Uncertainties in Knowledge Assessment

SYLVIA ENCHEVA
Department Haugesund
Stord/Haugesund University College
Bjornsosnsg. 45, 5528 Haugesund
NORWAY
sbe@hsh.no

SHARIL TUMIN
IT Department
University of Bergen
P.O. Box 7800, 5020 Bergen
NORWAY
edpst@it.uib.no

Abstract: - Various techniques for reasoning in the presence of inconsistent or incomplete information have been exploited lately. In this paper we focus on applying many-valued logic in practical deductive processes evaluating propositions being neither true nor false when they are uttered. Such situations occur while dealing with inconsistent and/or incomplete information. Application of many-valued logics allows the system to handle situations with such input. Many-valued logic is a generalization of Boolean logic and as such offers solution to a number of Boolean problems.

Key-Words: - Web-based assessment, learning, intelligent tutoring systems

1 Introduction

Intelligent tutoring systems apply a variety of methods for automated evaluation of students' knowledge. Most systems are forcing the student to point the correct answer (if the system suggests exactly one correct answer), to find all the correct answer (if the system suggests several correct answers), or to provide a solution (if the system suggests open ended questions). If a question is unanswered than it is treated as a wrongly answered question with respect to both grading and providing further guidance and/or help.

In this paper we discuss automated assessment of students' understanding of terms. Application of a many-valued logic allows the system to handle situations with inconsistent and/or incomplete input. Many-valued logic is a generalization of Boolean logic and as such offers solution to many Boolean problems [1].

The rest of the paper is organized as follows. Related work and statements from many-valued logic may be found in Section 2. The main results of the paper are placed in Section 3. The system architecture is described in Section 4. The paper ends with a conclusion in Section 5.

2 Background

Let P be a non-empty ordered set. If $sup\{x, y\}$ and $inf\{x, y\}$ exist for all $x, y \in P$, then P is called a *lattice* [2]. In a lattice illustrating partial ordering of knowledge values, the logical conjunction is identified with the meet operation and the logical disjunction with the join operation.

A three-valued logic is presented in [11]. Another three-valued logic, known as Kleene's logic is developed in [10] and has three truth values, truth, unknown and false, where unknown indicates a state of partial vagueness (Fig. 1). These truth values represent the states of a world that does not change. Two natural orderings concerning 'amount of knowledge' and 'degree of truth' are suggested, [4]. Thus applying 'knowledge' ordering we place the value unknown below both truth and false, while applying 'degree of truth' ordering results in unknown better than false, and truth better than unknown, i.e. false, unknown, truth.

The five-valued logic in Fig. 2 introduced in [3] is based on the following truth values: *uu* - unknown or undefined, *kk* - possibly known but consistent, *ff* - false, *tt* - true, *ww* - inconsistent.

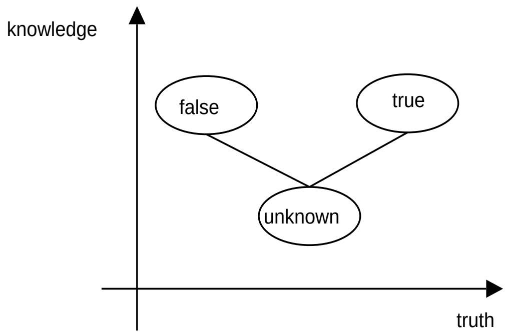

Fig. 1 Kleene's strong 3-valued logic

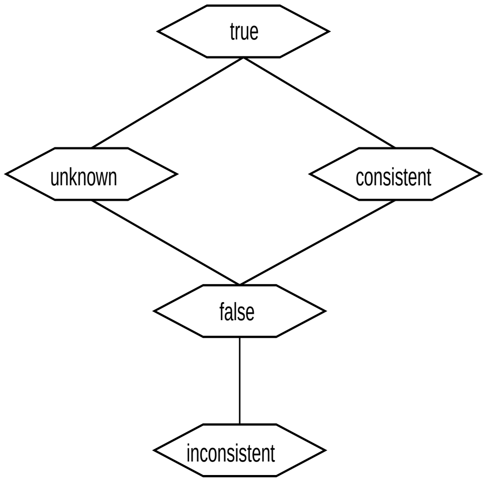

Fig. 2 Five-valued logic

A brief overview of a six-valued logic, which is a generalized Kleene's logic, has been first presented in [12]. The six-valued logic was described in more detail in [8]. In [4] this logic is further developed by assigning probability estimates to formulas instead of non-classical truth values.

