Software Tools in Logic Education: Some Examples

BELEN PEREZ-LANCHO, ELENA JORGE, ANA DE LA VIUDA and RAQUEL SANCHEZ, *Department of Computer Science, University of Salamanca, Pza. Merced s/n, 37008 Salamanca (Spain).*
E-mail: lancho@usal.es

Abstract

Computers are increasingly present in education and make many resources and activities available to teachers and pupils. New pedagogical resources development is very interesting for both. Our digital library Summa Logicae is overtly involved in innovation and pedagogical systematization. It includes some software tools for teaching logic developed by computer science students, and in this article we present two of these tools. The *MAFIA* tool is especially attractive for first year students and helps them to understand the basic concepts of logic in an interactive way using sematic tableaux. It also allows them to solve the crazy cases in Mafia which their fellow students from previous years proposed. The *Modelos de Kripke* tool, oriented to a more advanced level, serves for understanding the link between the properties of the accessibility relation and the modal formulas, which is at the basis of the current developments of modal logic.

Keywords: Information and Communications Technology (ICT), didactic sofware tools, semantic tableaux, Kripke models.

1 Introduction

Information and Communications Technology (ICT or TIC, in Spanish) is concerned with the use of technology in managing and processing information, especially in large organizations. In particular, it deals with the use of computers and software to convert, store, protect, process, transmit, and retrieve information. The impact of ICT concerns many aspects of society and also the educational environment which, as a living element, is transforming itself and adapting all the time. In this field especially, technical resources must be employed to improve the learning process, to support teaching, to reach more people and to bridge geographical gaps. The classical education model based on knowledge transmission must be changed to another more practical and collaborative one, where the students have an active role, the teacher supervises their work and technology allows them to obtain many resources and activities that simplify education and assessment. Active involvement of the teaching staff is encouraged, by means of projects where teachers interact with their pupils and can get a real feedback about their own performance. The use of ICT can help learners to be creative, is a useful aid to problem solving, provides ready access to a world of knowledge and research, and improves the quality of presentation [5]. It also helps to improve pupils' attainment and capacity to learn.

2 Education in Spanish Universities

A few years ago, Spanish educational policy began to promote the introduction of ICT as a ''*culture of quality*'' at Spanish universities, by establishing an institutional quality

doi:10.1093/jigpal/jzm025

system that guarantees and revitalizes educational activity. This is also one of the keys to confront the challenge of the imminent incorporation to the European Higher Education Area (EHEA, or EEES in Spanish). Other changes introduced by the EHEA are oriented to facilitate student mobility by adopting a common structure for most of the programmes and to introduce a final project in all of them.

The final project (Proyecto Fin de Carrera or PFC in Spanish) is already mandatory at the end of Computer Science Studies. Students must develop and present a complete piece of software that includes the lifecycle phases of analysis, design, implementation and testing. At present this fact raises a serious problem because there are many more students looking for topics than proposals. In recent years we have proposed a number of interactive tools for teaching particular aspects of logic. Thus, a student that has attended logic courses develops a tool that helps others to understand and apply the concepts. Then, although they are not expert programmers, they have enough knowledge, great motivation and the advantage of their recent student experience, which allows them to surmise very well the requirements of the final user. Though the primary purpose of the tools is for them to be used by Computer Science students, they can also be useful for other degrees, such as Mathematics or Philosophy, for example. Within this institutional academic context some of our pupils have developed didactic tools such as the ones we present in this article.

3 Summa Logicae

The development of new pedagogical resources is of paramount importance, but storing, searching and managing a great amount of information are also crucial. Summa Logicae *http://logicae.usal.es* came into being with this idea in mind. It is a digital library, a reference repository for logic students and researchers alike, especially Spanish speakers, overtly involved in innovation and pedagogical systematization. This library has been the result of several research projects referred to in the acknowledgment section.

Articles and texts about Logic are grouped into five main branches: *Fundamentals*, *Logical Systems*, *Applications*, *Studies about Logic* and *Exercises*. It also includes other categories: a *Software* section, with tools for teaching logic, a selection of *Links*, a *Glossary* of logic terms and another section for *Working groups*, specifically devised for students.

