First International Congress on Tools for Teaching Logic Masters theses for providing feedback to the Logic Classroom

G. Aguilera I.P. de Guzmán M. Ojeda
A. Valverde
aciego@ctima.uma.es
Dpt. Matemática Aplicada
Universidad de Málaga

September 6, 2000

Abstract

In this work we present di¤erent graphical implementations of tools for teaching logic, which have been developed under the supervision of our group and have shown useful for our teaching to students of the subjects of Logic for Computer Science in the Faculty of Computer Science of the University of Málaga.

1 Introduction

The content of the subjects of "Logic for Computer Science" (LCS)in the Faculty of Computer Science of the University of Málaga are strongly directed to the study of automated deduction systems, for Classical Logic in LCS I and for Non-Classical Logics in LCS II. One of the …rst things that the student has to learn is that automated reasoning is guided by syntactic rules and is not based on "intuition". Our teaching experience has shown the usefulness of using automated tools in the classroom. The programs needed have been developed by our students as Master theses under our supervision, as teachers of the subjects mentioned above. The programs generated in those MSc theses have been used as a pedagogical tool for helping students to learn Logic.

Natural Deduction is a special case because it is not an automatic method, the student has a limited range of rules but he can choose them using his own strategy. This is an intermediate level between the way in which he normally reasons and the automatic way of the Automated Theorem Provers (ATPs). The tool developed for this case provides an environment for making deductions in a natural deduction system which helps students to keep on the syntactical level, o¤ers the possibility of save deductions as "derived rules" and, in addition, has a package with useful examples of ways of making deductions as well as making new deductions in a higher level.

One of the methods studied is the TAS[1] methodology, which has been developed by our research group GIMAC. All the other tools are related with it. This methodology has proven to be an e¢cient way for constructing automated theorem provers not only for classical logic but also for temporal, modal and multi-valued logics. Implementations of those ATPs for di¤erent logics have been made: TAS-D [2], [3], TAS-Kt [4], TAS-Fnext [5], TAS-M3 [10]. The programs have been written using graphical environments such as Windows or KDE in Linux. The main feature of TAS is that it works on the syntactic tree of the formula by means of repeated dynamic transformations. This dynamism is di¢cult to transmit to the students using a blackboard or slides. This way, the graphical environments have been chosen not only for commodity reasons for the users but, specially, because of the possibility of seeing the syntactic trees of the formulae changing dynamically step by step. It has proven be very useful for helping students to understand how the method works.

Another implementation of TAS-D has been made using ¢-trees. This program has been written in the functional language CAML and shows graphically every step in the algorithm. This tools has been useful this year because is the …rst time we have taught the algorithm TAS-D using ¢-trees.

2 A tutorial for natural deduction

"Graphical environment for helping natural deductions with pedagogical applications" [7] is a program for helping students to learn Natural Deduction. Using this tool, students can test the correctness of their own deductions.

Using the program helps student to convince themselves of the importance of being strict with the syntax. The graphical environment, developed for a Windows operative system using C++, provides a friendly and visual framework for making deductions.

[1] TAS stands for Syntactic Tree Transformations (in Spanish).

Moreover, deductions can be saved and used as derived rules in subsequent deductions. This possibility allows the students making deductions at a higher level of complexity, for they can load saved deductions and/or use them as examples.

The system gives error messages to the user in the cases of bad use of the syntax or when trying to incorrectly apply some rule. The graphical presentation of the deduction helps students to understand additional hypotheses. The election of the rule or the derived rule in the data-base with a menu system provides a comfortable framework for making deductions.

3 Graphical implementation(s) of TAS

"Syntactic tree transformations" [6] was the …rst implementation of the algorithm TAS-D made in a graphical environment, speci…cally, it was implemented using C++ for Windows. This implementation was based on a previous version in text mode for DOS. This graphical version has been the most used because it shows the TAS methodology in its easiest way, that is, applied to Classical Propositional Logic.

The most useful options of the tool, from a pedagogical point of view, are the possibility of tracing the execution of the algorithm, showing step by step the resulting syntactic tree,and also the facility of applying every step of the algorithm in an interactive mode. If the analysed formula is not valid, then a counter model is provided.

Other implementations have added new capabilities that had been theoretically incorporated to the algorithm. This is the case of "E¢cient treatment of the information of the implicants in automatic deduction" [8], which uses ¢-lists instead of ¢-sets and the case of "Automatic deduction based on ¢-trees" [1], the latter has been made using C++ in Linux for a KDE desktop. It has proven to be specially useful in the present year because this year we have taught the TAS methodology using ¢-trees instead of using the standard syntactic trees.

One of the implementation of TAS methodology applied to a non Classical Logic was "An automatic prover for temporal logic: TAS-K_t^+" [9]. This program o¤ers an automatic tool for proofs in the temporal logic K_t^+, which is the future fragment of the minimal temporal logic K_t^+ introduced by J. Lemmon. In this logic there are no restrictions on time, so the future is not a line but a tree. Because of this, our graphical tool is especially useful, not only for understanding the steps in the algorithm but also for seeing the graphical representation of the countermodel as a tree of instants.

All previous implementations have been developed using the imperative language C++, recently a new implementation in CAML has been released. This implementation is interesting because of several reasons: on one hand a functional programming language has been used and then the very code has pedagogical interest by itself, since it can be used as a description of the algorithm in the classroom. On the other hand this implementation uses the data-structure of ¢-tree to represent boolean formulas, since it has proven to be more adequate than the data-structure of syntactic tree for the implementation of TAS reductions.

References

[1] Manuel Aguilera Hidalgo. Demostración automática basada en ¢-árboles. Master's thesis, Universidad de Málaga, España, 2000.

[2] G. Aguilera, I. P. de Guzmán, and M. Ojeda-Aciego. Increasing the e¢ciency of automated theorem proving. Journal of Applied Non-Classical Logics, 5(1):9–29, 1995.

[3] G. Aguilera, I. P. de Guzmán, M. Ojeda-Aciego, and A. Valverde. Reductions for non-clausal theorem proving. Theoretical Computer Science, 2000. To appear. Available at http://www.ctima.uma.es/aciego/TR/tas-tcs.pdf.

[4] I. P. de Guzmán and M. Enciso. A new and complete theorem prover for temporal logic. In Proceedings of the IJCAI Workshop on Executable Temporal Logics, Montreal (Canada), August 1995.

[5] I. P. de Guzmán, M. Enciso, and C. Rossi. Just one approach for several temporal logics in computing: the topological semantics. In Proceedings of the TIME'95 Workshop, FLAIRS, Melbourne, Florida (USA), April 1995.

[6] Miguel A. Ferrero GarcÌa. Transformaciones de árboles sintácticos. Master's thesis, Universidad de Málaga, España, 1994.

[7] Reyes García Rosado. Implementación de un entorno grá…co de ayuda a la deducción natural con aplicación a la docencia. Master's thesis, Universidad de Málaga, España, 1999.

[8] Antonio Jiménez Vela. Tratamiento e…ciente de la información de los implicantes en demostración automática. Master's thesis, Universidad de Málaga, España, 1998.

[9] Juan F. Moncada García. Demostrador automático para lógica temporal: TAS-Kt+. Master's thesis, Universidad de Málaga, España, 1996.

[10] I. P. de Guzmán, M. Ojeda-Aciego, and A. Valverde. Reducing signed propositional formulas. Soft Computing, 2(4):157–166, 1999.