ICOT Technical Memorandum: TM-0576

TM-0576

General-Purpose Reasoning Assistant System EUODHILOS and Its Applications

by
H. Sawamura and T. Minami

July, 1988

ICOT
Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan
(03) 456-3191 – 5
Telex ICOT J32964

Institute for New Generation Computer Technology

General-Purpose Reasoning Assistant System EUODHILOS and its Applications

Every universe of discourse has its logical structure
S. K. Langer (1925)

Hajime Sawamura and Toshiro Minami
International Institute for Advanced Study of Social Information Science (IIAS-SIS),
FUJITSU LIMITED, 140 Miyamoto, Numazu, Shizuoka 410-03, JAPAN
hajime@iias.fujitsu.junet tos@iias.fujitsu.junet

Summary

A new dimension of computer-assisted reasoning research is being explored. It aims at a general-purpose reasoning assistant system that allows a user to interactively define the syntax and inference rules of a formal system and to construct proofs in the defined system. We have named such a system EUODHILOS, an acronym reflecting our philosophy or observation that *every universe of discourse has its logical structure*, which turns out to spell and sound like a Greek philosopher's name.

In these days, various logics play important and even essential roles in computer science and artificial intelligence. In fact, they have made use of a variety of logics, including first-order, higher-order, equational, temporal, modal, intuitionistic, type theoretic logics and so on. However, implementing an interactive system for developing proofs is a daunting and laborious task for any style of presentation of these logics. For example, one must implement a parser, term and formula manipulation operations (such as substitution, replacement, juxtaposition, etc.), definitions, inference rules, rewriting rules, proofs, proof strategies and so on. Thus, it is desirable to find a general theory of logics and a general-purpose reasoning assistant system that captures the uniformities of a large class of logics so that much of this effort can be expended once and for all. We aim at building an easy to use and general reasoning system which handles as many of these logics as possible. There are two major subjects to be pursued for such an interactive system. One is reasoning-oriented human-computer interface that can be established as an aspect of reasoning supporting facilities. An easy to use system with good interface would be helpful for one to conceive ideas in reasoning. The other subject is the kind of reasoning styles suitable for human reasoners which should be taken into account. More generally, reasoning (proving) methodology, which reminds us of programming methodology, needs to be investigated.

We believe that a general-purpose reasoning assistant system incorporating these points should cater to the mathematician or programmer who wants to do proofs, and also to the logician or computer theorist who wants to experiment with different logical systems.

A brief system summary of EUODHILOS

In what follows, a brief summary of a current system is presented.

Purpose

Research and development of an easy to use and general-purpose reasoning assistant system which supports definition of formal systems and construction of proofs in the systems so defined.

Fundamental design considerations

- Realization of a general reasoning sytem, based on the philosophy that every universe of discourse has its logical structure.
- Support of logical thought, symbolic or logical manipulations done by human reasoners
- Provision of an easy to use environment for supporting proof constructions
- Environment for experimenting logical model construction and methodology of science

Functional features

We list the main features of EUODHILOS and explain them briefly.

(1) Formal system description language

A formal system (logical system), in general, consists of a language system and a derivation system. In EUODHILOS, a language system to be used is designed and defined by a user, using so called definite clause grammar (DCG), and then the bottom-up parser and unparser for the defined language are automatically generated, which are to be internally used in all the phases of symbol manipulations. At the same time the internal structures of the expressions of the language are constructed as well. These functions greatly lighten a user's burden in setting up his own language. A derivation system consists of an inference system and a rewriting system. They are given in a natural deduction style presentation by a user. Especially, an inference rule is stated as a triple consisting of three elements, where the first is the derivations of the premises of a rule, the second the conclusion of a rule, and finally the third the restrictions that are imposed on the derivations of the premises, such as variable occurrence condition (eigenvariable). Well-known typical styles of logic presentations such as Hilbert's style, Gentzen's style, Equational style could be treated within this framework.