In this article we present two examples of software tools. They are included in our digital library, in the Software section, and were also presented at the *Second International Congress on Tools for Teaching Logic (SICTTL)* held in Salamanca in September 2006. The use of the tools in the current academic year will allow the applications to be tested, bugs to be detected and fixed, and new versions developed.

For a first level course *MAFIA* is an interactive tool to help students to understand the basic concepts of classical propositional logic using semantic tableaux. It was developed by Elena Jorge. Ana de la Viuda created the second tool, *Modelos de Kripke* (Kripke Models) oriented to an intermediate level student and which instructs about some aspects of modal logic. At present both applications are only available in Spanish.

4 Example 1: MAFIA

MAFIA is an interactive tool based on propositional semantic tableaux. Students can use it to establish the consistency/inconsistency of a set of propositional formulas, to classify single formulas, to check the consequence of a formula from a set of hypothesis, and also to draw conclusions from a given set of premises. The latter is the situation we frequently find in

everyday life, where we have to reach our own conclusion from given data. It also contains a set of Mafia problems proposed by the students to be solved by the user, i.e. little crime stories that give the Godfather some clues to identify the traitors.

The software has been developed to support an introductory course which teaches the basic notions of logic. Following some of the guidelines for logic education proposed by ASL [1], we are trying to spend some time on amusing logic problems, on real-life examples and we are also producing courseware in logic to help our teaching.

The first part of the course is devoted to explaining the informal notion of consistency, as well as the concept of ''*logically correct argument*'' as one which is immune to counter-examples; also some informal strategies for producing counter-examples to fallacious arguments are presented. The propositional logic is then introduced as an example of a formal language where semantics and syntax are well defined from a mathematical point of view and where the logically valid arguments are tested and even produced in a deductive calculus.

Introducing formal logic from an intuitive and informal starting point means shedding encumbrances. The first step is to eliminate the natural language by defining a formal one, after that the intuitive concept of consistency as compatibility of beliefs has to be substituted by the semantic concept of satisfiability, where we use mathematically defined interpretations instead of the informal idea of situations. Then the intuitive concept of consequence is replaced by the semantic one, where we employ the concept of truth under an interpretation. Finally, even the concept of truth is abandoned in favour of the notion of formal proof, as a syntactic manipulation of the formulas. To each intuitive concept there corresponds a semantic and a syntactic one, the soundness and completeness theorem then show that they are equivalent.

INTUITIVE	*SEMANTIC*	*SYNTACTIC*
Consistency	*Satisfiability*	*Consistency*
Consequence	*Consequence*	*Deducibility*

In the sequel we review the calculus rules and the strategies for building a proof tree and also we present the main features of the software tool.

4.1 Propositional semantic tableaux

The semantic tableaux [9], [2] are a semantic but systematic method of finding a model of a given set of formulas. It is the best system for beginners, especially if the informal notion of consequence is based on consistency, that of an argument for which it is impossible to find a situation where the hypotheses are true and the conclusion false. It is a refutation system in the sense that a theorem is proved by rejecting its negation. Moreover to prove $\Gamma \Vdash \varphi$ we build the semantic tableau of $\Gamma \cup \{\neg\varphi\}$ and refute it.

We have implemented the common tableau calculus for propositional logic, as presented in [6], with two kinds of rules: expansion rules and closing ones. We build the tableau by expanding the formulas on a tree; while β rules (β for ''*branching*'') result in branching, α and σ rules (α for ''*and*'' and σ for ''*simplification*'') develop vertically.

