Artificial Intelligence 231 (2016) 17–38

Contents lists available at ScienceDirect

Artificial Intelligence

www.elsevier.com/locate/artint

Automated conjecturing I: Fajtlowicz's Dalmatian heuristic revisited

C.E. Larson [a,*,1], N. Van Cleemput [b,c]

[a] Department of Mathematics and Applied Mathematics, Virginia Commonwealth University, Richmond, VA 23284, United States
[b] Department of Mathematics, European Centre of Excellence NTIS (New Technologies for the Information Society), University of West Bohemia, Univerzitní 8, 306 14 Plzeň, Czech Republic
[c] Department of Applied Mathematics, Computer Science and Statistics, Ghent University, Krijgslaan 281 – S9 – WE02, 9000 Ghent, Belgium

ARTICLE INFO

Article history:
Received 22 December 2013
Received in revised form 9 October 2015
Accepted 12 October 2015
Available online 2 November 2015

Keywords:
Automated conjecturing
Automated conjecture-making
Mathematical discovery
Automated scientific discovery
Dalmatian heuristic

ABSTRACT

We discuss a new implementation of, and new experiments with, Fajtlowicz's Dalmatian conjecture-making heuristic. Our program makes conjectures about relations of real number invariants of mathematical objects. Conjectures in matrix theory, number theory, and graph theory are reported, together with an experiment in using conjectures to automate game play. The program can be used in a way that, by design, advances mathematical research. These experiments suggest that automated conjecture-making can be a useful ability in the design of machines that can perform a variety of tasks that require intelligence.

1. Introduction

We have reimplemented Fajtlowicz's useful but little-known Dalmatian heuristic for the automation of mathematical conjecture-making (this heuristic, for instance, has never been referenced in the pages of *this journal*). The heuristic is general and can be used to conjecture relations between real number invariants of any objects, mathematical or otherwise. We include examples of conjectures in number theory, matrix theory, graph theory and the characterization of game positions. One of the number theory conjectures implies, and is stronger than, Goldbach's Conjecture. Some of the number theory conjectures seem to imply the Riemann Hypothesis. And some of the graph theory conjectures would advance the lower bound theory of the independence number of a graph, a widely-studied NP-hard graph invariant. We have also implemented an idea, suggested to us by Barry Mazur, to include existing theorems in the program; when used in this way the program is guaranteed to produce statements that are not implied by existing mathematical knowledge.

Our program often makes interesting and useful conjectures on the basis of only a few examples. Humans, ordinarily and of necessity, make decisions based on very limited data. A general automated conjecture-making module that can make plausible and useful guesses based on limited data may be a central architectural feature in the design of machines that are intelligent. Guesses can be used, for instance, to constrain a search of possible actions. Fajtlowicz introduced his Dalmatian heuristic for the automation of mathematical conjecture-making more than 20 years ago [1]. Simply put, the heuristic is to produce a considered mathematical statement if it is both true—with respect to some given examples (matrices, integers,

* Corresponding author. Tel.: +1 804 828 5576; fax: +1 804 828 8785.
E-mail addresses: clarson@vcu.edu (C.E. Larson), nico.vancleemput@gmail.com (N. Van Cleemput).
[1] The authors dedicate this article to Prof. Justin Leiber, teacher, mentor, friend and inspiration.

http://dx.doi.org/10.1016/j.artint.2015.10.002
0004-3702/

graphs, etc.)—and if the statement gives new information about those objects, in particular, if it says something about at least one of the objects which is not implied by any other stored statement or conjecture.

It was very successful—both in limiting the number of conjectures produced by earlier versions of his GRAFFITI program and in producing conjectures of interest to research mathematicians. His student DeLaVina reimplemented the heuristic in a program that produces conjectures that have led to research and publications by mathematicians [2]; otherwise the heuristic has not been used. Fajtlowicz made some experiments to demonstrate the domain independence of the Dalmatian heuristic; nevertheless, the predominant and best-known uses of the heuristic—in the programs of Fajtlowicz and DeLaVina—has been in the production of graph theory conjectures. But the heuristic is not specific to the production of graph theory conjectures.

Our program is open-source, written in Python and C, and implemented as a Sage package. Details about the acquisition and use of our program, the Sage open-source mathematical computing environment, and how to reproduce our results are relegated to Appendix A.

Our experiments in implementing and applying this heuristic, including in domains where the authors have no more knowledge than anyone who has browsed a textbook or reference book, lead us to make several conclusions, which we will elaborate and discuss.

1. Successful mathematical discovery heuristics can be applicable in a variety of mathematical domains.
2. Good conjectures can be based on very limited data.
3. Mathematical discovery programs should aim to produce conjectures that address and advance pre-existing mathematical questions.
4. Intelligent conjecture-making programs for a domain do not require developer expertise in that domain.

Some of these conclusions should be surprising and, we hope, inspire new research in automated scientific discovery.

We see conjecture-making—and conjecture-revision in the face of contradictory data (counterexamples)—as a central feature of intelligence. We make guesses, based on our previous experience in relevantly similar situations, learn that our guesses are wrong, revise them, and test them against our experience.

2. Background & related work

Turing, famously, proposed the idea of designing intelligent machines as an engineering problem, and proposed a test for evaluating the success of such machines. In 1948 he suggested designing machines to do mathematical research as a starting point: mathematical research certainly requires intelligence and, it would be a good starting point as mathematical research would "require little contact with the outside world" [3]. In the 1950s Newell and Simon developed the Logic Theorist program that could prove (some) theorems in first-order logic, and went on to predict that a computer would discover and prove an important mathematical theorem within another decade [4]. Success did not come quite that quickly—but there has been significant progress in many areas of automating mathematical discovery, and there is no theoretical impediment to continued improvement. There is every reason to believe that Newell and Simon's prediction will be achieved—and likely sooner rather than later.

The automation of theorem proving is by far the largest and best-developed area of automated mathematical discovery research. A highlight in this area was the 1996 computer proof of the Robbins Conjecture [5]. More recently Timothy Gowers, a Fields Medalist, and likely the most accomplished mathematician to do research in automated mathematical discovery has, together with Mohan Ganesalingam, developed a theorem-proving program.[2]

Research on automated conjecture-making was initiated by Wang in the late-1950s [6]. His Program II produced thousands of statements in propositional logic that could be considered as conjectures or potential theorems. His program included heuristics for deciding which statements to output. Evaluated as a tool for advancing mathematical research, Wang's program was a failure. He wrote:

> It was at first thought that these crude principles are sufficient to cut down the number of theorems to a degree that only a reasonably small number of theorems remain. It turns out that there are still too many theorems. The number of theorems printed out after running the machine for a few hours is so formidable that the writer has not even attempted to analyze the mass of data obtained [6].

What Wang really wanted was for his program to produce a limited number of statements of interest to logicians. Wang selected a few statements to include in publication—but what was really needed was a way for the program itself to identify the interesting, useful or important statements.

The first program to make conjectures leading to published mathematical research was Fajtlowicz's GRAFFITI program [7–10,1]. An early version of GRAFFITI was called the "Sorcerer's Apprentice" [11] because the program, like Wang's, produced a large number of statements. In the Goethe poem (and the Disney *Fantasia* version with Mickey Mouse) a sorcerer's apprentice intends to use his master's spells to animate a broom to help him carry a bucket of water but he ends up with

[2] A preprint of their paper is available at: arxiv.org/abs/1309.4501.