(2) Proof constructing facilities

Proving activity is often said to be quite similar to programming activity, or rather, the former is sometimes identified with the latter. EUODHILOS has the following unique facilities which support constructions of proofs in the defined formal system.

- Sheets of thought

This originated from a metaphor of work or calculation sheet and is apparently analogous to the concept of sheet of assertion which is due to C. S. Peirce. It allows one to draft a proof, to compose proof fragments or detach a proof, or to reason using lemmas, etc.

Technically, a sheet of thought is a window with multi-functions for reasoning in the multi-window environment of a Personal Sequential Inferential machine (PSI).

- Tree-form proof

As mentioned above, inference and rewriting rules are presented in a natural deduction style. This naturally induces a construction of a proof into a tree-form proof. Consequentlly it leads to representing a proof structure explicitly, that is, proof visualization (cf. program visualization).

- Proving methodology

It is desirable that reasoning or proof construction can be done along the natural way of thinking of human reasoners. Therefore EUODHILOS supports the typical method for reasoning, that is, top-down reasoning, bottom-up reasoning and reasoning in a mixture of them. They are accomplished interactively on several sheets of thought. It is planed to incorporate not only such a proving methodology but also methodology of science (e.g., Lakatos' mathematical philosophy of science, Kitagawa's relativistic logic of mutual specifications, etc.).

(3) Human-computer interface for reasoning

In EUODHILOS the following facilities are available as human-computer interface for ease in communicating and reasoning with a computer, in particular facilities for inputing formulas and formula visualization.

- Formula editor

This is a structure editor for logical formulas and makes it easy to input, modify and display complicated formulas. In addition to ordinary editing functions, it provides some proper functions for formulas.

- Software keyboard and Font editor

These are used to make and input special symbols often appearing in various formal systems. It is a mater of course that provision of special symbol which reasoners are accustomed to use makes it possible to reason as usual on a computer.

- Goods for reasoning

Independently of a logic under consideration, various reasoning tools such as decision procedures become helpful and useful in reasoning processes. In a sense it may also play a role

of a model which makes up for a semantical aspect of reasoning. Currently, a logic calculator for Boolean logic is realized as a desk accessary.

In addition to the above features, multi-window environment, mouse, icon, pop-up menu, etc., are exploited in the implementation of EUODHILOS.

Implementation

EUODHILOS is implemented in ESP language on PSI-II/SIMPOS, and its size is about 4MB. The system configuration of EUODHILOS is illustrated in Fig. 1.

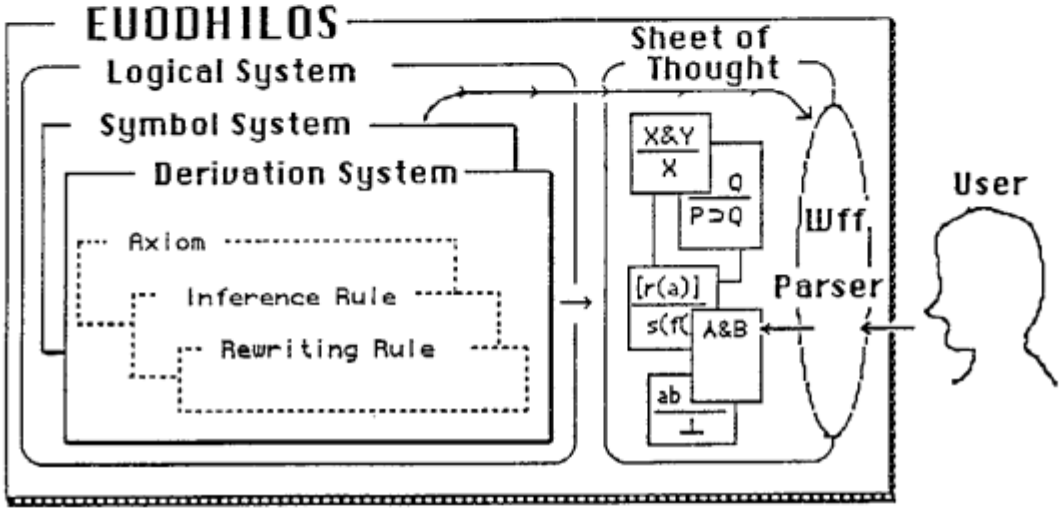


Fig. 1 An illustration of EUODHILOS

Experiences and experiments with EUODHILOS

Logics and proof examples that we have dealt with so far on EUODHILOS are :