Tableau rules tell us what we can do but not what we should do, therefore some strategies are needed for building an efficient tableau, for example:

1. Apply α-rules first.
2. Choose to apply a β-rule over a formula when one of the branches closes easily.

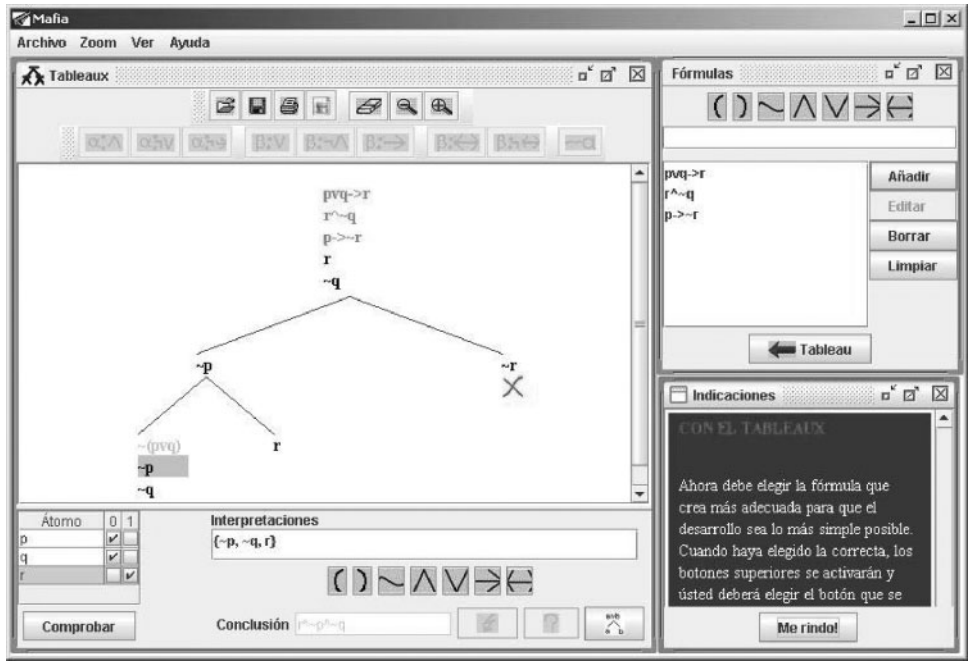

FIG. 1. Solving semantic tableaux.

3. Stop when the problem is solved (only an open branch is needed to prove consistency).
4. When neither 1, 2 or 3 applies, expand the more complex formula.

4.2 The software

MAFIA is a didactic software tool that can be used as a support for first year students of logic. They learn tableaux calculus in the classroom and this software helps them to easily understand the main concepts and to produce their own trees.

The first thing we find when we open the application is a *online tutorial* to guide them throughout. It is divided into two parts, the first one, more technical, explains how to use it, and the second one presents the theoretical concepts that the software can handle.

After that, the application provides an easy *graphical environment* that helps the user to understand the concepts related to the construction and interpretation of semantic tableaux. We can use it in different ways depending on the kind of problem we wish to solve. The options are: (1) *Satisfiability*, (2) *Consequence*, (3) *Classification of formulas* and (4) *Finding the conclusion*. After selecting one of them the user must write the formulas in the upper right edit window. Under it there is a help window which provides some explications. Once the formulas are introduced the tableau can be solved. There are two possibilities for solving it: *Manual* (step by step, where each formula, node and rule to apply must be selected by the user) or *Automatic* (''*Me rindo*'' or ''*I give up*'' button).

To illustrate this main functionality, Fig. 1 shows an example of the fourth option, solving a semantic tableau for a set of formulas and drawing a conclusion from the intersection of the open branches. An interpretation can be seen under the graph.

