Explanation as Proof
Position Paper for PTHG-20
September 7, 2020

Eugene C. Freuder
Insight Centre for Data Analytics
School of Computer Science and Information Technology
University College Cork
Cork, Ireland
eugene.freuder@insight-centre.org

Introduction

Explanation is a hot topic in artificial intelligence. For constraint satisfaction problems most attention has focused on explaining failure when confronted with overconstrained problems. The most obvious form of explanation for why there is no solution to a constraint satisfaction problem is to list all possible instantiations of the variables and show that each involves at least one conflict. This, of course, is of dubious practical value for people seeking to understand why their problem is unsolvable, in particular how they might modify the problem to achieve a solution.

There have been many approaches to explanation in constraint programming (Freuder 2018), including determining minimal sets of conflicts and tracing the reasoning that led to failure. But all these approaches bring to mind to some degree a child looking up at a parent, who has just delivered a careful explanation, and saying plaintively "But why?"

This paper suggests an approach that might satisfy such children — at least if they are future mathematicians. It is the approach that has been used forever in the mathematical and computational sciences, namely that we provide a *proof* of unsatisfiability. Of course, many, if not all, of the other approaches to explanation just alluded to could be regarded as proofs, and proving unsolvability has been studied outside the context of explanation, e.g. (Gaur and Kahn 2020). (Veksler and Strichman 2010) describe a solver that generates its own proof of unsolvability. What is intended here is something akin to a "narrative" human proof, where the reasoning provides some high-level insight into the "why".

This approach can be compact. It does not require running a solver to failure, and thus might be useful in situations where that is computationally expensive or even impractical. It could provide useful insights into the nature of the problem, the nature of the failure, and the options for altering the problem to permit success. Producing such proofs appears to be a good opportunity for human-computer collaboration.

Of course, automating the production of such proofs is distinctly non-trivial, and I do not attempt it here. I simply want to raise it as an aspiration. One approach might be to try to 'translate' formal proofs into more human-friendly terms. Another might be to recognize and then automate the application of common principles, especially from combinatorial mathematics. The "inferences" provided by constraint propagation might also play a role.

In the rest of this position paper I will illustrate and further explore the idea of explanation as proof with a few simple examples.

First we consider two unsolvable variations on the well-known Queens Problem (the problem of placing 8 queens on a standard 8 by 8 chessboard such that no two attack one another).

Nine Queens

Consider the problem of placing 9 queens on a regular 8 by 8 chessboard such that no two attack one another. We could try all 64^9 possible placements (including those that stack queens on top of one another) and demonstrate that none succeed; but that would be a bit tedious. Consider instead this proof:

The Pigeonhole Principle from combinatorial mathematics says that if we want to place n objects into m containers, where $m<n$, at least one container must contain more than one object. In this case, if we want to place 9 queens into the 8 rows of the chessboard, at least one row must contain more than one queen. But queens in the same row will attack each other. So there is no solution. Q.E.D.

This proof also suggests a natural way to try to transform our problem into a solvable one: lose one of the queens. We are then left with the classic 8-Queens problem, which is, in fact, solvable.

Three Queens

We will give a proof that we cannot place 3 queens on a 3 by 3 board such that no two attack one another:

There are 9 places to put the first queen. The center square attacks all others so we can't use that. We observe by inspection that each of the other squares attacks 6 others, and no two attack exactly the same 6 others. Imagine we've successfully placed 3 queens such that no two attack each other. The 6 squares a queen attacks cannot include the square the queen itself is on, of course, nor can they include the squares where the other two queens are placed. There are only 9 squares so each queen must attack exactly those 6 squares that the 3 queens do not occupy themselves. But this contradicts the fact that no two squares attack exactly the same set of other squares. Q.E.D.

This is a bit more complicated than invoking the Pigeonhole Principle, but it still provides a high level explanation that involves some "structural" insight into the nature of the difficulty.