The six-valued logic distinguishes two types of unknown knowledge values - permanently or eternally unknown value and a value representing current lack of knowledge about a state [7].

A lattice showing a partial ordering of the truth values *false*, *unknown f*, *unknown*, *unknown t*, *contradiction*, *true* by degree of knowledge is presented in Fig. 3. The knowledge lattice illustrates how the truth value of a formula that has a temporary truth value can be changed as more knowledge becomes available. Suppose a sentence has a truth value *unknown f* at one point of time and *false* at another. Its truth value is then determined as *false*, i.e. the system allows belief revision as long as the revision takes place in an incremental knowledge fashion.

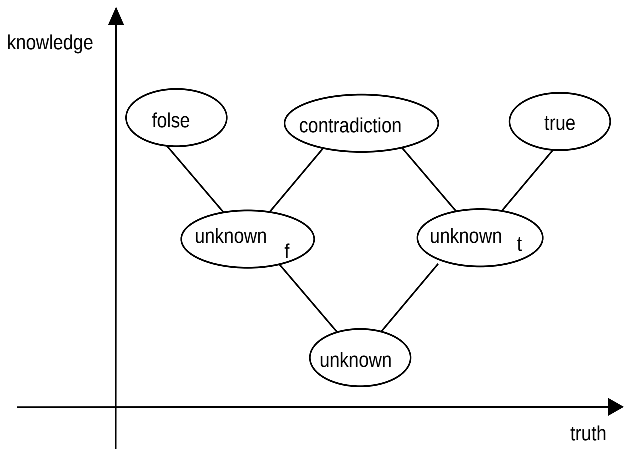

Fig. 3 Knowledge lattice

3 Tests

3.1 Tests with two questions

We consider a multiple-choice test assessing student's understanding of new terms. The test consists of two questions. According to the result of a multiple-choice test, understanding of a term is achieved if a student gives a correct answer to each of the two questions about that term. Such tests are placed after a new term has been introduced in the theoretical part of the system.

Modus ponens (P, P $\rightarrow$Q) $\rightarrow$ Q can be applied if there is no doubt about the truth-status of P. Most tests are based on the understanding that a 'correct answer to one question about a term' (P) implies 'understanding of that term' (Q).

However, a single question is not enough, since a student can get a correct answer by guessing. To minimize the amount of opportunities for such occurrences we propose use of two questions.

Based on the theory of six-valued logic in [5], and [6] we propose the following:

- Two correct answers imply understanding of that particular term.
- One correct answer and one unanswered question imply some doubt about the student's understanding of that particular term.
- One correct answer and one incorrect answer imply doubt about the student's understanding of that particular term.
- Two unanswered questions imply uncertainty about the student's understanding of that particular term.

- One incorrect answer and one unanswered question imply doubt about the student's understanding of that particular term.
- Two incorrect answers imply lack of understanding of that particular term.

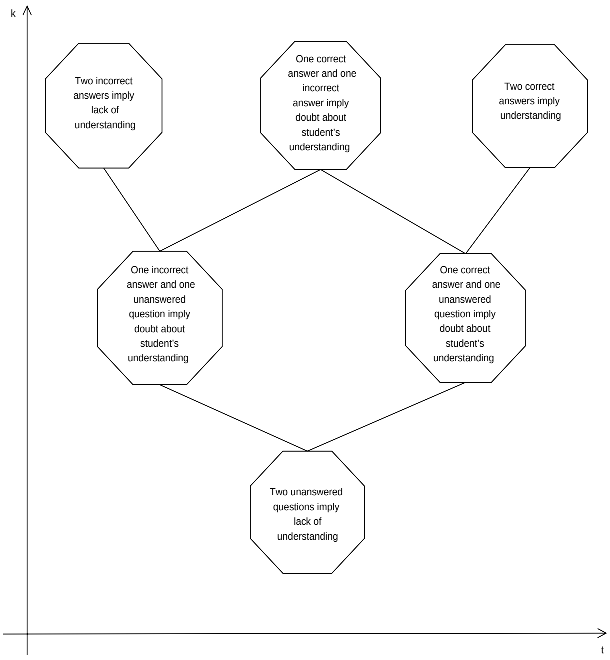

Fig. 4 Partial ordering

A lattice showing a partial ordering of the above described cases is presented in Fig. 4. The knowledge lattice illustrates how the truth value of a formula that has a temporary truth value can be changed as more knowledge becomes available.