(1) First Order Logic (NK)
- Various pure logical formulas
- Unsolvability of the halting problem, etc.

(2) Propositional Modal Logic (T)

(3) Intensional Logic (IL) (with reflective proof)

(4) Combinatory Logic

(5) Martin-Löf's intuitionistic Type Theory

These proof experiments with different logical systems have shown the potential and usefulness of EUODHILOS in the realm of logics often appearing in computer science. In the future we plan to attack:

- Logic of knowledge
- Various logics of programs (including Hoare's logic, Dynamic logic, etc.)
- Non-monotonic logic
- Relevant logic

:

Future research directions

- Extension of EUODHILOS

  e.g., making the logic description language more expressive
- Mathematical investigation of logic description language
- Investigation of higher-level supporting functions for reasoning

  e.g., developing a language for proof strategies, incorporating metatheory, etc.
- Maintaining a relational dependency among various theories
- Opening up a new application field of reasoning by EUODHILOS
- Improvement and refinement of human-computer interface for the reasoning system

etc.

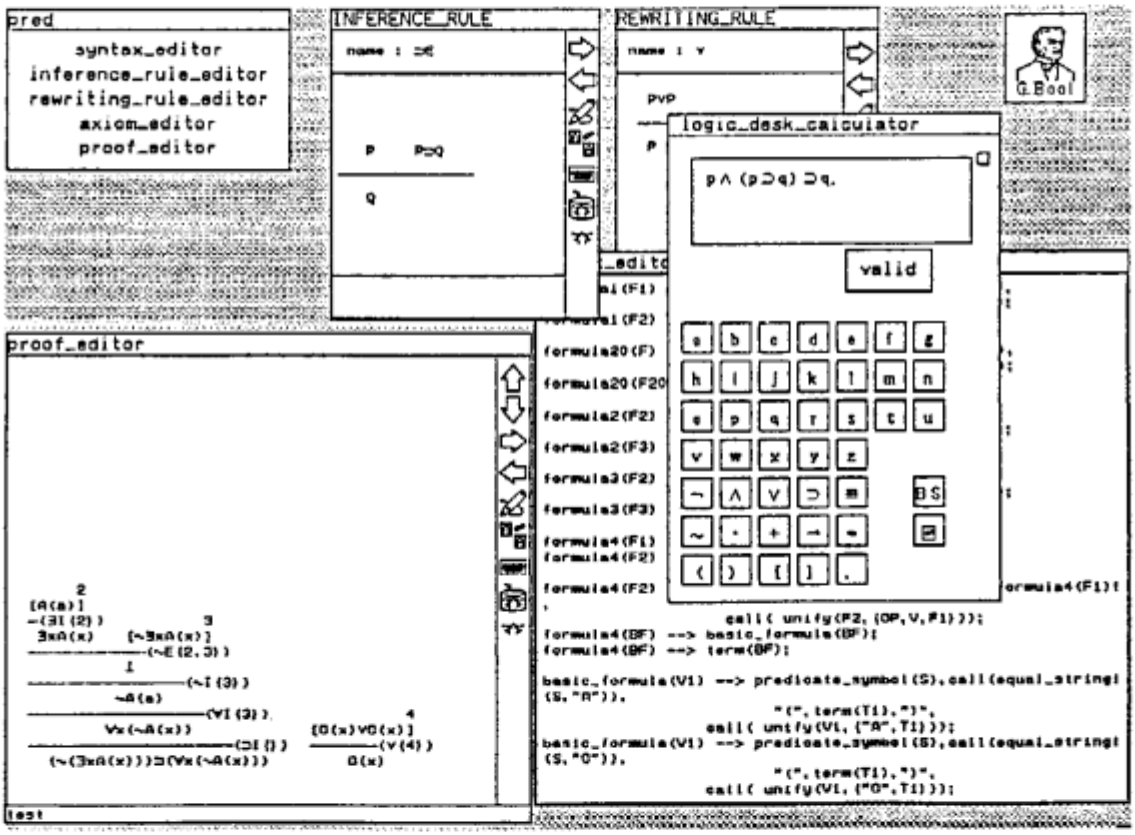
Appendix

In this appendix three formal logics are taken up and typical screen layouts showing proof examples are exhibited.

First-order logic with NK

(1) deMorgan s law

$\vdash \sim\exists x A(x) \supset \forall x \sim A(x)$



(2) Smullyan's logical puzzles (originally examples in combinatory logic)

Axioms :

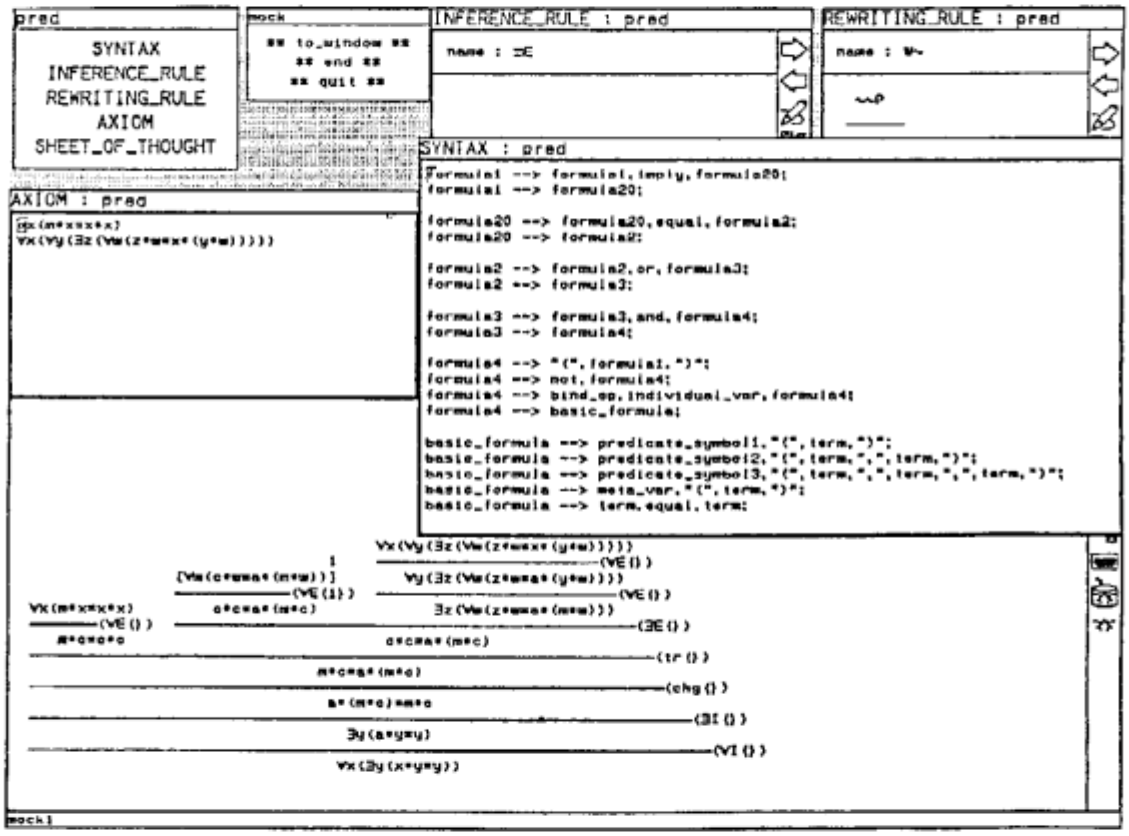
1. $\forall x\ m \bullet x = x \bullet x$ *(Mockingbird condition)*