Interestingly, a little googling turned up an alternative proof [https://math.stackexchange.com/questions/2617260/nonattacking-queens-on-3x3-board-why-is-there-no-solution]:

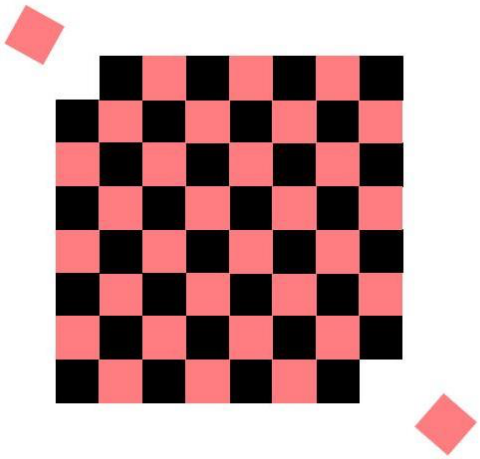
"Clearly, the center square cannot be occupied, since it attacks every other square.
If a corner square is occupied, this leaves exactly two squares that are not attacked; specifically, those that are one knight's move away from the occupied corner. But these two squares mutually attack each other, so it is impossible to place two more queens. Hence no queen can occupy a corner. Having eliminated the center and the corners, we conclude that three queens must be placed among the four edge squares, but again this is trivially impossible since any occupied edge attacks the other three edge squares." Q.E.D.

This provides an alternative insight into the structure of the problem. While these two proofs complement one another, they also raise the general question of what makes one explanation "better" than another. While we could proposes some general measures, like "shorter is better", it might be best to embrace diversity and seek insight into the different advantages different explanations can provide in different circumstances (Wallace and Freuder 2001).

Mutilated Chessboard

Now another chessboard problem, but one involving dominos instead of queens.

Consider the problem of covering a chessboard that has had two opposite corners removed with 31 dominos.



(Figure from http://cognitivepsychology.wikidot.com/problem-solving:insight.)

This can be viewed as a constraint satisfaction problem with 31 variables (the placement of the 31 dominoes) each of which has 108 possible values (placement positions) for a total of 108^{31} possible ways to place all the dominos (including placing dominoes on top of each other). None of these will work as a covering of the board, the problem is unsolvable; but it would take quite a while to prove unsolvability by checking all the 108^{31} possiblilities individually. However, there is a concise proof of unsolvability:

Each domino, wherever placed, covers one black square and one pink square in the above figure. So 31 dominos will cover 31 black squares and 31 pink squares. But there are 32 black squares and 30 pink squares. So we can't cover all the squares with 31 dominos. Q.E.D.

The Mutilated Chessboard has been used to illustrate the importance of representation for reasoning (Kaplan and Simon 1990). In particular, it is harder for people to determine that the problem is unsolvable if it is presented as covering a grid of 62 squares all of the same color. By extension the problem also illustrates the importance of representation for explanation.

Note that this explanation also suggests how we might alter the problem to make it solvable. If we remove adjacent, as opposed to opposite, corners, the problem becomes solvable.

Conclusion

"Narrative" proofs of unsatisfiability for constraint satisfaction problems can be concise and insightful. It is challenging to consider automating their production for practical problems. However, as AI becomes more pervasive in everyday life, human-scale, automated, narrative proofs for human-scale problems may prove both desirable and achievable.

References

Freuder, E. (2018). Progress towards the Holy Grail. *Constraints, 23.* 158–171.

Gaur, D. and Khan, M. (2020). Testing Unsatisfiability of Constraint Satisfaction Problems via Tensor Products. *International Symposium on Artificial Intelligence and Mathematics*. https://isaim2020.cs.ou.edu/papers.html.

Kaplan, C. and Simon, H. (1990). In search of insight. *Cognitive Psychology, 22.* 374-419.

Veksler, M. and Strichman, O. (2010). *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence.* 204-209.

Wallace, R. and Freuder, E. (2001). Explanations for whom?. *CP 2001 Workshop on User-Interaction in Constraint Satisfaction.* http://www.cs.ucc.ie/~osullb/cp01/.