumentation from machine-oriented inference structures. Ingredients of our method are techniques to build representations to suitable degrees of abstraction and explicitness, and a module for their interactive and adaptive exploration. The presented techniques are not only relevant for the interactive use of theorem provers, but they also have the potential to support the functionality of dialog-oriented tutorial systems.

This paper is organized as follows. We first provide some background information about presentation of machine-found proofs in natural language. Then we introduce empirical motivations that substantiate divergent demands for human adequate presentations. We describe techniques for building representations meeting these psychological requirements, followed by selection options aiming at summarizations. Then we describe a module for interactive and adaptive exploration. Finally, we illustrate our approach by a moderately complex example.

2 Background

The problem of obtaining a natural language proof from a machine-found proof can be divided into two subproblems: First, the proof is transformed from its original machine-oriented formalism into a human-oriented calculus, which is much better suited for presentation. Second, the transformed proof is verbalized in natural language.

Since the lines of reasoning in machine-oriented calculi are often unnatural and obscure, algorithms (see, e.g., [1, 14]) have been developed to transform machine-found proofs into more natural formalisms, such as the *natural deduction (ND) calculus* [7]. ND inference steps consist of a small set of simple reasoning patterns, such as forall-elimination ($\forall x P(x) \Rightarrow P(a)$) and implication elimination, that is, modus ponens. However, the obtained ND proofs often are very large and too involved in comparison to the original proof. Moreover, an inference step merely consists of the syntactic manipulation of a quantifier or a connective. [11] gives an algorithm to abstract an ND proof to an *assertion level* proof, where a proof step may be justified either by an ND inference rule or by the application of an assertion (i.e., a definition, axiom, lemma or theorem).

One of the earliest proof presentation systems was [2]. Several theorem provers have presentations components that output proofs in pseudo-natural language using canned text (e.g., [3, 4]). Employing several isolated strategies, [5] was the first system to acknowledge the need for higher levels of abstraction when explaining proofs. *PROVERB* [12] expresses machine-found proofs abstracted to the assertion level and applies linguistically motivated techniques for text planning, generating referring expressions, and aggregation of propositions with common elements. Drawing on *PROVERB*, we are currently developing the interactive proof explanation system *P.rex* [6], which additionally features user adaptivity and dialog facilities. [9] is another recently developed NLG system that is used as a back end for a theorem prover.

In order to produce reasonable proof presentations, many systems describe some complex inference steps very densely, and they leave certain classes of proof steps implicit in their output, for example, by abstracting from intermediate inference steps that are recoverable from inductive definitions, or by omitting instantiations of axioms. However, leaving out information on the basis of purely *syntactic* criteria, as this has been done so far, easily leads to incoherent and hardly understandable text portions. In order to get control over the inferability and comprehensibility in presenting inference steps, an explicit model is required which incorporates semantic and pragmatic aspects of communication, which is what we try to achieve by our approach.

3 Empirical Motivation

Issues in presenting deductive proofs, as a special case of presenting argumentative discourse, have attracted a lot of attention in the fields of psychology, linguistics, and computer science. Central insights relevant to deductive argumentation are:

(1) "Let ϱ be an equivalence relation. Therefore we have ϱ is reflexive, we have ϱ is symmetric, and we have ϱ is transitive. Then we have ϱ is symmetric and we have ϱ is reflexive. Then $\forall x : x \varrho x$. Thus we have $h_0 y_0 \varrho h_0 y_0 \ldots$ "

(1') "Let ϱ be an equivalence relation. Thus we have $h_0 y_0 \varrho h_0 y_0 \ldots$"

(2) "Let ϱ be a transitive relation and let $\neg(a \varrho b)$. Let us assume that $c \varrho b$. Hence we have $\neg(a \varrho c)$."

(2') "Let ϱ be a transitive relation and let $\neg(a \varrho b)$. Let us assume that $c \varrho b$. Since ϱ is transitive, $\neg(a \varrho b)$ implies that $\neg(a \varrho c)$ or $\neg(c \varrho b)$ holds. Since we have $\neg(a \varrho b)$ and $c \varrho b$, $\neg(a \varrho c)$ follows."

