Assistant tools for teaching FOL to CF Conversion

Foteini Grivokostopoulou, Isidoros Perikos, Ioannis Hatzilygeroudis
School of Engineering
Department of Computer Engineering & Informatics
University of Patras
26500 Patras, Hellas (Greece)
{grivokwst, perikos, ihatz}@ceid.upatras.gr

Abstract. The FOLtoCF system is an interactive web-based system for learning to convert first order logic (FOL) formulas into Clause Form (CF). FOL to CF conversion is a fundamental part of using FOL for making inferences. In this paper, we present two tutor assistant tools integrated with that system. The first, called tutoring manager, helps the tutor to manage the teaching material and monitor the progress of students. It helps tutors to investigate students' answers and errors made by providing useful statistics. Also, it gives a graphical view of them for an easier understanding of difficulties that students face. The second tool, the difficulty estimating expert system, aims at helping tutors in determining the difficulty level of a formula's conversion process. This is based on the complexity of the FOL formula. Experimental results show that the difficulty estimating system is quite successful.

Keywords: Tutor assistant tool, Student progress statistics, Difficulty level estimation, First Order Logic, Clause Form

1 Introduction

Logic is one of the fundamental topics taught in computer science and/or engineering departments. In most such departments, logic is taught as a means for constructing formal proofs in a natural deduction style. However, teaching logic as a knowledge representation and reasoning (KR&R) vehicle is also basic in all introductory artificial intelligence (AI) courses. A basic KR language is First-Order Logic (FOL), the main representative of logic-based representation languages [3], which is part of almost any introductory AI course. To make automated inferences, Clause Form (CF), a special form of FOL, is used. Students usually find difficulties in converting complex FOL formulas into CF [7].We have constructed tools for helping tutors in teaching logic as a KR&R language and more specifically tools for learning the conversion of NL sentences into FOL ones [12][11].

There are several systems [1][2][8][13] for teaching propositional logic (PL), but most of them teach how to construct formal proofs using natural deduction. Logic-ITA [14] deals with propositional logic in the same sense as above, but it is addressed to both students and teachers and uses intelligent techniques to automatically adapt to their needs. None of the above deals with PL as a KR&R language and cannot determine the difficulty level of a formula's conversion.

In a previous work [6], we dealt with teaching the FOL to CF conversion using a simple tool implemented in Java. However, that system does not offer any tools for helping the tutors.

In this paper, we introduce two assistant tools that aim at assisting tutors in their tasks. The first one helps tutors in managing and updating the teaching material of the system. In addition, it can help them monitor student's progress and trace learning gaps and difficulties the students may face. The second tool is an expert system that aims at helping the tutor in determining the difficulty level of a formula's conversion process. The determination of the difficulty of the exercises material is a vital aspect for an e-learning system. However, in most systems the difficulty levels of the exercises are determined by the tutor, when inserting them into the system. The determination of an exercise difficulty level is a time-consuming task for the tutor and to some degree non-consistent. Therefore, such a system can be of great help.

The paper is organized as follows. In Section 2, related work is presented. Section 3 deals with the FOL to CF conversion, by presenting the process through an example. Section 4 presents the revised architecture of the FOLtoCF system. Section 5 and Section 6 focus on the assistant tools. Section 7 presents and discusses experimental results. Finally, Section 8 concludes and provides directions for future research.

2 Related Work

There are several tools for teaching logic and logic-based reasoning, but most teach natural deduction. A number of them are characterized as logic educational software. ProofWeb [8] is a system for teaching logic as natural deduction, based on the higher order proof assistant Coq [4]. Moreover, some systems provide strategies for proving propositions and use those strategies to provide hints or worked out examples (e.g. Fitch [1], AProS [13] and Pandora [2]).

Logic-ITA [14] is an intelligent teaching assistant system for Logic. It deals with propositional logic and provides students with an environment to practice formal proofs. It expands Logic Tutor (an earlier version) with new tools which are designed to assist the tutor. The teacher can use those tools to manage the teaching configuration settings and the teaching material. Also, they help him/her monitor the students' results and their progress. P-Logic Tutor [10] is a kind of intelligent tutoring system aiming at teaching students fundamental aspects of PL and theorem proving. P-Logic tutor also provides an environment in which the tutor can track student learning activities. However, none of them explicitly deals with the FOL to CF conversion and provides no information about the difficulty of the FOL formula's conversion into CF.

