IJITCE

ISSN 2347- 3657

International Journal of Information Technology & Computer Engineering

Email : ijitce.editor@gmail.com or editor@ijitce.com

International Journal of
Information Technology & Computer Engineering

ISSN 2347–3657
Volume 10, Issue 2, 2022

An Intelligent Tutoring System for Automata Theory: A Proposed Framework

Luo Da, Den Hui, Zhang Youhu

Department of Graduate School, Beijing University of Chinese Medicine, Beijing
, Department of Ophthalmology, China-Japan Friendship Hospital, Beijing

Abstract: Humans are being displaced in almost every aspect of their existence on this planet by computers and the advances they have made in technology. Since the discovery of Artificial computers that function on the same principle as the human brain and learn with changes in experience over time, the concept of intelligent computers attempting to emulate the human brain has developed. Learners benefit from intelligent tutors since it is hard for a human tutor to focus on every student in a classroom setting. As a result of a review of current intelligent tutoring systems for undergraduate courses in Automata Theory for computer science students, this article presents an improved approach.
Keyword: FLAT, Automata Theory, and Intelligent Tutoring System.

Introduction

An concept from distant education that allowed students to study at their own speed and in any location came up with the notion of giving instruction online, which was a result of research in education. Education and Artificial Intelligence have been bolstered and advanced by scholars who have worked in both fields. Traditional computer-aided teaching systems have developed into intelligent tutoring systems (CAI). Intelligent tutoring systems (ITS) were originally known as Intelligent Computer Assisted Instruction (ICAI) when CAI was designed to act intelligently (ITS). [11]

Computational modules on cognitive sciences, cognitive learning, computational linguistics and artificial intelligence, as well as mathematics and regular language, are used in computer-based learning environments to construct intelligent systems that are well-specified computationally. [1]

Following are some of the original ITS needs outlined by researchers in [21]:

(a) A working knowledge of the field (Expert model)

learner self-awareness, or b) (Student model)

b) An understanding of instructional methods (Tutor model)

Although the overall structure has remained the same, a new module has been introduced.

Intelligent tutoring systems currently rely on the four-model design, which maintains the previous three models while adding a fourth.

The following are four examples of each design type:

(a) Database of knowledge

a) Student's Representation

b) A model for teaching

Interface (d)

The knowledge base is identical to the preceding designs' domain model. Consolidation of declarative, procedural, and metacognitive information is the goal of a tutor or coach.

This is an internal model that describes the cognitive processes, metacognitive methods, and psychological traits of each learner.
A comparable module exists in the other architectures as a pedagogical model for this one. It selects an efficient route through its knowledge representation based on a model of the learner's current understanding to produce expert behaviour by the learner.
This module combines three categories of information: knowledge about the patterns of interpretations and actions inside conversations; domain knowledge required for presenting material, and knowledge needed for conveying intent.
Student knowledge is 'actively constructed' rather than passively learned from textbooks and lectures, according to this theory. In this way, each student will create their own unique form of knowledge since the construction builds recursively on the learner's prior knowledge (facts, thoughts, and beliefs). Constructivist teaching methods are expected to be more effective than conventional methods because they openly address the unavoidable process of knowledge production. [24]

Computer Science teaching is challenging to use constructivist theory because of these reasons.

• Because sensory data from class must be integrated into a student's existing framework that is too superficial, CS concepts are constructed using constructivist theory.

There is a lot of frustration and the idea that computer science is difficult since models have to be built from scratch.

• Success in academic computer science courses is not always connected with autodidactic programming expertise. These pupils, like physics students, undoubtedly arrive to school with preconceived notions that aren't going to hold up under the scrutiny of a classroom environment.

• Students who prefer a more introspective or social learning approach may be discouraged by the actual feedback they get while working on a computer. [24]

FLAT is included in most Computer Science curricula because it provides students with an understanding of the mathematical foundations of computing, as well as its power and limits, as well as how to use it. the numbers 20, 10, and 19, respectively.

Theoretical computer science, formal languages and automata theory (FLAT), and other similar terms are taught in many computer science curricula. Designing Finite Automata (FAs), Push-Down Automata (PDA) and Turing machines to identify languages of the appropriate class is a common requirement in these courses.

