TEACHING, LEARNING AND AI

LUCIAN M. IONESCU

ABSTRACT. Teaching and Learning occur concomitantly, with various weights, in any interaction between two systems.

In this article we will explore some general aspects, in order to better understand how to plug-in Mathematica, as a mathematical software, to a Math college course, like Calculus III.

The role of formal languages, especially adaptive grammars, is emphasized, as the "other side" of the approach focusing on automata.

CONTENTS

1. INTRODUCTION

Pondering on a process, from time to time, helps clarify and improve; this is the purpose of this essay-like article, a "research en passant", to address a few aspects regarding the role of modern Computer Algebra Systems, including a free-form interface providing AI-characteristics, in the teaching and learning process, e.g. in Mathematics Education.

This preliminary work is based on a recent presentation at the ISU Undergraduate Symposium [1]. Future versions will build on top (or rather add at the bottom) of it, continuing to address together with [2], some borderline topics of Research and Development in Mathematics.

Date: January 2020.

2. What is "Teaching / Learning"?

Due to the broadness of the question, We'll point to some aspects less emphasized, needed in the sequel.

Here, Teaching / Learning is an iterative process in which a system A conditiones a system B, with changes of their own "transition tables", and will be modeled as a Machine Learning process, via a sequence of layers ...

Main example is how a Machine learns how to recognise the digits $0, ..., 9$, from an image, say 728 x 728 [3].

2.1. The Language aspects.

Automata / Languages: each system adapts its grammer / extends its language, to achieve "resonance" (in some sense ...), or severe the connection if dissonance is increased.

2.2. Teaching Calc III with Mathematica.

Todays model of College Courses makes use of textbooks for traditional courses. Electronic version of this allows for coupling the course with an online homework-grading system, like WebAssign.

The use of calculators has generalized from, e.g. TI-83/89, to full Mathematical Software like Mathematica.

In a nutshell, students need to:

1) Learn from the textbook the *concepts* of the theory, introduced via definitions, their *properties* via Propositions and Theorems, and *methods* needed for solving various *types of problems.* We will call this the "human interface" to the Theory.

2) Learn from a Mathematica textbook ("User's Manual") the syntax for the mathematical concepts needed to solve problems in the above Theory. We call this the Machine Interface.

Mathematica has also a powerfull free-form interpretar capable of "bridging" the gap between the two interfaces (1) and (2). Unfortunately, this "free-form" is not always reliable; but all the "bad" has its good side: the student needs to be pro-active, inventive, and explore what Mathematica's free-form provides, to make his/her way to the needed syntax, or ... ask the Teacher (Human).

For example, in Calc III based on Stewart's textbook [4], Ch. 16, Stokes Theorem relates the concepts of *flux of a vector field* (VF) F through a *surface* S, and *circulation* of its *curl* $curl(F)$ through its *boundary.*

• **Layer 1**:

- The *application interface* (application language) here consists of the words: flux, circulation; - the *mathematics interface* onto which the application (physics) is mapped consists in: curve, surface, line integral, surface integral, curl, vector field.

View this "Phys -¿ Math" as 1st layer of a Neural Network model (to be built in Student's brain :).

• **Layer 2**

The above Math interface is explained in the textbook, and is mapped onto some computational prescriptions (algorithms/formulas etc.), typically explained in previous sections and Chapters of the textbook. This would be solving the problem by hand.

For example, Problem 16.8/3 (?), asks to "Check Stokes Theorem for ...".

using a Math Software, e.g. Mathematica, should allow to focus on the Application, after of course some other preliminary problems checking the student understood the math behind it, solving "by-hand".

But in any case, some trivial integrations will be done by Calculator, e.g. simple integrals using TI-89.

So, focusing on the application, one has to "bridge-the-gap" with Mathematica.

• **Layer 3**:

Let's consider the concrete example above.

a) Surface defined by (algebraic) conditions (equations) ...

b) Curve is implicitly contained in the description of the surface: corresponds to intersection: *solving a system of equations.*

c) *Vector field* defined by components;

Does Mathematica have such capabilities? i.e. to "define" the *surface* S as above, VF F and then ask for the *surface integral* of F over S?

2.3. **Programming as a Teaching Process.** This is in essence a process where the *student teaches the Machine* (Mathematica), by programming it, and bridging the gap. It is a valuable experience for the student.

On the other hand, one can have a team of Mathematica Programmers do this, and define a *custom made grammer / language module* for Calc III. Ch. 16 relies on Ch. 15, which in turn relies on Ch. 14,13,12; but below Ch.12, the standard capabilities of Mathematica are already in place (an equivalent of TI-89, for instance).

In such a case, the Student becomes mearly a *User* of the Math-Software, having to understand only the concepts, e.g. Ch. 16.6-9.

2.4. **Pedagogical outcomes.** This shows that the Calc III Teacher may choose to subject the student to various scenarios: from computational / by-hand solutions, to programmer and machine teacher, or just scientist / application user / designer, by allowing various resources to be used by the Student.

2.5. **Mathematica Developers Goal.** The above discussion suggests that *textbook oriented Expert Systems* may be built on top of Mathematica; e.g. "Calc III Expert System" programmed in Wolfram Mathematica, and available in Wolfram Alpha, as if we would chat with a Mathematical-savvy online chatbot [5].

3. Conclusions and Further Developments

This was just a sketch of the roles in the Teaching/Learning Trinity Student, teacher, Machine (as assistant); what needs to be done is to clarify the "Diagramatic" of this, and "standardise" the *protocols* for the various interfaces.

In other words, "Who is teaching Who/What?" and as a result, adapts by Learning (conform to Neural networks model of Machine Learning).

Starting by building on the above example would help maintain the clarity, before adventuring into a wild generalization in terms of cross-over algebras, evolutive grammars etc.

Also, in the process of "bridging gaps", i.e. connect interfaces via some "plug-ins" or "drivers", one would try to "hide" the implementation as neatly as possible (no visible loose wires, unneeded switches etc.): *user friendly, minimal interfaces*!

References

[1] L. M. Ionescu, "Teaching mathematics with Technology", ISU CTLT Symposium, Intentional Technologies: Reclaiming our Spaces, Jan. 8, 2020, https://ctlt.illinoisstate.edu/symposium/2020/#tabs-accord3

[2] L. M. Ionescu, "A few misconceptions in Mathematics", www.hpm-americas.org/2014/04/01/april-2014-illinois-state-university-meeting-information/
see also: https://my.ilstu.edu/~lmiones/presentations_drafts.htm

[3] 3Blue1Brown,"But what is a neural network?", https://www.youtube.com/watch?v=aircAruvnKk

[4] Stewart, Calculus with Early Transcendentals, v.6.

[5] Dan Shewan, "10 of the most innovative chatbots", https://www.wordstream.com/blog/ws/2017/10/04/chatbots