# Using Interaction and Visualisation for Teaching Deductive Reasoning

**Ulrich Endriss**

*Department of Computer Science, King's College London,
Strand, London WC2R 2LS, UK, Email: endriss@dcs.kcl.ac.uk*

In this paper we discuss how computers can be deployed to support teaching deductive reasoning at university level. These considerations led to the development of the interactive theorem proving assistant and learning environment WinKE.

We start out by describing common difficulties in teaching logic and deductive reasoning and thereby justify the need for computer support in that field. We identify interaction (with a formal system) and visualisation (of abstract concepts) as two important components in a helpful pedagogical tool for teaching deductive reasoning.

Thereafter the WinKE software, which deploys the logical calculus KE, is described. Important features include a comfortable graphical user interface, various levels of user support, automated deduction, visualisation of counter models, editing problem files, and many more. We conclude with a short discussion of related work, comparing WinKE with some other tools for similar tasks.

Keywords: **Interactive Learning Environments, IT Education, Logic**

## 1 Teaching Logic and Deductive Reasoning

Elementary logic nowadays is widely regarded as one of the key ingredients for a theoretically well founded Computer Science education. For most students concerned, however, logic is not their field of specialisation, and it often imposes great difficulties on them to even acquire a working knowledge of the very basics. Such problems are by no means unique to teaching logic, but indeed very common in many areas that require imparting knowledge about abstract concepts. This is well known and various examples for developing interactive learning environments or intelligent tutoring systems to help overcoming these problems are reported in the literature, like e.g. [7] (computability and Turing machines) and [4] (formal languages and automata).

The typical task in deductive reasoning is to show whether a certain formula (the conclusion) logically follows from a set of other formulas (the premises). The most popular deduction systems used for teaching are Tableau, and natural deduction. In Tableau a proof is performed by extending the initial set of formulas by further formulas obtained from the application of a number of analytical rules. These formulas are structured as a tree. KE [3] is a Tableau-like system, which combines the representational advantages of Tableau with some of the semantical advantages present in the natural deduction method. (It would be beyond the scope of this paper to discuss the various deduction methods in detail. A prominent textbook covering Tableau, natural deduction, and various other systems is [5].) Still, even when using KE, following a deduction that involves constructing a proof tree with several branches can be very difficult. Also, students cannot really be expected to solve anything but trivial exercises during tutorials or exams. This is not so much due to conceptual difficulties inherent in the reasoning systems as such, but a simple consequence of the representational complexity of systematic proofs—whatever method is used. Paper seems not to be a suitable medium for carrying out systematic logical deductions.

## 2 Interaction and Visualisation

Computers clearly *are* suited to carry out formal proofs in logical systems. This is exactly what is done in the area of automated theorem proving and there exist a number of very useful theorem provers that are used in many computer science applications. Such programs however are not particularly useful for teaching purposes. The knowledge required by a student to help her or him to really understand how a deduction method functions can only be gained through interaction with that method.

The intellectual challenge of finding a proof for a theorem should still be left to the student, also when a computer program is used, but the program can assist the student in handling the data (formulas and branches) involved. A flexible graphical user interface allows to concentrate locally on interesting aspects of a complex proof as well as grasping the structural essence of the entire deduction.

This suggests that aspects of interaction and visualisation constitute important guidelines for the design of a tutoring system implementing a logical calculus like KE. To allow a user's interaction with a formal reasoning system such a software tool firstly has to serve as an electronic drawing board simulating what is traditionally done on paper. Furthermore it could report errors or give hints. Visualisation firstly means making the structure of a proof as transparent as possible to the user. Therefore, every information relevant to any part of a proof should be easily accessible. On the other hand displaying all that information at the same time would not serve the intended use, but make it difficult to overview a proof tree as a whole. With visualisation we also mean providing graphical representations of other abstract concepts apart from the proof tree itself. This will be done in the case of counter models as described in the next section.