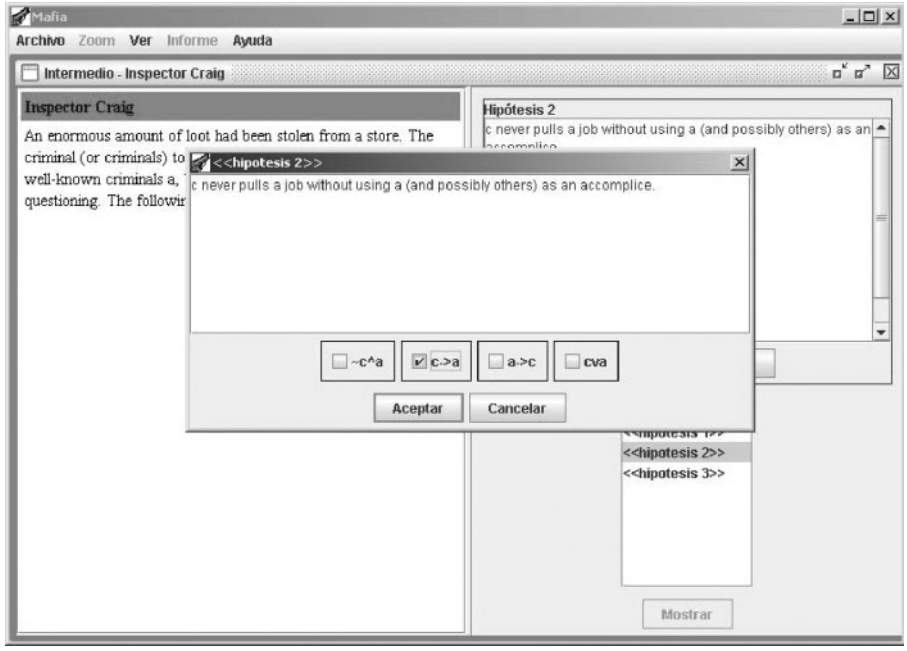

FIG. 2. Solving a problem.

Added to the graphical tool, *MAFIA* provides a *library* of amusing *problems*, some crime situations where the Godfather must decide who is going to wear the cement shoes. Problems can be solved at three different difficulty levels. At the *beginner* one, the tool solves everything and the user can only see the steps he or she must follow. At the *intermediate* level, the application provides some clues in case the student does not know how to continue with the problem. At the *expert* level users have to solve the whole problem by themselves. In Fig. 2 we show an example of solving a problem where the correct formulation of a textual hypothesis must be selected. A report with the problem and its solution can also be saved as an *html* file.

Finally, the tool also offers a facility for editing and *creating new problems* easily, which can be used to increase the library and also to evaluate the students. Quite frequently they find an automatic way to solve everything the teacher poses, but do not really understand what these things are used for. If we give them the possibility of posing some practical problems, they will probably have to look for more information and in the end they will understand the concepts correctly (see Fig. 3).

5 Example 2: Modelos de Kripke

The application entitled *Modelos de Kripke* was conceived as a didactic tool which could help the students to understand Kripke models, the current way to represent the semantics of modal logic [7], [8].

In particular, this tool allows the representation of elemental Kripke models as well as the verification of some of the properties of their accessibility relation. The main goal is to relate

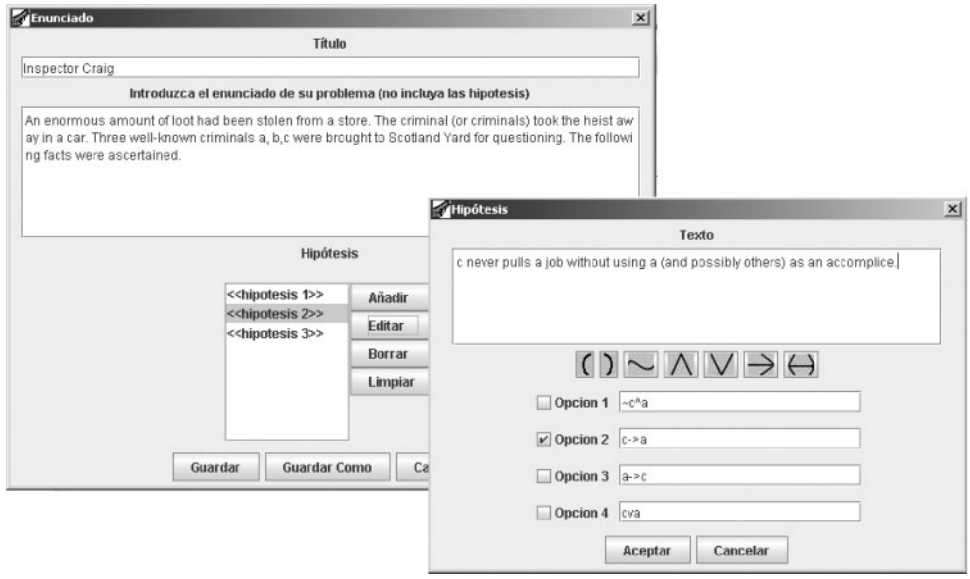

FIG. 3. Creating a new problem.

