An Interactive Theorem Proving Assistant

Ulrich Endriss

Department of Computer Science, King's College London,
Strand, London WC2R 2LS, UK, Email: endriss@dcs.kcl.ac.uk
URL: http://www.dcs.kcl.ac.uk/~endriss/WinKE/

Abstract. This paper describes WinKE, an interactive proof assistant, which is based on the KE calculus. The software has been designed to serve as a tutoring system supporting the teaching of logic and theorem proving through KE.

1 Introduction

The KE calculus [4] is a refutation system close to the common method of semantic tableaux. The main difference between the two is, that KE is explicitly *not* cut-free. Its analytic cut rule, *PB*, is the only branching rule of the system. Elsewhere [3] it has been argued that KE might be better suited for teaching elementary classical logic than for instance Tableau. The first logic textbook based on KE has been published recently [5].

Even though KE proofs are essentially shorter than Tableau proofs [3, 4], the traditional way of manually building up such trees is – within the teaching context – hardly feasible for examples exceeding, say, five branches, and is in general very time consuming and error-prone. The use of a proof assistant with a strong graphical user interface can help to overcome such problems. It may be used for demonstration purposes during classes or as an interactive learning environment for students working on their coursework. WinKE has been designed to meet those requirements. First of all, it serves as a 'drawing board' for constructing KE proof trees. On top of that various levels of user support are provided, ranging from basic bookkeeping facilities to a fully automated theorem prover for propositional and first order logic. WinKE's design was strongly inspired by the work described in [6]. The program runs under Windows and has been implemented in LPA WinProlog.

In the sequel WinKE's interface and its most important features are described. The last section briefly compares WinKE with other programs of similar objectives.

2 Interface and Graphic Tools

WinKE's interface consists of four windows (see Fig. 1), all of which are opened after the program has been started. The large window is used to display the currently active proof tree. Whenever a particular action requires a specific formula

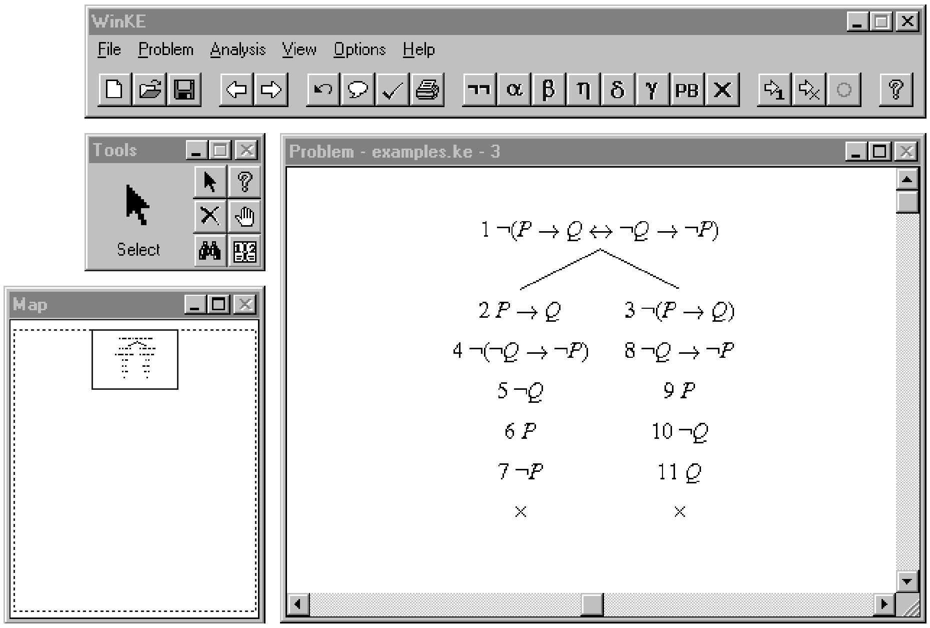

Fig. 1. The WinKE Interface

to be selected, this is done by clicking on that formula on the tree using the mouse. In what way the system will react on such a selection depends on the graphic tool chosen. A graphic tool can be selected from the graphic tool box, just as in any standard graphics software for Windows. The main window contains all the menus to call dialogues for the user's interaction with the program. The buttons on that window provide shortcuts to menu options likely to be used frequently. Finally, the window in the lower left-hand corner can be used as a viewer to navigate around large proof trees that do not fit onto a single screen.

