An Intelligent Tutoring System for Teaching FOL Equivalence

Foteini Grivokostopoulou, Isidoros Perikos, Ioannis Hatzilygeroudis

University of Patras, Department of Computer Engineering &Informatics, 26500, Hellas (Greece)

{grivokwst,perikos,ihatz @ceid.upatras.gr}

Abstract. In this paper, we present an intelligent tutoring system developed to assist students in learning logic. The system helps students to learn how to construct equivalent formulas in first order logic (FOL), a basic knowledge representation language. Manipulating logic formulas is a cognitively complex and error prone task for the students to deeply understand. The system assists students to learn to manipulate and create logically equivalent formulas in a step-based process. During the process the system provides guidance and feedback of various types in an intelligent way based on user's behavior. Evaluation of the system has shown quite satisfactory results as far as its usability and learning capabilities are concerned.

Keywords: Intelligent Tutoring System, Teaching Logic, First Order Logic, Logic Equivalence

1 Introduction

The advent of the Web has changed the way that educational material and learning procedures are delivered to the students. It provides a new platform that connects students with educational resources which is growing rapidly worldwide giving new possibilities to students and tutors and offering better, cheaper and more efficient and intensive learning processes. ITSs constitute a popular type of educational systems and are becoming a fundamental mean of education delivery. Their main characteristic is that they provide instructions and feedback tailored to the learners and perform their tasks mainly based on Artificial Intelligence methods. The teacher's role is also changing and is moving from the face-to-face knowledge transmission agent to the specialist who designs the course and guides and supervises the student's learning process [10]. ITSs have been used with great success in many challenging domains to offer individualized learning to the students and have demonstrated remarkable success in helping students learn challenging content and strategies [18].

Logic is considered to be an important domain for the students to learn, but also a very hard domain to master. Many tutors acknowledge that AI and logic course contains complex topics which are difficult for the students to grasp. Knowledge Representation & Reasoning (KR&R) is a fundamental topic of Logic. A basic KR&R language is First-Order Logic (FOL), the main representative of logic-based representation languages, which is part of almost any introductory AI course and textbook. So, teaching FOL as a KR&R language is a vital aspect. Teaching and learning FOL as KR&R vehicle includes many aspects. During an AI course the student's learn to translate Natural Language (NL) text into FOL, a process also called *formalization*. A NL sentence is converted into a FOL formula, which conveys the sentence's meaning and semantics and can be used in several logic processes, such as inference and equivalency creation. Equivalency is a fundamental topic in logic. It characterizes two or more representations in a language that convey the same meaning and have the same semantics. Manipulating FOL formulas is considered to be a hard, cognitive complex and error prone process for the students to deeply understand and implement. In this paper, we present an intelligent tutoring system developed to assist students in learning logic and more specifically to help students learn how to construct equivalent formulas in FOL. The system provides interactive guidance and various types of feedback to the students.

The rest of the paper is structured as follows. Section 2 presents related work. Section 3 presents the logic equivalences in FOL. Section 4 presents the system architecture and analyzes its functionality. Section 5 presents the logic equivalent learning. More specifically describes the learning scenarios, the student's interaction and the feedback provided by the system. Section 6 presents the evaluation studies conducted and the results gathered in real classroom conditions. Finally, Section 7 concludes our work and provides directions for future work.

2 Related work

There are various systems created for teaching for helping in teaching logic [8] [19]. However, most of them deal with how to construct formal proofs, mainly using natural deduction. Logic Tutor [1] is an intelligent tutoring system (ITS) for learning formal proofs in propositional logic (PL) based on natural deduction. As an intelligent system, it adapts to the needs of the students via keeping user models. In [4], an intelligent tutoring system is developed for teaching how to construct propositional proofs and visualize student proof approaches to help teachers to identify error prone areas of the students. All the above systems, although deal with learning and/or teaching logic, they are not concerned with how to use FOL as a KR&R language.

