Australian Journal of Basic and Applied Sciences, 8(4) Special 2014, Pages: 20-23

AENSI Journals

Australian Journal of Basic and Applied Sciences

ISSN:1991-8178

Journal home page: www.ajbasweb.com

Teaching Theory of Computation – An Experience

M.A. Iliana, I.M. Nazri, J. Suriati

Faculty of Computer, Media & Technology Management, TATI University College, 24000Terengganu, Malaysia

ARTICLE INFO

Article history:
Received 20 November 2013
Received in revised form 24 January 2014
Accepted 29 January 2014
Available online 5 April 2014

Keywords:
Automata; flow; assessment

ABSTRACT

The course of theory of computation is important in the computer science curriculum as this course is necessary in the teaching of courses on computer design, artificial intelligence, and the analysis of algorithms. The concepts learnt in this course can be applied in compiler design and natural language as well. Teaching the course on theory of computation is challenging because there are difficulties in understanding the automata, computability and complexity theory concepts as the course is mathematical in nature. Additionally, the conventional approach to the course is that given a sequence of machine and then come out with the corresponding language theory. The standard references also recommended the same approach. The authors observed that this approach has failed to establish the context required for the study. An approach to teaching the course on theory of computation is attempted by the authors. This flow of teaching makes students understand the concepts and relate to the real time applications easily. Course assessment was done in term of regular tests, quizzes, and final examination. The authors have observed that these has resulted in increased learning of students as seen in their performance and decrease pressure during examination as students were comfortable with the learning. The paper discusses the experience of the course teachers and the results obtained.

To Cite This Article: M.A. Iliana, I.M. Nazri, J. Suriati., Teaching Theory of Computation – An Experience. ***Aust. J. Basic & Appl. Sci.,*** *8(4):20-23, 2014*

INTRODUCTION

The course of theory of computation is important in the computer science curriculum as this course is necessary in the teaching of courses on computer design, artificial intelligence, and the analysis of algorithms. The concepts learnt in this course can be applied in compiler design and natural language as well. Teaching the course on theory of computation is challenging because there are difficulties in understanding the automata, computability and complexity theory concepts as the course is mathematical in nature. The conventional approach to the course is that given a sequence of machine and then come out with the corresponding language theory. The standard references also recommended the same approach.

Learning the course was difficult for students since teaching was monotonous, no participation of students during the teaching sessions and poor understanding of the subject since it is mathematical in nature.

It is observed that the student lack in problem solving ability, the subject is categorized as high-risk; and it required more preparation time during the examination.

An attempt has been made to overcome the problems, so the subject begins with the discussion of mathematics such as sets, functions, relations, graphs, and proof techniques which gives the strong foundation to understand the concepts of automata, computability and complexity theory.

2. Course Design:

As an introductory course to undergraduate students, the course on automata, computability and complexity theory has been designed where the purpose is to teach automata, computability and complexity followed by the properties of the respective languages, and the course delivery is attempted with the following course objectives (CO's) written according to Bloom's Taxonomy.

Course Objectives:

- Explain and proof the capabilities and limitations of machines with different type of hardware: finite automata, push-down machines, Turing machines.
- Explain the concepts of decidability and undecidability

Corresponding Author: M.A. Iliana, Department of Computer Science, Faculty of Computer, Media and Technology Management, TATI University College, 24000 Terengganu, MALAYSIA.
E-mail: iliana@tatiuc.edu.my

- Do mathematical proofs for abstract machine.
- Differentiate problems according to complexity.

Classroom Delivery:

Fig. 1 displays the flow of teaching the course in a classroom. For example, the regular language is taught, then the finite state machine, and its properties along with the applications (Hopcroft, J.E., Motwani, R. and Ullman, J.D., 2000). This flow will help the students to understand how a language is represented using regular expressions (Linz, P., 2012) and constructing the finite state machine, exploring its properties and relate to the real time applications (Martin, J.C., 2011).

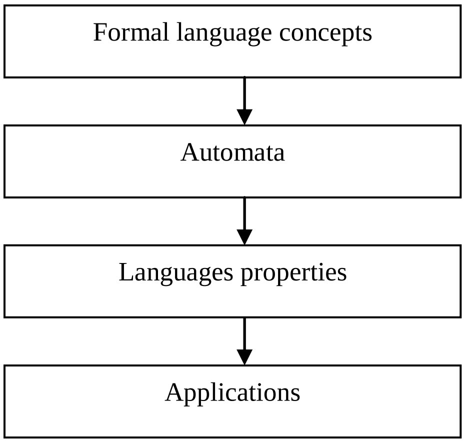


Fig. 1: Content delivery flow.

3. Assignment:

At the end of every selected chapter, an assignment was given, for instance, assignment 2 was to implement the Turing machine (TM) learnt in respective chapter.

Extracted Assignment 2: Give sequence of configurations that TM enters when started on the indicated input string (Sipser, M., 2013). The algorithm is as follows:

M_2 = "on input string w:

1) Sweep left to right across the tape, crossing off every other 0.
2) If in stage 1 the tape contained a single 0, *accept*.
3) If in stage 1 the tape contained more than a single 0 and the number of 0s was odd, *reject*.
4) Return the head to the left-hand end of the tape.
5) Go to stage 1."

4. Assessment:

The criteria of assessments are given in Table 1.

Table 1: Criteria of Assessment.

Assessment	Percentage, %
Quiz	10
Test	20
Assignment	30
Final Examination	40

Tests:

There were two tests conducted. The objective was to test the problem solving ability of a student.

