March 1988

Explaining Modal Logic Proofs

Amy Felty
University of Pennsylvania

Greg Hager
University of Pennsylvania

Recommended Citation

Amy Felty and Greg Hager, "Explaining Modal Logic Proofs", . March 1988.

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-88-18.

Explaining Modal Logic Proofs

Abstract

There has recently been considerable progress in the area of using computers as a tool for theorem proving. In this paper we focus on one facet of human-computer interaction in such systems: generating natural language explanations from proofs. We first discuss the X proof system - a tactic style theorem proving system for first-order logic with a collection of inference rules corresponding to human-oriented proof techniques. In X, proofs are stored as they are discovered using a structured term representation. We describe a method for producing natural language explanations of proofs via a simple mapping algorithm from proof structures to text. Nonclassical or specialized logics are often used in specialized applications. For example, modal logics are often used to reason about time and knowledge, and inheritance theories are often developed for classification systems. The form of, and explanations for, proofs in these systems should be tailored to reflect their special features. In this paper, we focus on the extension of X to incorporate proofs in modal logic, and on the different kinds of explanations of modal proofs that can be produced to meet the needs of different users.

Comments

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-88-18.

EXPLAINING MODAL LOGIC PROOFS

Amy Felty
Greg Hager

MS-CIS-88-18
LINC LAB 105

Department of Computer and Information Science
School of Engineering and Applied Science
University of Pennsylvania
Philadelphia, PA 19104

March 1988

Appearing in the proceedings of the IEEE 1988 International Conference on Systems, Man and Cybernetics, Beijing and Shenyang China, August 1988.

Acknowledgements: This research was supported in part by DARPA/ONR grants NOO14-85-K-0807, NOOO14-85-K-0018, NSF grants MCS-83-05221, MCS-8219196-CER, IRI84-10413-AO2 and U.S. Army grants DAA29-84-K-0061, DAA29-84-9-0027.

EXPLAINING MODAL LOGIC PROOFS

Amy Felty and Greg Hager
Department of Computer and Information Science
University of Pennsylvania
Philadelphia, PA 19104-6389

Abstract

There has recently been considerable progress in the area of using computers as a tool for theorem proving. In this paper we focus on one facet of human-computer interaction in such systems: generating natural language explanations from proofs. We first discuss the χ proof system—a tactic style theorem proving system for first-order logic with a collection of inference rules corresponding to human-oriented proof techniques. In χ, proofs are stored as they are discovered using a structured term representation. We describe a method for producing natural language explanations of proofs via a simple mapping algorithm from proof structures to text.

Nonclassical or specialized logics are often used in specialized applications. For example, modal logics are often used to reason about time and knowledge, and inheritance theories are often developed for classification systems. The form of, and explanations for, proofs in these systems should be tailored to reflect their special features. In this paper, we focus on the extension of χ to incorporate proofs in modal logic, and on the different kinds of explanations of modal proofs that can be produced to meet the needs of different users.

Appearing in the proceedings of the IEEE 1988 International Conference on Systems, Man, and Cybernetics, Beijing and Shenyang China, August 1988.

1 Introduction

One common route in the formal verification of correctness is to axiomatize the system under study, and then verify its properties using proof theory. The desire to provide computer-aided facilities for the construction of these proofs has led to the development of several systems for interactive and semi-automatic construction of proofs in various logics. Examples include LCF[6] and Nuprl [2]. These systems typically provide a collection of inference rules that correspond to human-oriented proof techniques such as indirect proof and case analysis. This allows construction of natural proofs, encourages user involvement in the search for proofs, and facilitates understanding of the resulting proofs. The χ proof system [3], built by the first author, is a theorem proving system built on these principles. χ includes several additional facilities, including a mechanism for producing explanations from proofs.

The explanation of a proof can take numerous equivalent forms depending on taste, background, and the type of information to be conveyed. A basic criterion for presenting a proof is the ability to disregard uninteresting detail and present only the most relevant facts. Furthermore, if a proof is specific to a given domain, the explanation should be presented in terms of concepts which have meaning within that domain. This is particularly true in theories where logical operations correspond to specialized inferences. For instance, in a first-order theory of inheritance, it makes sense to explain a proof in terms of inheritance rather than the basic logical operations. Along the same lines, explaining a proof of a statement in a nonclassical logic will require specialized treatment of any operators which are not found in first-order logic.