KRRT [2] is a web–based system the main goal of which is helping students to learn FOL as a KR&R language. The student gives his/her FOL proposal sentence and the system checks its syntax and whether is the correct one. NLtoFOL [7] is a web-based system developed to assist students in learning to convert NL sentences into FOL. The student can select a NL sentence and interactively convert it in a step based approach into the corresponding FOL. In [6], we deal with teaching the FOL to CF (Clause Form) conversion, via a web-based interactive system. It provides a step-by-step guidance and help during that process. Organon [5] is a web-based tutor for basic logic courses and helps the students during practice exercises. All the above systems, although deal with learning (or teaching) logic, they do not deal with logic equivalency and how to assist students to learn how to construct logically equivalent formulas. As far as we are aware of, there is only one system that claims doing the latter. It is called IDEAS [11] and deals with rewriting formulas from propositional logic into disjunctive normal form. A student is called to transform a formula by applying one transformation rule at a time. The system provides feedback to the student. Also, the system provides a tool [12] for proving equivalences between propositional logic formulas. However, it is restricted to propositional logic and does not deal with FOL.

3 Logical Equivalences in FOL

FOL is the most widely used logic-based knowledge representation formalism. Higher order logics are difficult to handle, whereas lower order logics, such as those based on propositional calculus, are expressively poor. FOL is a KR&R language used for representing knowledge in a knowledge base, in the form of logical formulas, which can be used for automatically making inferences. Logical formulas or sentences explicitly represent properties of or relations among entities of the world of a domain. In logic, two logical formulas *p* and *q* are logically equivalent if they have the same logical content. Logical equivalence between *p* and *q* is sometimes expressed as $p \leftrightarrow q$. Logical equivalence definition in FOL is the same as in propositional logic, with the addition of rules for formulas containing quantifiers. Table 1 presents rules of logical equivalence between FOL formulas.

Table 1. Rules of logical Equivalence for FOL

Equivalence	Name
$p \wedge T \leftrightarrow p$, $p \vee F \leftrightarrow p$	Identity Laws
$p \vee T \leftrightarrow T$, $p \wedge F \leftrightarrow F$	Domination Laws
$p \vee p \leftrightarrow p$, $p \wedge p \leftrightarrow p$	Idempotent Laws
$\neg(\neg p) \leftrightarrow p$	Double Negation Law
$p \vee q \leftrightarrow q \vee p$, $p \wedge q \leftrightarrow q \wedge p$	Commutative Laws
$(p \vee q) \vee r \leftrightarrow p \vee (q \vee r)$, $(p \wedge q) \wedge r \leftrightarrow p \wedge (q \wedge r)$	Associative Laws
$(p \Rightarrow q) \leftrightarrow (\neg p \vee q)$	Implication Elimination
$\neg(p \vee q) \leftrightarrow \neg p \wedge \neg q$, $\neg(p \wedge q) \leftrightarrow \neg p \vee \neg q$	De Morgan's Laws
$\forall x\, P(x) \leftrightarrow \neg\exists x\, \neg P(x)$, $\neg\exists x\, P(x) \leftrightarrow \forall x\, \neg P(x)$	De Morgan's FOL
$p \vee (q \wedge r) \leftrightarrow (p \vee q) \wedge (p \vee r)$ $p \wedge (q \vee r) \leftrightarrow (p \wedge q) \vee (p \wedge r)$	Distribution Laws
$\forall x\, (P(x) \wedge Q(x)) \leftrightarrow \forall x\, P(x) \wedge \forall x Q(x)$ $\exists x\, (P(x) \vee Q(x)) \leftrightarrow \exists x\, P(x) \vee \exists x Q(x)$	Distribution Laws FOL

4 System Architecture and Function

The architecture of our system is depicted in Fig.1. It consists of five units: *Domain Model (DM), Student Model (SM), Student Interface (SI)*, *Interface Configuration (IC)* and *Intelligent Data Unit (IDU)*.

Domain Model (DM) contains knowledge related to the subject to be taught as well as the actual teaching material. It focuses on assisting students to learn how to create FOL-equivalent formulas and so syntax of FOL and equivalence rules constrains are stored in the domain model.

Student Model (SM) unit is used to record and store student related information. Also contains the system's beliefs regarding the student's knowledge of the domain and additional information about the user, such as personal information and characteristics. SM enables the system to adapt its behavior and its pedagogical decisions to the individual student who uses it [3]. Also it sketches the cognitive process that happens in the student learning sessions.

Student interface (SI) is the interactive part of the system. Through SI, a student initially subscribes to the system. During subscription, the required personal information, such as name, age, gender, year of study and email are stored. After subscription, the student can anytime access the system. SI is also responsible for configuring the interface to adapt to the needs of the specific session.