In [11], a work that deals with the difficulty of the conversion of NL into FOL is presented. It is an expert system that automatically determines the difficulty level of a sentence's conversion process. It takes as input the corresponding FOL formula of a NL sentence and gives as output an estimation of the difficulty of its conversion process. The estimation of the difficulty level is based on a set of parameters, like the number and the type of the quantifier(s), the number of the implications and the different connectives of the FOL expression.

However, according to our knowledge, there isn't any effort to determine the difficulty level of a FOL formula's conversion process into CF.

3 FOL to CF Conversion

FOL is the most widely used logic-based knowledge representation formalism. FOL is a KR language used for representing knowledge in a knowledge base, in the form of logical formulas. To make automated inferences using the resolution principle (the strongest inference rule), their Clause Form (CF) is used. The FOL to CF conversion is a well-defined process that can be automated within a computer. However, most of existing systems perform it in one step manner so that, one cannot see all the steps of the conversion. Therefore, they are not suitable for use in teaching the conversion process. The conversion process in our system includes six steps, as presented below through an example: the conversion of FOL sentence

$$\text{“}(\forall x)\ dog(x) \Rightarrow (\exists y)(cat(y) \wedge chases(x,y))\text{”}.$$

Step1: Eliminate connectives $\Leftrightarrow$ and $\Rightarrow$, using the equivalences: $P \Leftrightarrow Q \equiv (P \Rightarrow Q) \wedge (Q \Rightarrow P)$ and $P \Rightarrow Q \equiv \neg P \vee Q$

Result: $(\forall x)\ \neg dog(x) \vee (\exists y)(cat(y) \wedge chases(x,y))$

Step2: Reduce negation scope using the following equivalences:
$\neg(P \vee Q) \equiv \neg P \wedge \neg Q$, $\neg(P \wedge Q) \equiv \neg P \vee \neg Q$, $\neg \forall x P \equiv \exists x \neg P$, $\neg \exists x P \equiv \forall x \neg P$, $\neg\ \neg\ P \equiv P$

Not applicable here

Step3: Rename variables (so that each quantifier has its own unique variable name) and transform into Prenex Normal Form (move all quantifiers to the left of the formula)

Result: $(\forall x)\ (\exists y)\ \neg dog(x) \vee (cat(y) \wedge chases(x,y))$

Step4: Remove existential (Skolemization) and universal quantifiers. The process is as follows:

(a) if there are universal quantifiers whose scope includes the scope of an existential quantifier replace each occurrence of its variable with a (Skolem) function whose arguments are the variables of those universal quantifiers,

(b) If there are no such universal quantifiers, replace each occurrence of its variable with a (Skolem) constant.

(c) After skolemization, remove all universal quantifiers.

Result: $\neg dog(x) \vee (cat(sk\text{-}f(x)) \wedge chases(x, sk\text{-}f(x)))$

Step5: Transform the produced formula into CNF (Conjunctive Normal Form), using the equivalence:
$(P \wedge Q) \vee R \equiv (P \vee R) \wedge (Q \vee R)$

$(\neg dog(x) \vee cat(sk\text{-}f(x))) \wedge (\neg dog(x) \vee chases(x, sk\text{-}f(x)))$

Step6: Extract clauses and rename variables.

As many clauses as the number of conjunction connectives in the produced formula plus one are extracted.

Result: {¬dog(x), cat(sk-f(x))}
{¬dog(x), chases(x, sk-f(x))}

4 System Architecture

The architecture of the system is depicted in Fig. 1. It consists of four units: the Tutor Interface (TI), the Tutoring Manager (TM), the Difficulty Estimating System (DES) and the System Database (SD). Through TI, the tutor interacts with tutoring manager (TM) unit. TM is responsible for the teaching process and helps the tutor in examining students' progress.TM unit offers to the tutor the capability of exercise management. So, the tutor can create new exercises-sentences and also edit or even delete existing ones. Also, the tutor can monitor the knowledge level of students and the common errors of student's answers. Moreover, the system can present useful statistics about the student learning process.