In this paper, we first present the generation of explanations of proofs in χ, and then show how χ can be extended to produce explanations for proofs in modal epistemic logic [1], a nonclassical logic originally developed by philosophers to describe certain language constructions involving knowledge and belief [10]. Variations on standard epistemic logics have been employed in AI [14] and in distributed systems [8] as well as other areas. We then demonstrate the explanation algorithm on a variant of the well-known wise men problem[15]. We present two different levels of explanation, and discuss other possible extensions and variations.

2 Constructing Proofs and Generating Explanations

In the χ proof system, proofs are stored using a structured term representation. Explanations are generated by mapping these proof terms to natural language text. This mapping algorithm is simple, yet flexible in that it is possible to generate different kinds of explanations to meet the needs of different users. The form and content of a given explanation depends upon the information extracted from its corresponding proof term, and the manner in which that information is mapped to strings of text.

In this section, we first discuss the proof system used in χ and present the term representation for proofs. We then briefly describe the proof construction component used to search for proofs. This component is quite important since it is at this stage that proof terms are built, and the information used later in generating explanations is recorded. By changing the proof at this level, we may change the explanation. Finally, we present the

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{R(a,a) \longrightarrow R(a,a) \qquad P(a) \longrightarrow P(a)}{R(a,a), R(a,a) \supset P(a) \longrightarrow P(a)}\ \text{forwardchain}}{R(a,a), \forall y(R(a,y) \supset P(y)) \longrightarrow P(a)}\ \text{all-L}}{\forall w\ R(w,w), \forall y(R(a,y) \supset P(y)) \longrightarrow P(a)}\ \text{all-L}}{\forall w\ R(w,w) \longrightarrow \forall y(R(a,y) \supset P(y)) \supset P(a)}\ \text{imp-R}}{\forall w\ R(w,w) \longrightarrow \forall x[\forall y(R(x,y) \supset P(y)) \supset P(x)]}\ \text{all-R}}{\longrightarrow \forall w\ R(w,w) \supset \forall x[\forall y(R(x,y) \supset P(y)) \supset P(x)]}\ \text{imp-R}$$

Figure 1: A sequential proof tree for $\forall w\ R(w,w) \supset \forall x[\forall y(R(x,y) \supset P(y)) \supset P(x)]$.

algorithm for mapping proof terms to natural language explanations.

Sequential Trees In χ, proofs in first-order logic are constructed using a Gentzen style sequent system similar to the LK system [5]. A sequent, consisting of a set, Σ, of hypotheses, and a set, Δ, of conclusions, is written $\Sigma \longrightarrow \Delta$. Such a sequent has the interpretation, "from the formulas in Σ, we can prove one of the formulas in Δ." In χ, Gentzen's LK inference rules are available as well as several additional derived rules, providing the user with more choices in constructing proofs. These rules are described in [3]. For example, the first inference rule below is the LK rule to introduce a disjunction on the left of the sequent arrow, corresponding to an argument by case analysis. The second rule is a derived rule for modus ponens.

$$\dfrac{A, \Sigma \longrightarrow \Delta \qquad B, \Sigma \longrightarrow \Delta}{A \vee B, \Sigma \longrightarrow \Delta}\ \text{or-L}$$

$$\dfrac{A \longrightarrow A \qquad B, \Sigma \longrightarrow \Delta}{A, A \supset B, \Sigma \longrightarrow \Delta}\ \text{forwardchain}$$

Trees of sequents are constructed by applying inference rules which join a conclusion (parent) sequent with its premises (children) sequents. A proof of a formula A is a finite tree with the sequent $\longrightarrow A$ at the root and axioms at all the leaves. Axioms are trivially true sequents of the form $A \longrightarrow A$. Figure 1 gives an example proof of a statement about a reflexive relation R and a predicate P.