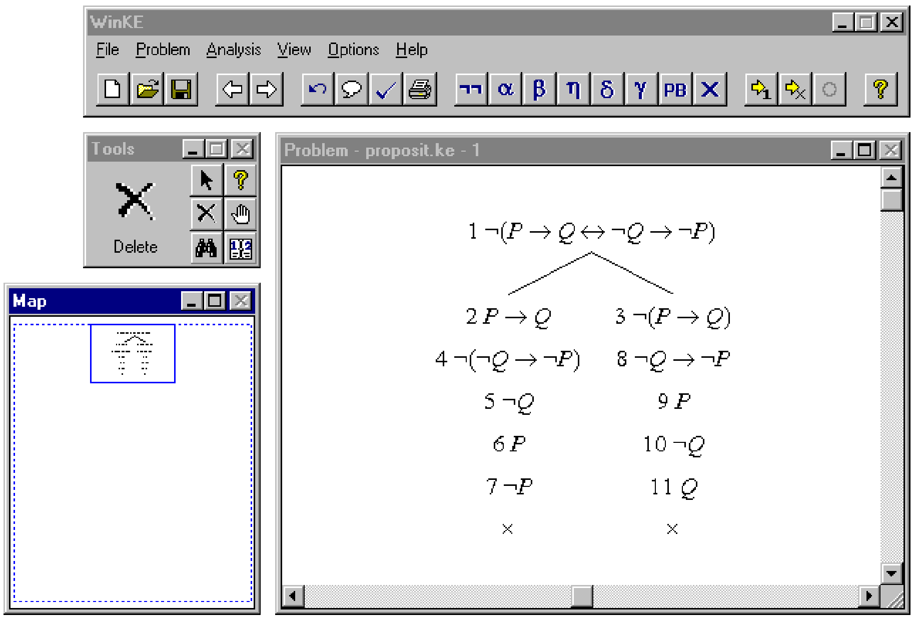

WinKE
File Problem Analysis View Options Help
PB
Tools
Delete
Map
Problem - proposit.ke - 1
1 $\neg(P \to Q \leftrightarrow \neg Q \to \neg P)$
2 $P \to Q$
3 $\neg(P \to Q)$
4 $\neg(\neg Q \to \neg P)$
8 $\neg Q \to \neg P$
5 $\neg Q$
9 $P$
6 $P$
10 $\neg Q$
7 $\neg P$
11 $Q$
×
×

**Fig. 1.** WinKE displaying a simple KE proof

## 3 The WinKE Software

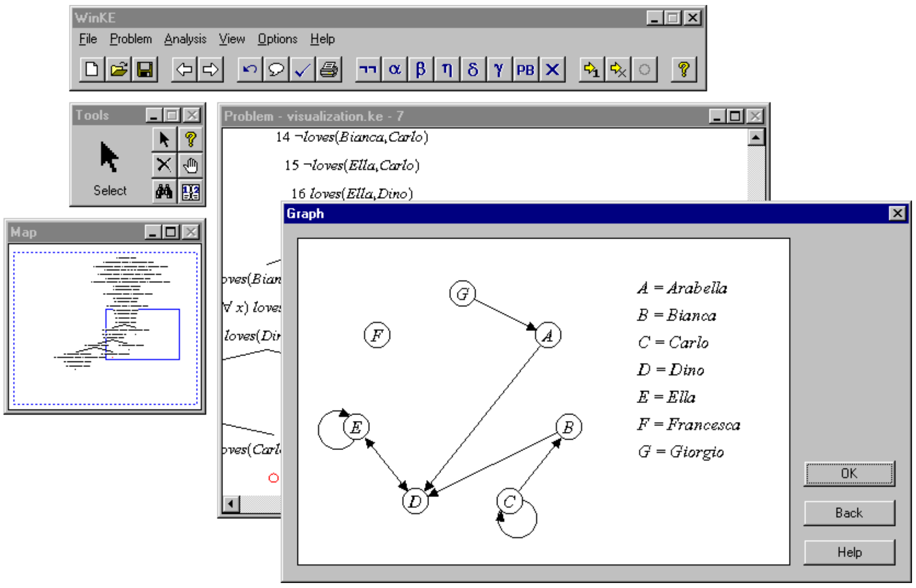
WinKE is a pedagogical theorem proving assistant based on the KE calculus, which provides a suitable learning environment for interacting with a formal deduction method. It is running under Windows and shares common standards with other Windows software products, which makes it easier to learn how to use the system.

A large graphical window is used to display proof trees. Fig. 1 shows a sample KE proof for the theorem $(P \to Q) \leftrightarrow (\neg Q \to \neg P)$. A second, smaller graphical window represents a viewer that can be used to navigate around trees not fitting onto a single screen. Interaction between the user and the program takes place through mouse clicks, menus, buttons, and dialogs. The system's particular reaction on a mouse click is determined by the choice of a graphic tool from the fourth window.

The different graphic tools can be used to delete formulas, to display information on the status of a particular formula, or to query the system for hints about possible further proof steps. The default choice is a selection tool, which highlights the clicked objects. Actions requested by choosing a menu option or pressing a button are carried out with respect to the currently selected objects. Typically, such an action will be the application of a KE rule.