Quiz:

There were two quizzes conducted at the end of each selected chapter. The objective is to train the students to type of questions.

Assignment:

There were three assignments given at the end of every selected chapter. Here the objectives was to make the students analyze how an automata, a Turing machine (TM) and a post correspondence problem (PCP) work. Table 2 shows the assignment rubrics for evaluation of the assignment 2.

Final Examination:

A three hour final examination was conducted; the question paper is prepared based on the learning objectives. The objective is to test the overall understanding of students towards theory of computation course.

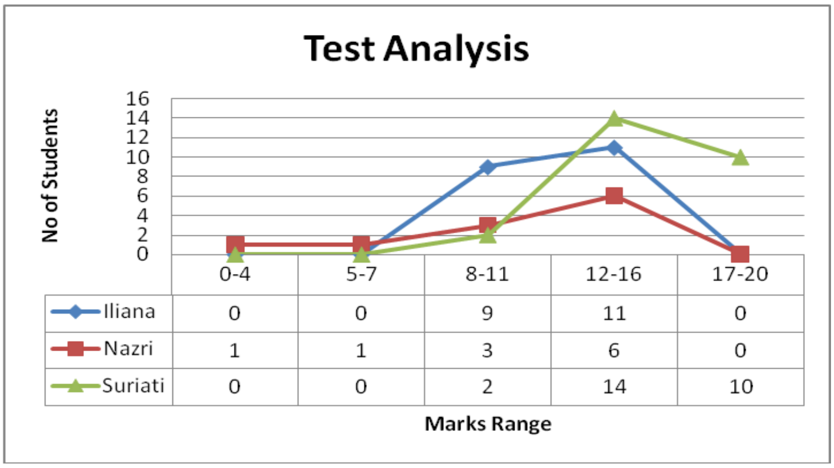
Table 2: Assignment Rubrics for Evaluation

Criteria	Bad	Weak	Moderate	Good	Excellent
Condition	Both conditions are wrong	One of the conditions is partially correct.	One of the conditions is correct.	One condition is correct, the other is partially correct.	Both conditions are correct.
Input/Output alphabet	Input/Output alphabet is wrong.	Either input or output alphabet is partially correct.	Either input or output alphabet is correct.	Three-fourths of input or output alphabet is correct.	Input/Output alphabet is correct.
Input/Output, sequence of states conclusion	All input/output and sequence of states conclusion are wrong.	Either input or output and sequence of states are partially correct.	Either input or output and sequence of states conclusion are correct.	Either input or output and sequence of states is/are correct and partially correct.	All input/output and sequence of states conclusion are correct.
Sequence of states	All sequences of states are wrong.	Only one sequence of states is correct.	Two sequences of states are correct.	Three sequences of states are correct.	All sequences of states are correct.
Tuple, states, and transition function	Tuple, states and transition function are wrong.	Tuple or one state or transition function is correct.	Either tuple and transition function or states are correct.	States and tuple or transition function are correct.	Tuple, states and transition function are correct.

Result and Observation:

The main intention was to enhance the problem solving ability.

- Tests and Overall Students' Performance Analysis:



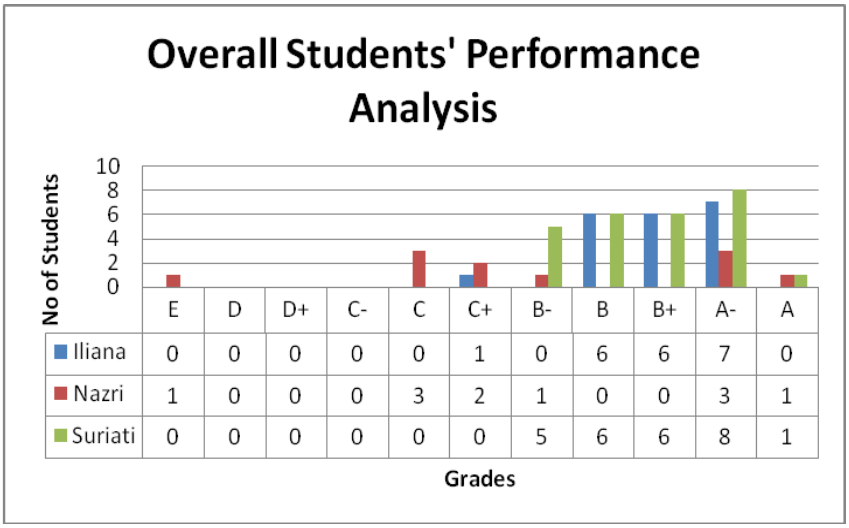


Fig. 2: The above graphs bring out the uniform assessment and teaching methodology followed by three different course instructors.

Observations made by the authors:

The flow of teaching makes students understand the concepts and relate to the real time applications easily. Through the assignments given, students gain better understanding of concepts and can increase the problem solving ability.

Conclusion:

An approach to the teaching the course on theory of computation is attempted by the authors. This flow of teaching makes students understand the concept and relate to the real time applications easily. The authors have observed that these has resulted in increased learning of students as seen in their performance and decrease pressure during examination as students were comfortable with learning. The paper discusses the experience of the course teachers and the results.

REFERENCES

Hopcroft, J.E., R. Motwani and J.D. Ullman, 2000. "Introduction to Automata Theory, Languages, and Computation", Addison Wesley.

Linz, P., 2012. "An Introduction to Formal Languages and Automata", Jones & Bartlett Learning.

Martin, J.C., 2011. "Introduction to Languages and Theory of Computation". McGraw Hill.

Sipser, M., 2013. "Introduction to the Theory of Computation", Cengage.