A Structured Representation for Sequential Proofs We represent proof trees as recursive term structures. The single node proof tree $A \longrightarrow A$ is represented by the term `axiom(`A`)`. Each inference rule is encoded by a function symbol indicating the rule that was applied, with arguments for the proof terms of the premises of the rule, and possibly other arguments to encode the information necessary for an application of the rule. For readability, we leave out such auxiliary arguments in this paper. For example, an application of the forwardchain inference rule is represented by the term `forwardchain(`T_1`,`T_2`)` where T_1 and T_2 are the term representations for the proof trees for the premises, $A \longrightarrow A$ and $B, \Sigma \longrightarrow \Delta$, respectively. (In this case T_1 is `axiom(`A`).`) Each inference rule has a similar corresponding representation (see [12]). The proof term for the example above is:

```
imp_r(all_r(imp_r(all_l(all_l(forwardchain(axiom(R(a,a)),
                                           axiom(P(a))))))))
```

Proof Construction in χ The χ proof system employs tactics and tacticals as the mechanism for proof search and construction. In general, in tactic style theorem provers, primitive tactics implement inference rules, while tacticals provide a mechanism for building compound tactics by composing primitive tactics in various ways. Tactics and tacticals promote modular design in the construction of theorem provers, and provide flexibility in controlling both interactive and automatic aspects of the search for proofs (see [6,4]).

In χ the primitive tactics implement the inference rules of our modified LK proof system. (For more information on the use of tactics and tacticals in χ, see [12].) The ability to introduce new tactics into the theorem proving environment provides a mechanism for incrementally enhancing the interactive proof environment with new inference rules. For example, derived rules such as the forwardchain rule above enhance user interaction during proof search by providing human-oriented proof techniques. As we will see, this capability also enhances the explanation facility. New inference rules introduce new proof structures, which will be mapped to the appropriate natural language text.

The Explanation Algorithm The explanation algorithm is implemented by providing, for each inference rule, a corresponding function which takes the explanations of the proofs of the premises and puts them together, possibly adding more text, to construct the explanation for the proof of the conclusion. This algorithm will only "lexicalize" inference rules to produce a skeletal outline for proofs. We will not lexicalize formulas, though this could be done fairly easily and would often produce more readable text.

To illustrate how the mapping is achieved, consider the or-L inference rule above. A term of the form `or_l`(T_1, T_2) represents a proof using case analysis. If T_1 is mapped to $text_1$ which is the text for the proof of the left premise arguing that Δ follows from A and Σ, and T_2 is mapped to $text_2$ which argues that Δ follows from B and Σ, then the text below is one possible mapping of `or_l`(T_1, T_2) to the conclusion that Δ follows from $A \vee B$ and Σ.

> We have two cases. Case 1: Assume A. $text_1$ Case 2: Assume B. $text_2$ Thus, in either case, we have Δ.

As another example, if $text$ is the explanation for T_2 representing a proof of $B, \Sigma \longrightarrow \Delta$ in the proof term `forwardchain(`T_1`,`T_2`)`, the resulting explanation is simply:

> By modus ponens we have B. $text$

All inference rules can be given such interpretations. Using this mapping algorithm the following explanation is generated for the simple example above.

> Assume $\forall w\ R(w, w)$. Assume $\forall y(R(a, y) \supset P(y))$. Let $w = a$ in $\forall w\ R(w, w)$. Let $y = a$ in $\forall y(R(a, y) \supset P(y))$. By modus ponens we have $P(a)$. Since a was arbitrary we have $\forall x[\forall y(R(x, y) \supset P(y)) \supset P(x)]$.

Even here, we have made choices in the presentation. An experienced logician would probably suppress the statements about the instantiation of variables since they can be inferred from context. This can be achieved by modifying the mapping to text of the all-L rule.

3 Modal Logic Proof Systems

The modal language we will consider is a slightly modified form of that found in [15]. We consider some set of n agents and discuss what these agents "know" by introducing modal operators of the form K_i $(i = 1, \ldots, n)$. We form the language by considering all sentences constructed from some set of propositions, the usual logical connectives, and the (monadic) modal operators. Thus we have statements of the form K_iA and $K_i \neg K_j(A \vee B)$ with the informal interpretations "agent i knows that A" and "agent i knows that it is not the case that agent j knows either A or B," respectively. The semantic interpretation for K, developed initially in [10], is that to know something is to see that it is true in all possible configurations of the world imaginable given what is currently known. Thus, the modal operator is not interpreted truth-functionally—the truth of K_iA depends not just on the truth of A, but its truth in all states consistent with the current one. Different logics result by adopting different notions of a consistent state.