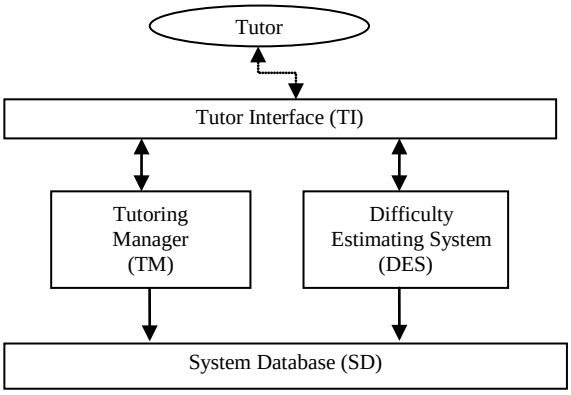

Fig 1. Architecture of the Tutoring Assistant System

The Difficulty Estimating System (DES) is an expert system that aims to help the tutor to determine the difficulty level of FOL to CF conversion. DES automatically determines the difficulty level of a formula without tutor intervention. Tutor can accept or modify the result of DES.

Finally, in System Database (SD) the FOL sentences and their corresponding estimated difficulty levels are stored.

5 Tutoring Manager

The tutoring manager (TM) offers the capability of the tutor to manage the exercise-FOL formula. An exercise consists of the following parts:

- Its name (Natural Language)

- Its body (FOL formula)
- Its difficulty level (1-5)
- Its correct answer (CF formula (s))

The tutor can create a new exercise-FOL formula and also edit or even delete existing ones. To create a new question-answer pair, the tutor has to insert a new FOL formula. The difficulty level is automatically determined. The tutor can modify it. Also, the correct answers are automatically determined and inserted into SD. Also, the tutor can see all the exercises and all information concerning them. So, the tutor can change the exercises' information at any time according to his/her teaching needs.

Another facility that is available to the tutor is monitoring a student's learning process. As mentioned above, while the student is using the FOLtoCF system, all the necessary information is stored in SD. The tutor can monitor the errors of the student made during the FOL to CF process, the exercises that the student has tried, the time spent and the knowledge level obtained. The system can present some statistics about the students, which are the following:

- The percentage of correct answers to exercises per difficulty level per student.
- The number of errors made per sentence.
- The type of errors made per sentence per difficulty level per student.
- The percentage of errors made per difficulty level per student.
- The percentage of correct answers to exercises per difficulty level for all students.

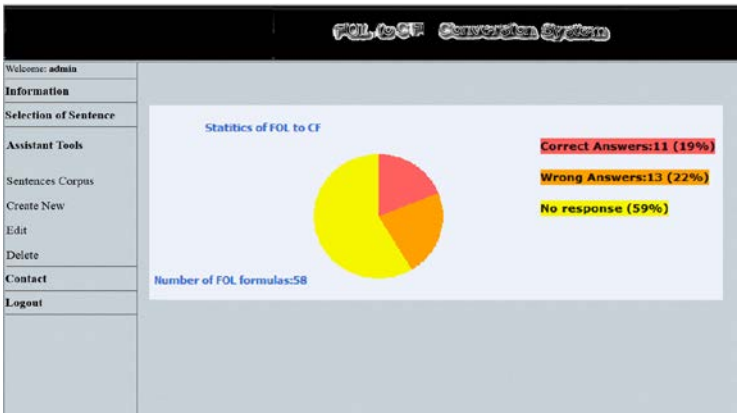

Fig 2. Statistics of Students

In Fig. 2, some statistics about students is illustrated. As mentioned above, the tutor can evaluate not only the students, but also the exercises-formulas and indirectly the teaching process. So, the tutor can check the quality (e.g difficulty level) of each exercise that he/she has designed for the evaluation of students.