Suppose a student takes a test second time. An intelligent agent is making a decision about the student's knowledge based on the outcomes of both tests and applying relations among truth values as given in [5].

However, the student can start a new assessment of his/her understanding of that particular term at any time he/she wants.

3.2 Tests with four questions

For establishing the current level of mastering a particular skill we propose a multiple choice test with four questions. Every answer can be correct (*c*), incorrect (*i*) or partially correct (*p*).The case where no answer is provided is denoted by (*n*).

Thus we obtain the following answer combinations:

- Four correct answers (*cccc*)
- Three correct answers and one partially correct answer (*cccp)*
- Three correct answers and one unanswered question (*cccn*)
- Three correct answers and one incorrect answer (*ccci*)
- Two correct answers and two partially correct answers (*ccpp*)
- Two correct answers, one partially correct answer and one unanswered question (*ccpn*)
- Two correct answers, one partially correct answer and one incorrect answer (*ccpi*)
- Two correct answers, one unanswered question and one incorrect answer (*ccni*)
- Two correct answers and two unanswered questions (*ccnn*)
- Two correct answers and two incorrect answers (ccii)
- One correct answer and three partially correct answers (*cppp*)
- One correct answer, two partially correct answers, and one unanswered question (*cppn*)
- One correct answer, two partially correct answers, and one incorrect answer (*cppi*)
- One correct answer, one partially correct answer, one unanswered question, and one incorrect answer (*cpni*)
- One correct answer and three unanswered questions (*cnnn*)
- One correct answer, two unanswered questions, and one partially correct answer (*cnnp*)
- One correct answer, two unanswered questions and one incorrect answer (*cnni*)
- One correct answer, one unanswered question and two incorrect answers (*cnii*)
- One correct answer, two incorrect answers, and one partially correct answer (*ciip*)
- One correct answer and three incorrect answers (*ciii*)
- Four partially answers (*pppp*)
- Three partially answers and one unanswered question (*pppn*)
- Three partially answers and one incorrect answer (*pppi*)
- Two partially answers and two unanswered questions (*ppnn*)
- Two partially answers, one unanswered question and one incorrect answer (*ppni*)
- Two partially answers and two incorrect answers (*ppii*)

- One partially answer and three unanswered questions (*pnnn*)
- One partially answer, two unanswered questions and one incorrect answer (*pnni*)
- One partially answer, one unanswered question and two incorrect answers (*pnii*)
- One partially answer and three incorrect answers (*piii*)
- Four unanswered questions (*nnnn*)
- Three unanswered questions and one incorrect answer (*nnni*)
- Two unanswered questions and two incorrect answers (*nnii*)
- One unanswered question and three incorrect answers (*niii*)
- Four incorrect answers (*iiii*)

All answer combinations are grouped in five lattices with respect to the five truth values.
Any two nodes, in Fig. 5, Fig. 6, Fig. 7, and Fig. 8, connected by an edge differ in one answer only. Going upwards from one level to another in the lattices in Fig. 5, Fig. 6, Fig. 7, and Fig. 8 increases the level of certainty.