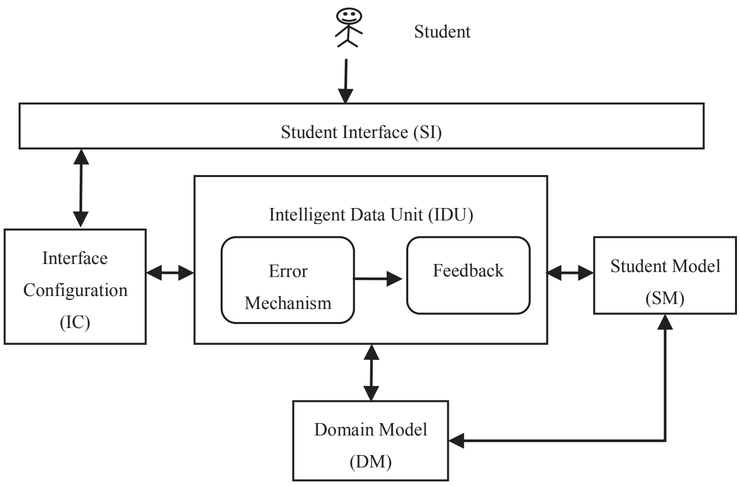

Fig.1. System architecture and its components.

Interface Configuration (IC) unit is responsible for configuring the student interface during the learning sessions, based on the guidelines given by the intelligent data unit. So, the student interface is dynamically re-configured to adapt to the needs of the specific session.

Intelligent Data Unit (IDU) interacts with IC and its main purpose is to provide guidance and feedback to the students and help during application of the logical equivalence rules. It is a rule-based system that based on the input data from user interface decides on which reconfigurations should be made to the user interface or which kind of interaction will be allowed or given to the user. It is also responsible for tracing user's mistakes and handling them in terms of appropriate feedback to the student.

IDU deals with a student's actions for each equivalence exercise as follows:

1. Let the student select an equivalence rule to apply to the FOL formula
2. Check if the selected current equivalence rule can be applied.
   - If it can, allow the student to insert his/her answer to the current rule and go to 2.
   - Otherwise, inform the student that the selected rule is not applicable, provide proper feedback and allow select a new answer.
3. Check the student's answer (formula) to the selected rule
   - If it is correct, inform his/her and go to 1.
   - Otherwise: (a) Determine the error(s) made by the student. (b) Provide feedback based on the error(s) and the corresponding equivalence rule. (c) Allow the student to give a new answer for the selected rule and go to 1.

5 User interaction

The student interface of the system is dynamically reconfigured during a conversion process. After the student enters the system, he/she can select any of the existing FOL formulas/exercises and then starts its conversion into an equivalent formula. This process is made in a step-based approach where the student, at each step, has to select and implement a logic equivalence rule (see above, Table 1). At each step the student can request the system's assistance and feedback (which is based on student's actions and knowledge state). Initially, the student has to select a proper equivalence rule to implement. All the equivalence rules are presented at the working area of student's interface. The student can select a rule and apply it to the formula. If the rule cannot be applied, the system provides proper feedback messages notifying with the reason why it cannot be applied. In contrast, if the rule can be applied, a proper work area is created and the student can manipulate the formula and transform it by applying the selected rule. Then the student can submit the answer (FOL formula). After the student gives an answer, the system informs him/her whether the answer is correct or incorrect. If it is incorrect, the system performs an analysis of the student's answer to find and recognize the errors made by the student. After that the student can submit the new formula derived by the rule application.

As an example, consider the FOL formula "(∀x)~likes(x,snow) ⇒~skier(x)". Initially the student selects to apply the *implication elimination* of equivalency as illustrated in Fig. 2. The system analyses the formula and recognizes that the selected law can be applied. So, proper configurations are made on the interface and the student can insert his/her answer, which is the equivalent formula derived from the application of the rule. After the student submits his/her answer, the system analyzes it and recognizes that the implication is not removed correctly and generates the proper feedback message(s). The feedback messages are linked to the help button and the student can look at them by clicking on it.

5.1 Feedback

The behavior of the system is modeled to consist of two (feedback) loops, the inner and the outer loop respectively [16]. The main role of the inner loop is to provide feedback to the student as a reaction to his/her actions during an exercise, whereas the role of the outer loop is to select the next exercise corresponding to the student's knowledge state. The inner loop of the system is responsible for analyzing the student's answer and provides the proper feedback messages. The feedback provided, in order to enhance its effectiveness, refers to different levels of verification and elaboration. Verification concerns the confirmation whether a student's process is correct or not, while the elaboration can address the answer and related topics, discuss particular error(s) and guide the student towards the correct answer [15].