6 Difficulty Estimating System

We have developed the Difficulty estimating system (DES), which is an expert system. It aims at helping the tutor in determining the difficulty level of a FOL to CF conversion. DES takes as input a set of values corresponding to difficulty estimation parameters and automatically determines the conversion difficulty of the corresponding FOL formula. Values are extracted from that formula by an analysis process. The structure of DES is illustrated in Fig. 3. DES is a rule-based expert system implemented in Jess, which an expert system shell [5]. It consists of the Difficulty Parameters Fact Base (DFB), the Difficulty Estimation Rule Base (DRB) and the Jess inference engine (JESS IE). DFB contains the values of the estimation parameters and the DRB contains the rules for estimating difficulty levels. DFB and DRB constitute the Knowledge Base (KB) of the system. FA is a tool that analyses a FOL formula and automatically extracts the values of the estimation parameters for a FOL formula, which are stored as facts in DFB and used as input to DES.

The process of estimating the difficulty level of the FOL to CF process of a FOL formula is as follows:

1. The FOL formula is analyzed via the FOL Analyzer tool (FA) to extract values for the difficulty estimation parameters
2. The extracted values are transformed into Jess facts and stored in DFB
3. Jess IE is triggered and deduces the difficulty level of the corresponding FOL to CF conversion process

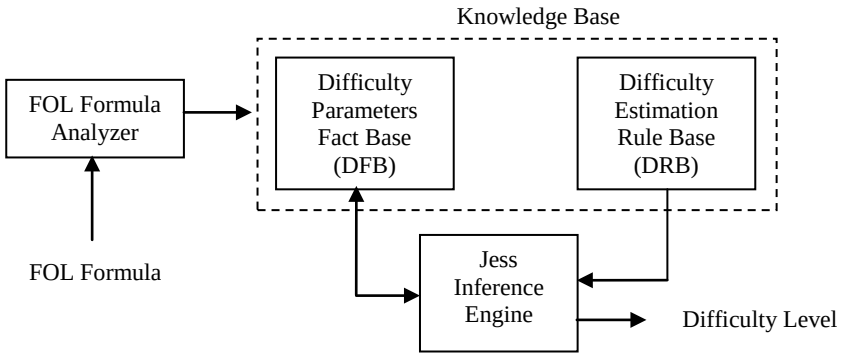

Fig 3. The structure of Difficulty Estimating System

The development of DES is based on expert-tutor. The expert offers knowledge based on experience. Most tutors empirically estimate the difficulty of a FOL sentence's conversion. In co-operation with the expert-tutor, we tried to specify which factors/parameters have an impact on the difficulty level of a sentence's FOL to CF conversion. Finally, we came up with the following difficulty estimation parameters:

- The number, the type, the order and the position of quantifier (s)
- The number of the implication symbols
- The number of the different connectives

- The type of negation(s)
    - Negation in front of a group of atoms
    - Negation in front of a quantifier.
- Whether the LHS of an implication is a group of atoms

where by "connectives" we mean $\{\wedge, \vee, \neg\}$.

Table 1. Rules for determining difficulty levels

Rules	**Num of ∀, ∃**	**∀**	**∃**	**Order ∀∃**	**PNF**	**Num of ⇒**	**Num of different ∧, ∨, ¬**	**¬ in front of group**	**¬ in front of ∀, ∃**	**Group at LHS of ⇒**	**Difficu-lty Level**
1.	0	N	N	*	*	0	≤ 1	*	*	*	VE
2.	1	N	Y	*	*	0	0	*	*	*	VE
3.	1	N	Y	*	*	0	≥ 1	*	*	*	VE
4.	≤ 2	Y	Y	*	Y	0	≥ 1	N	N	*	E
5.	≤ 2	Y	-	*	*	0	≥ 1	N	N	*	E
6.	≤ 2	-	Y	*	*	0	≥ 1	N	N	*	E
7.	1	Y	-	*	*	1	≤ 1	N	N	N	E
8.	0	N	N	*	-	1	≥ 0	*	*	Y	E
9.	1	Y	N	*	*	1	≥ 1	Y	Y	*	E
10.	2	Y	Y	Y	*	0	≥ 1	Y	Y	*	E
11.	2	N	Y	*	Y	1	≥ 1	N	N	*	M
12.	≥2	Y	N	*	Y	1	≥ 1	*	*	*	M
13.	≥2	Y	Y	N	N	1	≥ 1	Y	N	N	M
14.	≥2	Y	Y	Y	Y	1	≥ 1	N	Y	N	M
15.	2	Y	Y	Y	Y	1	≥ 1	N	N	Y	M
16.	2	Y	Y	Y	N	1	≥ 1	N	N	N	M
17.	<3	Y	-	*	Y	1	≥ 1	N	N	Y	M
18.	<3	Y	Y	Y	*	1	≥ 1	Y	N	N	D
19.	<3	Y	Y	Y	*	1	≥ 1	Y	N	N	D
20.	3	Y	Y	Y	N	≥1	≥ 1	N	Y	Y	A
21.	3	Y	Y	N	Y	≥1	≥ 1	Y	Y	Y	A