Proof trees displayed in the graphic window consist of graphical objects, which are either formulae or so-called branch markers used to refer to a certain branch of a tree. A branch marker is either represented as a circle (for open branches) or as a cross (for closed branches), placed below the last formula of that branch. Every formula is associated with a certain number, which can be used to refer to parent formulae.

The default graphic tool is the select tool. Clicking on a formula or a branch marker with the select tool will highlight that object. The user may then choose a particular action (by choosing a menu option) to be applied to the selected objects. This will typically be an application of a KE rule. Where necessary the user is prompted for further input (via a dialogue), e.g. the conclusion of a rule application. Then the tree is expanded accordingly. The formulae on a tree are automatically grouped in a space-saving and 'aesthetic' way, thus making sure the user can concentrate on the semantics of a proof tree, instead of its layout.

Two different graphic tools to delete formulae from a tree are provided, the delete and a retract tool. The former simply prunes the tree at the clicked formula, whereas the retract tool only deletes those formulae that logically depend on the clicked one, i.e. that could not have been derived without that formula being on the same branch. This is completed by a standard 'undo' option available from the menus.

The hint tool applied to an open branch marker will highlight all formulae that have not yet been analysed on the associated branch. Vice versa clicking a formula will highlight all open branch markers denoting a branch which that formula has not yet been analysed on. Finally, the bookkeeping tool will display the bookkeeping information available for each node. If that node is a formula, the bookkeeping information consists of the KE rule used to derive it, the parent formula(e), and possibly the sibling formula. In addition, formulae that are either analysed or subsumed on all open branches are marked. If the node clicked on is a closed branch marker, the bookkeeping tool reveals which pair of formulae has been used to close that branch. The button showing a question mark can be used to enter the WinKE help system directly at the section on graphic tools.

3 Deduction and Countermodels

Typically WinKE is used to perform a step-by-step deduction. The system provides three different modes, namely the supervisor, the pedagogue, and the assistant mode. In supervisor mode within the rule application dialogues (for an example see Fig. 2) any (syntactically correct) input is accepted, whereas in pedagogue mode the correctness of the rule applications is checked on-line. The same is true for the assistant, but here the user's input is reduced to a minimum. That means, for the simple rules (the propositional ones apart from PB), no input of the conclusion(s) is required as their derivation is straightforward given the premise(s). For the other rules the system gives a list of possible inputs to choose from (alternatively, the user may also type in a formula). In case the supervisor mode has been used, WinKE also provides off-line proof checking. This will display all errors on a tree in turn and allow to retract the wrong formulae directly. For novice users the pedagogue mode will be the most useful one. After some training, possibly in an exam-like context, the supervisor mode may be used. Once a student is familiar with the basics, the assistant mode provides a comfortable way for studying KE more profoundly, for example by comparing different ways of proving the same theorem.

For the on- as well as for the off-line checking the user may choose the level of error reporting. Only the very basic KE rules are checked in any case, in addition you may or may not add checking for beta simplification (subsumption), analytic application of PB, and/or checking of the order of rule applications (like for example: analyse an alpha formula before you split a branch using PB, etc.).

In particular to make the system a more convenient assistant, but also to be able to demonstrate proofs to novice users, the option to automatically derive (parts of) proofs has been added. You can either ask WinKE to perform the next