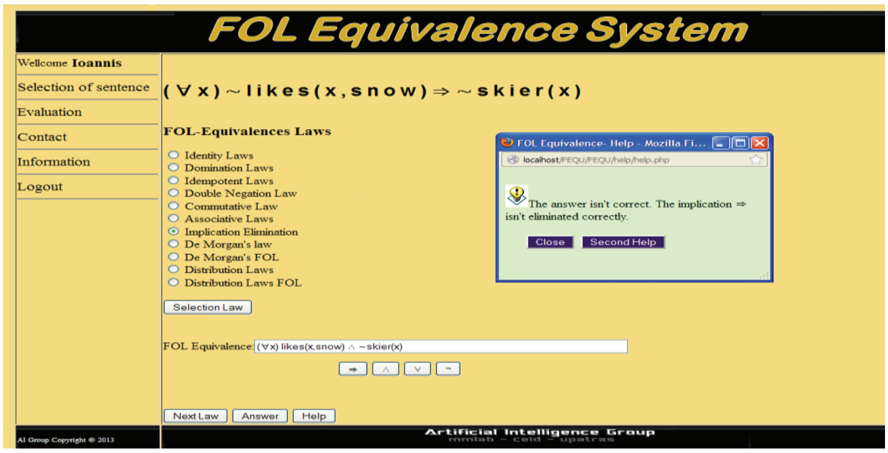

Fig.2. Student Interface

The categories and the types of feedback developed are based on combinations of the classifications of feedback presented in [13] and [16]. So, the main types of feedback offered to the students by the system are the following.

- ***Minimal feedback.*** The system informs the student if the answer is correct or not.
- ***Error-specific feedback.*** When a student's answer is incorrect the system provides the proper feedback based on the errors made, indicating what makes the answer incorrect and the reason why it does it.
- ***Procedural feedback.*** The system can provide a student with hints on what has to do to correct a wrong answer and also what to do next.
- ***Bottom-out hints.*** The system can decide to give the correct answer of a step to the student. This can be done after a student's request or after constantly failure rates and circumstances.
- ***Knowledge on meta-cognition.*** The system analyzes a student's interactions and behavior and can provide meta-cognitive guiding and hints.

The system implements an incremental assistance delivery. Initially, after a student's incorrect action, starts by delivering minimal feedback, just noticing that there are errors and inconsistencies in the student's action. Error-specific feedback is offered after a student's erroneous action. Research has shown that student's motivation for understanding and learning is enhanced when errors are made [9] and the delivery of proper feedback can help the students get a deeper understanding and revise misconceptions. While a student is striving to specify the correct action, the system scales up its assistance till the delivery of the correct action/answer. Providing the correct answer in logic exercises-procedures are consider an important part of the system's assistance. Indeed, student knowledge and performance can be improved significantly after receiving knowledge of correct response feedback, indicating the correct answer [17]. The system never gives unsolicited hints to the student. If the student's answer is incorrect, the proper feedback messages are available (linked) via the help button. So, the student can get those messages on demand, by clicking on the help button. The pedagogical assumption indicates that when the student has the control of the timing of the help provided by the system, there is a greater likelihood that the help messages are received at the right time and therefore be more effective for knowledge construction [14].

6 Evaluation

We conducted an evaluation study of the system during the AI course in the fall semester of the academic year 2011-2012 at our Department. 100 undergraduate students from those enrolled in the course participated in the evaluation study. The students had already attended the lectures covering the relevant logic concepts. The methodology selected to evaluate the system is a pre-test/post-test, experimental/control group one, where the control group used a traditional teaching approach. The students were divided into two groups of 50 students each one, of balanced gender, which were named group A and group B respectively. Group A was selected to act as the experimental group and group B as the control group. Group A (experimental) did some homework through the system, whereas Group B (control) did the homework without using the system and then submit the answers to the tutor and discuss them with him.

Initially, all students took a pre-test on logical equivalence concept. The test included 15 FOL formulas-exercises and the students were asked to provide equivalent FOL formulas. After that, the students of group B were given access to the system and were asked to study for a week aiming at one 20 minutes session per day. After that intervention, the students of both groups took a final post-test including 15 FOL formulas-exercises. The two tests consisted of exercises of similar difficulty level and the score ranged from 0 to 100.