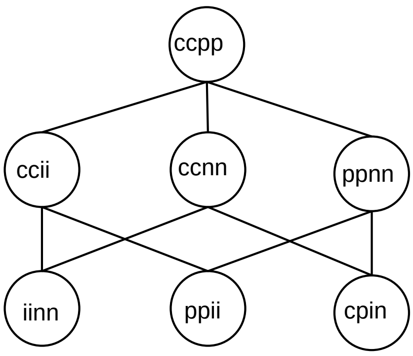

Fig. 5 Answer combinations related to the truth value *tt*

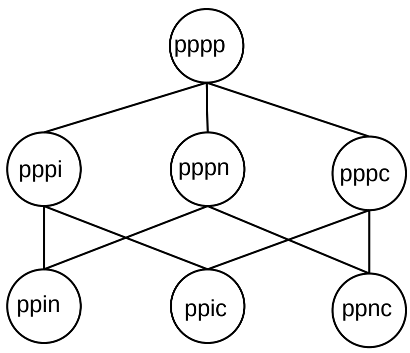

Fig. 6 Answer combinations related to the truth value *kk*

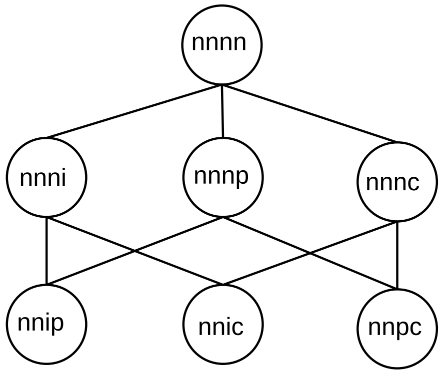

Fig. 7 Answer combinations related to the truth value *uu*

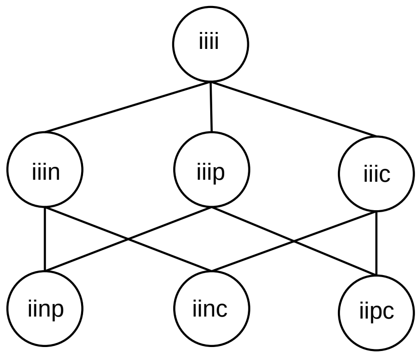

Fig. 8 Answer combinations related to the truth value *ff*

Any two nodes, in Fig. 9, connected by an edge differ in exactly two answers. Going upwards from one level to another in the lattices in Fig. 9 increases the level of knowledge.

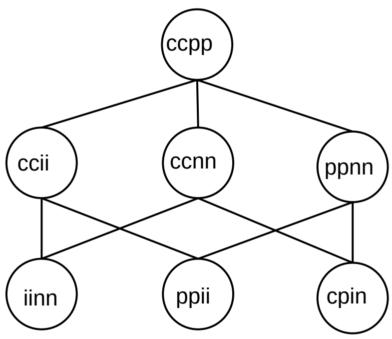

Fig. 9 Answer combinations related to the truth value *ww*

Application of seven-valued logic [3] supports a decision making process related to drawing conclusions in case a student is taking the same test several times. This way the system keeps a history of the learning progress of every student and is able to provide personalized guidance based on that history.

4 System Architecture

The system is implemented using the so-called LAMP Web server infrastructure and deployment paradigm. It is a combination of free software tools of an Apache Web server, a database server and a scripting programming platform on a Linux operating environment.

Behind this traditional three-tiers Web deployment is a service support sub-system. Communication framework based on XML-RPC is used to connect the Web application middle-ware and the intelligent assessment/diagnostic system together. The separation of these two units made it possible to modularly design and implement the system as loosely couple independent sub-systems.

An authenticated user receives a unique session key that is used to identify user in the system for that particular session. This session key is saved in the user Web browser cookie. All sessions' dynamic data stored in the database are indexed using this session key. A session key is used to index state variables assigned to a user for that particular session. The session key cookie and dynamic data stored are used together to keep user's states of interaction with the system within otherwise a stateless HTTP protocol.

The dynamic page publisher compiles a page to be presented to the user from a template file in relation to the user response, current state variables and activities history.

A template file contains the static declarations of a document. The variables in a particular template files are given values by the dynamic page publisher module during the production of an HTML document. The resulting HTML document is sent back to the user Web browser. This module also acts as a handler when a user requests a page or sends a form back to the Web server.

Each learning unit is atomic, self-contained and reusable. The dynamic page publisher makes use of these learning units to provide students with a dynamic and personalized learning material as a direct reaction to students' interaction to the system.