It's common for students to struggle with these FLAT ideas since they're so abstract and difficult to grasp (as there is a perception that the field is not only dated, but that it has little current applicability in the real world). [4] In order of appearance: [7] [28] [30]... [34] As a result, students may get demotivated and disappointed, and they may not remember much of what they've studied. It has been shown that a lack of problem-solving abilities is a key contributor to the challenges connected with the development of FAs. The logic mistakes that were shown to be common may be traced back to a lack of problem-solving skills. It is possible that pupils are relying on "plug and chug" procedures without a clear knowledge of how they work. Many teaching methods have been developed in response to these issues, as well as to aid in the development of stronger mental models of FAs [5]. [38]

• Active learning and constructivist teaching strategies: Students require quick feedback at each level of the process of developing FAs to better improve their problem-solving abilities (something that is very difficult to implement in a distance learning environment)

By offering an alternate perspective and letting students to interact with the ideas by testing them on multiple input strings with quick feedback, visualisation tools aid students in grasping FAs and how they function. Such methods are based on the idea that students can make abstract models more tangible when they are able to engage with them. [9]
Link to current computer science applications: This might include leveraging programming expertise as incentive for FA investigation and usage of its application to real-world issues. • Link to current computer science applications.

These systems give students with personalised guidance, adjusting the path through the needed content depending on the student's progress.
Students put off taking the course until their final semesters of an undergraduate course because they have only rudimentary or no prior understanding of computer science theory and arithmetic. The pupils become bogged down in the language if the educational content isn't motivating enough [14]. The issue here is that students may resort to rote learning if they do not grasp the topics [37].
A variety of hardware and software applications benefit greatly from automata theory's insights. Because they are abstract, these topics are often taught using a conventional lecture format, which works well for students who want to think things out for themselves.
Computer Engineering students, on the other hand, exhibit a significant predilection for active and sensory learning.

RELATED WORK:

Formal Languages and Automata Theory have been supported by a few famous instances of e-learning systems.

Since it was launched in 1998, FLUTE (Formal Languages and Automata Education) [42] has been educating students about automata theory via examples and dependency graphs connecting related subjects. However, this system is out of date and is no longer in service.

For automata theory, a smartphone-based multimedia learning system was created. Simulating an automaton, reading course notes, seeing presentations, taking a quiz and even having a help section that acts as a handbook are just some of the features available.

The difficulties of online education and the static acquisition of information have been addressed in certain ways in [41] [32] [2] [13]. They have been conceived and developed so that students may learn via exploration, rather than relying just on textbooks. This kind of method is often used in learning activities where students answer practical issues, so that the information acquisition involves the completion of an acceptable number of exercises..

As a result of these systems, students are given exercises from a repository, given the opportunity to contribute their own answers, given comments on their work, and even given the opportunity to create new challenges depending on their progress. There is no way for students to create their own exercises using these systems. They are limited to working with a collection of activities that are already in a repository. [18]

Motivated teachers seek for or develop visual aids to aid their pupils in their learning processes. As long as kids are interested in what they are learning, they will find a way to succeed. Animations have been suggested for the DFA, NFA and Turing machines. In addition, the recommended remedy has not been implemented, and no explanation is offered as to how to create these storyboards

An introduction to FSMs is provided by these first two components. In one component, learners are given a brief movie-style introduction to the subject matter. In addition, there is a more in-depth hypertext explanation of the fundamental principles. Students may use this component to get a general understanding of the material. Finite automation and a Turing machine simulator are included in the course, which also includes two games for students to apply what they've learned. [25]

It is composed of several components, including an animated (movie-like) welcome component, a hypertext introduction to computation theory topics, a finite state machine (FSM) simulator, a Turing machine (TM) simulator, a self-assessment component for online collaborative learning, and three other components showing visual examples of automata, such as video, of Hamada's e-learning system for automata and the theory of computation These detailed resources are tough for novice automata students to understand since they were developed to accommodate a wide range of learning styles. Students are stumped as to where to begin their studies. [27]