Fig. 1. Straightforwardly presented proof portions and suitable improvements.

- Logical consequences of certain kinds of information are preferably conveyed implicitly through exploiting the discourse context and default expectations.
- Human performance in comprehending deductive syllogisms varies significantly from one syllogism to another.

The study in [17] demonstrates that humans easily uncover missing pieces of information left implicit in discourse, most notably in sequences of events, provided this information conforms to their expectations in the given context. Similarly to the expectations examined in that study, which occur frequently in everyday conversations, a number of elementary and very common inferences are typically left implicit in mathematical texts, too, including straightforward instantiations, generalizations, and associations justified by domain knowledge.

Another presentation aspect is addressed by studies on human comprehension of deductive syllogisms (see the summary in [13]). These studies have unveiled considerable performance differences among individual syllogisms (in one experiment, subjects made 91% correct conclusions for modus ponens, 64% for modus tollens, 48% for affirmative disjunction, and 30% for negative disjunction). The consequences of this result are demonstrated by the elaborate essay in [18], which presents a number of hypotheses about the impacts that human resource limits in attentional capacity and in inferential capacity have on dialog strategies. These hypotheses are acquired from extensive empirical analysis of naturally occurring dialogs and, to a certain extent, statistically confirmed. One that is of central importance for our investigations says that an increasing number of logically redundant assertions to make an inference explicit are made, in dependency of how hard and important an inference is (modus tollens being an example for a hard inference which requires a more detailed illustration).

In the following, we demonstrate that these crucial issues in presenting deductive reasoning are insufficiently captured by current techniques. Consider the portions of straightforwardly presented proofs produced by an earlier version of *PROVERB*, (texts (1) and (2) in Fig. 1), each of which can be improved significantly, as demonstrated by texts (1') and (2'), correspondingly. Text (1) should be presented more concisely, while parts of text (2) require more explanation. In (1'), the addressees' knowledge about definitions (here, concerning equivalence relations), and their capabilities to mentally perform some sort of simple inference steps such as conjunction eliminations and elementary substitutions are exploited. In (2'), the involved application of the transitivity axiom is exposed more explicitly through separating the descriptions of the instantiation of the theorem (in reversed direction as a modus tollens) from the disjunction elimination inference, thereby reintroducing the facts not mentioned in immediately preceding utterance parts. Altogether, these examples show some crucial deficits in current proof presentation techniques:

- A large number of easily inferable inference steps is expressed explicitly.
- Involved inferences, though hard to understand, are presented in single shots.

The first deficit suggests the omission of contextually inferable elements in the proof graph, and the second demands the expansion of compound inference steps into simpler parts.

4 Content Determination

In order to obtain presentations similar to (1'), and (2') in Fig. 1, we propose the application of an optimization process that enhances an automatically generated proof at the assertion level. Through this process, pragmatically motivated expansions, omissions, and short-cuts are introduced, and the audience is assumed to be able to mentally reconstruct the details omitted with reasonable effort. In a nutshell, the modified proof graph is built through two subprocesses:

- Building expansions
  Compound assertion level steps are expanded into elementary applications of deductive syllogisms, while marking the original larger steps as summaries.
- Introducing omissions and short-cuts
  Shorter lines of reasoning are introduced by skipping individual reasoning steps, through omitting justifications (marked as inferable) and intermediate reasoning steps (marking the 'indirect' justifications as short-cuts).

In the following, we explain these subprocesses in more detail.