2. $\forall x \forall y \exists z \forall w\ z \bullet w = x \bullet (y \bullet w)$ *(Composition)*

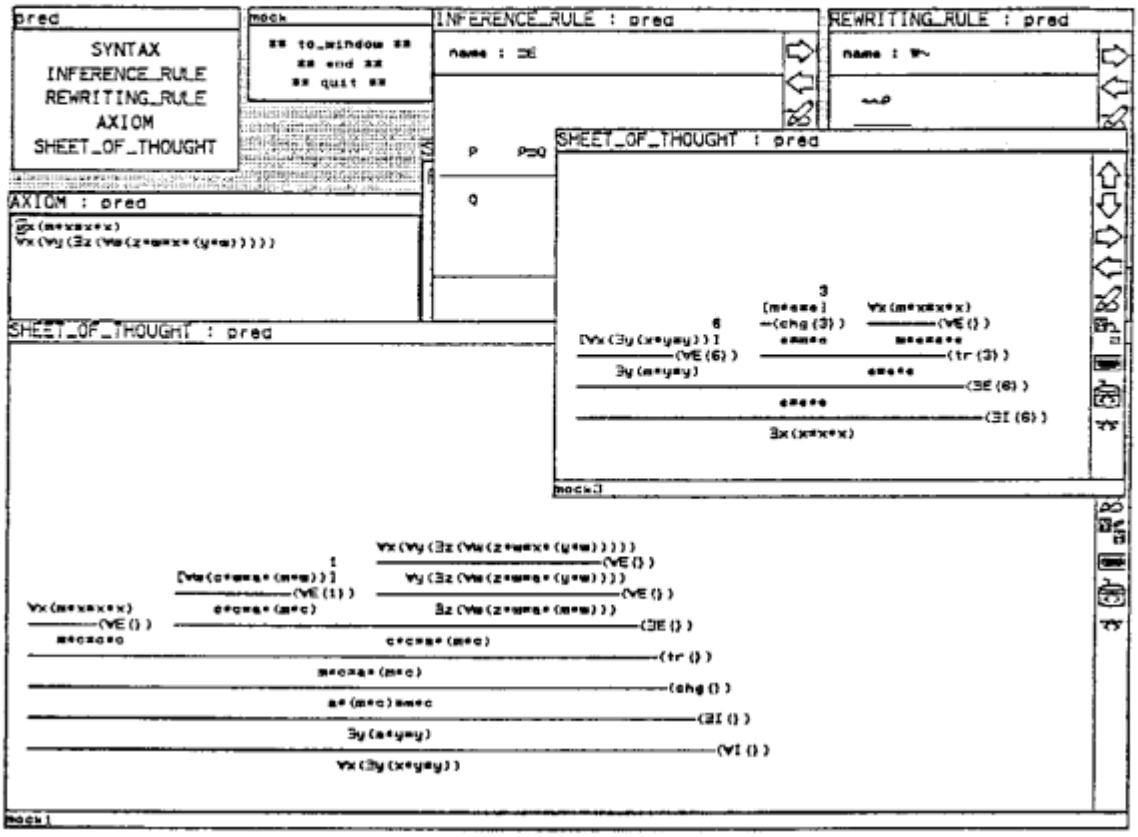
Theorems :