Modal logic can be translated to first-order logic [9], and so may be considered a specialized theory in first-order logic. The translation of a propositional logic of knowledge to first-order logic uses the following general rules: 1) consider the initial modal statement relative to some situation w_0, 2) translate a formula consisting of one or two operands joined by a standard logical connective as that connective applied to the translations of its operands relative to the situation of the original formula, 3) translate a primitive proposition A relative to situation w as $A(w)$, 4) translate K_iA relative to situation w as $\forall w'(w \, R_i \, w' \supset B)$ where B is the translation of A relative to w'. If B is the translation of A relative to w_0, and T is the conjunction of the axioms describing the possible-world topology, the final formula to be proven is $T \supset \forall w_0 B$. For example, the statement proved in Figure 1 is the translation of the modal statement $K_iP \supset P$ in a possible worlds topology which is reflexive.

A Modal Gentzen Style Proof System While translation is a viable approach to automating modal logic, the resulting first-order proofs contain details which are irrelevant to the thrust of the proof. For example, in the explanation of the proof in Figure 1, the reasoning about possible worlds shows up explicitly, and the modal operators are lost. We will develop a sequent system that presents a compressed version of the proof for the first-order translation of a modal formula. This allows us to present explanations couched at the level of the actual model theory (which the first-order statements represent), or in terms of the original modal statement.

In our modal Gentzen system, we will ornament formulas with a world term denoting the situation in which the formula is to be interpreted, e.g. $(A \vee B)_w$. The ornamentation on the formula, and a set of relational constraints indicating the possible-world structure will encode the current frame of the proof. Validity is ensured by ornamenting the initial formula with an arbitrary initial world term which appears nowhere else in the proof and appealing to universal generalization over the class of models generated from that term.

We will extend the definition of sequent to include a set, $\mathfrak{R}$, indicating the currently known facts about the possible-world relation. A sequent will now be written $\Re; \Sigma \longrightarrow \Delta$ and has the interpretation, "given the possible-world configuration $\Re$, the possible-world theory, and the formulas in Σ, we can prove one of the formulas in Δ." We map the propositional Gentzen rules for classical logic to rules of our system by maintaining the world denotation. In addition, we include introduction rules for the modal operators which take the following form [7]:

$$\frac{[\Re \Vdash^s w\, R_i\, x] \;\; \Re; A_x, \Sigma \longrightarrow \Delta}{\Re; (K_i A)_w, \Sigma \longrightarrow \Delta}\ \mathrm{K}_i\text{-L} \qquad \frac{\Re, w\, R_i\, x; \Sigma \longrightarrow \Delta, A_x}{\Re; \Sigma \longrightarrow \Delta, (K_i A)_w}\ \mathrm{K}_i\text{-R}$$

K$_i$-R requires the proviso that x does not appear as an ornamentation in the lower sequent. K$_i$-L requires the proviso that $w\, R_i\, x$ follows, in the theory of the system s, from the statements contained in $\Re$, i.e. $\Re \Vdash^s w\, R_i\, x$ as indicated on the inference figure. For example, if s is a system whose possible-world theory is transitive, the following inference is acceptable.

$$\frac{w\, R_i\, x, x\, R_i\, y; A_y, \Sigma \longrightarrow \Delta}{w\, R_i\, x, x\, R_i\, y; (K_i A)_w, \Sigma \longrightarrow \Delta}\ \mathrm{K}_i\text{-L}$$

We will also extend term structures to accommodate the additional inference rules. The single node modal proof tree $\Re; A_w \longrightarrow A_w$ is represented by the term `axiom(`A`,`w`,`$\Re$`)`. A proof tree whose last inference rule is K$_i$-R will be represented as `Ki_r(`T`,`w`)`. The argument T is the proof term encoding all the information for the proof of the premise and w is the ornamentation on the formula $K_i A$ in the conclusion. The proof term for K$_i$-L is `Ki_l(`T`,`w`,`$\mathcal{P}$`)`. The extra argument $\mathcal{P}$ encodes the proof of the proviso.