Afterwards, we consulted the expert-tutor to acquire the necessary rules for the difficulty estimation based on the above parameters. The results of knowledge acquisition are depicted in Table 1. There are 21 rules for determining the difficulty level. The difficulty level is classified in one of the following five classes: (a) very easy (VE), (b) easy (E), (c) medium (M), (d) difficult (D) and (e) advanced (A). Also, in Table 1, PNF stands for "Prenex Normal Form" (all quantifiers are already at the left hand side of the formula), LHS for "Left Hand Side", "Y" for "Yes", "N" for "No" and "*" stands for "don't care".

In Table 2, some example FOL formulas/sentences and corresponding difficulty levels of their conversion processes as produced by DES are presented.

Table 2. Examples of difficulty levels estimated by DES

First Order Logic (FOL)	Clause Form	Difficulty Level	Rule applied
¬happy(john) ∨¬happy(peter)	{¬happy(john),¬happy(peter)}	Very Easy	1
(∃x) apple(x) ∧red(x)	{apple(sc_x)} {red(sc_x)}	Very Easy	3
(∀x) gardener(x) ⇒likes(x,sun)	{¬gardener(x),likes(x,sun)}	Easy	7
(∀x)(∀y)(∀z) (ancestor (x, y) ∧ancestor (y,z) ⇒ancestor(x,z))	{¬ancestor (x, y) , ¬ancestor (y,z),ancestor(x,z)}	Medium	12
(∀x)(student (x) ⇒ (∃y)(student(y) ∧ loves(x,y)))	{¬student(x), student(sk-f(x))} {¬student(x), loves (x, sk-f(x))}	Medium	16

7 Experimental Results

We used DES for a number of 94 FOL formulas, found in textbooks and web resources. DES results were compared to the results of the expert-tutor on the basis of the extracted factual information. To evaluate DES, we used four metrics, commonly used for this purpose: *accuracy*, *precision*, *sensitivity* and *specificity*. The metrics for two output classes are defines as following:

$$Acc=\frac{TP+TN}{TP+FN+TN},\ Prec=\frac{TP}{TP+FP},\ Sen=\frac{TP}{TP+FN},Spec=\frac{TN}{TN+FP}$$

Where *TP*, *TN*, *FP* and *FN* denote the number of the true positives, true negatives, false positive and false negatives, respectively. By "positive", we mean that a case belongs to the class of the corresponding difficulty level and by negative that it doesn't.

In case of multi-class classification, as ours, the above metrics are calculated as follows:

$$acc=\frac{\sum_{i=1}^{m} acc_i}{m},\ prec=\frac{\sum_{i=1}^{m} prec_i}{m},\ Sen=\frac{\sum_{i=1}^{m} sen_i}{m},\ Spec=\frac{\sum_{i=1}^{m} spec_i}{m}$$

The evaluation results are presented in Table 3 and show an acceptable performance of DES.

Table 3. Evaluation metrics for DES

Metrics	**Difficulty Class**					**Average**
	Very Easy	Easy	medium	Difficult	Advanced	
Accuracy	0.9	0.89	0.89	0.89	0.96	0.906
Precision	0.55	0.92	0.83	0.5	0.5	0.66
Sensitivity	1	0.75	0.91	0.2	0.5	0.672
Specificity	0.92	0.99	0.96	0.89	0.97	0.946

The results show a very good performance of DES. From the corpus of 94 FOL formulas that were used the system correctly identified the difficulty of 77 FOL formulas.

