Learning difficulties experienced by students in a course on formal languages and automata theory

Article *in* ACM SIGCSE Bulletin · January 2009

DOI: 10.1145/1709424.1709444 · Source: DBLP

N. Pillay
University of Pretoria

Learning Difficulties Experienced by Students in a Course on Formal Languages and Automata Theory

Nelishia Pillay
School of Computer Science
University of KwaZulu-Natal
Pietermaritzburg Campus,
Pietermartizburg, KwaZulu-Natal, South Africa
pillayn32@cs.ukzn.ac.za

*Abstract***:** Students taking courses on formal languages and automata theory (FLAT) usually do not find these courses interesting and experience difficulty in grasping the different concepts. While there has been a vast amount of research into methodologies to assist students to conceptualize FLAT topics, there has been no research into the actual learning difficulties experienced by students with the different topics. This paper reports on the findings of a study conducted to identify these learning difficulties for some of the FLAT topics.

Categories and Subject Descriptors: K.3.2 [Computer and Information Science Education]: Computer Science Education
General Terms: Experimentation
Keywords: Formal language and automata theory, learning difficulties

1. INTRODUCTION

Students generally find the content of FLAT courses boring and difficult. A lot of research has been conducted into teaching methodologies to make these courses more interesting and assist students in conceptualizing the different topics. For example, Berque et al. [1] propose the use of pen-based computing to promote active learning in FLAT courses. Furthermore, a number of visualization tools ([2], [4], [8]) such as JFLAP [7] and Thoth [5] to help students visualize the different concepts have been developed. Some studies have examined how programming can be used to motivate students ([9], [10]). Devedzic et al. [3] and Pillay et al. [6] suggest the use of intelligent tutoring systems as a means of assisting students learn FLAT concepts. However, the actual difficulties experienced by students have not been investigated. The study presented in this paper is a first attempt at identifying these difficulties for some of the FLAT topics.

The following section describes the course used to conduct this study. Section 3 describes the learning difficulties experienced by students. A discussion and categorization of these difficulties is presented in section 4. Section 5 summarizes the findings of the study and describes future work.

2. COURSE DETAILS

A third year first course on FLAT was used to identify learning difficulties experienced by students with the following FLAT topics:

Regular Languages

1. Constructing finite acceptors, i.e. DFAs and NFAs.
2. Converting NFAs to DFAs.
3. Minimization of DFAs.
4. Defining regular expressions (REs).
5. Converting regular expressions to NFAs.
6. Converting NFAs to regular expressions.
7. Constructing regular grammars.
8. Converting regular grammars to finite acceptors.
9. Converting finite acceptors to regular grammars.
10. Properties of regular languages
11. Non-regular languages and the pumping lemma.

Transducers

1. Constructing Mealy and Moore machines.
2. Converting Mealy machines to Moore machines.
3. Converting Moore machines to Mealy machines.

Context-Free Languages

1. Context-free grammars
2. Ambiguity and grammars
3. Simplifying grammars.
4. Chomsky normal form
5. Greibach normal form
6. Constructing pushdown automata, i.e. npdas and dpdas.
7. Converting context-free grammars to npdas.
8. Converting npdas to context-free grammars.

Recursively Enumerable Languages

1. Turing machines acceptors

2. Turing machines transducers
3. Combining Turing machines
4. Linear-Bounded Turing machines
5. Multi-tape Turing machines
6. Nondeterministic Turing machines

The pre-requisite for this course is sixteen credits of Mathematics at second year level. Thirteen students enrolled for the course and the duration of the course is a semester. The students attended four forty-five minute lectures a week and one forty-five minute tutorial. In order to promote active learning during lectures, students were firstly presented with different concepts and then required to work on exercises covering the concepts. The lecturer and the students then worked on the exercises together and the students were given feedback. The weekly tutorial reinforced the concepts covered in lectures. At the end of the tutorial the students were required to submit their solutions. The solutions were marked and individual feedback was given. JFLAP was made available to students to help them visualize the different concepts and construct solutions. The use of JFLAP was not compulsory. The students sat for three forty-five minute tests during the semester. Each test covered different sections of the course.

Student solutions to the three tests and weekly tutorials were analyzed to identify learning difficulties experienced by students. The findings of this analysis are presented in the following section.