A Gentzen Proof for the Wise Men Problem To illustrate our modal system, we turn to the wise men problem as stated in [15]. In this puzzle, a king has three advisors and he wishes to determine who is the wisest. He devises a test in which he paints a white dot on each advisor's forehead, and then tells them that *at least one of them* has a white dot on his forehead. The solution involves the advisors reasoning about what the others know. The first advisor, upon seeing only white dots, is forced to admit he does not know whether he has a white dot. The second advisor, on hearing this but still seeing two white dots, is also forced to admit ignorance. The final advisor, based on the admissions of his colleagues, is now able to conclude he has a white dot.

For illustration purposes, we have shortened the puzzle to involve a queen with two advisors. The reasoning remains essentially the same. In the following, interpret p_i as the proposition, "advisor i has a white dot on her head." K_i is, of course, interpreted as, "advisor i knows that", and O is interpreted as, "it is common knowledge that." The proof system will include rules for the modal operators K_1, K_2 and O. These modalities all have a reflexive, transitive possible-world relation, and, in addition, the theory for O states that if, for any i, $w\, R_i\, w'$, then $w\, R_O\, w'$. That is, the possible-world relation for O is the superset of all individual possible-world relations. The axioms for the advisor puzzle (adapted from [15]) are given below.

1. It is common knowledge that someone has a white dot: $O(p_1 \vee p_2)$. However, we will use the logically equivalent form $O(\neg p_2 \supset p_1)$ since it results in a better explanation.

$$\begin{prooftree}
\AxiomC{$\Re_1; (p_2)_x \longrightarrow (p_2)_x$}
\RightLabel{K$_1$-L}
\UnaryInfC{$\Re_1; (K_1 p_2)_x \longrightarrow (p_2)_x$}
\AxiomC{$\Re_2; (\neg p_2)_y \longrightarrow (\neg p_2)_y$}
\AxiomC{$\Re_2; (p_1)_y \longrightarrow (p_1)_y$}
\RightLabel{forwardchain}
\BinaryInfC{$\Re_2; (\neg p_2)_y, (\neg p_2 \supset p_1)_y \longrightarrow (p_1)_y$}
\RightLabel{O-L}
\UnaryInfC{$\Re_2; (\neg p_2)_y, O(\neg p_2 \supset p_1)_w \longrightarrow (p_1)_y$}
\RightLabel{K$_1$-L}
\UnaryInfC{$\Re_2; (K_1 \neg p_2)_x, O(\neg p_2 \supset p_1)_w \longrightarrow (p_1)_y$}
\RightLabel{K$_1$-R}
\UnaryInfC{$\Re_1; (K_1 \neg p_2)_x, O(\neg p_2 \supset p_1)_w \longrightarrow (K_1 p_1)_x$}
\RightLabel{neg-L}
\UnaryInfC{$\Re_1; (K_1 \neg p_2)_x, O(\neg p_2 \supset p_1)_w, (\neg K_1 p_1)_x \longrightarrow$}
\RightLabel{thinning}
\UnaryInfC{$\Re_1; (K_1 \neg p_2)_x, O(\neg p_2 \supset p_1)_w, (\neg K_1 p_1)_x \longrightarrow (p_2)_x$}
\RightLabel{or-L}
\BinaryInfC{$\Re_1; (K_1 p_2 \vee K_1 \neg p_2)_x, O(\neg p_2 \supset p_1)_w, (\neg K_1 p_1)_x \longrightarrow (p_2)_x$}
\RightLabel{O-L}
\UnaryInfC{$\Re_1; O(K_1 p_2 \vee K_1 \neg p_2)_w, O(\neg p_2 \supset p_1)_w, (\neg K_1 p_1)_x \longrightarrow (p_2)_x$}
\RightLabel{K$_2$-L}
\UnaryInfC{$\Re_1; O(K_1 p_2 \vee K_1 \neg p_2)_w, O(\neg p_2 \supset p_1)_w, (K_2 \neg K_1 p_1)_w \longrightarrow (p_2)_x$}
\RightLabel{K$_2$-R}
\UnaryInfC{$\Re_0; O(K_1 p_2 \vee K_1 \neg p_2)_w, O(\neg p_2 \supset p_1)_w, (K_2 \neg K_1 p_1)_w \longrightarrow (K_2 p_2)_w$}
\RightLabel{imp-R,and-L}
\UnaryInfC{$\Re_0; \longrightarrow (O(K_1 p_2 \vee K_1 \neg p_2) \wedge O(\neg p_2 \supset p_1) \wedge K_2 \neg K_1 p_1 \supset K_2 p_2)_w$}
\end{prooftree}$$

