USE OF VISUAL TOOLS IN DISTANCE TEACHING OF COMPUTATIONAL MODELS

Judith Gal-Ezer[1], and Mark Trakhtenbrot[2]

Abstract *- We present visual interactive tools for distance teaching in course "Automata Theory and Formal Languages" (AFL) that explores mathematical models of computation. Using the tools, students can interactively explore the various classes of automata and grammars, their properties, expressive power, and transformations. The tools help to get prompt answers to typical questions, and to gain a better sense of concepts that would otherwise remain purely formal. This is especially important in distance education, where traditional difficulties in learning theoretical material become even more severe. Our tools address cognitive and communication aspects of asynchronous learning process. The need in such tools has been recently widely recognized.*

Index Terms – Computer science education, Distance teaching, Visualization and animation.

PROBLEMS IN LEARNING THEORETICAL CS

Theoretical computer science courses (such as AFL) are characterized by intensive use of formal methods. Students have to understand the intuition and motivation behind the various mathematical models, their properties and relationships; learn the relevant construction and validation methods; get familiar with various proof techniques; and master the ways to apply the gained knowledge in problem solution. Altogether, this causes severe difficulties in the learning process; they are even more crucial in distance education due to rare or no face-to-face tutorial sessions.

WHEN TOOLS MAY HELP?

A good way to address reasoning-related problems (raised whereas proofs play a central role) is the use of appropriate instructive examples that help to highlight typical misunderstandings and misconceptions, to analyze them and to show how they might lead to wrong results.

On the other hand, construction of models and their validation, as well as understanding of how the various algorithms work, is better achieved by visualization and animation. For this, interactive software tools are best suited. Using such tools, it is also much easier to explore the impact of model modifications on its properties and on results of various model transformations. See [1-2] for tools examples.

TOOLS' CAPABILITIES

Our tools provide a wide variety of capabilities:
visual creation and editing of models (state-transition diagrams and grammar trees);
static checks of model properties, to reveal various kinds of errors and incompleteness (e.g. unused grammar symbols);
simulation of finite automata (deterministic and non-deterministic), this includes continuous or stepwise run, stepping back, animation of state-transition diagrams;
interactive visual derivation in context-free grammars, including animation of derivation trees;
transformations, with **traceability** between elements of the original and the resulting models:

- determinization and minimization of finite automata
- grammar simplifications, translation into normal forms
- translations between regular expressions, automata, grammars

These features are supported on **two levels of details**. In the first, only final result is provided. Student can compare with it his own solution (e.g. by comparing simulation traces) and if needed, make modifications and have another try. The second level provides all solution stages, with explanations and references to the course textbook.

Sharing of models created with the tools facilitates **better communication** between all course participants (e.g. submitting questions and solutions by students, checking and posting assignment solutions by tutors).

CONCLUSION AND FUTURE WORK

Students provided positive initial feedback on the tools. We plan to further extend them, to carry out a thorough comparison with other existing tools, and to perform a detailed study of the effectiveness of their use.

REFERENCES

[1] Lepore M. and Spencer D., "FinITE - Finite automata Interactive Theory Exploration, v.0.98b", Worcester Polytechnic Institute, 1995

[2] Rodger S.H., "Using Hands-on Visualizations to Teach Computer Science from Beginning Courses to Advanced Courses", *Second Program Visualization Workshop*, June 2002, pp. 103-112.

[1] Judith Gal-Ezer, Open University of Israel, 16 Klausner St., Tel-Aviv 61392, Israel galezer@openu.ac.il
[2] Mark Trakhtenbrot, Open University of Israel, 16 Klausner St., Tel-Aviv 61392, Israel, marktr@openu.ac.il , and Holon Academic Institute of Technology, 52 Golomb St., POB 305, Holon 58102, Israel