Kripke models, propositional modal formulas and certain binary relation properties. It is a friendly tool, because it is very visual and easy to use.

It consists of two well specified parts. The first one allows the user to define a model, check the truth of arbitrary propositional modal formulas in every world of the model and check the accomplishment of a fixed set of binary relation properties. The second one is a model generator device that permits the user to generate Kripke models fulfilling a combination of characteristics.

5.1 Propositional modal logics

Modal logics are usually presented as logics of qualified truth; i.e., where sentences gain a new self-reflexive dimension on top of their descriptive and denotative ones. Useful adjectives are: Necessary, obligatory, always, from now on, after an action, known, believed, etc. The concepts of necessity and possibility have been studied by philosophers since Aristotle, but the trend we want to incorporate is certainly more recent. In the 1960s, Kanger, Hintikka and Kripke conceived the relational structures we use today. With this semantics, it is simply natural to focus on the study of the accessibility relation of the model. Reflexive, symmetric, transitive, serial or Euclidean relations are not only properties of relations that naturally arise in mathematics and computer science, but the accessibility relations of the standard models of common logical systems such as $S4$. For instance, a determination theorem is typically formulated as follows: A modal formula is a theorem of $S4$ if and only if it is true in all reflexive and transitive Kripke models.

There are many modal logics, but the best known and studied are the axiomatic normal modal logics obtained when adding to the minimal logic K any combination of the axiom scheme $D := \Box\varphi \rightarrow \Diamond\varphi$, $T := \Box\varphi \rightarrow \varphi$, $B := \varphi \rightarrow \Box\Diamond\varphi$, $4 := \Box\varphi \rightarrow \Box\Box\varphi$,

and $5 := \Diamond\varphi \rightarrow \Box\Diamond\varphi$. The following basic theorem goes a long way towards explaining the great success that relational semantics achieved from the beginning.

1. D is valid in the class of models where R is serial;
2. T is valid in the class of models where R is reflexive;
3. B is valid when R is symmetric;
4. 4 is valid for transitive R;
5. 5 is valid for Euclidean R.

The mathematical proof of this theorem is very easy but we can use the tool *Modelos de Kripke* to visualize its content as well as to find counter-examples to show that the relationship between validity of the formula and the property of the relation goes just in one direction; that is, there are models of T with non-reflexive relations, models of B which are non-symmetric, etc. Therefore, it helps students to understand why the semantics of frames is needed. The software application is also very useful when proving that a modal logic is strictly contained in another and to find out the different modalities of a given modal logic.

5.2 Definition of models

This part of the application allows the user to define a Kripke model by specifying the universe of the structure, its accessibility relation and the interpretation of the propositional letters; namely, the number of points (from 2 to 6), the number of letters (1 or 2), the relationship between the points and the letters' value at each point. Once the model has been defined, the user will be able to check some properties of the accessibility relation, to evaluate several axioms for each letter and to evaluate any modal formula introduced by the user at each point of the defined model.

In Fig. 4, we can see how the application looks when we are defining a Kripke model.

In this example a model W with four points (also called worlds or states) and two letters (p and q) has been selected in the lower right window. In the upper right one the user has defined a relationship R between the points and also the points where the variables p and q are true (called p^A and q^A).

$W = \{1, 2, 3, 4\}$ is the universe of the structure, a set of points
$R = \{(1, 1), (1, 2), (2, 1), (2, 2), (3, 3), (3, 4), (4, 3)\}$ is the accessibility relationship, pairs of related points
$p^A = \{1, 4\}$ are the points where p is true
$q^A = \{2, 3\}$ are the points where q is true.