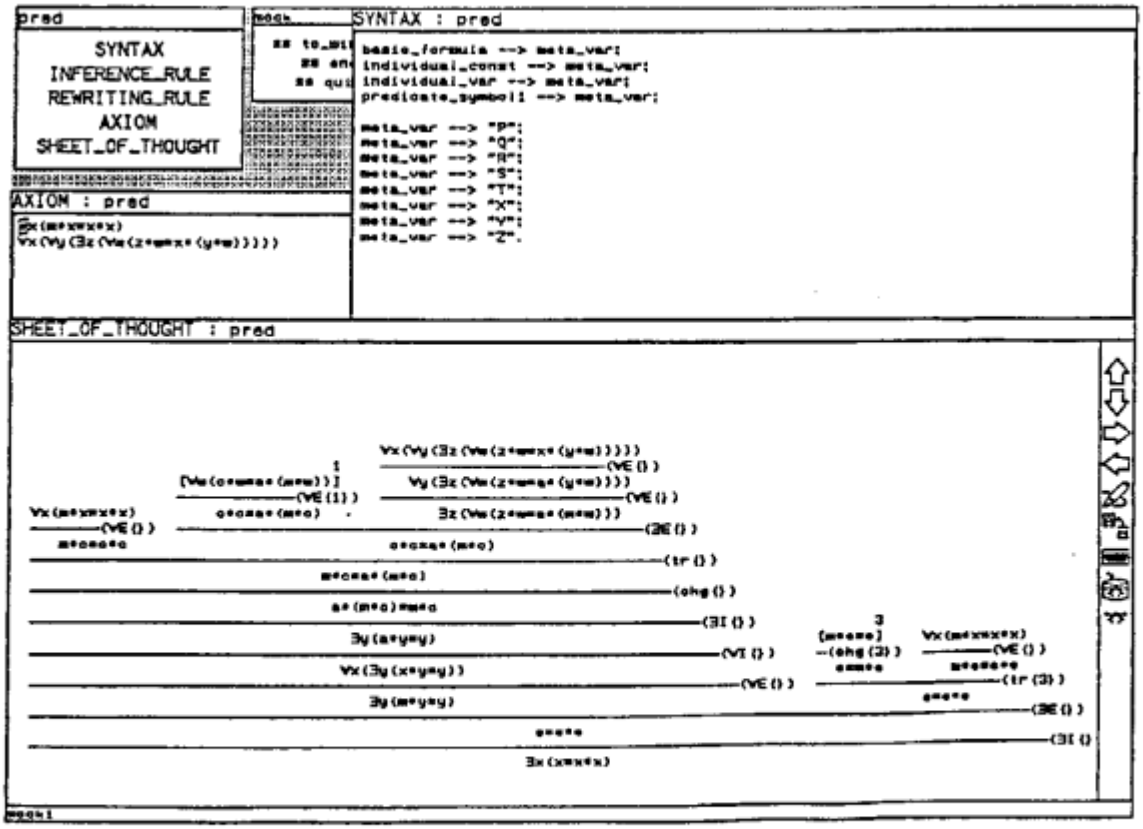
1. $\vdash \forall x \exists y (x \bullet y = y)$ *(Every bird of the forest is fond of at least one bird)*

2. $\vdash \exists x\ (x \bullet x = x)$ *(At least one bird is egocentric or narcissistic)*



(continued)


pred
SYNTAX
INFERENCE_RULE
REWRITING_RULE
AXIOM
SHEET_OF_THOUGHT
mack
** to_window **
** end **
** quit **
INFERENCE_RULE : pred
name : ⊃E
P P⊃Q
Q
REWRITING_RULE : pred
name : ¬¬
¬¬P
AXIOM : pred
SHEET_OF_THOUGHT : pred
SHEET_OF_THOUGHT : pred
mack1
mack2


pred
SYNTAX
INFERENCE_RULE
REWRITING_RULE
AXIOM
SHEET_OF_THOUGHT
mack
SYNTAX : pred
basic_formula --> meta_var;
individual_const --> meta_var;
individual_var --> meta_var;
predicate_symbol1 --> meta_var;
meta_var --> "P";
meta_var --> "Q";
meta_var --> "R";
meta_var --> "S";
meta_var --> "T";
meta_var --> "X";
meta_var --> "Y";
meta_var --> "Z".
AXIOM : pred
SHEET_OF_THOUGHT : pred
mack1