Figure 2: A proof tree for the wise queen problem. Here $\Re_0 = \emptyset$, $\Re_1 = \{w\, R_2\, x\}$, and $\Re_2 = \{w\, R_2\, x, x\, R_1\, y\}$.

In χ, we allow the user to choose either form by including a tactic for an inference rule which treats implication as the equivalent disjunction.

2. It is common knowledge that the first advisor knows whether the second advisor has a white dot or not: $O(K_1 p_2 \vee K_1 \neg p_2)$.
3. Advisor 2 knows that her colleague has no information as to the color of her own dot: $K_2 \neg K_1 p_1$.

We have axiomatized the puzzle asymmetrically—in general, either advisor could determine the color of her spot if her colleague speaks first. However, since the dual axioms are never used, we have not included them here.

We wish to demonstrate that Advisor 2 can determine she has a white dot: $K_2 p_2$. The proof tree in Figure 2 establishes the necessary conclusion. The term representation for this proof is:

```
imp_r(and_l(and_l(K2_r(K2_l(O_l
   (or_l(K1_l(axiom(p₂,x,ℜ₁)),
         thin(neg_l(K1_r(K1_l(O_l
            (forwardchain(axiom(¬p₂,y,ℜ₂),
                          axiom(p₁,y,ℜ₂))))))))))))))
```

4 Explanations of Modal Proofs

To extend the explanation algorithm to modal logic, we must add text generation functions whose contribution to the explanation will depend on the meaning of the corresponding modal operator. This contribution can vary depending on context and the amount of detail desired in the explanation. For example, the rules for the knowledge operators embody the fact that if $K_i A$ is true (in world w), then A is also true (in world x), as long as x is

R_i-related to w. The simplest explanations, which we will demonstrate first, assume that the reader is familiar with these rules, so that if K_iA is true, it will not be necessary to explicitly state A before using it. Such a reader will not need to know the details of how the possible worlds are related. This kind of detail would add unnecessary clutter to the proof.

A Skeletal Explanation We explain an instance of the K$_i$-R rule by simply concluding K_iA after explaining the proof of A. This reflects the modal rule of necessitation. The function for K$_i$-R takes an input argument *text* which is the explanation for the proof for $\Re, w\, R_i\, x; \Sigma \longrightarrow \Delta, A_x$ (the premise of the rule) and produces the following text as the proof of the conclusion:

> *text*. Thus K_iA.

The K$_i$-L function is even simpler. In the conclusion of this rule, K_iA is an assumption which takes the form A in the premise. Since these two are equivalent to our reader, the explanation for the premise and conclusion will be the same, i.e. the function K$_i$-L takes the input text and returns it unchanged. The explanation functions for the common knowledge operator will be defined in the same way as those for K_i.

Returning to the wise queen example, the explanation generated using the above functions is:

> Assume:
>
> 1. $O(K_1p_2 \vee K_1\neg p_2)$
> 2. $O(\neg p_2 \supset p_1)$
> 3. $K_2\neg K_1p_1$
>
> We have two cases:
>
> Case 1: K_1p_2
>
> Case 2: $K_1\neg p_2$. By modus ponens, we have p_1. Thus K_1p_1. Hence, we have a contradiction.
>
> Thus, in either case, we have p_2. Hence, K_2p_2.