4.1 Level of Abstraction

The purpose underlying the expansion of assertion level steps is to decompose presentations of complex theorem applications or involved applications of standard theorems into easier comprehensible pieces. This operation is motivated by performance difficulties humans typically have in comparable discourse situations. At first, assertion level steps are completely expanded to the natural deduction (ND) level according to the method described in [11]. Thereafter, a partial recomposition of ND steps into inference steps encapsulating the harder comprehensible deductive syllogisms, modus tollens and disjunction elimination steps, is performed, in case the sequence of ND rules in the entire assertion level step contains more than one of these. To do this, the sequence of ND rules is

$$(1) \quad \frac{\forall x, y, z : ((x \varrho y \wedge y \varrho z) \Rightarrow x \varrho z) \quad \neg(a \varrho c) \quad b \varrho c}{\neg(a \varrho b)}\text{Assertion}$$

$$(2) \quad \cfrac{\cfrac{\cfrac{\cfrac{\forall x, y, z : ((x \varrho y \wedge y \varrho z) \Rightarrow x \varrho z)}{(a \varrho b \wedge b \varrho c) \Rightarrow a \varrho c}\forall E \qquad \neg(a \varrho c)}{\neg(a \varrho b \wedge b \varrho c)} \Rightarrow E}{\neg(a \varrho b) \vee \neg(b \varrho c)}\text{NR} \qquad b \varrho c}{\neg(a \varrho b)} \vee E$$

$$(3) \quad \cfrac{\cfrac{\forall x, y, z : ((x \varrho y \wedge y \varrho z) \Rightarrow x \varrho z) \quad \neg(a \varrho c)}{\neg(a \varrho b) \vee \neg(b \varrho c)}\text{Modus Tollens} \qquad b \varrho c}{\neg(a \varrho b)} \vee E$$

Fig. 2. An involved assertion level inference at several degrees of abstraction.

broken after each but the last occurrence of a modus tollens or disjunction elimination, and the resulting subsequences of ND steps are composed into a sequence of reasoning steps at some sort of *partial assertion* level. This sequence is then inserted in the proof graph as a potential substitute for the original assertion level step, which is marked as a *summary*. An example for such an expansion and partial recomposition is shown in Fig. 2 ($\forall E$, $\Rightarrow E$, $\vee E$, and NR stand for the ND rules, forall-elimination, implication elimination, disjunction elimination, and natural rewrite, respectively). If $b \varrho c$ and $\neg(a \varrho c)$ hold for a transitive relation ϱ, $\neg(a \varrho b)$ is derivable by a single assertion level step ((1) in Fig. 2). Through expansion to the ND level ((2) in Fig. 2) and recomposition encompassing deductive syllogisms ((3) in Fig. 2), the modus tollens inference step "$\neg(a \varrho c)$ implies $\neg(a \varrho b)$ or $\neg(b \varrho c)$" is separated from the disjunction elimination "Thus, $b \varrho c$ yields $\neg(a \varrho b)$". Note that, in contrast to modus tollens, modus ponens would be composed with disjunction elimination into a single step.

4.2 Degrees of Explicitness

Unlike expanding summaries, creating omissions and short-cuts is driven by communicatively motivated *presentation rules*. They express aspects of human reasoning capabilities with regard to contextually motivated inferability of pieces of information on the basis of explicitly mentioned facts and relevant background knowledge [8]. These rules provide an interface to stored assumptions about the intended audience. They describe the following sorts of situations:

Cut-prop: omission of a *proposition* (premise) appearing as a reason
Cut-rule: omission of a *rule* (axiom instance) appearing as a method
Compactification: short-cut by omitting an *intermediate* inference step

These *reduction* rules aim at omitting parts of a justification that the audience is considered to be able to infer from the remaining justification components of the same line of the proof, or even at omitting an entire assertion level step that is considered inferable from the adjacent inference steps. In order for these rules to apply successfully, presentation preferences and conditions about the addressees' knowledge and inferential capabilities are checked.

The functionality of the reduction rules can be explained by a simple example. If trivial facts, such as $0 < 1$, or axioms assumed to be known to the audience, such as *transitivity*, appear in the set of justifications of some inference step, they are marked as *inferable* ($0 < 1$ through *Cut-prop*, and *transitivity* through *Cut-rule*). Consequently, the derivation of $0 < a$ can simply be explained by $1 < a$ to an informed audience. Moreover, single facts appearing as the only non-inferable reason are candidates for being omitted through applying *Compactification*. If, for instance, $0 < a$ is the only non-inferable reason of $0 \neq a$, and $0 < a$, in turn, has only one non-inferable reason, $1 < a$, the coherence maintaining similarity between $0 < a$ and $1 < a$ permits omitting $0 < a$ in the argumentative chain. Altogether, $0 \neq a$ can be explained concisely by $1 < a$ to an informed audience.