8 Conclusion and Future Work

The FOLtoCF system is a web based interactive system for helping students to convert First order logic (FOL) formulas into Clause Form (CF). In this paper, two teaching assistant tools are presented. The first one is the Teaching Manager tool, which is developed to help the tutor to manage and update the teaching material of the system. Moreover, can help him/her monitor student's progress and trace learning gaps and difficulties that the students may face. Also the tool offers sets of graphs and statistics and based on them the tutor can accordingly reconfigure the contents of the system by adding specific sentences.

The second is the Difficulty estimation expert system, which is used to automatically determine a FOL formulas difficulty concerning its conversion into CF. DES is a rule-based expert system that takes as input a FOL formula, analyses it in terms of connectives, quantifiers, negations types and order specification and based on theses parameters determines the FOL formulas conversion difficulty-complexity into CF.

A more interesting direction is to use a method similar to that in [9] could be investigated. This means to use a student-based approach instead of a sentence-based approach or a combination of them.

Acknowledgement

This work was supported by the Research Committee of the University of Patras, Greece, Program "Karatheodoris", project No C901.

References

1. Barwise, J., Etchemendy, J.: Language, Proof and Logic, Center for the Study of Language and Information, (2002)
2. Boda, K., Ma, J., Sinnadurai, G., Summers, A.: Pandora: A reasoning Toolbox using natural Deduction Style. Logic journal of the IGPL, 15(4): pp. 293--304 (2007)
3. Brachman, R.J., Levesque, H.J.: Knowledge Representation and Reasoning. Elsevier, Amsterdam (2004)
4. Coq Development Team. The Coq Proof Assistant User's Guide. Version 8.3 (2010)
5. Friedman-Hill, E.: Jess in Action: Rule-Based Systems in Java, Manning Publications Company, (2003)
6. Hatzilygeroudis, I., Giannoulis, C., Koutsojannis, C.: A Web Based Education System for Predicate Logic. Proceedings of the IEEE International Conference on Advanced Learning Technologies (ICALT '04), pp. 106--110, (2004)
7. Hatzilygeroudis, I.: Teaching NL to FOL and FOL to CL Conversions. In: Proceedings of the 20th International FLAIRS Conference, Key West, FL, pp. 309--314. AAAI Press, Menlo Park (2007)
8. Hendriks, M., Kaliszyk, C., Van Raamsdonk, F., Wiedijk, F.: Teaching logic using a state-of-the-art proof assistant. Acta Didactica Napocensia, 3(2): pp. 35--48, (2010)
9. Koutsojannis, C., Beligiannis, G., Hatzilygeroudis, I., Papavlasopoulos, C., Prentzas, J.: Using a hybrid Al approach for exercise difficulty level adaptation, International Journal of Continuing Engineering Education and Life-Long Learning, 17(4-5), pp. 256--272 (2007)
10. Lukins, S., Levicki, A., Burg, J.: A tutorial program for propositional logic with human/computer interactive learning. In SIGCSE 2002, ACM, New York, pp. 381--385, (2002)
11. Perikos, I., Grivokostopoulou, F., Hatzilygeroudis, I., Kovas, K.: Difficulty Estimator for Translating Natural Language into First Order Logic, Proceedings of the Third International Conference on Intelligent Decision Techologies (KES-IDT 2011), Volume 10, Part I, pp. 135--144 (2011)
12. Perikos, I., Grivokostopoulou, F., Hatzilygeroudis, I.: Teaching assistant tools for NL to FOL Conversion Proceedings of the IADIS International Conference e-Learning 2011, 20 - 23 July, Rome, Italy, pp. 337--345 (2011)
13. Sieg, W.: The AProS project: Strategic thinking & Computational logic. Logic journal of the IGPL 15(4), pp. 359--368 (2007)
14. Yacef, K .: The Logic-ITA in the classroom: a medium scale experiment. International Journal on Artificial Intelligence in Education, pp.41--60 (2005)