Teaching Computational Logic: Technology-enhanced Learning and Animations

Luboš Popelínský, Eva Mráková, Marek Stehlík, and Zdeněk Zavrtal

Faculty of Informatics, Masaryk University,
Brno, Czech Republic

Abstract. In this article we present methods and tools used for computational logic teaching at the Faculty of Informatics, Masaryk University. University Information System offers a comprehensive learning management system. Teaching materials developed using this system are described. Special attention is paid to electronic testing: creation of electronic tests and their usage for training and testing students' knowledge. A data mining tool used for their analysis is also introduced. We then describe both non-interactive and interactive animations used in computational logic courses.

1 Introduction

Teaching computational logic at the Faculty of Informatics involves especially two courses, bachelor course *Introduction to Logic and Logic Programming* and master course *Computational logic.*

The goal of the bachelor course is an introduction to propositional logic (including non-classical logics) and first-order logic as well as to computational logic, namely to resolution principle, logic programming and to deductive and inductive inference. The course is annually attended by approximately 500 students and is taught via lectures and demonstration exercises. To finish the course succesfully, a student has to pass two questionnaires, a midterm written exam and a final written exam.

The master course concerns about advanced parts of computational logic. It is focused on automatic provers for propositional and predicate logic (tableaux provers and resolution), methods for inductive inference in first-order predicate logic and inductive logic programming and on tableaux provers for non-classical logics (modal and intuitionistic). The course also comprises introduction to description logic and to tableaux proves in this logic. The course is annually attended by approximately 80 students and is taught via lectures and demonstration exercises. To succeed, a student has, similarly as for the introductory course, to pass questionnaires, midterm and final exams.

Technology-enhanced learning (TEL) comprises tools and methods for computer-assisted tutoring. In the process of building the courses as well as in tutoring we used various tools of TEL. In this paper we focus on two, most important: to animations, both non-interactive and interactive, and creation and evaluation of online tests. We also describe analytic tool for mining in this kind of educational data. Animations that are used in the lectures are also freely available to students and thus extend the space for excercises, which is, in case of the bachelor course, rather small. Online tests are used both for practising knowldge (students can work with them repeatedly but the results do not influence the final grade) but also for evaluation of students. DZEMUj, a data mining tool for analysing results of online tests serve teachers to find easy/difficult questions as well as clusters and relations among questions. All these tools seem to be necessary because of huge number of students. Usage of TEL enables to increase students' computational logic competence.

Learning management system (LMS) which is a part of Masaryk University Information System (IS MU) offers assistance for creation, evaluation and analysis of online tests. The team of LMS IS MU is also helpful in more specific tasks of TEL, in our case in implementing animations.

For the purpose of comparison of our work with other projects, the Organon [1] LMS should be mentioned. It aims to handle logic exercises as well as to manage the administration of teaching. In comparison with our system, Organon is focused only on basic logic courses, it is a stand-alone system and we have larger set of animations.

To compare our collection of animations to those available on the web, e.g. [2, 3], our collection is topic-specific, compact, searching and management are easy and all animations have similar design.

In Section 2 we introduce tools for technology-enhanced learning that are used at Masaryk University Brno. Then we show how some of those tools – especially questionnaires – have been used in both courses. Section 3 describe tools for analysis of questionnaire results. Section 4 focus on animations that have been implemented. We conclude with discussion of the tools employed for teaching logic and with discussion of future work.

2 Technology-enhanced Learning and Logic Teaching

2.1 Technology-enhanced Learning at MU

Technology-enhanced learning at Masaryk University is supported by University Information System (IS MU). It provides comprehensive Learning Management System [4, 5] which includes

- various discussion forums including those for every course;
- interactive curricula for the courses;
- management tools for homeworks and students' projects;
- management tools for questionnaires (online tests).

Discussion forum of a course is mainly used for a discussion among its students. However, teachers can also take part in these discussions. Interactive curricula help teachers to create courses' curricula and keep them up-to-date without a necessity to do it separately (e.g. in HTML). Homework management tools enable collection of electronic homeworks via web. It also offers some supplementary functions like plagiary detection. Electronic tests (questionnaires) may contain questions of all types common in LMS and it is possible to use them also for writting exams.