A More Detailed Explanation For a reader not so familiar with the axioms and inference rules of modal logic, this explanation is probably too skeletal. Since modal logic is not truth-functional in the classical sense, the surface structure of a proof does not directly mirror the underlying model theory, though one is often interested in the actual model-theoretic underpinnings of a modal statement[11]. As we have pointed out, a direct explanation of a first-order translation is too general, so we will develop an algorithm which gives a "deeper" explanation of the proof which is specialized to possible-world semantics. We will borrow terminology from Moore [14] and Hintikka [10], and refer to possible worlds as "situations" or "states of affairs." A state of affairs can be thought of as the set of propositions which are true in that situation. The possible-world relation will be interpreted as linking consistent states of affairs. The new function for K$_i$-R will return the text string:

> Let situation x be an arbitrary state of affairs consistent with situation w. *text*. Since situation x is consistent with situation w, we have K_iA in situation w.

Here, *text* is the explanation of the premise sequent, x is the possible-world variable ornamenting the formula A in the premise, and w is the world ornamenting K_iA in the conclusion. Since the proof terms fully represent sequential proof trees, all the necessary information (including these world variables) will be present and obtainable from the input arguments to the explanation function.

The new K$_i$-L function generates the following explanation:

> We have A in situation x. *text*.

Once again the explanation functions for the common knowledge modal operator O will be similar to those for K_i, except that the phrase "We have" will be replaced by "From common knowledge we have."

To obtain this kind of deeper explanation, in addition to these new functions for the knowledge operators, the explanation functions for the remainder of the inference rules must give the information about the situation. This is accomplished by adding the text "in situation w" after every formula that is inserted into the text, where w is the ornamentation on the formula. Using these explanation functions, the text for our example is:

> Assume:
>
> 1. $O(K_1p_2 \vee K_1\neg p_2)$ in an initial situation w.
> 2. $O(\neg p_2 \supset p_1)$ in an initial situation w.
> 3. $K_2\neg K_1p_1$ in an initial situation w.
>
> Let situation x be an arbitrary state of affairs consistent with situation w. We have $\neg K_1p_1$ in situation x. From common knowledge we have $K_1p_2 \vee K_1\neg p_2$ in situation x.
>
> We have two cases:
>
> Case 1: K_1p_2 in situation x. We have p_2 in situation x.
>
> Case 2: $K_1\neg p_2$ in situation x. Let situation y be an arbitrary state of affairs consistent with situation x. We have $\neg p_2$ in situation y. From common knowledge we have $\neg p_2 \supset p_1$ in situation y. By modus ponens we have p_1 in situation y. Since situation y is consistent with situation x, we have K_1p_1 in situation x. Hence we have a contradiction.
>
> Thus, in either case we have p_2 in situation x. Since situation x is consistent with situation w, we have K_2p_2 in situation w.

The second explanation provides two kinds of additional information. First, it provides more detail about the chain of inference. For example, in the first explanation K_1p_1 contradicts the fact that Advisor 2 knows that $\neg K_1p_1$ (given by the third assumption). In the second explanation, the additional inference to obtain $\neg K_1p_1$ is stated explicitly, and the contradiction follows from the resulting formula. Second, information from the possible-world ornamentations is used to give explicit reference as to how situations are related. For example, in Case 2, we conclude $\neg p_2 \supset p_1$ in situation y from $O(\neg p_2 \supset p_2)$ in situation w. This follows from earlier statements that situation w is consistent with situation x, and situation x is consistent with situation y. While this is more detailed than the first explanation, it assumes some familiarity on the part of the reader about how worlds are related. For example, the reader must understand that the possible-world relation is transitive. A slightly more detailed explanation would result from explicitly stating the consistency conditions that prevailed in order to apply K_i-L. We can generate such an explanation by using the information in $\mathcal{P}$ (the proof of the proviso) when explaining instances of the K_i-L rule. In this kind of explanation, the inferences used to determine that the situation of the premise is consistent with the situation of the conclusion would be explicitly stated.

5 Discussion

We have presented a technique for the generation of text explanations for proofs in modal logic. These explanations were generated from structures corresponding to Gentzen style proofs in a modified sequent system. Proofs in this system were represented via a recursive term structure, and explanations generated by a simple mapping from these term structures to text strings.