To make the learning experience more fun, [27] adds an additional component that helps the student discover their preferred method of learning.
In the same way as other IT systems, SELFA-Pro is made of a number of modules that work together to achieve the overall system goal. The FLAT problem solver module, the linker

module, and the interface module make up the proposed system. Each module has been meticulously crafted to do a particular set of activities using just the data that is required. It is the goal of the issue solver module to take the student's exercise (declarative knowledge) and assess the problem or circumstance needs before using the knowledge to solve the problem (i.e. execute procedural operations). The linker module will provide advice on how to repair an issue after it has identified one (i.e. attempts to place conditional knowledge into specific execution). In accordance with the linker module's recommendations, the interface module gives students mechanisms for sending exercises and for creating the most suitable view of system outputs. [18]

PROBLEM IDENTIFICATION:

It is assumed that the student has a basic understanding of formal languages and automata theory in order to benefit from the teaching aids currently available. Since no mechanism has been created to help a pupil who does not comprehend the essential ideas.

Simulators and emulation seem to be the offered answers in the current systems, which aid students in putting the ideas and information they've acquired to use. There isn't a single system that makes it easier for beginners to learn about automata theory topics.

Mathematical foundations of computer science are assumed to have been taught in the past, but students tend to forget these concepts or rely on rote learning to get a passing grade, which prevents them from progressing to a higher level course or semester in the traditional classroom.

Assuming that all students have the same level of comprehension and that a teacher is unable to spend enough time with each one, most E-learning systems for this purpose use recorded lectures, scanned textbooks as study materials, and scanned handwritten notes, which are considered to be standard for the acquisition of knowledge in the traditional classroom-based scenario.

For the most part, ITS built in computer science focus on offering individualised answers to issue questions relevant to information learned by students, rather than on the learning module itself. For many students, however, this isn't a viable option since they are frustrated by the fact that they must go back to their textbooks or notes for notes and study material while using the system the conventional way.

PROPOSED SOLUTION:

An ITS for formal languages and automata theory may be developed by using the four current model ITS designs [16] [wing components:

For further information, see the17] [39], which are endorsed by numerous scholars [16] [17] [39]. It is composed of the follo following resources:

Theoretical framework for instruction

- The User Interface

Fig: Proposed Solution of ITS for Automata Theory

Knowledge base: The ITS for automata theory will include a knowledge base made up of learning objects that may be mapped to the preferences and requirements of learners based on their many intelligences. When creating the parts of the knowledge base, attention must be made since it contains domain knowledge, restrictions, rules, and difficulties linked to the topic. When the learner model is modified, the knowledge base should be updated accordingly.

The various needs, preferences and strengths of the eight Intelligences can be seen below which is taken from [12]

Intelligence Area	**Strengths**	**Preferences**	**Needs**
Verbal / Linguistic	Writing, reading, memorizing dates, thinking in words, telling stories	Write, read, tell stories, talk, memorize, work at solving puzzles	Books, tapes, paper diaries, writing tools, dialogue, discussion, debated, stories, etc.
Mathematical/ Logical	Math, logic, problem-solving, reasoning, patterns	Question, work with numbers, experiment, solve problems	Things to think about and explore, science materials, manipulative,
Visual / Spatial	Maps, reading charts, drawing, mazes, puzzles, imagining things, visualization	Draw, build, design, create, daydream, look at pictures	video, movies, slides, art, imagination games, mazes, puzzles, illustrated book
Bodily / Kinesthetic	Athletics, dancing, crafts, using tools, acting	Move around, touch and talk, body language	Things to build, movement, tactile experiences, hands-on learning, etc.
Musical	Picking up sounds, remembering melodies, rhythms, singing	Sing, play an instrument, listen to music, hum	Sing-along time, musical instruments, etc.
Interpersonal	Leading, organizing, understanding people, communicating, resolving conflicts, selling	Talk to people, have friends, join groups	group games,
Intrapersonal	Recognizing strengths and weaknesses, setting goals, understanding self	Work alone, reflect pursue interests	self-paced projects, choices
Naturalistic	Understanding nature, making distinctions, identifying flora and fauna	Be involved with nature, make distinctions	Order, same/different, connections to real life and science issues, patterns

Table: Strengths, preferences and needs of the Eight Intelligences

Many of these intelligences can still be used in automata theory education, even if they can't be applied to all of them. Identifying learning objects for each form of intelligence in automata theory ideas would be a future project in this field.