2.2 TEL and Computational Logic

In both computational logic courses we take advantage of many features provided by the LMS in IS MU. We focus on the questionnaire management tool and describe (in more details) how it is used for training and testing students' knowledge of computational logic.

The questionnaire tool in the LMS is used for creating three kinds of tests: online mock tests, online tests, and also written tests (on printed papers). For all of these tests it is necessary to create sets of questions first. Division of questions into the sets is quite important. The LMS generates tests according to parameters related to selected sets of questions and thus the tests are well-balanced only for reasonably created sets.

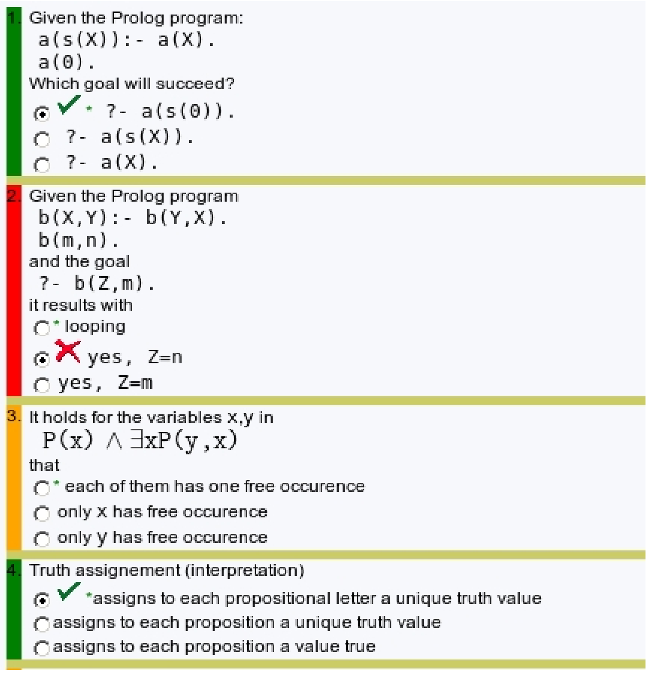
For both computational logic courses we preferred to create separate sets of questions for online mock tests and for online tests (although it is not necessary). The particular set of questions was created according to individual topics and with regard to balance of the generated tests. The sets of questions for the course Computational Logic are the following:

- Resolution
- Tableaux proofs
- Inductive inference
- Modal logic: theory
- Modal logic: tableaux
- DCG: theory
- DCG: examples
- Description logic

Having at least one set of questions, it is possible to create and configure a questionnaire. Questionnaires can be reconfigured easily. Parameters of a questionnaire are the following:

- questionnaire's name;
- list of sets of questions used;
- number of questions generated from each set;
- ordering of questions and ordering of offered answers in tests;
- questionnaire's type (e.g. online mock test, online test, written test) and further parameters related to each type.

Each student has a unique online test generated by the LMS according to these parameters. Online tests are available via authorised IS MU web pages during a specified time period. For both online mock tests and online tests it is possible to set a total time available for solving and the correct answer labelling in the finished tests. Figure 1 shows a part of an online test that had been filled and evaluated. If required, grades of the filled online tests are automatically transferred into a Teacher's Notebook which is a part of IS MU.



1. Given the Prolog program:

```
a(s(X)):- a(X).
a(0).
```

Which goal will succeed?

● ✓ * `?- a(s(0)).`
○ `?- a(s(X)).`
○ `?- a(X).`

2. Given the Prolog program

```
b(X,Y):- b(Y,X).
b(m,n).
```

and the goal

```
?- b(Z,m).
```

it results with

○ * looping
● ✗ `yes, Z=n`
○ `yes, Z=m`

3. It holds for the variables x,y in

$$P(x) \wedge \exists x P(y,x)$$

that

○ * each of them has one free occurence
○ only x has free occurence
○ only y has free occurence

4. Truth assignement (interpretation)

● ✓ *assigns to each propositional letter a unique truth value
○ assigns to each proposition a unique truth value
○ assigns to each proposition a value true

Fig. 1. A Part of Evaluated Online Test.