Once the model is defined its graph will be seen in the drawing window and, according to the values and relations defined by the user, the truth values of certain simple modal formulas ($\Diamond\rho$, $\Box\rho$, $\Box\Diamond\rho$, $\Box\Box\rho$) will appear at the points of each model. The properties of the accessibility relationship (in this case, serial and symmetric) and some axioms which are fulfilled will be shown in the "*Propiedades*" table (Fig. 5). Choosing the other tab of this table it is also possible to write another propositional modal formula and to check if the formula is true or false at each point of the model (Fig. 6).

5.3 Generation of models

The generator allows the production of every possible model that fulfils a set of conditions established by the user. The first condition is the number of points and letters in the model.

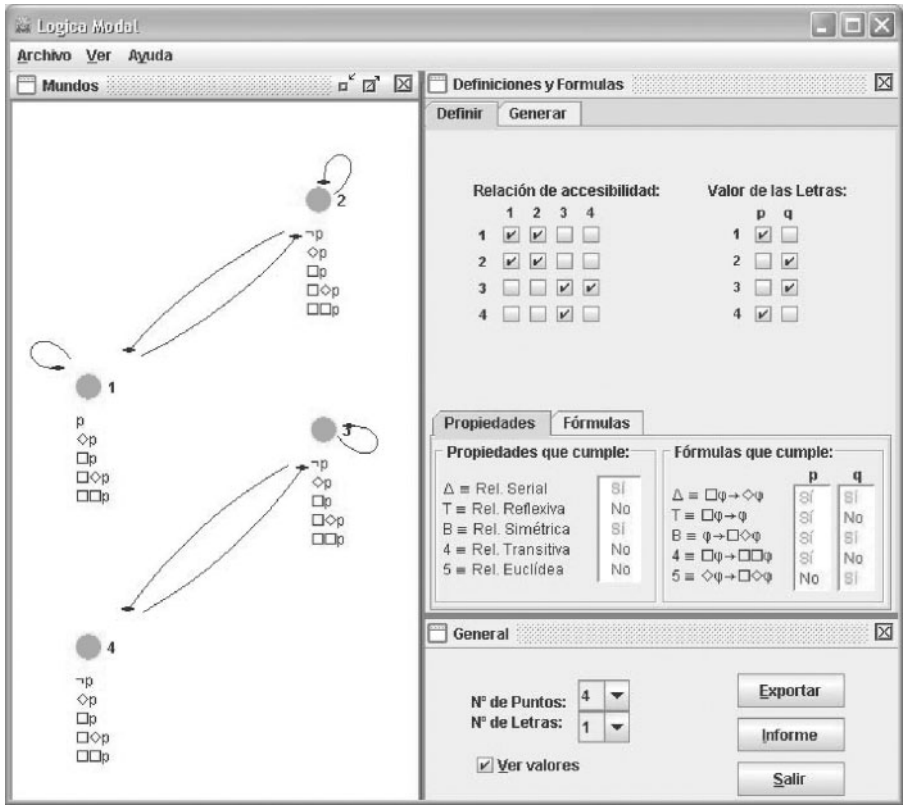

FIG. 4. Defining a model.

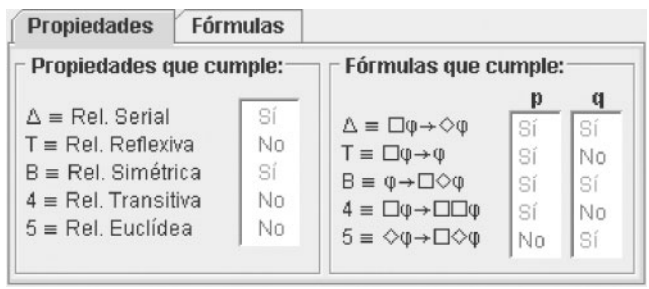

FIG. 5. Properties of a defined model.

It is possible to generate the models determining the fulfilment/non-fulfilment/indifference of a series of properties or to specify a modal formula and to require all the models to accomplish it.

Once the properties for the models and their worlds have been defined, the application generates every possible model that fulfils these requirements. Then, all of them will be listed, and finally, as the user selects each of them, they will appear displayed in the corresponding drawing window.