(3) Unsolvability of the halting problem

Premises :

1. $\exists x(A(x) \,\&\, \forall y(C(y) \supset \forall z D(x,y,z))) \supset \exists w(C(w) \,\&\, \forall y(C(y) \supset \forall z D(w,y,z)))$

   (Church's thesis)

2. $\forall w(C(w) \,\&\, \forall y(C(y) \supset \forall z D(w,y,z)) \supset \forall y \forall z((C(y) \,\&\, H(y,z) \supset H(w,y,z) \,\&O(w,g)) \,\&\, (C(y) \,\&\, \sim H(y,z) \supset H(w,y,z) \,\&\, O(w,b))))$

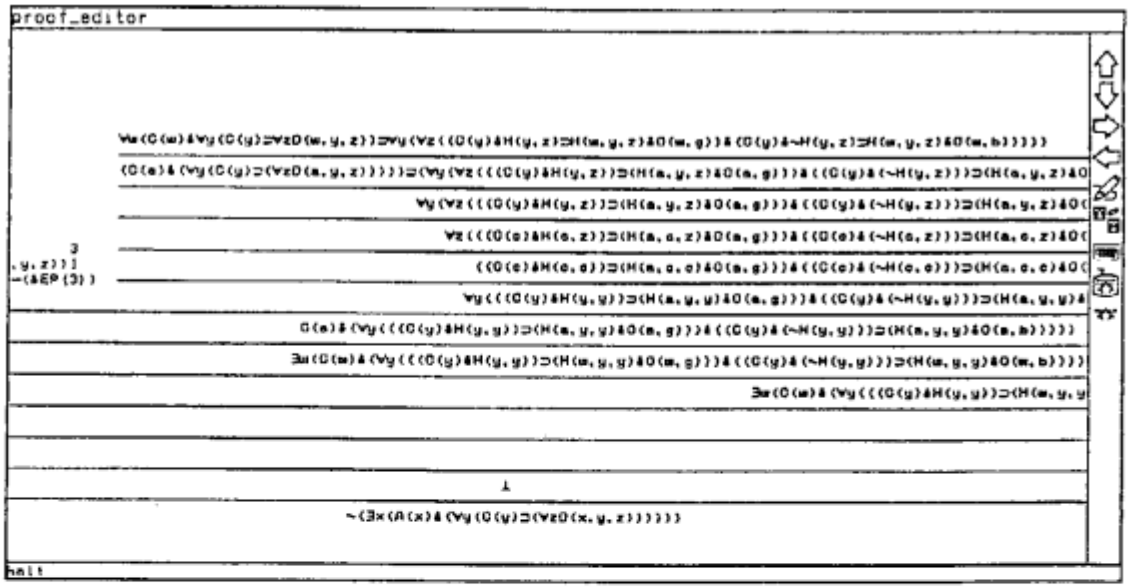
3. $\exists w(C(w) \,\&\, \forall y((C(y) \,\&\, H(y,y) \supset H(w,y,y) \,\&\, O(w,g)) \,\&(C(y) \,\&\, \sim H(y,y) \supset H(w,y,y) \,\&\, O(w,b)))) \supset \exists v(C(v) \,\&\, \forall y((C(y) \,\&\, H(y,y) \supset H(v,y) \,\&\, O(v,g)) \,\&\, (C(y) \,\&\, \sim H(y,y) \supset H(v,y) \,\&\, O(v,b))))$