Written tests are part of both midterm and final exams. Preparation of these exam tests is similar to creation of the online tests but the tests are not finally solved via web interface. Instead, written tests are printed according to layout parameters set in the appropriate questionnaire.

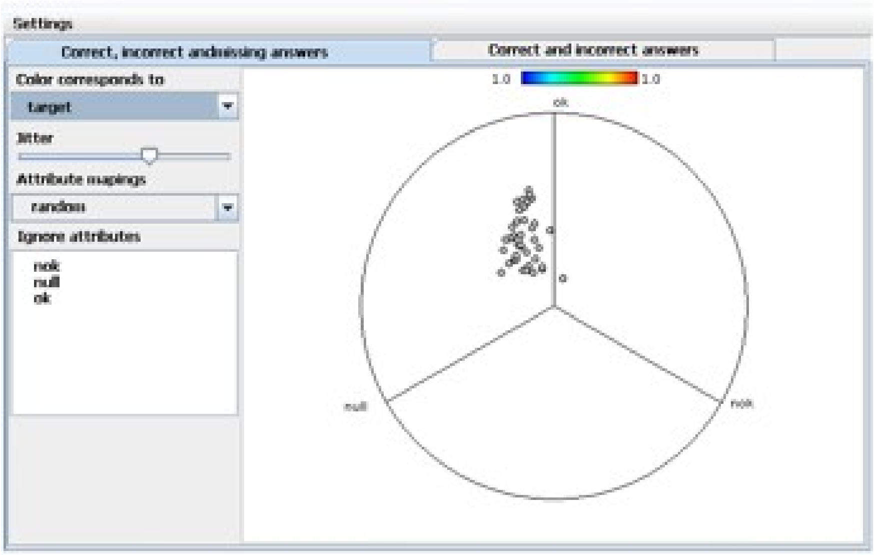
Each student gets a unique printed test and a special answer sheet. A student fills in their answer sheet with their student's ID, the ID of their unique test and marks their answers. Answer sheets are then scanned and automatically recognized. The data obtained are transferred back into the questionnaire tool and possibly into the Teacher's Notebook.

Finally, the same information for both written tests and online tests is stored in the LMS. Tools available in the system enable to inspect and analyze the individual tests as well as the individual questions or the overall results of any exam.

3 Analysing Questionnaires by Means of Data Mining

DZEMUj [6, 7], a tool for data mining in results of on-line tests, has been developed in Knowledge Discovery Lab FI MU and is seemlesly connected with the LMS IS MU. It offers a set of machine learning tools, namely decision tree classifier, cluster analysis and association rule miner, and also visualization tools for this data analysis. DZEMUj thus enables to find the most difficult and/or the easiest test queries, to discover associations between the answers and also between the behaviour of the students.

For visualization of clusters (see Fig. 2) we use RadViz [8].


Settings
Correct, incorrect andmissing answers
Correct and incorrect answers
Jitter
1.0
1.0
ok
null
nok

Color corresponds to: target

Attribute mapings: random

Ignore attributes:
nok
null
ok

Fig. 2. RadViz for Cluster Visualization.

It allows to display n-dimensional data in two dimensions. Attributes – the number of correct, incorrect and missing answers – are put as anchors on a circle. A distance of a point (question) inside the circle from an anchor corresponds to a value of the attribute. It is also possible by clicking to a point to display the particular question. DZEMUj has been tested on data from the LMS IS MU since January 2008 and nowadays is running. DZEMUj is quite efficient: e.g. for 507 students of the bachelor course on logic, 80 different questions and 10 questions in a student test, the analysis finished in less then 1 minute. DZEMUj is implemeted in Python, Java and R (`www.r-project.org`) and is independent on LMS used.

4 Animations

Animations of various proofs, constructions and other techniques are useful supplementary tutoring tools. The topic presented via animations is easier to understand, the lectures are more attractive and freely available animations make the learning easier for students. We decided to create a collection of animations suitable for computational logic teaching. These animations were inserted into the Study Materials section of the LMS IS MU to be available for all students of the computational logic courses and are also available on `http://www.fi.muni.cz/kd/projects/animationsForLogic`.