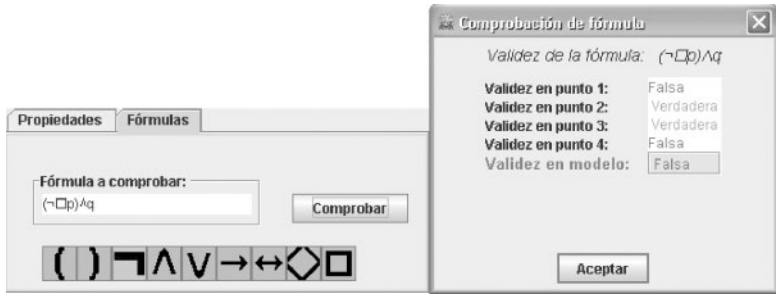

FIG. 6. Checking formulas.

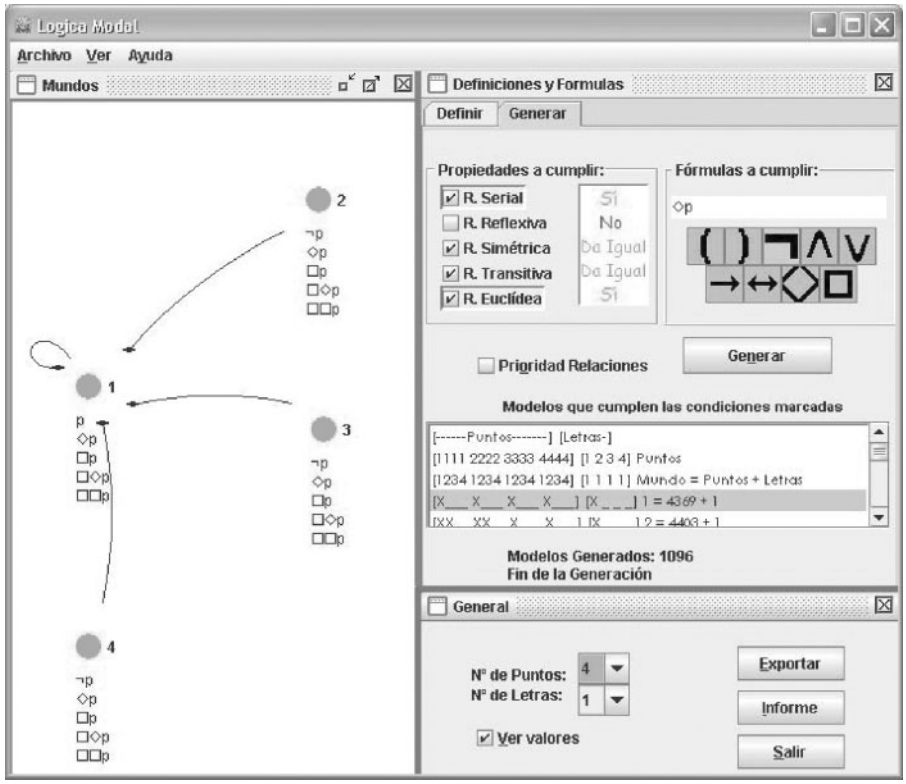

FIG. 7. Generating models.

We can see in Fig. 7 how the application has generated every model with the serial and the Euclidean properties and without the reflexive one (we are not concerned about the symmetric and transitive properties). In all the generated models, the formula $\Diamond \rho$ is true.

Both in the definition and generation of models, the program allows the user to save, at every moment, the model which is displayed in the drawing window in a *jpg* image format. Moreover, if the user wishes, he will also be able to generate an *html* report containing all the information, characteristics and properties of the model he is working with at the moment.

Besides, the application also includes a *Help Guide* that will be accessible for the user every time he needs to settle a doubt while using the program.

6 Conclusions

Implementation of the tools described has already begun during the current academic year 2006-07 with students from Computer Science and Philosophy. The evaluation phase will be started shortly; this phase requires a quantitative study of its indexes of use and of the benefits it provides, together with a review of the quality indicators established in the overall evaluation processes. At present we can affirm that the initiative has been received with great interest and diverse positive aspects can be highlighted.