4. $\exists v(C(v) \,\&\, \forall y((C(y) \,\&\, H(y,y) \supset H(v,y) \,\&\, O(v,g)) \,\&\, (C(y) \,\&\, \sim H(y,y) \supset H(v,y) \,\&\, O(v,b))) \supset \exists u(C(u) \,\&\, \forall y((C(y) \,\&\, H(y,y) \supset \sim H(u,y)) \,\&\, (C(y) \,\&\, \sim H(y,y) \supset H(u,y) \,\&\, O(u,b)))$

Conclusion :

$\vdash \sim\exists x(A(x) \,\&\, \forall y(C(y) \supset \forall z D(x,y,z)))$

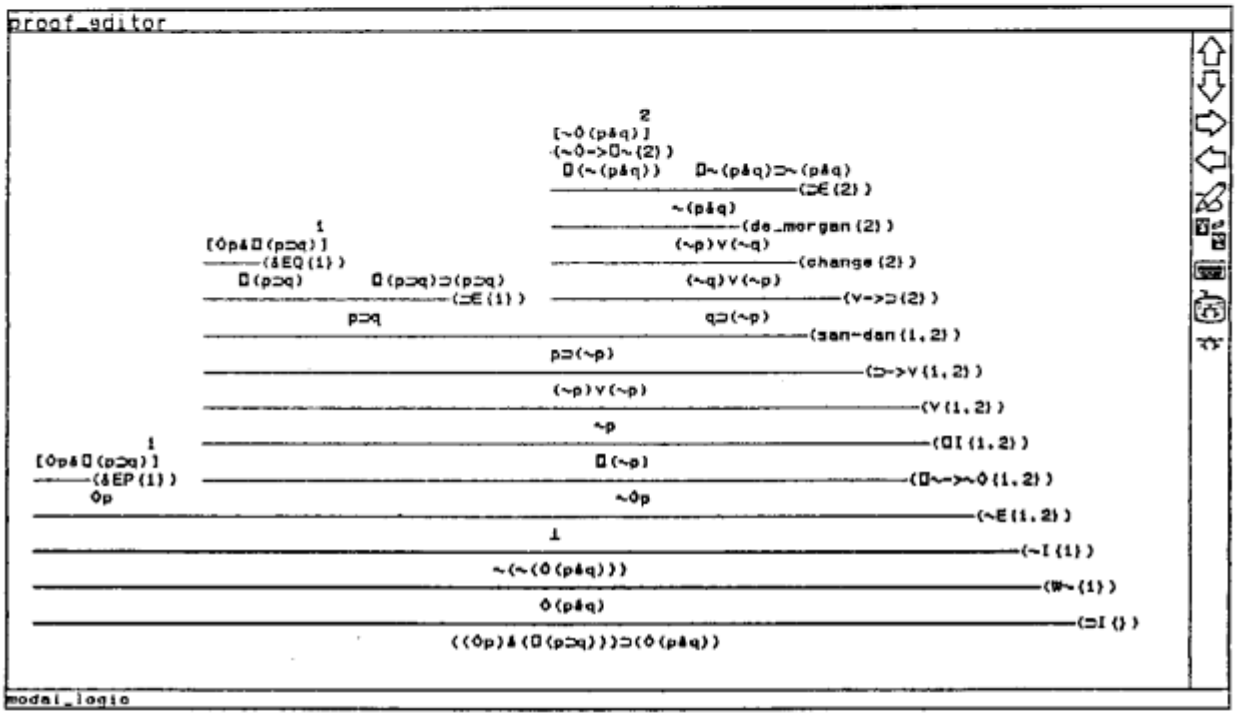
(no algorithm to solve the halting problem exists)



Propositional modal logic

$\vdash \Diamond p \wedge \Box(p \supset q) \supset \Diamond(p \wedge q)$

(A strong correctness assertion is implied from a termination assertion and a weak correctness assertion)



Martin-Löf's intuitionistic type theory

$\vdash \sim\sim(P \vee \sim P)$ $(\equiv (P \vee (P \supset \bot) \supset \bot) \supset \bot)$

(The law of excluded middle cannot be refuted)