Student Model:

The student model is the internal model that will keep track of the following information about the learner:

- The current degree of understanding of the subject by asking questions regarding the subject's preparatory themes. Assuming a clean slate would be a far cry from the current state of affairs, though.

A questionnaire based on Gardner's theory of multiple intelligence [15] is used to determine a person's intelligence type, which may then be used to design learning routes appropriately.

In order for the knowledge base to update the learner model and make expert judgements, it needs the ability to retain and remember information.

It is important to note that the Bloom's learning goals [22] have not been met, and a 100 shows that the principles of the constructivist theory [6] have been fulfilled, where the learner develops new knowledge.

Pedagogical model:

Learner profiles are used to match several learning routes and discover the most efficient one, so that a learner advances in such a way that produces expert behaviour.

User interface:

Integration of the three models into a unified user interface will ensure that it does not overload the student with too much information, while at the same time helping to maintain the learner's pace and redirecting them to other links utilising adaptive hypermedia [33].

Evaluation module:

These milestones for the learner may be referred to as ITS assessment modules that test the student's knowledge on a regular basis to ensure that the learning objectives have been met. From simple multiple-choice quizzes to more advanced methods like employing simulators and emulators to solve issues, there are a wide range of assessment methods available. The next level of assessment would be to see how well students can apply what they've learned by designing their own games and puzzles. In order to make the principles more understandable, they should be applied to real-world scenarios.

In this case, the knowledge base would assist students in solving issues by highlighting incorrect answers and providing clues to the right ones, much as a human tutor would.

Regularly, the student model will update the profiles of students and their potential learning routes based on assessment findings.

Conclusion and future work:

Problems in the current intelligent tutoring systems for automata theory have been discovered and a solution has been offered to address these issues.

In the future, this system will be designed and developed, and the results of its assessment will be reported by comparing its usage with those of the conventional classroom method. These findings may be used to other theoretical areas, such as the Mathematical Foundation of computer science, Digital Logic, Compiler Design etc. if results are sufficient.

References

"Computerized learning environments that include research in discourse psychology and cognitive science as well as computational linguistics," by A. C Grasser, X. Hu and D. McNamara in Experimental cognitive psychology and its applications. 2005 marks the end of a decade of conduct..

Computers & Education, Volume 53, Number 2, 2009, Pages 243-251, "Using Co-Lab to Build System Dynamics Models: Students' Actions and On-Line Tutorial Advice,"
[3]
Concise Review and Synthesis of Recent Research in Intelligent Computer-Assisted Instruction by C. Dede

24:329-353 (1986), International Journal of Man-Machine Studies (IJMS).

A psychological theory of learning based on constructivism is presented in the second volume of Constructivism: Theory, Perspectives, and Practice, pp. 8-33.

International Conference on Computer Systems and Technologies, 2007: D. Chuda, "Visualization in the Education of Theory of Computer Science," p. 58.

[1] "Automata and Formal Languages: An Introduction," by David Kelley. Prentice-Hall Inc., Upper Saddle River, NJ (USA), 1995.

[2] Sleeman and Brown (1982), Intelligent tutoring systems: Cambridge: Academic Press, published by D.

[3] Multiple intelligences and learning styles," in Emerging Perspectives on Learning and Teaching with Technology, edited by M. Orey, 2003, CreateSpace Independent Publishing Platform (D.W. West, J.P. Pearce and M.Chance).

[4]

[5] volume 20, issue 5, pp. 36-46, E. Fernandes and A. Kumar (2005a).An introduction to subprogram implementation in the Journal of Computing Sciences in Colleges,

[6]

[7] Theoretical Computer Science Teaching at the Undergraduate Level: Experiences, Observations, and Proposals for Improving the Status Quo, UCLA Computer Science Department, Los Angeles, 1988.