First, it has been noted that the posing of this type of application as a final paper for computer science students is a very enriching experience at all levels. The students who develop the tools play a truly active role in the learning process and, as is defended in the so-called ''*innovative educational thought*'', they become collaborators in the development of their own courses through critical reflection [3]. They obtain many advantages: a deeper knowledge of the subject, the practical application of their computer knowledge and the possibility of developing tools that will be used by their fellow students. We stress that the way they manage to focus the tool is highly appropriate, because they are well aware of the needs and doubts the future users face.

From the point of view of the students who use the tools, we would emphasise two relevant aspects. On the one hand, greater motivation for the subject has been detected in computer science courses, since this type of interactive tool brings logic closer to their ''*environment*'', and offers them the possibility of using their computer to see direct, specific and amusing applications of logic, thus allowing them to illustrate the more abstract parts of the subject easily. Moreover, they are good critics of the tools and their suggestions help to detect errors and improve the tools. In philosophy courses, this type of application also helps students to get to know the new technologies.

The *MAFIA* tool is especially attractive for first year students and helps them to understand the basic concepts of logic, such as those of consistency and consequence, in an interactive way. With the tool they can solve the crazy cases in Mafia which their fellow students from previous years proposed.

The *Modelos de Kripke* tool, oriented to a more advanced level, serves for understanding the link between the properties of the accessibility relationship and the modal formulas, which is at the basis of the current developments of modal logic.

Finally, from the teachers' point of view, this type of tool makes their task easier and enlivens the teaching process. Furthermore these tools help them to develop and put into practice the skills associated with the new role that the EHEA is assigning them: the ability to incorporate new technologies, arouse interest, stimulate curiosity and creativity and to provide pupils with appropriate opportunities to take responsibility for their own learning.

Acknowledgement

We have relied on the inestimable direction, supervision and collaboration of María Manzano. Her texts, ideas and teaching material have been considered as a basis for these tools.

Summa Logicae is the result of four research projects: the first one was ''*Tools for Teaching*'', an ALFA project (América Latina Formación Académica) for the innovation and systematization of the teaching process that takes Logic as its main subject matter (ALR/B7011/94-6.0285.2). After that, we expanded and updated it with new projects and contributions from several authors, thanks to the support of ''*Summa Logicae en el Siglo XXI*'' (BFF2000-1273), ''*C@lculus*'' (BFF2003-08998) and ''*Las lógicas de la red*'' (HUM2006-12848-C02-01) financed by the Spanish MCYT (Ministry of Science and Technology) and EU's FEDER funds.

References

[1] ASL Committee (1995). Guidelines for Logic Education. The Bulletin of Symbolic Logic, Vol. 1, n. 1, March 1995. *http://www.ucalgary.ca/phil/asl-cle/guidelines.ps*.
[2] Fitting, M. (1996). First-order logic and automated theorem proving (2^{nd} ed.). Springer-Verlag, New York, Inc.
[3] Gavari, E. (2006). Los principios rectores del Espacio Europeo de Educación Superior Virtual. Revista Electrónica de Teoría de la Educación: Educación y Cultura en la Sociedad de la Información Vol. 7, n. 2. Universidad de Salamanca.
[4] Goldblatt, R. (1992). Logics of Time and Computation (2^{nd} ed.). CSLI Lecture Notes, n. 7. CSLI Publications, Stanford.
[5] HM Inspectorate of Education (2004). Using ICT in learning and teaching. Astron BXXXXX 09/04. *http://www.hmie.gov.uk/documents/publication/hgiosict.pdf*
[6] Manzano, M. y A. Huertas (2004). Lógica para principiantes. Alianza Editorial.
[7] Manzano, M. (2006). Lógica Modal. Summa Logicae, *http://logicae.usal.es*
[8] Popkorn, S. (1994). First Steps in Modal Logic. Cambridge University Press.
[9] Smullyan, R. M. (1995). First-Order Logic. Courier Dover Publications.

Received 1 June 2007