3. LEARNING DIFFICULTIES

This section discusses the difficulties experienced by students in learning regular languages, transducers, context-free languages and Turing machines. In addition to the topic-specific errors listed below, a general difficulty experienced by students was the conceptualization of proofs to theorems such as the Pumping Lemma.

3.1 Regular Languages

Students were able to conceptualize the workings of finite acceptors, however they experienced difficulties in constructing solutions for some languages. This learning difficulty can be attributed to immature problem-solving skills which need to be developed further.

The main difficulty that students experienced in converting NFAs to DFAs was dealing with epsilon transitions, namely, calculating δ^* in NFAs that contained many epsilon transitions. For example, consider Figure 1.

$\delta^*(C,c)$ would be C and B due to the epsilon transition connecting C to B. The error that students made was not including the B. This needs to be taken into consideration when teaching the conversion algorithm.

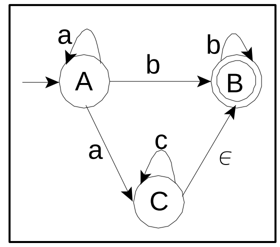

Figure 1: Example NFA

In applying the minimization algorithm to reduce the number of states in a DFA, a number of students combined two states together instead of separating them for one particular example. An example of a DFA to be minimized is illustrated in Figure 2 and the corresponding solution is depicted in Figure 3. An example of the error typically made by students in converting the DFA in Figure 2 to that Figure 3 is to combine E and F, for example, into one state.

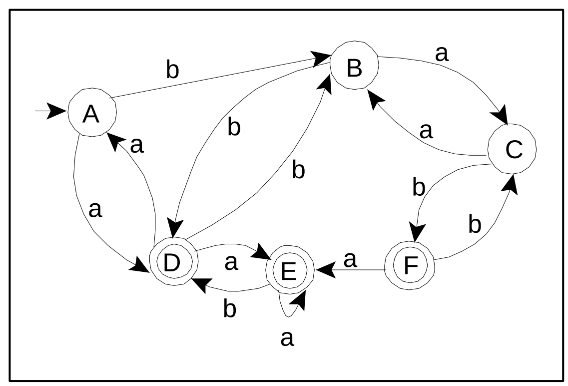

Figure 2: DFA to be minimized

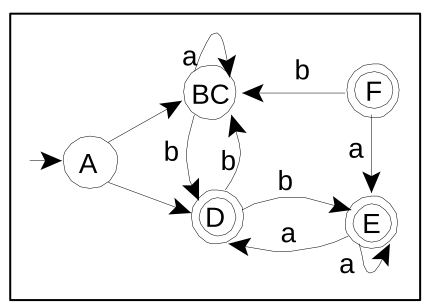

Figure 3: Minimized DFA

Students experienced similar difficulties in formulating regular expressions as they did with constructing finite acceptors, namely, their solutions contained logical errors.

This again can be attributed to a lack of problem–solving skills which need to be further developed. Students did not find difficulties with converting regular expressions to NFAs. However, converting NFAs to regular expressions proved to be challenging and students made errors when applying the conversion algorithm. The most common error was omitting components of the regular expression. In addition to this some students concatenated components of the regular expression instead of taking the union and vice versa. Figure 4 depicts an NFA to be converted to an RE.

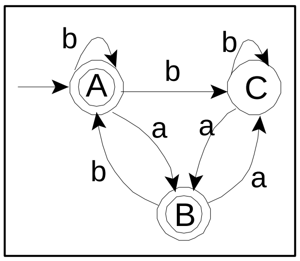

Figure 4: NFA to be converted to an RE

The first two steps of converting this NFA to an RE is illustrated in Figure 5. An example of a concatenation error made by students would be to concatenate *a* to *bb*b* on the edge from B to C in Figure 5. An omission error made by students was leaving out the *bb*b* component of the regular expression on this edge.