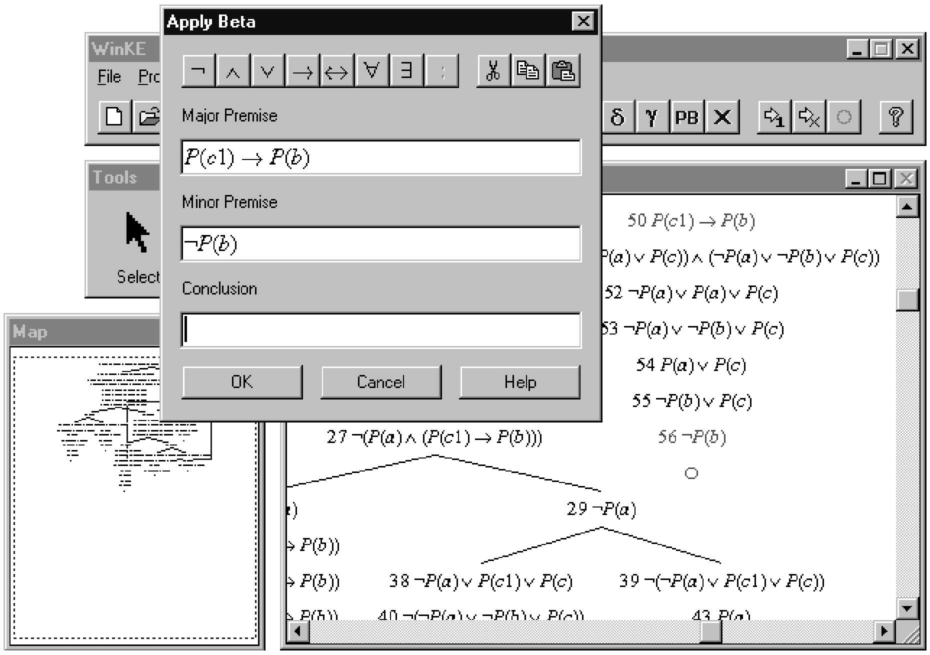

Fig. 2. Applying a Beta Rule

proof step on a selected branch automatically, to finish a branch, or to complete an entire proof.

For consistent sets of formulae, i.e. if there are open branches that cannot be closed, WinKE can automatically derive the description of a countermodel. Moreover, for certain classes of problems a graphical visualization of a countermodel may be displayed. If for instance the countermodel just contains a single 2-ary predicate and the number of terms appearing is limited, the positive atoms in the model can be represented as edges in a graph. Another example where visualization is possible is the class of (simple) 'pigeon hole' problems.

4 Additional Features

KE problems are saved in files, either as problems, proofs, or incomplete proofs. Within the program you can jump between different problems of the same file. Problem files are edited in the same environment as they are worked on. You have the option to cut and paste from existing problems when defining new ones. This offers a comfortable way for teachers to write up and test new exercises. Students could be encouraged to make their own experiments trying different sets of formulae.

Every problem is associated with a text of arbitrary length. Also that text can be edited and read directly within WinKE. In the context of a student exercise it might contain hints for finding a solution or a reference to a page of a textbook. Other features available include printing and generating LaTeX descriptions of proof trees. Parts of the functionality of WinKE can be made password protected, for example to disable automated proving, the assistant mode, or the proof checker. The tool is completed by a comprehensive on-line help system.

5 Conclusion

WinKE's principal task is to support teaching in the context of an introductory course on elementary classical logic. The software is complementary to the logic textbook [5], which is based on KE. Evaluation copies of the software are available on request.

Other logic tutors include popular programs like Tarski's World [1] and Hyperproof [2]. Using Tarski's World students are asked to verify first order formulae stating propositions about simple worlds inhabited by geometric objects, but unlike WinKE the program does not deploy a systematic proof procedure. Hyperproof is used to construct proofs of statements about that same geometric world applying a natural deduction like calculus. As it is restricted to examples of that particular domain it is difficult to be compared with WinKE. WinKE has been designed to simulate an existing proof procedure. In that sense it is supportive of the teaching process. For Hyperproof, on the contrary, teaching is more likely to be centered around the software.

The Tableau II program [7] is based on semantic tableaux and therefore much closer to WinKE than the other two systems. As far as interface and usability are concerned WinKE clearly offers noticeable advantages over Tableau II.

Acknowledgments. This work has partly been supported by CARID (Centro di Ateneo per la Ricerca e l'Innovazione Didattica) at the University of Ferrara. The author would like to thank Jeremy Pitt, Marcello D'Agostino, Marco Mondadori, and Dov Gabbay for their help and support, and two anonymous referees for their valuable comments.

References

1. J. Barwise and J. Etchemendy. *Tarski's World*. CSLI Publications, Stanford, 1991.
2. J. Barwise and J. Etchemendy. *Hyperproof*. CSLI Publications, Stanford, 1994.
3. K. Broda, M. D'Agostino, and M. Mondadori. A solution to a problem of Popper. In *The Epistemology of Karl Popper*. Kluwer Academic Publishers. To appear.
4. M. D'Agostino and M. Mondadori. The taming of the cut. Classical refutations with analytic cut. *Journal of Logic and Computation*, 3:285–319, 1994.
5. M. Mondadori and M. D'Agostino. *Logica*. Edizioni Scolastiche Bruno Mondadori, Milan, 1997. English translation in preparation.
6. J. Pitt. MacKE: Yet another proof assistant & automated pedagogic tool. In P. Baumgärtner *et al.*, editors, *Theorem Proving with Analytic Tableaux and Related Methods (TABLEAUX'95)*, vol. 918 of *LNAI*, pages 324–337. Springer-Verlag, 1995.
7. M. Potter and D. Watt. Tableau II: A logic teaching program. Technical report, Oxford University Computing Services, Learning and Resource Centre, Oxford, 1988.