In order to analyze students' performance, an independent *t*-test was used on the pre-test. The mean and standard deviations of the pre-test were 45.18 and 14.73 for the experimental group, and 47.34 and 14.01 for the control group. As the *p*-value (Significant level) was $0.567 > 0.05$ and $t = 0.46$, it can be inferred that those two groups did not significantly differ prior to the experiment. That is, the two groups of students had statistically equivalent abilities before the experiment. In Table 2 and Table 3 the descriptive statistics and the t-test results from assessment of students' learning performance are presented. The results revealed that the mean value of the pre-test of the experimental group is higher than the mean value of the pre-test of the control group. The Levene's test confirmed the equality of variances of the control and experimental groups for pre-test ($F = 0.330$, $p = 0.567$) and post-test ($F = 3.016$, $p = 0.086$). Also the t-test result ($p=0.000 < .05$) shows a significant difference between the two groups. Thus, it implies that the students in the experimental group got a deeper understanding in manipulating FOL formulas and created correctly equivalent formulas for more FOL formulas exercises than the control group.

Table 2. Descriptive Statistics of Pre-test and Post-test

	Group	N	Mean	SD	SE
Pre-Test	Group A	50	45.18	14.73	2.08
	Group B	50	47.34	14.01	1.98
Post-Test	Group A	50	51.74	18.17	2.57
	Group B	50	71.56	15.43	2.18

Table 3. t-test results

Equality of variance		**F-test for variance**		**t-Test for mean**			
		F	Sig.	t	df	Sig.(2-tailed)	MD
Pre-Test	Equal	0.33	0.567	-0.751	98	0.454	-2.16
	Unequal			-0.751	97.756	0.454	-2.16
Post-Test	Equal	3.016	0.086	-5.879	98	0.000	-19.8
	Unequal			-5.879	95.49	0.000	-19.8

In the second part of the evaluation study, the students of group B, who had used the system, were asked to fill in a questionnaire. The questionnaire was made to provide both qualitative and quantitative data. It included questions for evaluating the usability of the system, asking for the students' experience and their opinions about the impact of system in learning and understanding logical equivalence. The questionnaire consisted of nine questions and the results are presented in Table 4. Questions Q1-Q6 were based on a Likert scale (1: not at all, 5: very much). Questions 7-8 were open type questions and concerned strong and weak points of the system or problems faced and also improvements that can be made to the system. Finally, question 9 was about spent time to cope with the system and had three possible answers: less than 15 min, 15-30 min and more than 30 min. Their answers show that 72% of the students needed less than fifteen minutes and only 12% of them needed more than 30 min.

Table 4.Questionnaire Results.

		ANSWERS (%)				
Q	QUESTIONS	1	2	3	4	5
1	How you rate your overall experience?	0	0	20	28	52

2	How much the system did assisted you to learn logical equivalence?	0	0	18	32	50
3	How helpful was the feedback provided?	0	4	12	36	48
4	Did you find the interface of the system helpful?	0	0	28	36	36
5	When stuck, did the system provide enough help so that you could fix the problem(s)	0	2	14	34	50
6	Do you feel more confident in dealing with logical equivalence transformations?	0	4	16	38	42

The students' answers to Q1-Q6 indicate that the majority of the them enjoyed interacting with the system and 82% of them believe that the system helped them in learning FOL equivalences. Also, 84% of them found the feedback provided by the system very useful and that assisted them in manipulating FOL formulas and creating equivalent ones.

7 Conclusions and Future Work

Logic is acknowledged by tutors to be a hard domain for students to grasp and deeply understand. It contains complex cognitive processes and students face many difficulties to understand and correctly implement them. Manipulating FOL formulas and transforming them into equivalent forms is a fundamental topic in logic, but also hard and error prone for students.

In this paper, we introduce an intelligent tutoring system developed to help students in learning how to deal with FOL equivalent formulas. It provides the student an interactive way to manipulate FOL formulas and transform them into equivalent form(s) by applying equivalence rules (or chain of rules) or proper combinations of them. The student, at each stage of the transformation, gets proper guidance and feedback by the system on his/her actions. Regarding the usefulness of the system, the reactions of the students were very encouraging. An evaluation study was conducted to test the system impact on student’s learning. The results revealed that the experimental group outperformed the control group significantly on the post-test exercises. According to the results, the students of the experimental group got a deeper understanding of the logical transformations and significantly enhanced their knowledge. Moreover, the system helped the students to improve their logic conceptual understanding and also to increase their confidence in handling equivalence.