The user authenticator authenticates a user during login and creates initial session contact in the system if the user provides correct credentials. It is realized by creating a unique session key and saving that key in the database of the server and a cookie in the client. The module also provides user authorization during an active user's session. The user authenticator module is responsible for session cleanup at user's logoff.

The users stack profiler keeps track of user activities history in a stack like data structure in the database. Each event, like for example response/result of a test or a change of learning flows after following a hint given by the system, is stored in the database. This module provides the percepts to the intelligent modules of the software agents' sub-system. The users stack profiler communicates directly with the agents by sending messages over the XML-RPC communication channel. By using some common data stored in the database, the users stack profiler indirectly affects the behaviour of the user's agents and vice verse.

The application middleware and the software agents run independently of each other. As such, they can be situated on different servers. The middleware implement the Web side of the system while the software agents implement the decision side of users learning process. Given a certain response to a particular test at a particular user state, what best action can be taken to increase the probability that the user will learn a particular unit of knowledge? This decision is done by the intelligent diagnostics agent.

The intelligent assessment agent is a special agent that implements six-valued logic table of inference for testing students' understanding of a particular basic term by using a pair of questions after the term has been introduced. The agent does an early diagnostic about absorption of knowledge. A response given by a particular student from a test will give the system an indication about the state of learning of a particular term. Each of the different truth values of a response triggers different rule-based reaction. This agent helps to implement a part of an intelligent tutoring system, which differ from the intelligent diagnostics agent. The intelligent assessment agent facilitates students' early absorption/assimilation of new terms.

5 Conclusion

This paper presents an intelligent sub-system assessing students initial understanding of new terms and skills learning. The decision making process is based on a many-valued logic.

Our motivation for employing many-valued logic is that this way the system will provide better personalized help to the students and course builders will receive more detailed information about the effectiveness of their learning materials. In our opinion assessment rules for students' understanding of new terms, shortly after they have been introduced, should differ from the ones used for marking students' achievements in a subject.

References:

[1] F. R. de Carvalho, I. Brunstein, J. M. Abe , Decision Making based on Paraconsistent Annotated Logic. *Frontiers in Artificial Intelligence and Applications*, Nakamatsu, K., Abe J. M. (eds) Vol. 132, 2005, pp.55-62.

[2] B. A. Davey and H. A. Priestley, *Introduction to lattices and order*, Cambridge University Press, Cambridge, 2005.

[3] U. Ferreira, A Five-valued Logic and a System, *Journal of Computer Science and Technology*, Vol. 4(3), 2004, pp. 134-140.

[4] M. Fitting, Kleene's Logic, Generalized. Journal of Logic and Computation, 1(6) (1991, pp. 797-810.

[5] M. Fitting, Kleene's three-valued logics and their children. *Fundamenta Informaticae*, Vol. 20, 1994, pp. 113-131.

[6] M. Fitting, Tableaus for many-valued modal logic. *Studia Logica*, Vol. 55, 1995, pp. 63-87.

[7] O. N. Garcia and M. Moussavi, A Six-Valued Logic for Representing Incomplete Knowledge. *Proceedings of ISMVL*, IEEE Computer Society Press, Charlotte, NC, USA, 1990, pp. 110-114.

[8] J. García-Duque, M. López-Nores, J. Pazos-Arias, A. Fernández-Vilas, R. Díaz-Redondo, A. Gil-Solla, Y. Blanco-Fernández and M. Ramos-Cabrer, A Six-valued Logic to Reason about Uncertainty and Inconsistency in Requirements Specifications. *Journal of Logic and Computation*, Vol. 16(2), 2006, pp. 227-255

[9] J. Kacprzyk and M. Roubens, Non Conventional Preference Relations in Decision Making. *LNMES* , Vol. 301, 1988.

[10] S. Kleene, *Introduction to Metamathematics*. D. Van Nostrand Co., Inc., 1952.

[11] J. Lukasiewicz, On Three-Valued Logic. *Ruch Filozoficzny*, 5, English translation in Borkowski, L. (ed.) 1970. Jan Lukasiewicz: Selected Works. Amsterdam: North Holland. 1920.

[12] M. Moussavi and O. N. Garcia, A Six-Valued Logic and its application to artificial intelligence. *Proceedings of the Fift Southeastern Logic Symposium*, 1989.