4.1 Non-interactive animations

The first part of our collection is composed of animations which are not interactive [9, 10]. They are based on carefully selected demonstrative examples and their step-by-step solutions. The series of steps are fixed and a user can only go forward and backward in the proof but cannot change the process of the animation. Every step is completed with an explanation and the relevant part of the construction is highlighted. The collection comprises animations for the following topics:

- interpretations, truth tables;
- normal forms (CNF, DNF);
- prenex NF, Skolemization, unification;
- resolution and its refinements;
- SLD trees;
- hypotheses space and version space;
- learning decision trees;
- propositional, predicate and modal tableaux.

Figure 3 shows one animation stage of the version-space construction. The version space is constructed for the learning set containing one negative <`sm,blue,circ`> and one positive example <`big,red,squr`>. Currently the positive example is processed which is indicated by the highlighting in the figure.

4.2 Interactive Animations

Interactive animations are more suitable for students' exercises and experiments. Users have the possibility to enter their own examples (i.e. formulas to be proved) and are able to control the process of an animation, e.g. select the steps of a proof.

Figure 4 shows a stage of a tableau-proof interactive animation and we shortly describe the tool in the following.

The application enables construction of analytical tableaux of logical formulas. A signed logical formula is entered in the input field of the application.

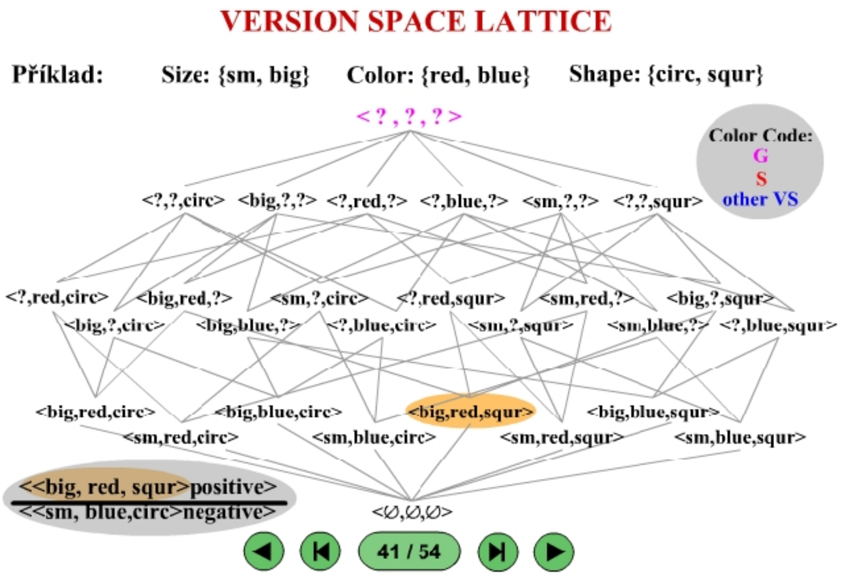


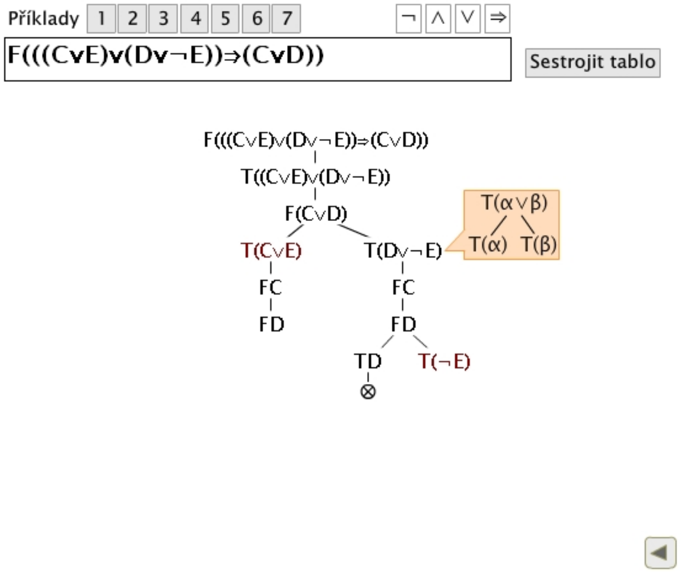
Fig. 3. Version Space Construction.