Multiple intelligence theory has educational implications for the classroom, according to H. Gardner and T. Hatch, "Educational implications of multiple intelligences," in Educational researcher, 1988, pp. 4–10.
Henri L. Burns and Charles G. Capps, "Foundations of Intelligent Tutoring Systems," in Foundations of intelligent tutoring Systems, Hillsdale Erlbaum Associates pp. 1-19 (1988).
Learning challenges for intelligent tutoring systems, New York, Springer-Verlag, 1988: H. Mandl and A. M. Lesgold, "Toward a philosophy of curriculum to be used in constructing intelligent instructional systems,"

An investigation of the use of formal languages and automata theory as an example of knowledge-based systems to increase learning has been published in Knowledge-Based Systems, 2017.

Introduction to Languages and the Theory of Computation, New York, NY, USA: McGraw-Hill, Inc., 2007.

Addison-Wesley Longman Publishers, Inc., Boston (MA), 2006. Introduction to Automata Theory, Languages and Computation, by John E. Hopcroft, Richard Motwani, and John D. Ullman.

J. R. Hartley and D. H. Sleeman, "Towards more intelligent tutoring systems," International Journal of Man-Machine Studies, vol. 2, pages 215-236, 1973.

There are 15 authors listed in this section: LW Anderson, DR Krathwohl, KW Cruikshank, RE Mayer, PER Pintrich, J Raths and M C Wittrock [15] [16] [17] [18] [19] An updated version of Bloom's Taxonomy of Educational Objectives: A Taxonomy for Learning, Teaching, and Assessing

Computer science education: a constructivist perspective, presented at the twenty-ninth annual SIGCSE technical conference in Atlanta in 1998, by M. Ben-Ari

in Theory of Computation: An Integrated Virtual Environment for Active and Collaborative e-Learning," in IEEE TRANSACTIONS ON LEARNING TECHNOLOGIES, vol. 1, no 2.

A Study of a Learning Style Index to Support an Intelligent and Adaptive Learning System, in Intelligent and Adaptive Educational-Learning Systems, Achievements and Trends, Springer Berlin Heidelberg, 2013, pp.109-132, by M. Hamada, K. Nishikawa and J. Brine [18].

IEEE International Conference on Engineering Education (AICERA) 2012, Kottayam, India: "Activity-based teaching and learning in formal languages, automata theory - An experience," by M. Vijayalaskhmi and. K. Karibasappa.

M. Hamada, "Web-based Active E-Learning Tools for Automata Theory," in ICALT 2007, 2007.

"Web-based Education for All: A Tool for Development of Adaptive Courseware," by P. Brusilovsky, J. Eklund, and E. Schwarz, in the Seventh International, 1998Methods and methods of adaptive hypermedia by P. Brusilovsky, published in User Modeling and User-Adapted Interaction, 6(2):87-129, 1996. [8]

[8] "Affective Computing — A Rationale for Measuring Mood with the Mouse and Keyboard," International journal of occupational safety and ergonomics: JOSE •, vol. 9, no. 4, 2003, pp. 539–51.

[9]

[10] The learning preferences of engineering students from two viewpoints, P. Rosati, Frontiers in Education, Tempe (Arizona): 1998.

[11]

[12] In "Rote Versus Meaningful Learning," R. E. Mayer argues for the need of meaningful learning above rote learning.

[13] Advances in Intelligent Tutoring Systems, Berlin, Springer-Verlag (2010), pp. 63-80, R. Nkambou and R. Mizoguchi and J. Bourdeau

[14] A visual and interactive automata theory course stressing the breadth of automation was presented at the 10th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education, ITiCSE, Caparica in Portugal in 2005.

[15] [16] T. Murray, "Authoring intelligent tutoring systems: an examination of the current state of the art," International Journal of Artificial Intelligence in Education, vol. 10, pp. 98-129, 1999.

[16]

[17] V. Devedzic, J. Debenham, and D. Popovic, "An Intelligent Tutoring System for Formal Languages,"

[18]

[19] Vol. 3, no. 2, 2000, Educational Technology & Society.

[20]

[21] Automata Theory in an Active and Collaborative Environment: A Multimedia Learning System on a Smartphone for Automata Theory," presented at the 11th International Conference on Electronics (ICECCO) in 2014.