The presentation rules are matched against proof lines by traversing the proof graph from its leaf nodes and successively continuing to the root node, without back-tracking. In doing this, *Cut-prop* and *Cut-rule* mark locally inferable justification components, and *Compactification* adds alternative justifications through short-cuts.

5 Generating Condensations

In order to convey the information specified completely in view of the assumptions made about the audience summaries are avoided and inferables omitted. Depending on the target item, giving such an explanation in all details required for full understanding may result in a long text. Therefore, it is better to present a reduced first-shot contribution, which can be further investigated interactively, according to user reactions. All possible reductions amount to relaxing the degree of completeness in which the information is presented. In accordance with the aspects of variations focused on, there are two kinds of condensations for obtaining higher degrees of abstraction:

- A sequence of inferences is abstracted into a set of propositions consisting of its conclusion and its premises, while the method how the conclusion is obtained, that is, the underlying sequence of inferences, is omitted. If there is evidence that some of the premises are more important or of more interest to the audience than the remaining ones, larger sets of premises can be reduced to subsets of these. In particular, this measure comprises preferring summaries over detailed exposition of involved inference steps.
- Moreover, in case these inferences constitute the expansion of a pre-designed proof method [15], which underlies the construction of a partial proof, the functionality of that method can be expressed by a descriptive phrase.

Four alternatives are examined, in ascending order of information reduction:

1. Omitting the way how a piece of knowledge (a domain regularity) is applied.
2. Omitting that piece of knowledge.
3. Omitting premises of the inference (eventually, only some of them).

4. Omitting intermediate inference steps.

The choice among these options is based on assumptions about the audience and on the resulting balance of textual descriptions. In [10] we have defined and motivated some strategies for that.

6 Interactive Exploration

Automatically found proofs can be explored interactively with the system *P. rex*. It allows for three types of user interaction: A *command* tells the system to fulfill a certain task, such as explaining a proof. An *interruption* interrupts the system to inform it that an explanation is not satisfactory or that the user wants to insert a different task. In clarification dialogs, finally, the user is prompted to give *answers* to questions that *P. rex* asks when it cannot identify a unique task to fulfill. In this paper, we concentrate on interruptions.

The user can interrupt *P. rex* anytime to enter a new command or to complain about the current explanation. The following speech acts are examples for messages that can be used to interrupt the system:

`(too-detailed :Conclusion` C`)`
The explanation of the step leading to C is too detailed, that is, the step should be explained at a more abstract level.

`(too-abstract :Conclusion` C`)`
The explanation of the step leading to C is too abstract, that is, the step should be explained at in more detail.

`(too-implicit :Conclusion` C`)`
The explanation of the step leading to C is too implicit, that is, the step should be explained more explicitly.

`(too-difficult :Conclusion` C`)`
The explanation of the step leading to C is too difficult.

In *P. rex*, `too-difficult` is considered as an underspecified interruption. When the user complains that the derivation of a conclusion C was too difficult, the dialog planner enters a clarification dialog to find out which part of the explanation failed to remedy this failure. During the clarification dialog, the system tries to distill whether the user failed to follow some implicit references or whether the explanation was to abstract. The control of the behavior of the dialog planner is displayed in Fig. 3.

7 An Example

We demonstrate the functionality of our model by the presentation of a well-known proof, Schubert's Steamroller [16]:

Axioms:

(1) Wolves, foxes, birds, caterpillars, and snails are animals, and there are some of each of them. Also there are some grains, and grains are plants.