WinKE provides three different teaching modes. The one best suited for beginners is the *pedagogue mode*. In that mode every rule application will be checked on-line. In *supervisor mode* on the other hand the user's input is only checked for basic syntactical correctness to prevent typing errors. At any stage a proof built up that way may be checked off-line. Finally, the *assistant mode* is suited for users already familiar with the basics. It offers advanced support in the sense that the user's input is reduced to a minimum; all trivial steps are carried out automatically. In the other cases the system provides a set of formulas to choose from. On top of the *assistant mode* WinKE implements a fully automated theorem prover for first order logic.

An important concept in deductive reasoning is that of a counter model. If a formula is not a tautology, there exists a model, which exemplifies a case where the formula is false. For certain classes of formulas it is possible to visualise counter models, e.g. models of formulas with a single 2-ary predicate can be visualised as graphs, where the arrows correspond to instances of that predicate (see Fig. 2).



14 $\neg loves(Bianca,Carlo)$

15 $\neg loves(Ella,Carlo)$

16 $loves(Ella,Dino)$

$A = Arabella$
$B = Bianca$
$C = Carlo$
$D = Dino$
$E = Ella$
$F = Francesca$
$G = Giorgio$

**Fig. 2.** Visualisation of a counter model

The software comes with a collection of sample problems, partly taken from [6]. Further examples can be edited directly within WinKE. Like that lecturers can easily set up there own exercise files and, given the automated deduction feature, also have a comfortable way of testing them directly. Students have the chance to experiment with many different sets of formulas in a supportive environment. Parts of the program, like the automated theorem prover, certain teaching modes, or the hint tool, can be disabled, i.e. it is possible to tailor the system's functionality to users (novices or advanced students) and situations (exercises, exams, experiments).

WinKE has been written entirely LPA's WinProlog (Logic Programming Associates Ltd), which combines classical Prolog syntax with a Windows programming environment that allows the realisation of a comfortable graphical user interface.

## 4 Related Work and Conclusion

We already mentioned the existence of a variety of systems, like the ones reported in [4] and [7], to support teaching in other areas of theoretical computer science. Also for logics and reasoning several programs have been developed, some of them, like the *Tableau* system [9] or *MacKE* [8] for similar purposes as WinKE. In fact, WinKE originated from an evaluation of *MacKE* and the subsequent redesign process. *Tableau* is based on the method of analytic tableaux. It has successfully been used in teaching for many years, but the DOS-interface does not comply with today's standards anymore. Also, interacting with the system is not as intuitive as it could be. A useful feature, which on the other hand is not present in WinKE, is the storing of information on how individual students performed in previous sessions.

Other logic tutors include *Tarski's World* [1] and *Hyperproof* [2]. The former mainly supports learning how to translate ‘real world’ situations into logic. The ‘real world’ in that particular case is a chessboard inhabited by some simple geometric objects. The *Hyperproof* program 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 is difficult to be compared with WinKE.

To conclude, we believe WinKE provides a useful tool in overcoming some of the problems common in teaching logic and deductive reasoning. This view is supported by an (informal) classroom evaluation that has been carried out at the University of Ferrara and the feedback from users at other universities.

Further information on the software (including how to obtain an evaluation copy) and on the KE calculus can be found at http://www.dcs.kcl.ac.uk/~endriss/WinKE/.

*Acknowledgements.* The author would like to thank Marco Mondadori, Marcello D'Agostino, Jeremy Pitt, and Dov Gabbay for their help and support. This research has been funded by the Academic Centre for Research and Didactic Innovation (CARID) at the University of Ferrara, Italy.

## 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] 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.
[4] V. Devedzic and J. Debenham. An intelligent tutoring system for teaching formal languages. In B. Goettl et al., editors, *Intelligent Tutoring Systems. Proceedings of ITS'98*, LNCS 1452, pages 514—523. Springer-Verlag, 1998.
[5] M. Fitting. *First-Order Logic and Automated Theorem Proving*. Springer-Verlag, second edition, 1996.
[6] M. Mondadori and M. D'Agostino. *Logica*. Edizioni Scolastiche Bruno Mondadori, Milan, 1997.
[7] C. Pape and P. H. Schmitt. Visualizations for proof presentation in theoretical computer science education. In Z. Halim et al., editors, *Proceedings of the International Conference on Computers in Education*, pages 229—236. Association for the Advancement of Computing in Education, 1997.
[8] J. Pitt. MacKE: Yet another proof assistant & automated pedagogic tool. In P. Baumgaertner et al., editors, *Theorem Proving with Analytic Tableaux and Related Methods. Proceedings of Tableaux'95*, LNAI 918, pages 324—337. Springer-Verlag, 1995.
[9] M. Potter and D. Watt. *Tableau II: A logic teaching program*. Technical report, Oxford University Computing Services, Learning and Resource Centre, Oxford, 1988.