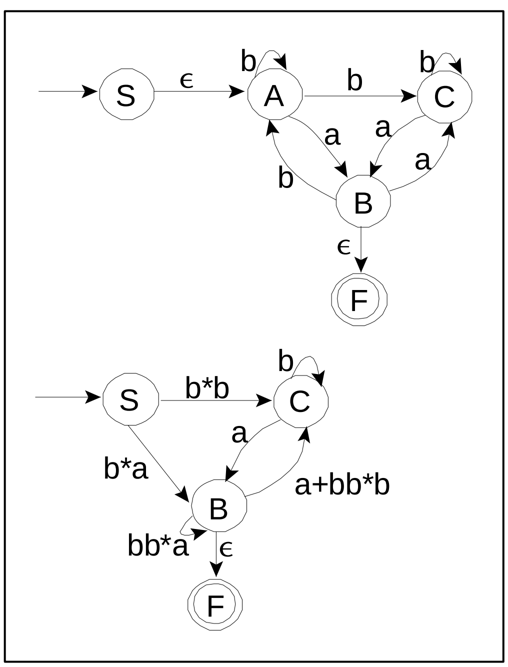

Figure 5: Partial conversion of the NFA to an RE

The main problem that students experienced with regular grammars was constructing grammars. Two students confused left-linear and right-linear grammars. The main error made when converting finite acceptors to regular grammars was the omission of epsilon from the production rule for accept states. Majority of the students did not have any difficulties in converting regular grammars to finite acceptors. Four students did not understand the algorithm for converting regular grammars to finite acceptors.

A majority of the students made logical errors when proving that a language is regular and using the Pumping Lemma to show that a language is non-regular. These could be attributed to a lack of problem-solving skills and an understanding of the Pumping Lemma.

3.2 Transducers

The main error that students made with transducers was constructing Mealy machines that accepted strings that were longer than that required for the problem. The Mealy machine for the even-2-parity problem is depicted in Figure 6. Some of the student solutions included transitions at states D, E, F and G from the state back to itself producing an output of 0 for any input. This produces output strings longer than that required for the problem. A few students made logical errors in constructing Mealy and Moore machines. One did not understand the concept of a transducer and included an accept state in the machine. Some solutions contained minor errors such as the omission of edge labels. Students did not seem to have any difficulties in converting Mealy machines to Moore machines and vice versa. A few students made minor errors, such as incorrect edge labels, when performing conversions.

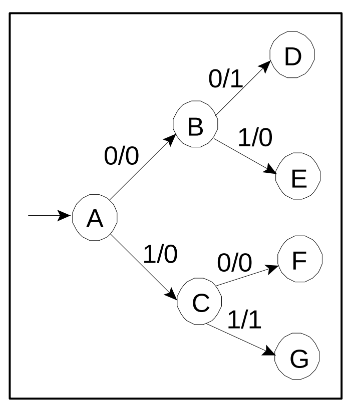

Figure 6: Mealy machine for the even-2-parity problem

3.3 Context-Free Languages

The main problem experienced by majority of the students was the construction of context-free grammars and pushdown automata (npdas and dpdas). Student solutions were partially correct and contained logically errors.

Students also experienced difficulties in simplifying grammars, namely, removal of useless productions and a recursive start symbol. Useless productions were not removed and recursive start symbols were not replaced. The conversion of simplified grammars to Chomsky normal and Greibach normal form did not pose any problems for students.

Similarly, a majority of the students did not experience difficulties in converting context-free grammars to pushdown automata and vice versa. Three students had problems in understanding the algorithm for converting nondeterministic pushdown automata to context-free grammars.

3.4 Recursively Enumerable Languages

Students possessed knowledge of Turing machine acceptors, transducers and the different types of Turing machines and understood how they worked. The major difficulty experienced was constructing of Turing machines for different problems. This again can be attributed to a need to further develop students' problem-solving skills to a level that will enable them to deal with such problems.

4. DISCUSSION

The main difficulty experienced by students for all FLAT topics was problem solving. Visualization tools such as JFLAP assist students in conceptualizing the different FLAT aspects and checking their solutions, however these tools do not directly aid them in further developing their problem-solving skills to the level necessary for FLAT courses. In order to develop this skill the students need to work on additional problems with feedback at each stage. It could be that a forty-five minute tutorial was insufficient and additional tutorials with immediate feedback are necessary. Intelligent tutoring systems are suitable for this purpose, however, the development of these systems for FLAT is still in its infancy and a lot more research needs to be done before they make their way into computer laboratories.

On a smaller scale students also experienced problems with understanding epsilon transitions when converting NFAs to DFAs, reducing the number of states in a DFA, removing useless productions when simplifying context-free grammars and converting NFAs to regular expressions.