However there are some points that the system could be improved. A direction for future research would be the development of an automatic assessment mechanism to assess the student's performance during the learning interaction with the system. This could help the system better adapt to the student.

Acknowledgements

This work was supported by the Research Committee of the University of Patras, Greece, Program “KARATHEODORIS”, project No C901.

References

1. Abraham, D., Crawford, L., Lesta, L., Merceron, A.,Yacef, K.: The Logic Tutor: A multimedia presentation. Electronic Journal of Computer-Enhanced Learning, (2001)
2. Alonso, J.A., Aranda, G.A., Martin-Matceos, F.J.: KRRT: Knowledge Representation and Reasoning Tutor. In: Moreno Diaz, R., Pichler, F., Quesada Arencibia, A. (eds.) EUROCAST LNCS, vol. 4739, pp.400–407, Springer, Heidelberg (2007)
3. Brusilovsky, P.: Student model centered architecture for intelligent learning environment. In Proc. of Fourth International Conference on User Modeling, Hyannis, MA, pp.31-36 (1994)
4. Croy, M., Barnes, T., Stamper, J.: Towards an Intelligent Tutoring System for propositional proof construction. In: Brey, P., Briggle, A., Waelbers, K. (eds.) European Computing and Philosophy Conference, pp.145–155,Amsterdam (2007)
5. Dostálová, L., Lang.J.: Organon - the web-tutor for basic logic courses Logic Journal of the IGPL 15(4), pp.305-311, (2007)
6. Grivokostopoulou, F., Perikos I., Hatzilygeroudis I.: A Web-based Interactive System for Learning FOL to CF conversion In Proc. of the IADIS International Conference e-Learning 2012, Lisbon, Portugal, pp.287-294, (2012)
7. Hatzilygeroudis, I., Perikos, I.: A web-Based Interactive System for Learning NL to FOL Conversion. New Directions in Intelligent Interactive Multimedia Systems and Services-2 Studies in Computational Intelligence, vol. 226, pp.297-307 (2009)
8. Hendriks, M., Kaliszyk, C., van Raamsdonk, F., and Wiedijk, F.: Teaching logic using a state-of-the-art proof assistant. Acta Didactica Napocensia, 3(2): 35-48, (2010)
9. Hirashima, T., Horiguchi, T., Kashihara, A., Toyoda, J.:Error-Based Simulation for Error-Visualization and Its Management. J. of Artificial Intelligence in Education, vol.9, pp.17-31 (1998).
10. Huertas, A.: Teaching and Learning Logic in a Virtual Learning Environment. Logic Journal of the IGP 15(4), pp.321–331 (2007)
11. Lodder, J., Passier, H., Stuurman, S.:Using IDEAS in teaching logic, lessons learned. Computer Science and Software Engineering, In Proc. of International Conference Computer Science and Software Engineering, vol. 5, pp.553–556 (2008).
12. Lodder, J., Heeren, B.: A teaching tool for proving equivalences between logical formulae. In Third International Congress on Tools for Teaching Logic, pp.66-80, (2011).
13. Narciss, S.: Feedback strategies for interactive learning tasks. In J. M. Spector, M. D. Merrill, J. J. G. Van Merrienboer, M. P. Driscoll (Eds.), Handbook of research on educational communications and technology 3rd ed., pp.125-143,(2008)
14. Renkl, A., Atkinson, R. K., Maier, U. H., Staley, R.:From example study to problem solving: Smooth transitions help learning. J. of Experimental Education,vol.70, pp.293–315, (2002).
15. Shute, V.J. :Focus on formative feedback, Review of Educational Research, Vol. 78, No. 1, pp.153–189 (2008).
16. VanLehn, K.: The behavior of tutoring systems. International Journal of Artificial Intelligence in Education, 16,pp.227-265
17. Wang, S-L., Wu, P-Y.: The role of feedback and self-efficacy on Web-based learning: the social cognitive perspective, Computers & Education vol.51 pp.1589-1598(2008).
18. Woolf, B. P. :Building intelligent interactive tutors: Student-centered strategies for revolutionizing e-learning. Burlington MA: Morgan Kaufman Publishers (2009).
19. Sieg, W., Scheines, R.: Computer Environments for Proof Construction. Interactive Learning Environments 4(2), pp. 159–169 (1994).