Typing of the logical connectives is supported by a virtual keyboard. The application includes seven model examples. The construction of a tableau starts by clicking at the *Sestrojit tablo* button.

At the beginning, only the tableau root is displayed. After clicking on a reducible node – which is highlighted – the proof is extended with the expansion of this node according to the appropriate elementary tableau. This elementary tableau is also displayed and disappears after a while. Node reduction is not strictly systematic, the children nodes can be reduced before their parent nodes. All sequential reduction steps are saved in a history file, so it is possible to go back to previous states or to go forward. All tableau states remain in the history and stay interactive. It is possible, after returning to a previous state, to choose another way of the proof. The history adapts to these changes and saves new sequences of states.

During a reduction step every branch of the tableau is evaluated. If a branch contains a contradiction (Tp and Fp nodes), it is contradictory and the appropriate symbol is displayed at the end of this branch. Whenever all nodes have been reduced, the tableau proof is finished and the root formula is classified as a contradiction, a tautology or as satisfiable.

Our interactive animations are prepared in cooperation with the team of the LMS IS MU. They were implemented in Adobe Flash using ActionScript 2.0 language. Currently, there are interactive animations for tableaux proofs of propositional formulas available. In the future we suppose to further extend this collection.



F(((C∨E)∨(D∨¬E))⇒(C∨D))

Fig. 4. A Stage of a Tableau Proof Interactive Animation.

5 Conclusions

In this paper we have discussed computational logic teaching at the Faculty of Informatics, Masaryk University. Technology-advanced learning is greatly exploited especially thanks to the considerable support of the LMS in the University Information System. We have focused on the improvement of training and testing the students' computational logic knowledge using a part of the LMS – the questionnnaire management tool.

Animations of proofs and other techniques offer the other improvement of computational logic teaching. We have introduced our sets of non-interactive animations and we have presented the currently developed interactive animation tool for tableaux proofs.

In the near future, we would like to prepare more online mock tests, one for each topic presented in lectures. We also plan to extend our set of animations. Current candidate topics for animations are tableaux proofs in descripion logic and checking consistency of an ontology in descripion logic.

Acknowledgements

This work has been partly supported by Faculty of Informatics, Masaryk University Brno and Czech Ministry of Education under the project FRVŠ 2363/2010 Innovation of the course Computational Logic.

References

1. Dostálová, L., Lang, J.: ORGANON – The Web Tutor for Basic Logic Courses. Logic Journal of IGPL (2007)
2. Center for Open and Sustainable Learning: Logic Tutorial and Animations (NEEDS). `http://www.folksemantic.com/resources/44120`
3. The Math Forum, Drexel: MathTools. `http://mathforum.org/mathtools/`
4. Čuhel, M., Obšívač, T., Brandejs, M.: Advantages of Versatile E-learning Tools. In: Proceedings of the IADIS International Conference, CELDA 2009, Rome, IADIS Press (2009) 521–522
5. Brandejs, M., Brandejsová, J., Novotný, G.: E-learning at Masaryk University – Outcomes of Fruitful Effort. In: ICETA 2008 Conference Proceedings, Košice, elfa, s.r.o. (2008) 231–236
6. Popelínský, L., Briatková, M., Kedaj, Z.: DZEMUj: A Tool for Mining in e-Learning Tests. Description and Experience. In: Proceedings of the 7th European Conference on eLearning ECEL, Reading, Academic Conferences Limited (2008) 299–303
7. Kedaj, Z., Nováková, L., Popelínský, L.: Visual Analytics for Educational Data Analysis. In: Proceedings of the 8th European Conference on eLearning ECEL, Bari (2009)
8. Fayyad, U., Grinstein, G.G., Wierse, A.: Information Visualization in Data Mining and Knwledge Discovery. Morgan Kaufmann Publishers (2002)
9. Vaňáček, J.: Animace ve výuce logiky. Faculty of Informatics, Masaryk University, Brno. (2008) (Bachelor's Thesis).
10. Zavrtal, Z.: Animace ve výuce logiky. Faculty of Informatics, Masaryk University, Brno. (2009) (Bachelor's Thesis).