We have selected a particular style of Gentzen proofs for a single modal logic. Other proof systems and logics may lead to different explanations. Also, we presented only two possible explanations. We can certainly obtain others by examining different kinds of mappings from proof terms to text. Yet another way to generate new explanations is to go back to the proof construction component and construct different proofs (thus obtaining different proof terms) for the same formula. Such new proofs may use alternative inference rules which will have their own encoding as proof terms, and thus their own mapping function to natural language text. The design of the proof construction component and the kinds of inference rules available play an important role in the generation of explanations.

Finally, we note that we have only presented a subset of the facilities available in χ for manipulating proofs of modal statements. χ also has facilities for integrating proofs automatically generated via traditional methods such as resolution for first-order logic. These proofs can be transformed into the sequential proof system used in the interactive environment. In order to do this, a technical device called expansion proofs [13] is employed. Expansion proofs can be extended to modal logic [7] thus allowing the automatic generation of modal proofs. By transforming these proofs to sequential proofs, we can produce explanations for modal statements in a completely automated fashion.

Acknowledgements The authors would like to thank Dale Miller for his help and guidance in doing this research, and for useful comments on a draft of this paper. This work has been supported by NSF AI Center grants NSF-MCS-83-05221, US Army Research office grant ARO-DAA29-84-9-0027, DARPA N000-14-85-K-0018, and DARPA/ONR N0014-85-K-0807.

References

[1] B. F. Chellas. *Modal Logic, an Introduction.* Cambridge University Press, Cambridge, 1980.

[2] R. L. Constable et al. *Implementing Mathematics with the Nuprl Proof Development System.* Prentice-Hall, 1986.

[3] A. Felty. *Using Extended Tactics to do Proof Transformations.* Master's thesis, University of Pennsylvania, December 1986. Technical Report MS-CIS-86-89.

[4] A. Felty and D. Miller. Specifying theorem provers in a higher-order logic programming language. In *Ninth International Conference on Automated Deduction*, Argonne Ill., May 1988.

[5] G. Gentzen. Investigations into logical deductions, 1935. In M. E. Szabo, editor, *The Collected Papers of Gerhard Gentzen*, pages 68–131, North-Holland Publishing Co., Amsterdam, 1969.

[6] M. J. Gordon, A. J. Milner, and C. P. Wadsworth. *Edinburgh LCF: A Mechanised Logic of Computation.* Volume 78 of *Lecture Notes in Computer Science*, Springer-Verlag, 1979.

[7] G. Hager. *Computational Aspects of Proofs in Modal Logic.* Master's thesis, University of Pennsylvania, December 1985. Technical Report MS-CIS-85-55.

[8] J. Halpern and Y. Moses. A guide to the modal logics of knowledge and belief: preliminary draft. In *Proceedings of the International Joint Conference on Artificial Intelligence 1985*, pages 480–490, Los Angeles, August 1985.

[9] C. Haspel. *A Study of Some Interpretations of Modal and Intuitionistic Logics in the First Order Predicate Calculus.* PhD thesis, Syracuse University, Syracuse, NY, 1972.

[10] J. Hintikka. *Knowledge and Belief.* Cornell University Press, 1962.

[11] J. Hintikka. *Models for Modalities.* D. Reidel Publishing Company, Boston, 1969.

[12] D. Miller and A. Felty. An integration of resolution and natural deduction theorem proving. In *Proceedings of the Fifth National Conference on Artificial Intelligence*, pages 198–202, AAAI, Morgan Kaufmann, Philadelphia, PA, August 1986.

[13] D. A. Miller. Expansion tree proofs and their conversion to natural deduction proofs. In R. E. Shostak, editor, *Seventh Conference on Automated Deduction*, pages 375–393, Springer-Verlag, Napa CA, May 1984.

[14] R. C. Moore. *Knowledge and Action.* Technical Report 191, SRI International, Menlo Park, October 1980.

[15] M. Sato. A study of Kripke-type models for some modal logics by Gentzen's sequential method. *Publications of the Research Institute for Mathematical Science*, 13:381–468, 1977.