When teaching the conversion of NFAs to DFAs more examples on calculating δ^* will be included. In addition to this students will be given exercises that enable them to focus on only calculating δ^* before applying the entire algorithm. Once they have grasped calculating δ^* with epsilon transitions they can move on to applying the overall conversion algorithm.

In applying the minimization algorithm some students have forgotten that there should be at most one outgoing arc for each element of the alphabet. As a result of this some students combined two states that should be separated for one of the examples. This will be re-emphasized in lectures.

Similarly, more examples requiring the removal of useless productions in simplifying context-free grammars will be included in both lectures and tutorials.

The use of visualization tools was not compulsory and thus students who experienced difficulties in conceptualizing the algorithm for converting NFAs to regular expressions may have not taken the initiative to utilize JFLAP. Thus, in the future a visualization tool will be used to present this algorithm to students during lectures. The difference between concatenating components of a regular expression and taking the union will also be emphasized in lectures and tutorials. Tutorial sessions requiring students to rectify each others' solutions may also be of benefit.

A general problem experienced by students was the conceptualization of proofs. Visualization tools have proven to be effective in assisting students in understanding the function and conversion algorithms of the different FLAT constructs. Thus visualization tools could assist students in understanding the different proofs. Future work will look at developing such tools..

5. CONCLUSION AND FUTURE WORK

The main aim of the study presented in this paper was to identify the difficulties experienced by students in learning FLAT topics. Previous studies have assumed that the main difficulty experienced by students is conceptualization of FLAT topics and propose the use of visualization tools to assist learners. However, this study has revealed that most of the errors made by students were logical errors due to immature problem-solving skills rather than lack of conceptualization or knowledge. Future work will focus on presenting students with exercises with immediate feedback to assist them in further developing their problem solving skills. The development of tools for this purpose will also be investigated. Future extensions of this study will also include identifying learning difficulties that students experience with other FLAT topics such as decidability.

REFERENCES

[1] Berque, D., Johnson, D.K., and Jovanovic, L. 2001. Teaching Theory of Computation Using Pen-Based Computers and an Electronic Whiteboard. ACM SIGCSE Bulletin inroads 33, 3, 169-172.

[2] Chesnevar, C.I., Cobo, M. L., and Yurcik, W. 2003. Using Theoretical Computer Simulators for Formal Languages and Automata Theory. ACM SIGCSE Bulletin inroads 35, 2, 33-37.
[3] Devedzic, V., Debenham , J., and Popvic D. 2000. Teaching Formal Languages by an Intelligent Tutoring System. Educational Technology and Society 3, 2, ISSN 1436-4522.
[4] Esmoris, A., Chesnevar, C. I., and Gonzalez M. P. 2005. TAGS: A Software Tool for Simulating Transducer Automata. International Journal of Electrical Engineering Education 42, 4, 338-349.
[5] Garcia-Osorio, C., Mediavilla-Saiz, I., Jimeno-Visitacion, J., and Garcia-Pedrajas N. 2008. Teaching Pushdown Automata and Turing Machines, inroads –SIGCSE Bulletin, 40, 3, 316.
[6] Pillay, N. and Naidoo, A. 2006. An Investigation into the Automatic Generation of Solutions to Problems in an Intelligent Tutoring System for Finite Automata. Proceedings of the 36th SACLA Conference - 2006, 84-93.
[7] Rodger, S.H., Wiebe, E., Lee, M. E. 2009. Increasing Engagement in Automata Theory with JFLAP. Inroads SIGCSE Bulletin 41, 1, 403-407.
[8] Vieira, L. F. M., Vieira, M. A. M., and Vieira, N. J. 2004. Language Emulator, A Helpful Toolkit in the Learning Process of Computer Theory. ACM SIGCSE Bulletin inroads 36, 1, 135-139.
[9] Wermelinger, M., and Dias, A. M. 2005. A Prolog Toolkit for Formal Languages and Automata. ACM SIGCSE Bulletin-inroads 38, 3, 330-334.
[10] Zingaro, D. 2008. Another Approach for Resisting Student Resistance to Formal Methods. inroads-SIGCSE Bulletin. 40, 4, 56-57.