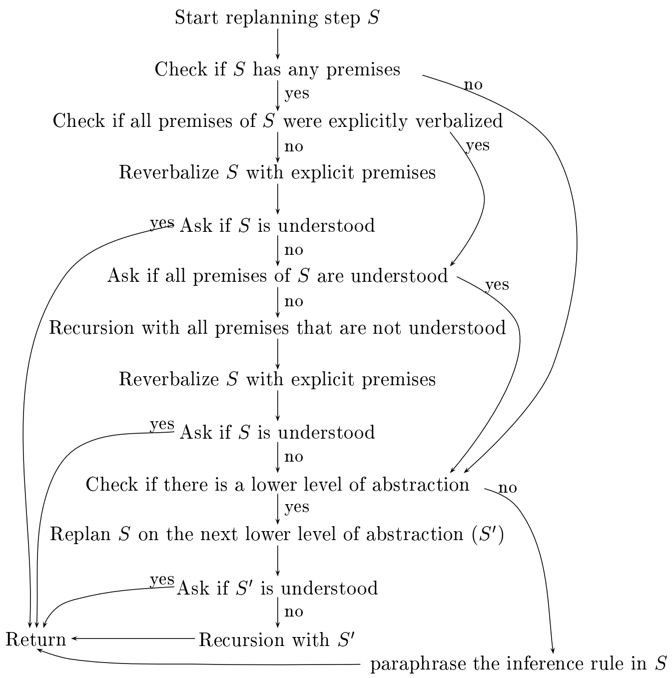

Fig. 3. The reaction of the dialog planner if a step S was too difficult.

(2) Every animal either likes to eat all plants or all animals much smaller than itself that like to eat some plants.
(3) Caterpillars and snails are much smaller than birds, which are much smaller than foxes, which in turn are much smaller than wolves. Wolves do not like to eat foxes or grains, while birds like to eat caterpillars, but not snails. Caterpillars and snails like to eat some plants.

Theorem:
(4) Therefore there is an animal that likes to eat a grain-eating animal.

Proving that theorem (4) is based on applying given pieces of simplified real world knowledge (1) to (3).

In a nutshell, the proof runs along the following lines: Through applying axiom (2) three times, it is first derived that birds eat plants, then that foxes do not eat grains and, finally, that foxes eat the smaller grain-eating birds, the last being the witness needed to prove theorem (4).

Within the theorem proving community, the steamroller problem is famous, because solving it requires several variables to be instantiated purposefully without having a guidance how to do this through the formulation of the theorem to be proved — it has only existentially quantified variables in it, but no constants. Until some years ago, automated theorem provers were unable to apply this technique with sufficient degrees of efficiency, so that they were originally unable to solve this problem. For our purposes, this problem is attractive for completely different reasons: its definition is easily comprehensible without mathematical knowledge, and a full-detailed solution path is sufficiently complex so that exploring it interactively seems to be well motivated.

Apparently, the proof sketch given above is far from being a complete and fully comprehensible explanation of the proof, since many details that are necessary to understand how the central axiom is applied in each case are not mentioned. The task of the presentation module is then to suitably mediate between such a concise proof sketch and a fully expanded proof description:

System: The proof runs through applying three times the rule that animals either eat plants or all plant eaters smaller than themselves. It is first derived that foxes do not eat grain. Similarly, it is derived that birds eat grain, and finally, that foxes eat birds. Since foxes eat birds, an animal is known that eats a grain-eating animal, q.e.d.

User: More detail, please.

System: The fact that foxes do not eat grain follows from the assumptions that wolves do not eat grain and foxes are smaller than wolves, because animals who do not eat plants eat plant eaters smaller than themselves. Thus, either foxes do not eat grain or they are not smaller than wolves must hold. Hence, only the first alternative is valid. ...

User: Less detail, please.

System: Birds do not eat plant eaters because it is assumed that they do not eat snails, but snails are smaller than birds and they eat plants. And foxes eat birds, because they are plant eaters, and smaller than foxes.

User: Why do birds not eat plant eaters?

System: For animals, it holds that (1) they eat all plants or (2) they eat all plant eaters smaller than themselves. (2) cannot hold for birds, because snails eat plants and they are smaller than birds, but it is known that birds do not eat snails. Hence (1) holds for birds.

8 Conclusion

The analysis of human proof explanations shows that certain logical inferences are only conveyed implicitly drawing on the discourse context and default expectations. Moreover, different syllogisms call for different presentation strategies to account for human performance. In this paper, we proposed the partial assertion level as an appropriate representation of proofs to plan the content of an explanation and different degrees of explicitness and condensation. Then, driven by the unfolding dialog, a reactive planner allows for an interactive, user-adaptive navigation through the proofs.

So far we have implemented *P.rex* and some tools for mediating between levels of abstraction. We are currently investigating manipulations of the proof structure to realize different degrees of explicitness. We will soon incorporate this work into a newly starting project on dialog-oriented tutoring systems. Moreover, we believe that our approach also proves useful for argumentative dialog systems in general.

References

[1] Peter B. Andrews. Transforming matings into natural deduction proofs. In *Proceedings of the 5th International Conference on Automated Deduction*, pages 281–292. Springer Verlag, 1980.

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

[3] Yann Coscoy, Gilles Kahn, and Laurent Théry. Extracting text from proofs. In Mariangiola Dezani-Ciancaglini and Gordon Plotkin, editors, *Typed Lambda Calculi and Applications*, number 902 in Lecture Notes in Computer Science, pages 109–123. Springer Verlag, 1995.

[4] Bernd Ingo Dahn, J. Gehne, Thomas Honigmann, and A. Wolf. Integration of automated and interactive theorem proving in ILF. In William McCune, editor, *Proceedings of the 14th Conference on Automated Deduction*, number 1249 in LNAI, pages 57–60, Townsville, Australia, 1997. Springer Verlag.

[5] Andrew Edgar and Francis Jeffry Pelletier. Natural language explanation of natural deduction proofs. In *Proceedings of the 1st Conference of the Pacific Association for Computational Linguistics*, Vancouver, Canada, 1993. Centre for Systems Science, Simon Fraser University.

[6] Armin Fiedler. Using a cognitive architecture to plan dialogs for the adaptive explanation of proofs. In Thomas Dean, editor, *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 358–363, Stockholm, Sweden, 1999. Morgan Kaufmann.

[7] Gerhard Gentzen. Untersuchungen über das logische Schließen I & II. *Mathematische Zeitschrift*, 39:176–210, 572–595, 1935.

[8] H. Grice. Logic and conversation. *Syntax and Semantics*, 3:43–58, 1975.

[9] Amanda M. Holland-Minkley, Regina Barzilay, and Robert L. Constable. Verbalization of high-level formal proofs. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI-99) and Eleventh Innovative Application of Artificial Intelligence Conference (IAAI-99)*, pages 277–284. AAAI Press, 1999.

[10] Hemlut Horacek. Tailoring inference-rich descriptions through making compromises between conflicting cooperation principles. *Int. J. Human-Computer Studies*, 53:1117–1146, 2000.

[11] Xiaorong Huang. Reconstructing proofs at the assertion level. In Alan Bundy, editor, *Proceedings of the 12th Conference on Automated Deduction*, number 814 in LNAI, pages 738–752, Nancy, France, 1994. Springer Verlag.

[12] Xiaorong Huang and Armin Fiedler. Proof verbalization as an application of NLG. In Martha E. Pollack, editor, *Proceedings of the 15th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 965–970, Nagoya, Japan, 1997. Morgan Kaufmann.

[13] Philip Johnson-Laird and Ruth Byrne. *Deduction*. Ablex Publishing, 1990.

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

[15] Erica Melis. AI-techniques in proof planning. In Henri Prade, editor, *Proceedings of of the 13th European Conference on Artifical Intelligence*, pages 494–498, Brighton, UK, 1998. John Wiley & Sons, Chichester, UK.

[16] Mark E. Stickel. Schubert's steamroller problem: Formulations and solutions. *Journal of Automated Reasoning*, 2:89–101, 1986.

[17] Manfred Thüring and Kurt Wender. Über kausale Inferenzen beim Lesen. *Sprache und Kognition*, 2:76–85, 1985.

[18] Marylin Walker. The effect of resource limitations and task complexity on collaborative planning in dialogue. *Artificial Intelligence*, 85:181–243, 1996.